



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

**Βελτιστοποίηση του Υπολογιστικού Πυρήνα
Πολλαπλασιασμού Αραιού Πίνακα με Διάνυσμα σε
Σύγχρονες Πολυπύρηνες Αρχιτεκτονικές
Υπολογιστών**

Διδακτορική Διατριβή

Βασιλείου Κ. Καρακάση

Διπλωματούχου Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών

Αθήνα,
Δεκέμβριος 2012



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

**Βελτιστοποίηση του Υπολογιστικού Πυρήνα Πολλαπλασιασμού
Αραιού Πίνακα με Διάνυσμα σε Σύγχρονες Πολυπύρηνες
Αρχιτεκτονικές Υπολογιστών**

Διδακτορική Διατριβή

Βασιλείου Κ. Καρακάση

Διπλωματούχου Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών

Συμβουλευτική Επιτροπή: Νεκτάριος Κοζύρης
Παναγιώτης Τσανάκας
Ανδρέας Σταφυλοπάτης

Εγκρίθηκε από την επταμελή επιτροπή την 19η Δεκεμβρίου 2012.

Νεκτάριος Κοζύρης
Αναπ. Καθηγητής Ε.Μ.Π.

Παναγιώτης Τσανάκας
Καθηγητής Ε.Μ.Π.

Ανδρέας Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Ανδρέας Μπουρντουβής
Καθηγητής Ε.Μ.Π.

Γιώργος Στάμου
Λέκτορας Ε.Μ.Π.

Δημήτριος Σούντρης
Επικ. Καθηγητής Ε.Μ.Π.

Ιωάννης Κοτρώνης
Επικ. Καθηγητής Ε.Κ.Π.Α.

Αθήνα,
Δεκέμβριος 2012

Βασίλειος Κ. Καρακάσης
Διδάκτωρ Εθνικού Μετσοβίου Πολυτεχνείου

Copyright © Βασίλειος Κ. Καρακάσης, 2012.
Με επιφύλαξη παντός δικαιώματος. All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Η έγκριση της διδακτορικής διατριβής από την Ανώτατη Σχολή των Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Ε. Μ. Πολυτεχνείου δεν υποδηλώνει αποδοχή των γνώμων του συγγραφέα (Ν. 5343/1932, άρθρο 202).

*Στους γονείς μου,
Κωνσταντίνο και Ειρήνη,
και στα αδέρφια μου,
Μάριο, Όλγα, Αλέξανδρο*

Περίληψη

Η διατριβή αυτή εστιάζει στην βελτιστοποίηση του υπολογιστικού πυρήνα Πολλαπλασιασμού Αραιού Πίνακα με Διάνυσμα (SpMV) στις σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών. Πραγματοποιούμε μία εις βάθος ανάλυση του πυρήνα και αναδεικνύουμε τα κυριότερα προβλήματα επίδοσης. Αυτό μας επιτρέπει να προτείνουμε ένα προχωρημένο σχήμα αποθήκευσης αραιών πινάκων, το σχήμα Compressed Sparse eXtended (CSX), το οποίο εστιάζει στοχευμένα στην ελαχιστοποίηση του μεγέθους αναπαράστασης του αραιού πίνακα. Το σχήμα αυτό μπορεί να παράσχει σημαντική βελτίωση της επίδοσης του SpMV σε μία πληθώρα διαφορετικών πινάκων και πολυπύρηνων αρχιτεκτονικών, διατηρώντας μία σημαντική σταθερότητα στην επίδοση. Τέλος, μελετάμε την επίδοση του SpMV υπό το πρίσμα της ενεργειακής του απόδοσης, με σκοπό την αναγνώριση των ρυθμίσεων εκτέλεσης που οδηγούν σε βέλτιστους συμβιβασμούς επίδοσης-κατανάλωσης ενέργειας.

Λέξεις Κλειδιά: high performance computing; scientific applications; sparse matrix-vector multiplication; multicore; data compression; energy-efficiency; SpMV; CSX; HPC

Abstract

This thesis focuses on the optimization of the Sparse Matrix-Vector Multiplication kernel (SpMV) for modern multicore architectures. We perform an in-depth performance analysis of the kernel and identified its major performance bottlenecks. This allows us to propose an advanced storage format for sparse matrices, the Compressed Sparse eXtended (CSX) format, which targets specifically the minimization of the memory footprint of the sparse matrix. This format provides significant improvements in the performance of the SpMV kernel in a variety of matrices and multicore architectures, maintaining a considerable performance stability. Finally, we investigate the performance of the SpMV kernel from an energy-efficiency perspective, in order to identify the execution configurations that lead to optimal performance-energy tradeoffs.

Keywords: high performance computing; scientific applications; sparse matrix-vector multiplication; multicore; data compression; energy-efficiency; SpMV; CSX; HPC

Περιεχόμενα

Περιεχόμενα	i
Κατάλογος σχημάτων	v
Κατάλογος πινάκων	ix
Κατάλογος Αλγορίθμων	xi
Αντί Προλόγου	xiii
Εκτεταμένη Περίληψη	xv
1 Εισαγωγή	1
1.1 Αραιά γραμμικά συστήματα	1
1.2 Οι επαναληπτικές μέθοδοι από υπολογιστικής απόψεως	4
1.3 Προκλήσεις των πολυπύρηνων αρχιτεκτονικών	6
1.3.1 Θέματα κατανάλωσης ενέργειας	8
1.3.2 Η αλγοριθμική φύση του πυρήνα SpMV	9
1.3.3 Η ενεργειακή απόδοση του πυρήνα SpMV	11
1.4 Συμβολή της διατριβής	11
1.4.1 Εις βάθος ανάλυση της επίδοσης και μοντέλα πρόβλεψης	12
1.4.2 Το σχήμα αποθήκευσης αραιών πινάκων CSX	12
1.4.3 Προς μία ενεργειακά αποδοτική υλοποίηση του SpMV	15
1.5 Δομή της διατριβής	16
2 Αποθήκευση Αραιών Πινάκων	19
2.1 Συμβατικά σχήματα αποθήκευσης	19
2.2 Εκμετάλλευση των πυκνών δομών του αραιού πίνακα	22
2.2.1 Σχήματα σταθερού μεγέθους μπλοκ	23
2.2.2 Σχήματα μεταβλητού μεγέθους μπλοκ	26

2.2.3	Άλλες προσεγγίσεις	30
2.3	Άμεση συμπίεση των δομών δεικτοδότησης του πίνακα	33
2.4	Εκμετάλλευση της συμμετρίας	35
3	Η επίδοση του πυρήνα SpMV	39
3.1	Η αλγοριθμική άποψη	39
3.2	Προκαταρκτικά πειραμάτων	43
3.2.1	Σουίτα πινάκων	43
3.2.2	Πλατφόρμες υλικού	44
3.2.3	Διαδικασίες και πολιτικές μετρήσεων	49
3.3	Ποσοτική αξιολόγηση	50
3.3.1	Σειριακή επίδοση	50
3.3.2	Πολυνηματική επίδοση	53
3.4	Σχετικές εργασίες	67
3.5	Περίληψη	68
4	Δυνατότητες βελτιστοποίησης των μεθόδων μπλοκ	71
4.1	Η επίδραση της συμπίεσης	71
4.2	Η επίδραση των υπολογισμών	75
4.2.1	Η τεχνική «vectorization» και το σχήμα των μπλοκ	76
4.3	Πρόβλεψη του βελτίστου μεγέθους μπλοκ	82
4.3.1	Το μοντέλο MEM	82
4.3.2	Το μοντέλο SPARSITY	83
4.3.3	Το μοντέλο MEMCOMP	84
4.3.4	Το μοντέλο OVERLAP	84
4.3.5	Αξιολόγηση της ακρίβειας των μοντέλων	86
4.3.6	Επεκτάσεις	90
4.4	Περίληψη	92
5	Το Σχήμα Αποθήκευσης Compressed Sparse eXtended	93
5.1	Η ανάγκη ενός ενιαίου σχήματος αποθήκευσης	94
5.2	Οι δομές δεδομένων του CSX	97
5.3	Ανίχνευση και κωδικοποίηση κανονικοτήτων	99
5.3.1	Εξόρυξη κανονικοτήτων από τον πίνακα	99
5.3.2	Επιλογή κανονικοτήτων προς τελική κωδικοποίηση	105
5.3.3	Κατασκευή των δομών δεδομένων του CSX	106
5.4	Παραγωγή του κώδικα SpMV	107
5.5	Αντιμέτωπιση του κόστους προεπεξεργασίας	111
5.6	Μεταφορά σε αρχιτεκτονικές NUMA	113
5.6.1	Βελτιστοποίηση των υπολογισμών	113
5.7	Αξιολόγηση της επίδοσης του CSX	115

5.7.1	Δυνατότητες συμπίεσης του CSX	116
5.7.2	Η επίδοση του CSX	117
5.7.3	Το κόστος προεπεξεργασίας του CSX	126
5.8	Ενσωμάτωση του CSX σε λογισμικό multiphysics	128
5.9	Περίληψη	130
6	Εκμετάλλευση συμμετρίας στους αραιούς πίνακες	133
6.1	Ο συμμετρικός πυρήνας SpMV	134
6.2	Ελαχιστοποίηση του κόστους αναγωγής	136
6.2.1	Ενεργό εύρος τοπικών διανυσμάτων	137
6.2.2	Δεικτοδότηση τοπικών διανυσμάτων	138
6.2.3	Εναλλακτικές μέθοδοι	141
6.3	Το σχήμα CSX για συμμετρικούς πίνακες	142
6.4	Αξιολόγηση της επίδοσης	143
6.4.1	Μέθοδοι τοπικών διανυσμάτων	145
6.4.2	Συμμετρικό CSX	147
6.4.3	Πίνακες μειωμένου εύρους ζώνης	148
6.4.4	Επίδραση επί της επαναληπτικής μεθόδου CG	151
6.5	Περίληψη	153
7	Θέματα ενεργειακής απόδοσης	155
7.1	Βασικές αρχές κατανάλωσης ισχύος του επεξεργαστή	156
7.1.1	Πηγές κατανάλωσης ισχύος	156
7.1.2	Γινόμενα Energy-Delay	159
7.1.3	Ενεργειακή απόδοση από πλευράς λογισμικού	159
7.2	Διελκυστίνα επίδοσης-ενέργειας στον πυρήνα SpMV	161
7.2.1	Πειραματική διαρρύθμιση	162
7.2.2	Χαρακτηρισμός των συμβιβασμών	163
7.3	Πρόβλεψη των βελτίστων ρυθμίσεων εκτέλεσης	168
7.3.1	Ομαδοποίηση των πινάκων	170
7.3.2	Κατασκευή του μετώπου Pareto της ομάδας	171
7.3.3	Κατηγοριοποίηση και αξιολόγηση	172
7.3.4	Περιορισμοί	176
7.4	Ανοιχτά θέματα	177
8	Συμπεράσματα	179
8.1	Ο πυρήνας SpMV: Θύμα της αλγοριθμικής του φύσης	179
8.2	CSX: Βιώσιμη λύση προς την υψηλή επίδοση του SpMV	181
8.3	Προς ένα ενεργειακά αποδοτικό πυρήνα SpMV	182
8.4	Μελλοντικές ερευνητικές κατευθύνσεις	183

Περιεχόμενα

Κατάλογος δημοσιεύσεων	185
Βιβλιογραφία	187
Ευρετήριο	199
Σύντομο Βιογραφικό Σημείωμα	203

Κατάλογος σχημάτων

1.1	Κατανομή του χρόνου εκτέλεσης της μη-preconditioned επαναληπτικής μεθόδου CG για διαφορετικές κατηγορίες προβλημάτων.	5
1.2	Οι τρέχουσες τάσεις στις σύγχρονες αρχιτεκτονικές υπολογιστών.	7
1.3	Επίδειξη της κλιμάκωσης του πυρήνα SpMV σε συνάρτηση με την κατανάλωση εύρους ζώνης μνήμης.	10
2.1	Το σχήμα αποθήκευσης Coordinate.	20
2.2	Το σχήμα αποθήκευσης Compressed Sparse Row.	21
2.3	Το σχήμα αποθήκευσης Blocked Compressed Sparse Row.	24
2.4	Το σχήμα αποθήκευσης Row Segmented Diagonal.	25
2.5	Το σχήμα αποθήκευσης Variable Block Length storage.	27
2.6	Το σχήμα αποθήκευσης Variable Block Row.	29
2.7	Το σχήμα αποθήκευσης Ellpack-Itpack, κατάλληλο για διανυσματικούς επεξεργαστές και σύγχρονες αρχιτεκτονικές γραφικών (GPUs).	32
2.8	Κωδικοποίηση run-length των στηλών του πίνακα.	34
2.9	Το σχήμα αποθήκευσης Symmetric Sparse Skyline.	35
2.10	Η εξάρτηση RAW στο διάνυσμα εξόδου κατά την παράλληλη εκτέλεση του συμμετρικού πυρήνα SpMV.	36
3.1	Τα μπλοκ διαγράμματα των πολυεπεξεργαστικών συστημάτων που χρησιμοποιήθηκαν για τις πειραματικές αξιολογήσεις σε αυτή την διατριβή (συνεχίζεται στην σελ. 47).	46
3.1	Τα μπλοκ διαγράμματα των πολυεπεξεργαστικών συστημάτων που χρησιμοποιήθηκαν για τις πειραματικές αξιολογήσεις σε αυτή την διατριβή.	47
3.2	Επιβαρύνσεις της επίδοσης του πυρήνα SpMV.	54
3.2	Επιβαρύνσεις της επίδοσης του πυρήνα SpMV.	55
3.3	Κλιμάκωση του πυρήνα SpMV στα συστήματα SMP.	56
3.4	Κλιμάκωση του πυρήνα SpMV στο σύστημα Dunnington χρησιμοποιώντας την πολιτική πλήρωσης των πυρήνων «share-all».	57

3.5	Η συμφόρηση στον δίαυλο μνήμης των αρχιτεκτονικών SMP. . . .	58
3.6	Η επίδοση του πυρήνα SpMV σε σχέση με την υπολογιστική ένταση (λόγος flops:byte).	59
3.7	Ανισορροπία στην κατανομή του φόρτου εργασίας του πυρήνα SpMV.	60
3.8	Μεταβολή του πίνακα flops:byte κατά μήκος του πίνακα.	61
3.9	Η ωφέλεια της αναδιάταξης του πίνακα.	62
3.10	Η κλιμάκωση του πυρήνα SpMV στις αρχιτεκτονικές NUMA. . .	63
3.11	Κορεσμός του δικτύου διασύνδεσης στις αρχιτεκτονικές NUMA. .	64
3.12	Διαφανής τοποθέτηση δεδομένων στις αρχιτεκτονικές NUMA. . .	66
3.13	Η επίδραση του διαμοιρασμού του διανύσματος εισόδου στις αρχιτεκτονικές NUMA.	67
4.1	Συσχέτιση του λόγου συμπίεσης και της βελτίωσης της επίδοσης. .	73
4.2	Κλιμάκωση της επίδοσης των σχημάτων αποθήκευσης σε μπλοκ. .	75
4.3	Η επίδραση της συμπίεσης.	76
4.4	Η επίδραση της τεχνικής «vectorization» στις αρχιτεκτονικές SMP. .	79
4.5	Η επίδραση της τεχνικής «vectorization» στις αρχιτεκτονικές NUMA. .	80
4.6	Ακρίβεια επιλογής των θεωρουμένων μοντέλων επίδοσης.	88
4.7	Ακρίβεια πρόβλεψης χρόνου εκτέλεσης των θεωρουμένων μοντέλων επίδοσης.	91
5.1	Το πλεονέκτημα της ανίχνευσης πολλαπλών κανονικοτήτων εντός του αραιού πίνακα.	95
5.2	Ο πίνακας byte ct1 που χρησιμοποιείται από το σχήμα CSX για την κωδικοποίηση της τοπολογικής πληροφορίας των μη μηδενικών στοιχείων του αραιού πίνακα.	98
5.3	Κωδικοποίηση του πίνακα από το CSX.	98
5.4	Η κωδικοποίηση ακεραίων μεταβλητού μήκους που χρησιμοποιείται από το σχήμα CSX.	99
5.5	Παράδειγμα της κωδικοποίησης run-length των στηλών.	100
5.6	Ανίχνευση 2-Δ κανονικοτήτων στο σχήμα CSX.	103
5.7	Οι κανονικότητες που ανιχνεύθηκαν και κωδικοποιήθηκαν από το CSX σε ένα ευρύ σύνολο αραιών πινάκων.	104
5.8	Η υποδομή της Just-In-Time συμπίλησης του σχήματος CSX.	110
5.9	Η δυνατότητα συμπίεσης του CSX.	117
5.10	Επιτάχυνση του CSX σε συστήματα SMP.	119
5.11	Η επιτάχυνση του CSX σε συστήματα SMP χρησιμοποιώντας την πολιτική πλήρωσης πυρήνων «share-nothing».	119
5.12	Επιτάχυνση του CSX στις αρχιτεκτονικές NUMA.	120
5.13	Ανά πίνακα επίδοση στο σύστημα Gainestown.	122

5.14	Η επίδοση των CSX και CSR σε πίνακες αναδιατεταγμένους κατά RCM.	126
5.15	Το κόστος προεπεξεργασίας του CSX.	127
5.16	Η ανανεωμένη διεπαφή πολλαπλασιασμού πίνακα-επί-διάνυσμα στο Elmer.	129
5.17	Διάρθρωση του χρόνου εκτέλεσης της βιβλιοθήκης του CSX. . . .	131
5.18	Επιτάχυνση του λογισμικού Elmer.	132
6.1	Η εξάρτηση RAW στο διάνυσμα εξόδου κατά την παράλληλη εκτέλεση του συμμετρικού πυρήνα SpMV.	135
6.2	Επιτάχυνση της απλοϊκής υλοποίησης του συμμετρικού SpMV. . .	136
6.3	Μέθοδοι τοπικών διανυσμάτων για την φάση αναγωγής του συμμετρικού πυρήνα SpMV.	139
6.4	Η πυκνότητα των ενεργών περιοχών των τοπικών διανυσμάτων. .	140
6.5	Η επιβάρυνση της φάσης αναγωγής.	141
6.6	Κωδικοποιούμενες υπό του CSX-Sym κανονικότητες σε ένα αραιό πίνακα 8×8	143
6.7	Επιτάχυνση του συμμετρικού SpMV με τις διαφορετικές μεθόδους τοπικών διανυσμάτων.	145
6.8	Διάρθρωση του χρόνου εκτέλεσης του συμμετρικού SpMV.	146
6.9	Επιτάχυνση του συμμετρικού SpMV με το σχήμα CSX-Sym.	148
6.10	Ανά πίνακα επίδοση του σχήματος CSX-Sym.	148
6.11	Ανά-πίνακα επίδοση επί των αναδιατεταγμένων αραιών πινάκων. .	150
6.12	Διάρθρωση εκτέλεσης της μεθόδου CG.	152
7.1	Συμβιβασμοί επίδοσης-κατανάλωσης ενέργειας του πυρήνα SpMV.	165
7.2	Οι βέλτιστοι συμβιβασμοί επίδοσης-κατανάλωσης ενέργειας δύο αραιών πινάκων.	169
7.3	Ομαδοποίηση πινάκων.	171
7.4	Κατασκευή του μετώπου Pareto της ομάδας.	172
7.5	Τα πρώτα τρία μέτωπα Pareto του πίνακα thermal2.	173
7.6	Προβλέψεις βελτίστων ρυθμίσεων εκτέλεσης.	175

Κατάλογος πινάκων

3.1	Η σουίτα πινάκων που χρησιμοποιήθηκε για την πειραματική αξιολόγηση.	45
3.2	Τεχνικά χαρακτηριστικά των πλατφόρμων υλικού που χρησιμοποιήθηκαν στις πειραματικές αξιολογήσεις.	48
3.3	Μετροπρογράμματα για την αξιολόγηση των προβλημάτων επίδοσης του πυρήνα SpMV.	52
4.1	Ακρίβεια επιλογής των θεωρουμένων μοντέλων επίδοσης.	87
4.2	Ακρίβεια πρόβλεψης χρόνου εκτέλεσης των θεωρουμένων μοντέλων επίδοσης του SpMV.	89
5.1	Τα κύρια χαρακτηριστικά των σημαντικότερων εναλλακτικών του CSR.	96
5.2	Οι εφαρμαζόμενοι από του CSX μετασχηματισμοί συντεταγμένων.	103
5.3	Σταθερότητα της επίδοσης του CSX.	123
5.4	Επίδοση του CSX έναντι του DMV.	124
5.5	Τα προβλήματα αναφοράς που χρησιμοποιήθηκαν για την αξιολόγηση της ενσωμάτωσης του CSX στο Elmer.	130
6.1	Βαθμός συμπίεσης του συμμετρικού σχήματος CSX.	144
6.2	Βελτίωση της επίδοσης του συμμετρικού SpMV λόγω αναδιάταξης του πίνακα.	149
7.1	Γεγονότα παρακολούθησης της επίδοσης που χρησιμοποιήθηκαν για την πρόβλεψη της κατανάλωσης ισχύος.	164
7.2	Αποτελέσματα cross-validation.	176
7.3	Ακρίβεια πρόβλεψης γινόμενων energy-delay.	176

Κατάλογος Αλγορίθμων

1.1	Αφ' υψηλού αναπαράσταση του πολλαπλασιασμού αραιού πίνακα με διάνυσμα.	10
2.1	Υλοποίηση του πυρήνα SpMV με χρήση του σχήματος COO. . . .	21
2.2	Υλοποίηση του πυρήνα SpMV kernel με χρήση του σχήματος CSR. . . .	22
2.3	Υλοποίηση του πυρήνα SpMV με χρήση του σχήματος BCSR. . . .	25
2.4	Υλοποίηση του πυρήνα SpMV με χρήση του σχήματος VBL. . . .	28
2.5	Υλοποίηση του πυρήνα SpMV με χρήση του συμμετρικού σχήματος αποθήκευσης SSS.	36
3.1	Υλοποίηση του πυρήνα SpMV χρησιμοποιώντας το σχήμα CSR (βλ. Κεφάλαιο 2, Ενότητα 2.1 για μία λεπτομερή περιγραφή). . . .	40
4.1	Υλοποίηση SIMD του πυρήνα BCSR για μπλοκ 1×2	78
4.2	Υλοποίηση SIMD του πυρήνα BCSR για μπλοκ 2×1	81
5.1	Κωδικοποίηση run-length των στηλών.	101
5.2	Ανίχνευση κανονικοτήτων στην εσωτερική αναπαράσταση του CSX.	102
5.3	Ανίχνευση, επιλογή και κωδικοποίηση των κανονικοτήτων στο σχήμα CSX.	106
5.4	Κατασκευή των δομών δεδομένων του CSX.	108
5.5	Βοηθητική συνάρτηση για την παραγωγή των μονάδων δέλτα. . . .	109
5.6	Το πρότυπο εκτέλεσης του πυρήνα SpMV με το σχήμα CSX. . . .	111
6.1	Παράλληλη υλοποίηση του συμμετρικού πυρήνα SpMV με χρήση του σχήματος αποθήκευσης SSS.	135
6.2	Ο αλγόριθμος Conjugate Gradient χωρίς την χρήση preconditioner.	151

Αντί Προλόγου

Πολλές οι σκέψεις και τα συναισθήματα που γεμίζουν το μυαλό μου φτάνοντας στο τέλος του «αγώνα» που λέγεται διδακτορικό. Ανακούφιση και ικανοποίηση για την τελειώσή του, προσδοκία για το μέλλον, νοσταλγία για τις όμορφες και δύσκολες στιγμές του. Κάθε τι που με τον ένα ή τον άλλο τρόπο σημαδεύει την πορεία σου είναι συνήθως άρρηκτα συνδεδεμένο με τον ανθρώπινο παράγοντα, άμεσα ή έμμεσα, διακριτικά ή εύγλωττα. Το διδακτορικό δεν θα μπορούσε να είναι εξαίρεση, όχι μόνο λόγω του πολυετούς του, αλλά και λόγω του δυναμικού και του φάσματος των συναισθημάτων που συχνά-πυκνά προκαλεί κατά την διάρκειά του. Δεν πρόκειται, λοιπόν, επ' ουδενί για ένα «one-man-show», αλλά αποτελεί την συνισταμένη της συνεισφοράς, υλικής, πνευματικής, ψυχολογικής, εμφανούς ή αφανούς, ενός συνόλου εξαιρετικών ανθρώπων.

Κατ' αρχήν, θα ήθελα να ευχαριστήσω τον επιβλέποντά μου, αναπληρωτή καθηγητή Νεκτάριο Κοζύρη, για την εμπιστοσύνη που μου έδειξε στο δύσκολο έργο του διδακτορικού, εν πρώτοις αποδεχόμενός με στο διδακτορικό πρόγραμμα, και εν συνεχεία, φροντίζοντας για την απρόσκοπτη και επί της ουσίας εκπόνηση της διατριβής μου, μέσω της συνεχούς επαφής ολόκληρης της ομάδας μας με την τελευταία τεχνολογία και της ενθάρρυνσης της ενεργού συμμετοχής μας στην ευρύτερη ερευνητική κοινότητα. Οφείλω, επίσης, την ευχαριστία μου και στους συνεπιβλέποντες, καθηγητή Ανδρέα-Γεώργιο Σταφυλοπάτη, για την στενή, αρμονική και εποικοδομητική συνεργασία μας κατά τα πρώτα χρόνια του διδακτορικού, και καθηγητή Παναγιώτη Τσανάκα.

Η διδακτορική αυτή διατριβή, όμως, ίσως να μην υπήρχε χωρίς την αμέριστη συμπαράσταση του Λέκτορα Γιώργου Γκούμα. Η συμβολή του Γιώργου ήταν καθοριστική, τόσο στο επίπεδο της οργάνωσης και της ερευνητικής ισχυροποίησης της ομάδας μας, όσο και συγκεκριμένα στην δική μου διατριβή. Από τα πρώτα μου βήματα στον απαιτητικό χώρο του High Performance Computing, ο Γιώργος βρισκόταν συνεχώς δίπλα μου, στις ατυχίες, που υπήρξαν αρκετές, και στις επιτυχίες, που υπήρξαν σημαντικές και ελπιδοφόρες για το μέλλον.

Θα ήταν μεγάλη παράλειψη στο σύντομο αυτό σημείωμα να μην αναφερθώ στα υπόλοιπα μέλη του εργαστηρίου, παλαιότερα και νεώτερα, που με το υψηλό τους επιστημονικό και πνευματικό επίπεδο δημιουργούν ένα σημαντικό ρεύμα ανόδου που συμπαρασύρει και εμπνέει κάθε νεοεισερχόμενο στην ομάδα. Δύσκολο να ξεχωρίσεις συγκεκριμένα πρόσωπα χωρίς ενδεχομένως να «αδικήσεις» κάποια άλλα, καθότι το ευχάριστο κλίμα εργασίας και συνεργασίας δημιουργήθηκε και συντηρείται συλλογικά. Από τον Βαγγέλη Κούκη, που απετέλεσε ένα σημαντικό έναυσμα ώστε να ενταχθώ στην ομάδα του CSLab¹, γνωρίζοντάς τον ως προπτυχιακός φοιτητής, τον Κορνήλιο Κούρτη, με τον οποίο συνεργαστήκαμε και με ενέπνευσε σε μεγάλο βαθμό ο τρόπος εργασίας του και αντίληψης των πραγμάτων, τον Γιώργο Τσουκαλά, τον Νίκο Αναστόπουλο, τον Κωστή Νίκα, τον Τάσο Νάνο, την Γεωργία Κουβέλη, τον Τάσο Κατσιγιάννη, τον Στέφανο Γεράγγελο, με τους οποίους περάσαμε όμορφες στιγμές τεχνικών ή μη αναλύσεων εντός και εκτός συνόρων, τον Θεοδωρή Γκούντουβα, ο οποίος συνέβαλε με την διπλωματική του εργασία στην βελτίωση και επέκταση καίριων σημείων της διατριβής μου, αλλά και τα υπόλοιπα μέλη των «Παραλλήλων» και «Κατανεμημένων», όλοι μαζί συνέβαλαν και συμβάλλουν στην δημιουργία του ευχάριστου κλίματος που δρα ως κινητήριο δύναμη για κάθε επιτυχία.

Οφείλω, όμως, να ευχαριστήσω θερμά και τους «εκτός του κάδρου» συμβάλλοντες, τους αεί φίλους μου από τα λυκειακά και προπτυχιακά χρόνια, Νίκο, Κώστα και Σωκράτη, που απετέλεσαν συχνά την διαφυγή και διέξοδο από την τύρβη της καθημερινότητας, και την κοπέλα μου, Ευανθία, που τα τελευταία χρόνια βρισκόταν συνεχώς δίπλα μου, μεταδίδοντάς μου την δύναμη και την αγάπη της, για να συνεχίσω στον δύσκολο δρόμο του διδακτορικού.

Τέλος, αλλά μάλλον ως ακρογωνιαίοι παρά ακροτελεύτιοι, οι γονείς μου, Κωνσταντίνος και Ειρήνη, και τα αδέρφια μου, Μάριος, Όλγα και Αλέξανδρος, αξίζουν την αμέριστη ευγνωμοσύνη και ευχαριστία μου για όλα όσα μου έχουν προσφέρει όλα αυτά τα χρόνια, υλικά, ηθικά, πνευματικά. Τους αφιερώνω αυτό το πόνημα.

Με βαθειά εκτίμηση,
Βασίλειος Κ. Καρακάσης

¹ Η διατριβή αυτή εκπονήθηκε στο Εργαστήριο Υπολογιστικών Συστημάτων του Τομέα Τεχνολογίας Πληροφορικής και Υπολογιστών της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσοβίου Πολυτεχνείου.

Εκτεταμένη Περίληψη

Η διατριβή αυτή εστιάζει στην βελτιστοποίηση του υπολογιστικού πυρήνα Πολλαπλασιασμού Αραιού Πίνακα με Διάνυσμα (SpMV). Ο πυρήνας SpMV αποτελεί βασικό συστατικό στοιχείο των επαναληπτικών μεθόδων επίλυσης αραιών γραμμικών συστημάτων. Η εκτέλεσή του είναι διαπιστευμένα ιδιαίτερα αργή και υπό συνθήκες μπορεί να αποτελέσει μέχρι και το 90% του συνολικού χρόνου εκτέλεσης της μεθόδου επίλυσης. Μάλιστα, χαρακτηρίστηκε πρόσφατα ως ένα από τα σημαντικότερα υπολογιστικά προβλήματα της δεκαετίας, των οποίων η βελτιστοποίηση αποτελεί σημαντική προτεραιότητα της κοινότητας των Υπολογιστικών Συστημάτων Υψηλών Επιδόσεων (High Performance Computing – HPC).

Το πρόβλημα επίδοσης του SpMV πηγάζει από τα εγγενή χαρακτηριστικά του αλγορίθμου. Πιο συγκεκριμένα, η έλλειψη χρονικής τοπικότητας των αναφορών στην μνήμη απαιτεί για κάθε χρήσιμη πράξη κινητής υποδιαστολής πολλές προσβάσεις στην ιεραρχία μνήμης (flop:byte ratio), με αποτέλεσμα η επίδοση του πυρήνα να εξαρτάται από την επίδοση του υποσυστήματος μνήμης. Με την έλευση των πολυπύρηνων αρχιτεκτονικών υπολογιστών, η πίεση στο υποσύστημα μνήμης αυξάνεται κατάκορυφα, καθώς όλα τα νήματα του πυρήνα ζητούν ταυτόχρονα δεδομένα, με αποτέλεσμα το διαθέσιμο εύρος ζώνης μνήμης να κοραίνεται και, τελικά, η συνολική επίδοση να εξαρτάται σχεδόν αποκλειστικά από το μέγεθος της αναπαράστασης του αραιού πίνακα στην κύρια μνήμη. Η διατριβή αυτή, σε πρώτη φάση, αναδεικνύει καθαρά και με σαφήνεια τα προβλήματα επίδοσης του πυρήνα SpMV. Η εικόνα της επίδοσης του συγκεκριμένου πυρήνα υπήρξε αρκετά συγκεχυμένη κατά το παρελθόν, με διάφορες εργασίες να αναδεικνύουν μία ποικιλία από πιθανές αιτίες για την ιδιαίτερα χαμηλή επίδοση, χωρίς όμως να τις ποσοτικοποιούν επακριβώς. Η διατριβή αυτή ξεκαθαρίζει όλα αυτά τα ζητήματα και ποσοτικοποιεί την επίδραση κάθε αλγοριθμικού χαρακτηριστικού του SpMV στην τελική επίδοσή του. Πιο συγκεκριμένα, αναδεικνύει την συμφόρηση στο υποσύστημα μνήμης ως το κυριώτερο πρόβλημα επίδοσης του SpMV, που πρέπει να αντιμετωπιστεί πρωτίστως και αποτελεσματικά για την επίτευξη υψηλής

επίδοσης.

Στην συνέχεια, μελετάται σε βάθος η συμπεριφορά των διαφόρων μεθόδων αποθήκευσης αραιών πινάκων σε μπλοκ και αναδεικνύεται με σαφήνεια η διεκκυστία μεταξύ του χρόνου που καταναλώνεται σε υπολογισμούς στον επεξεργαστή και του χρόνου που αντιστοιχεί σε λειτουργίες προσπέλασης της μνήμης. Οι μέθοδοι που οδηγούν σε μεγαλύτερη συμπίεση των δεδομένων του πίνακα θα πρέπει να αντιμετωπίσουν το αυξημένο κόστος της αποσυμπίεσης των δεδομένων (π.χ., πρόσβαση σε επιπλέον βοηθητικές δομές στην περίπτωση των σχημάτων αποθήκευσης σε μπλοκ), με αποτέλεσμα την σημαντική επιβάρυνση του χρόνου υπολογισμού σε μονονηματικές εκτελέσεις ή εκτελέσεις με μικρό αριθμό νημάτων, όπου το υποσύστημα μνήμης δεν έχει ακόμα κορεστεί. Αντιθέτως, καθώς ο αριθμός των νημάτων αυξάνεται και το εύρος ζώνης μνήμης εξαντλείται, το όφελος από την αυξημένη συμπίεση αρχίζει να γίνεται εμφανές, ενώ το επιπλέον κόστος αποσυμπίεσης καλύπτεται από τον αυξημένο πλέον χρόνο εξυπηρέτησης των προσβάσεων στην κύρια μνήμη. Ως αποτέλεσμα, οι μέθοδοι που οδηγούν σε αυξημένη συμπίεση «αποζημιώνουν» στις πολυνηματικές εκτελέσεις την υπολογιστική τους επιβάρυνση, οδηγώντας σε σημαντικά οφέλη επίδοσης. Εστιάζοντας περαιτέρω στις υλοποιήσεις των σχημάτων αποθήκευσης μπλοκ σταθερού μεγέθους, η διατριβή αυτή μελετά τα υπολογιστικά χαρακτηριστικά των διαφορετικών μπλοκ και πώς αυτά αλληλεπιδρούν με την υποκείμενη αρχιτεκτονική, επηρεάζοντας, τελικά, την επίδοση του πυρήνα. Αποτέλεσμα αυτής της μελέτης είναι η πρόταση μοντέλων επίδοσης, τα οποία λαμβάνουν υπόψη τους τόσο μέρος των προσβάσεων στην μνήμη όσο και το υπολογιστικό μέρος, με σκοπό την πρόβλεψη του μπλοκ που οδηγεί στην βέλτιστη επίδοση. Τα προτεινόμενα μοντέλα είναι ιδιαίτερα ακριβή και επιδεικνύουν μία σημαντική προσαρμοστικότητα στα διαφορετικά χαρακτηριστικά των υποκειμένων αρχιτεκτονικών υπολογιστών.

Έχοντας αναλύσει σε βάθος τόσο την υπολογιστική συμπεριφορά του SpMV όσο και των μεθόδων αποθήκευσης αραιών πινάκων σε μπλοκ, η διατριβή αυτή προτείνει ένα νέο σχήμα αποθήκευσης, ονόματι Compressed Sparse eXtended (CSX). Το σχήμα αποθήκευσης CSX εστιάζει με άμεσο τρόπο στην ελαχιστοποίηση του μεγέθους της αναπαράστασης του αραιού πίνακα. Πιο συγκεκριμένα, βασισμένο πάνω σε τεχνικές συμπίεσης δεδομένων, όπως κωδικοποίηση-Δ (delta encoding) και Run-Length Encoding (RLE), επιδιώκει την μείωση του όγκου των μεταδεδομένων (δείκτες γραμμής, αριθμοί στηλών) που απαιτούνται για την αποθήκευση ενός αραιού πίνακα κατά τον πιο διαδεδομένο τρόπο (μορφή αποθήκευσης Compressed Sparse Row (CSR)). Το σχήμα CSX μπορεί να ανιχνεύει και να κωδικοποιεί ένα σημαντικό αριθμό διαφορετικών κανονικότητας μη μηδενικών στοιχείων που εμφανίζονται στον αραιό πίνακα, όπως οριζόντιες, κατακόρυφες, διαγώνιες, δισδιάστατες κανονικότητες κ.ά. Για κάθε μία από αυτές τις κανονικότητες, το σχήμα CSX αποθηκεύει μόνο ένα σύ-

ντομο περιγραφητή (δύο byte) αντί για κάθε αριθμό στήλης των μη μηδενικών στοιχείων ξεχωριστά, με αποτέλεσμα να επιτυγχάνει σημαντική συμπίεση των μεταδεδωμένων του πίνακα. Σε περιπτώσεις μη ανίχνευσης κανονικότητας (π.χ., πολύ αραιός πίνακας με μη-μηδενικά στοιχεία σε ακανόνιστη διάταξη), το σχήμα CSX χρησιμοποιεί κωδικοποίηση-Δ, ώστε να μειώσει το μέγεθος των μεταδεδωμένων και σε αυτή την περίπτωση. Συνολικά, το σχήμα CSX επιτυγχάνει επίπεδα συμπίεσης κοντά στο θεωρητικό μέγιστο, με αποτέλεσμα να οδηγεί σε σημαντική βελτίωση της επίδοσης του πυρήνα SpMV. Ειδικότερα σε αρχιτεκτονικές συμμετρικής πρόσβασης στην κύρια μνήμη (SMP), οι οποίες πάσχουν περισσότερο από το πρόβλημα του περιορισμένου εύρους ζώνης μνήμης, το σχήμα CSX επιτυγχάνει πάνω από 50% βελτίωση της επίδοσης του SpMV κατά μέσο όρο. Σε αρχιτεκτονικές μη ομοιόμορφης πρόσβασης στην κύρια μνήμη (NUMA), όπου η συμφόρηση στην πρόσβαση στην κύρια μνήμη είναι σαφώς μικρότερη, μειώνουμε τον βαθμό συμπίεσης των δεδομένων, ώστε να αποφεύγεται το κόστος αποσυμπίεσης, ενώ παράλληλα, φροντίζουμε, ώστε οι επιλεγόμενες προς κωδικοποίηση κανονικότητες να είναι οι περισσότερο φιλικές προς τους υπολογισμούς. Με αυτό τον τρόπο, το σχήμα CSX επιτυγχάνει μία επιτάχυνση του πυρήνα SpMV μεγαλύτερη του 20% κατά μέσο όρο σε αυτές τις αρχιτεκτονικές.

Στην διατριβή αυτή δίνεται ιδιαίτερο βάρος στην ελαχιστοποίηση του χρόνου προεπεξεργασίας του πίνακα κατά την δημιουργία του σχήματος CSX. Το βασικό αρχικό μειονέκτημα του CSX ήταν ο σημαντικός χρόνος προεπεξεργασίας του πίνακα που απαιτείτο στην περίπτωση της κωδικοποίησης μη οριζόντιων κανονικοτήτων. Παραταύτα, χρησιμοποιώντας στατιστικές τεχνικές για τον δειγματισμό του πίνακα, καταφέραμε την σημαντική μείωση του χρόνου προεπεξεργασίας με σχεδόν μηδαμινό κόστος στην τελική επίδοση του πυρήνα SpMV. Ο χρόνος προεπεξεργασίας ανέρχεται πλέον σε μερικές δεκάδες πράξεις SpMV, πράγμα που καθιστά την χρήση του CSX χωρίς την offline μετάθεση της προεπεξεργασίας. Πράγματι, στο πλαίσιο της διατριβής αυτής, ενσωματώσαμε το σχήμα CSX στο πρόγραμμα προσομοίωσης Elmer και καταφέραμε να επιταχύνουμε μέχρι και 40% τον χρόνο εκτέλεσης του SpMV, παρά τον αρχικό κόστος «εκκίνησης» του CSX, και τον συνολικό χρόνο εκτέλεσης του επιλυτή κατά 15% κατά μέσο όρο.

Επεκτείνουμε, στη συνέχεια, το σχήμα CSX ώστε να υποστηρίζει αποδοτικά και συμμετρικούς πίνακες. Η συμμετρία αποτελεί ένα δελεαστικό χαρακτηριστικό για τον πυρήνα SpMV, μιας και δίνει την δυνατότητα περαιτέρω μείωσης του μεγέθους της αναπαράστασης του πίνακα, αποφεύγοντας την αποθήκευση των μη μηδενικών στοιχείων του άνω τριγωνικού υποπίνακα. Παραταύτα, η πολυνηματική εκτέλεση του συμμετρικού πυρήνα SpMV εισάγει εξαρτήσεις δεδομένων με συγκεκριμένα στοιχεία του διανύσματος εξόδου ανάλογα με την κατανομή των μη μηδενικών του πίνακα εισόδου. Η πιο συνήθης τα-

κτική αποφυγής αυτή της εξάρτησης είναι η χρήση επιμέρους τοπικών διανυσμάτων εξόδου ανά νήμα, τα οποία στο τέλος του υπολογισμού προστίθενται στο τελικό διάνυσμα εξόδου. Δυστυχώς, η μέθοδος αυτή εισάγει μία σημαντική επιβάρυνση, μιας και η πράξη της άθροισης διανυσμάτων είναι και αυτή ιδιαίτερα memory-intensive. Επιπλέον, η επιβάρυνση αυτή αυξάνεται γραμμικά με το πλήθος των νημάτων εκτέλεσης, με αποτέλεσμα, πολύ σύντομα, το υποσύστημα μνήμης να φτάνει σε κορεσμό και η επίδοση του πυρήνα να μην μπορεί να κλιμακωθεί. Η μέθοδος που προτείνεται για την επίλυση αυτού του προβλήματος βασίζεται στην παρατήρηση ότι τα τοπικά διανύσματα εισόδου γίνονται ολοένα και πιο αραιά, καθώς αυξάνονται τα νήματα υπολογισμού. Προτείνεται, επομένως, η χρήση μίας δομής δεικτοδότησης των επιμέρους διανυσμάτων, η οποία υποδεικνύει μόνο τα στοιχεία με εξαρτήσεις δεδομένων. Κατά την διάρκεια υπολογισμού του SpMV τα στοιχεία με εξαρτήσεις ανανεώνονται στα επιμέρους διανύσματα, ενώ τα υπόλοιπα ανανεώνονται απευθείας στο τελικό διάνυσμα εξόδου. Κατά την τελική φάση της άθροισης των επιμέρους διανυσμάτων, αθροίζονται μόνο τα στοιχεία με εξαρτήσεις, μειώνοντας έτσι σημαντικά τον φόρτο εργασίας. Ως αποτέλεσμα, ο συμμετρικός πυρήνας SpMV μπορεί να κλιμακωθεί. Η τεχνική αυτή σε συνδυασμό με την παραλλαγή του CSX για συμμετρικούς πίνακες επιταχύνει μέχρι και δύο φορές τον υπολογισμό του SpMV σε σχέση με το συμβατικό σχήμα αποθήκευσης CSR.

Τέλος, στο πλαίσιο αυτή της διδακτορικής διατριβής μελετώνται, σε αρχικό στάδιο, ζητήματα ενεργειακής απόδοσης του πυρήνα SpMV και πώς αυτά διασυνδέονται με το επίπεδο της επίδοσης. Πιο συγκεκριμένα, μελετάται η αλληλεπίδραση της επίδοσης του SpMV με την δυναμική κατανάλωση ενέργειας του επεξεργαστή, καθώς μεταβάλλονται η συχνότητα λειτουργίας του και η τοποθέτηση των νημάτων υπολογισμού στους πυρήνες του επεξεργαστή. Αναδεικνύονται διαφορετικές συμπεριφορές ανάλογα με την δομή και το μέγεθος του πίνακα εισόδου και μελετώνται σε μεγαλύτερο βάθος οι ισορροπίες (tradeoffs) επίδοσης-κατάναλωσης ενέργειας. Βασιζόμενοι σε απλές τεχνικές μηχανικής μάθησης, προτείνεται μεθοδολογία για την αναγνώριση των ιδανικών ισορροπιών βάσει των κύριων χαρακτηριστικών του πίνακα εισόδου. Η προτεινόμενη μέθοδος δίνει αρκετά ακριβή αποτελέσματα και αποτελεί ένα πρώτο βήμα στην κατεύθυνση της συνδυασμένης αποδοτικής υλοποίησης του SpMV.

Εισαγωγή

Η έρευνα σχετικά με τους αραιούς πίνακες είναι ενεργή από τις πρώτες ημέρες των υπολογιστικών συστημάτων. Η χρήση τους είναι στενά συνδεδεμένη με την επίλυση μεγάλων αραιών γραμμικών συστημάτων, των οποίων οι μαθηματικές και υπολογιστικές ιδιότητες, καθώς επίσης και οι δυνατότητες βελτιστοποίησής τους, κεντρίζουν ακόμα και σήμερα το ενδιαφέρον των ερευνητικών κοινοτήτων των εφαρμοσμένων μαθηματικών και των υπολογιστών. Η έλευση των συγχρόνων πολυπύρηνων αρχιτεκτονικών υπολογιστών έχει θέσει νέες προκλήσεις στην βελτιστοποίηση πυρήνων αραιών πινάκων, οι οποίοι τώρα είναι ακόμα πιο σημαντικοί για την αποδοτική εκτέλεση της επίλυσης μεγάλων αραιών γραμμικών συστημάτων. Σε αυτό το κεφάλαιο παρουσιάζουμε σύντομα τις διάφορες μεθόδους για την επίλυση μεγάλων αραιών γραμμικών συστημάτων και σκιαγραφούμε την σπουδαιότητα του υπολογιστικού πυρήνα του πολλαπλασιασμού αραιού πίνακα με διάνυσμα, ο οποίος απετέλεσε και το κυρίως κίνητρο γι' αυτή την εργασία. Συζητούμε επιπλέον τις κυριώτερες προκλήσεις που θέτουν οι σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών στην εκτέλεση αυτού του πυρήνα και παρουσιάζουν την δική μας προσέγγιση για την βελτιστοποίησή του.

1.1 Αραιά γραμμικά συστήματα

Οι αραιοί πίνακες είναι πεπερασμένοι πίνακες που κυριαρχούνται από μηδενικά στοιχεία. Αυτοί πίνακες προκύπτουν συνήθως κατά την διακριτοποίηση μερικών διαφορικών εξισώσεων στις μεθόδους πεπερασμένων στοιχείων (finite element methods – FEM) και συμμετέχουν συνήθως στην επίλυση μεγάλων γραμμικών συστημάτων της μορφής

$$Ax = b, \quad (1.1)$$

όπου A είναι ένας $n \times n$ πίνακας συντελεστών, b είναι το εκ δεξιών διάνυσμα και x είναι το διάνυσμα αγνώστων. Υπάρχουν δύο μεγάλες κατηγορίες επίλυσης γραμμικών συστημάτων: άμεσες και επαναληπτικές μέθοδοι επίλυσης.

Οι άμεσες μέθοδοι επίλυσης υπολογίζουν την ακριβή λύση ενός γραμμικού συστήματος, παραγοντοποιώντας τον πίνακα A , δηλ., εκφράζοντάς τον ως ένα γινόμενο δύο ή περισσότερων πινάκων, και εν συνεχεία βασιζόμενες στην ευθεία και όπισθεν αντικατάσταση υπολογίζουν την τιμή του αγνώστου διανύσματος x . Στις πιο γνωστές άμεσες μεθόδους επίλυσης περιλαμβάνονται οι μέθοδοι Gauss-Jordan, LU και Cholesky [Duff et al., 1989; Barrett et al., 1987; Davis, 2006]. Το υπολογιστικό κόστος αυτών των μεθόδων είναι ανάλογο του κόστους του πολλαπλασιασμού πίνακα-επί-πίνακα [Strassen, 1969], το οποίο μπορεί να γίνει απαγορευτικό για μεγάλα αραιά συστήματα. Για τους αραιούς πίνακες, ωστόσο, το κόστος αυτό είναι της τάξης μεγέθους των μη μηδενικών στοιχείων του πίνακα, αλλά σε αυτή την περίπτωση οι υλοποιήσεις των αμέσων μεθόδων δεν είναι τόσο εύκολες όσο στην περίπτωση των πυκνών πινάκων. Κατά την διάρκεια της παραγοντοποίησης του αρχικού πίνακα, νέα μη μηδενικά στοιχεία (στοιχεία fill-in) μπορεί να εισαχθούν σε σημεία που αρχικά υπήρχαν μηδενικά στοιχεία. Αυτό δυσκολεύει τόσο τις δομές αποθήκευσης του πίνακα όσο και την ίδια την διαδικασία της παραγοντοποίησης. Η αποθήκευση των αραιών πινάκων θα πρέπει να υποστηρίζει αποδοτικά την εισαγωγή των στοιχείων fill-in, ενώ η διαδικασία της παραγοντοποίησης θα πρέπει να προηγείται από ένα βήμα ελαχιστοποίησης του fill-in, με σκοπό την συνολική μείωση της επιβάρυνσης της εισαγωγής νέων στοιχείων [Saad, 2003]. Το αυξημένο κόστος των μεθόδων άμεσης επίλυσης ευνοεί την χρήση προσεγγιστικών επαναληπτικών μεθόδων για την επίλυση μεγάλων αραιών γραμμικών συστημάτων.

Οι επαναληπτικές μέθοδοι επίλυσης δεν υπολογίζουν την ακριβή λύση ενός γραμμικού συστήματος, αλλά αντιθέτως, προσπαθούν να υπολογίσουν μία καλή προσέγγιση της λύσης. Οι μέθοδοι αυτοί μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες [Barrett et al., 1987; Saad, 2003]:

- (α') τις *στατικές μεθόδους* (stationary methods) και
- (β') τις *προβολικές μεθόδους* (projection methods).

Οι στατικές μέθοδοι ξεκινούν από μία αρχική προσέγγιση x_0 της λύσης του συστήματος, την οποία στην συνέχεια βελτιώνουν επαναληπτικά προσπαθώντας να εκμηδενίσουν ένα ή περισσότερα στοιχεία του υπολειπόμενου διανύσματος (residual vector) $b - Ax$ κάθε φορά. Στην γενική της μορφή, μία στατική μέθοδος μπορεί να γραφεί ως εξής [Barrett et al., 1987]:

$$x^{(k+1)} = Bx^{(k)} + c \quad (1.2)$$

Σε κάθε επανάληψη k , η επόμενη προσέγγιση $x^{(k+1)}$ της λύσης υπολογίζεται πολλαπλασιάζοντας την τρέχουσα προσέγγιση με ένα αραιό πίνακα συντελεστών B , που έχει προκύψει από τον αρχικό πίνακα A . Οι πιο γνωστές στατικές μέθοδοι είναι οι Jacobi, Gauss-Seidel και η Successive Overrelaxation (SOR). Η

σύγκλιση αυτών των μεθόδων επαφίεται σε αυστηρές προδιαγραφές της δομής του αρχικού πίνακα και, γι' αυτό τον λόγο, δεν μπορεί να εγγυηθεί για όλους τους πίνακες [James και Riha, 1975; Ferziger και Peric, 2001; Saad, 2003].

Οι προβολικές μέθοδοι προσπαθούν να εξάγουν μία προσεγγιστική λύση \tilde{x} του συστήματος από τον υποχώρο \mathcal{K} του \mathbb{R}^n , απαιτώντας το υπολειπόμενο διάνυσμα $r = b - A\tilde{x}$ να κάθετο σε ένα υποχώρο \mathcal{L} του \mathbb{R}^n . Ξεκινώντας από μία αρχική εκτίμηση x_0 , οι μέθοδοι αυτές προχωρούν σε κάθε επανάληψη επιλέγοντας ένα ζεύγος υποχώρων \mathcal{K} και \mathcal{L} και προχωρούν το x στον \mathcal{K} , ώστε το νέο υπολειπόμενο διάνυσμα να είναι κάθετο στον \mathcal{L} . Στην πράξη, οι μέθοδοι αυτές μπορεί να θεωρηθούν ως μία γενίκευση της μεθόδου *steepest descent*, όπου σε κάθε επανάληψη η επιλεγόμενη λύση είναι εκείνη που ελαχιστοποιεί την απόσταση από το εκ δεξιών διάνυσμα b . Πράγματι, η μέθοδος *steepest descent* μπορεί να ληφθεί θέτοντας $\mathcal{K} = \mathcal{L} = r$ [Saad, 2003].

Οι πιο επιτυχημένες και διαδεδομένες προβολικές μέθοδοι είναι μέθοδοι των υποχώρων Krylov [Saad, 1981]. Οι μέθοδοι αυτές βασίζονται στους υποχώρους Krylov για την επιλογή της προσεγγιστικής λύσης σε κάθε επανάληψη. Ένας υποχώρος Krylov ορίζεται ως

$$\mathcal{K}_m(A, v) = \text{span}\{v, Av, A^2v, \dots, A^{m-1}v\}, \quad (1.3)$$

όπου ο τελεστής span υποδηλώνει το σύνολο όλων των γραμμικών συνδυασμών των διανυσματικών ορισμάτων του. Σε κάθε επανάληψη m , μία μέθοδος Krylov επιλέγει μία λύση από τον υποχώρο $\mathcal{K}_m(A, r_0)$, όπου A είναι ο πίνακας συντελεστών του συστήματος και r_0 το υπολειπόμενο διάνυσμα της αρχικής εκτίμησης x_0 . Μία μέθοδος Krylov μπορεί να θεωρηθεί ως μία πολυωνυμική προσέγγιση του A^{-1} . Πράγματι, εάν η μέθοδος συγκλίνει ύστερα από m επαναλήψεις σε μία λύση \tilde{x} και $x_0 = 0$, τότε πρέπει να ισχύουν οι παρακάτω εξισώσεις:

$$x \approx \tilde{x} \quad (1.4)$$

$$A^{-1}b \approx \left(\sum_{i=1}^{m-1} \alpha_i A^i \right) b \quad (1.5)$$

Το μειονέκτημα των επαναληπτικών μεθόδων συγκριτικά με τις άμεσες είναι η έλλειψη σταθερότητας. Η σύγκλιση των στατικών μεθόδων δεν μπορεί να εγγυηθεί για όλους τους πίνακες, ενώ ο ρυθμός σύγκλισης των προβολικών μεθόδων μπορεί να είναι πολύ αργός για να είναι πρακτικός. Για τον λόγο αυτό, είναι επιθυμητή η μετατροπή του αρχικού προβλήματος σε ένα ισοδύναμο με καλύτερα χαρακτηριστικά σύγκλισης. Η διαδικασία αυτή είναι γνωστή ως *preconditioning* ενός γραμμικού συστήματος.

Οι *preconditioned* Krylov μέθοδοι είναι αυτή τη στιγμή οι πιο διαδεδομένες μέθοδοι επίλυσης αραιών γραμμικών συστημάτων. Οι μέθοδοι αυτοί περι-

λαμβάνουν ένα μικρό και σαφώς ορισμένο σύνολο από υπολογιστικούς πυρήνες, πράγμα που ευνοεί τις αποδοτικές παράλληλες υλοποιήσεις. Ανάμεσα στις πιο γνωστές μεθόδους Krylon είναι η μέθοδος Generalized Minimum Residual (GMRES) και η μέθοδος Conjugate Gradient (CG), μαζί με τις παραλλαγές της, Bi-CG and Bi-CG Stabilized [Saad και Schultz, 1986; Hestenes και Stiefel, 1952; van der Vorst, 1992].

1.2 Οι επαναληπτικές μέθοδοι από υπολογιστικής απόψεως

Οι μέθοδοι Krylon για την επίλυση αραιών γραμμικών συστημάτων περιλαμβάνουν την εκτέλεση των ακόλουθων υπολογιστικών πράξεων [Saad, 2003; Hoemmen, 2010]:

(α') *Ανανέωση διανυσμάτων*. Η ανανέωση ενός διανύσματος είναι ένας συνδυασμός από βαθμωτό πολλαπλασιασμό και διανυσματική πρόσθεση. Δεδομένων δύο διανυσμάτων, x and y , και ενός βαθμωτού μεγέθους α , η πράξη της ανανέωσης διανύσματος είναι της μορφής

$$x = x + \alpha y.$$

Η πράξη αυτή είναι επίσης γνωστή και ως AXPY από τις αντίστοιχες υπορουτίνες της βιβλιοθήκης BLAS [Lawson et al., 1979].

(β') *Σημειακό γινόμενο*. Το σημειακό γινόμενο είναι το εσωτερικό γινόμενο δύο διανυσμάτων. Δεδομένων δύο διανυσμάτων x and y με n στοιχεία, το σημειακό γινόμενο υπολογίζεται ως

$$t = x^T y = \sum_{i=1}^n x_i y_i.$$

(γ') *Γινόμενο πίνακα επί διάνυσμα*. Πρόκειται για το γινόμενο ενός αραιού πίνακα A με ένα διάνυσμα x :

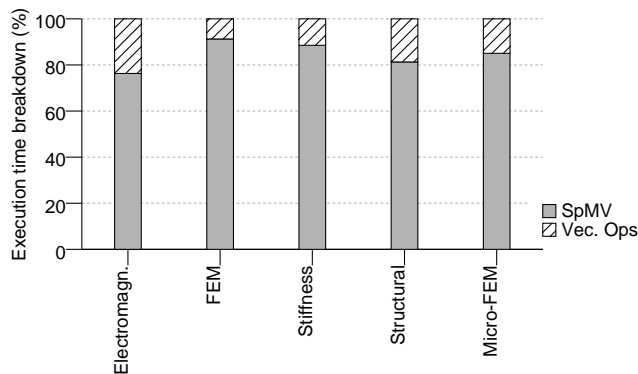
$$y = Ax.$$

Ο αραιός πίνακας A είναι ο πίνακας συντελεστών του συστήματος ή ο μετασχηματισμένος πίνακας συντελεστών στην περίπτωση των preconditioned Krylon μεθόδων.

(δ') *Πράξεις preconditioning*. Οι πράξεις αυτές σχετίζονται με την διαδικασία του preconditioning του συστήματος και συνήθως εμπεριέχουν την άμεση επίλυση ενός «εύκολου» γραμμικού συστήματος.

Κάθε μία από τις παραπάνω καλώς ορισμένες πράξεις αποτελεί ένα υπολογιστικό πυρήνα. Ο πιο χρονοβόρος πυρήνας σε μία μέθοδο Krylon είναι ο πυρήνας του Πολλαπλασιασμού Αραιού Πίνακα με Διάνυσμα, ή πιο απλά SpMV.

1.2. Οι επαναληπτικές μέθοδοι από υπολογιστικής απόψεως



Σχήμα 1.1: Κατανομή του χρόνου εκτέλεσης της μη-preconditioned επαναληπτικής μεθόδου CG για διαφορετικές κατηγορίες προβλημάτων.

Στο Σχήμα 1.1 παρουσιάζεται η κατανομή του χρόνου εκτέλεσης μία τυπικής υλοποίησης της μεθόδου CG σε μία σύγχρονη πολυπύρρηνη αρχιτεκτονική για διαφορετικούς πίνακες συντελεστών. Η πλειονότητα του χρόνου εκτέλεσης, ξεπερνώντας το 90% σε μερικές περιπτώσεις, καταναλώνεται στον πυρήνα SpMV. Η σπουδαιότητα του πυρήνα αυτού, επομένως, καθίσταται καθοριστικής σημασίας για την επιτάχυνση των επαναληπτικών μεθόδων επίλυσης, ενώ έχει χαρακτηριστεί πρόσφατα ως ένα από τα υπολογιστικά προβλήματα, των οποίων η βελτιστοποίηση θα παίξει σημαντικό ρόλο στον τομέα των επιστημονικών υπολογισμών την τρέχουσα δεκαετία [Colella, 2004; Asanovic et al., 2006].

Παραταύτα, η σπουδαιότητα των πράξεων preconditioning και των σημειακών γινομένων δεν θα πρέπει να υποεκτιμάται. Ένας ακριβός preconditioner, π.χ., η ατελής παραγοντοποίηση LU (ILU), μπορεί να καταναλώσει την πλειονότητα του χρόνου εκτέλεσης μίας επαναληπτικής μεθόδου. Παρόλα αυτά, καθώς οι preconditioners είναι στην πράξη άμεσες μέθοδοι επίλυσης, δεν συνδέονται άμεσα με την ίδια την επαναληπτική διαδικασία επίλυσης, όπως για παράδειγμα το πυρήνας SpMV και τα σημειακά γινόμενα, αντιθέτως, είναι κομμάτι του σημαντικού προβλήματος των αραιών μεθόδων επίλυσης για αραιούς πίνακες [Asanovic et al., 2006]. Επιπλέον, σε αντίθεση με τον πυρήνα SpMV και τις διανυσματικές πράξεις, δεν υπάρχουν πολύ αυστηρές προϋποθέσεις όσον αφορά στην αριθμητική ακρίβεια των υπολογισμών των preconditioners, επιτρέποντας επομένως αποδοτικές υλοποιήσεις με χαμηλότερη ακρίβεια [Buttari et al., 2008].

Τα σημειακά γινόμενα μπορούν να αποτελέσουν και αυτά σημείο συμφόρησης της εκτέλεσης σε ορισμένες περιπτώσεις, ιδιαίτερα σε παράλληλες εκτελέσεις. Η πράξη της αναγωγής (reduction) που απαιτείται για τον υπολογισμό

του τελικού αποτελέσματος έχει περιορισμένη παραλληλία, λογαριθμική του μεγέθους των διανυσμάτων n . Παραταύτα, καθώς το μέγεθος εισόδου αυξάνεται, ανεπιθύμητες επιπτώσεις μπορούν να προκύψουν, όπως αστοχίες στην κρυφή μνήμη ή αυξημένο κόστος επικοινωνίας, οι οποίες μπορούν να εμποδίσουν την εκμετάλλευση του διαθέσιμου παραλληλισμού.

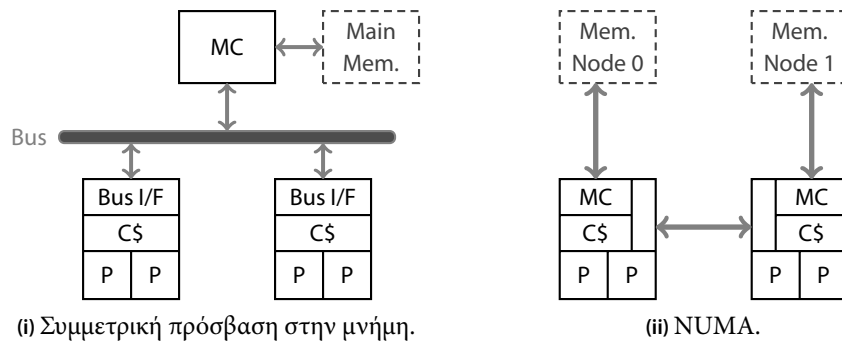
Η υλοποίηση επαναληπτικών μεθόδων υψηλών επιδόσεων για αραιούς πίνακες είναι όντως μία εξαιρετικά προκλητική διαδικασία με πολλαπλές όψεις, που απαιτούν προσεκτική και εις βάθος μελέτη. Η έλευση των πολυπύρηνων αρχιτεκτονικών υπολογιστών εισάγει επιπλέον προκλήσεις για την βελτιστοποίηση αυτών των μεθόδων, οι οποίες θα πρέπει να αντιμετωπισθούν επιτυχώς, ώστε να η υπολογιστική χωρητικότητα των νέων αρχιτεκτονικών να μπορεί να εκμεταλλευθεί.

1.3 Προκλήσεις των πολυπύρηνων αρχιτεκτονικών

Δεν έχει περάσει πολύς καιρός από την εποχή που η ιδέα για ολοένα και υψηλότερες συχνότητες στους επεξεργαστές εγκαταλήφθηκε από τους κύριους κατασκευαστές ολοκληρωμένων κυκλωμάτων, λόγω αφενός της ανάγκης για μεγαλύτερη ενεργειακή απόδοση και αφετέρου της ανάγκης για να διατηρηθεί την ίδια στιγμή ζωντανός ο νόμος του Moore. Το ενδιαφέρον της ερευνητικής και βιομηχανικής κοινότητας έχει από τότε μετατοπιστεί προς την ιδέα της ενσωμάτωσης πολλαπλών πυρήνων εντός του ίδιου φυσικού επεξεργαστή, εγκαινιάζοντας την εποχή των *πολυπύρηνων επεξεργαστών* και εισάγοντας νέες προκλήσεις. Η χρήση πολλαπλών πυρήνων ή νημάτων υλικού (hardware threads) μέσα στον ίδιο φυσικό επεξεργαστή επεξέτεινε το κενό μεταξύ του ρυθμού που ο επεξεργαστής μπορεί πλέον να καταναλώνει δεδομένα και του ρυθμού, με τον οποίο η κύρια μνήμη μπορεί να τα παρέχει, καθιστώντας το πρόβλημα του «memory wall» [Wulf και McKee, 1995] ακόμα πιο έντονο.

Οι αρχιτεκτονικές συμμετρικής πρόσβασης στην κύρια μνήμη (symmetric shared memory architectures) επηρεάζονται περισσότερο από την διαφορά ταχύτητας μνήμης-επεξεργαστή. Σε αυτές τις αρχιτεκτονικές, όλες οι αιτήσεις για πρόσβαση στην κύρια μνήμη και για επικοινωνία μεταξύ των επεξεργαστών δρομολογούνται μέσα από τον κοινό «εμπρόσθιο» δίαυλο (front-end bus) στον κεντρικό, εκτός τσιπ, ελεγκτή μνήμης και στους υπόλοιπους ομότιμους επεξεργαστές (Σχήμα 1.2i). Είναι φανερό ότι αυτή η κεντροκοποιημένη λογική, σε συνδυασμό με το χαμηλό εύρος ζώνης μνήμης και την συνακόλουθη υψηλή καθυστέρηση της εκτός-τσιπ επικοινωνίας, μπορεί εύκολα να γίνει το σημείο συμφόρησης μίας απαιτητικής σε εύρος ζώνης μνήμης εφαρμογής. Οι μεγάλες και σύνθετες ιεραρχίες κρυφής μνήμης, δυστυχώς, περιορίζουν αυτή την επίδραση μόνο βραχυπρόθεσμα.

1.3. Προκλήσεις των πολυπύρηνων αρχιτεκτονικών



Σχήμα 1.2: Οι δύο τρέχουσες τάσεις στις σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών: αρχιτεκτονικές συμμετρικής πρόσβασης στην μνήμη και αρχιτεκτονικές μη ομοιόμορφης πρόσβασης (NUMA).

Η ανάγκη της εξαγωγής περισσότερου παραλληλισμού από το υλικό, δεδομένου του χαμηλού ρυθμού εξέλιξης της ταχύτητας της κύριας μνήμης, απαιτεί μία λιγότερο κεντρική προσέγγιση. Οι αρχιτεκτονικές μη ομοιόμορφης πρόσβασης στην κύρια μνήμη (Non-Uniform Memory Access – NUMA) «μετατοπίζουν» των ελεγκτή μνήμης εντός του τσιπ του επεξεργαστή και χρησιμοποιούν εξειδικευμένο υλικό για την επικοινωνία μεταξύ των επεξεργαστών (Σχήμα 1.2ii). Η κύρια μνήμη, μολονότι κοινή, δεν είναι πια ομοιόμορφα προσπελάσιμη από κάθε επεξεργαστή του συστήματος: χωρίζεται σε πολλαπλούς κόμβους, που κάθε ένας ανατίθεται σε έναν επεξεργαστή. Το διαθέσιμο εύρος ζώνης είναι πλέον επαρκές για την επικοινωνία μεταξύ ενός επεξεργαστή και του τοπικού του κόμβου μνήμης, αλλά η πρόσβαση σε απομακρυσμένους κόμβους απαιτεί περισσότερα και πιο χρονοβόρα βήματα. Δύο προκλήσεις αναδεικνύονται με τις αρχιτεκτονικές NUMA για τους απαιτητικούς σε εύρος ζώνης πυρήνες:

- (α') Η αύξηση του διαθέσιμου εύρους ζώνης μνήμης μπορεί να αποκαλύψει αδυναμίες στο υπολογιστικό τμήμα του πυρήνα, οι οποίες σε άλλες περιπτώσεις θα κρύβονταν από την πολύ αργή πρόσβαση στην κύρια μνήμη.
- (β') Η επίδοση του πυρήνα μπορεί πλέον να είναι πολύ ευαίσθητη στην σωστή τοποθέτηση των δεδομένων του στους διαφορετικούς κόμβους μνήμης.

Η τελευταία πρόκληση προσθέτει ένα επιπλέον βάρος στον προγραμματιστή, ο οποίος μπορεί να χρειαστεί να τροποποιήσει σημαντικά τον πυρήνα, ώστε να μπορέσει να εκμεταλλευτεί πλήρως τις δυνατότητες NUMA ενός συστήματος.

Αυτές οι προκλήσεις θα πρέπει να αντιμετωπισθούν επιτυχώς, ώστε να γίνει εφικτή η υλοποίηση επαναληπτικών μεθόδων υψηλών επιδόσεων για τις μελλοντικές αρχιτεκτονικές υπολογιστών. Πιστεύουμε ότι η ισορροπία μεταξύ

επικοινωνίας¹ και υπολογιστικού φόρτου θα γίνεται ολοένα και πιο σημαντική την επόμενη δεκαετία, καθώς η πρόοδος στην τεχνολογία του υλικού θα επιτρέψει την συγχώνευση των ως τώρα αρχιτεκτονικών υψηλών επιδόσεων ειδικού σκοπού (π.χ., GPUs) με τους ευέλικτους επεξεργαστές υψηλών επιδόσεων γενικού σκοπού.

1.3.1 Θέματα κατανάλωσης ενέργειας

Η κατανάλωση ισχύος των σύγχρονων επεξεργαστών αποκτά αυξανόμενο ενδιαφέρον την τελευταία δεκαετία. Η σμίκρυνση της κλίμακας ολοκλήρωσης και η συνοδευόμενη αύξηση της συχνότητας λειτουργίας των κυκλωμάτων θα είχε οδηγήσει σε τέτοια πυκνότητα ισχύος που θα καθιστούσε την ψύξη του επεξεργαστή τουλάχιστον μη πρακτική, αν όχι προβληματική. Η συνειδητοποίηση αυτού του προβλήματος οδήγησε στην απότομη διακοπή της κλιμάκωσης της συχνότητας των επεξεργαστών, παρά τις φιλόδοξες προβλέψεις στις αρχές της δεκαετίας του 2000 [ITRS, 2001]. Η επίδοση, παραταύτα, δεν υποβιβάστηκε, καθώς περισσότεροι επεξεργαστικοί πυρήνες ενσωματώνονται πλέον στο ίδιο τσιπ, προσφέροντας ολοένα αυξανόμενα επίπεδα επίδοσης, εκμεταλλευόμενοι τον παραλληλισμό. Ωστόσο, καθώς η κλίμακα ολοκλήρωσης συνεχώς φθίνει προς μερικά μόνο νανόμετρα, η κατανάλωση ισχύος λόγω διαρροών έχει αρχίσει να κυριαρχεί στην συνολική κατανάλωση ισχύος των σύγχρονων επεξεργαστών [Ahmed και Schuegraf, 2011]. Ο διαμοιρασμός των πόρων σε επίπεδο υλικού και η ρύθμιση της συχνότητας μπορούν να βοηθήσουν στον έλεγχο της κατανάλωσης ισχύος των σύγχρονων επεξεργαστών και να αυξήσουν την ενεργειακή τους απόδοση.

Οι σύγχρονοι πολυπύρρηνοι επεξεργαστές διαμοιράζουν τους πόρους τους σε διαφορετικά επίπεδα, που κυμαίνονται από την σωλήνωση (pipeline) ως τα υψηλότερα επίπεδα της ιεραρχίας της κρυφής μνήμης και του ελεγκτή μνήμης. Ο αποδοτικός διαμοιρασμός των πόρων μεταξύ των νημάτων μιας πολυνηματικής εφαρμογής μπορεί να είναι ευεργετικός όσον αφορά στην συνολική κατανάλωση ισχύος, καθώς «αχρησιμοποίητα» μέρη του τσιπ μπορούν να απενεργοποιούνται, μειώνοντας σημαντικά την συνολική κατανάλωση ενέργειας [Kaxiras et al., 2001]. Παρομοίως, η επίδραση της κλιμάκωσης προς τα κάτω της συχνότητας του επεξεργαστή στην συνολική κατανάλωση ισχύος του μπορεί να είναι σημαντική, λόγω της υπερ-γραμμικής σχέσης τους [Brooks et al., 2000; Kaxiras και Martonosi, 2008]. Η συνακόλουθη απώλεια στην συνολική επίδοση μπορεί να ελαχιστοποιηθεί, διατηρώντας παράλληλα μία υψηλή ενεργειακή απόδοση, κλιμακώνοντας δυναμικά προς τα κάτω την συχνότητα σε μη κρίσιμα για την επίδοση τμήματα κώδικα [Kaxiras και Martonosi, 2008;

¹ Με τον όρο *επικοινωνία* εννοούμε ολόκληρη την εκτός-τσιπ κίνηση επικοινωνίας, συμπεριλαμβανομένων και των προσβάσεων στη μνήμη.

Curtis-Maury et al., 2008]. Δυστυχώς, όμως, δεν υπάρχει κάποιο μαγικό κλειδί που μεγιστοποιεί την επίδοση και ταυτόχρονα ελαχιστοποιεί την κατανάλωση ενέργειας. Υπάρχει ένα σύνολο βέλτιστων συμβιβασμών, αντιθέτως, που πρέπει να ισορροπήσει κανείς ανάλογα με τις συγκεκριμένες ανάγκες της κάθε αρχιτεκτονικής και του υποκειμένου συστήματος [Karakasis et al., 2011]. Η εύρεση ενός βέλτιστου συμβιβασμού επίδοσης-κατανάλωσης ενέργειας δεν είναι μία απλή διαδικασία και πρέπει να εμπλέκει τόσο το υλικό όσο και το λογισμικό. Το υλικό πρέπει να γίνει πιο ευέλικτο, ώστε να παρέχει δυνατότητες για την παρακολούθηση της κατανάλωσης ισχύος των διαφορετικών μερών του επεξεργαστή και για την δυναμική ενεργοποίηση/απενεργοποίηση των ενεργοβόρων μερών του. Το λογισμικό, από την άλλη πλευρά, πρέπει να χρησιμοποιεί αυτά τα χαρακτηριστικά αποδοτικά είτε στο επίπεδο του λειτουργικού συστήματος είτε μέσω ενός ενεργειακά ενήμερου περιβάλλοντος εκτέλεσης, ώστε να μπορεί να παρέχει τα μέσα για υψηλών επιδόσεων και ενεργειακά αποδοτικούς υπολογισμούς.

1.3.2 Η αλγοριθμική φύση του πυρήνα SpMV

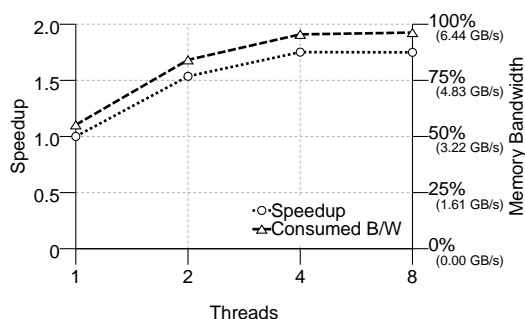
Οι πυρήνες πολλαπλασιασμού πίνακα επί διάνυσμα, τόσο πυκνών όσο και αραιών πινάκων, μπορούν να θεωρηθούν σαν μία ακολουθία σημειακών γινομένων μεταξύ των διανυσμάτων-γραμμών του πίνακα και του διανύσματος εισόδου. Ο Αλγόριθμος 1.1 δείχνει αυτή την αφ' υψηλού προσέγγιση. Ο αλγόριθμος διασχίζει μια φορά ολόκληρο τον πίνακα και πραγματοποιεί ένα σταθερό αριθμό υπολογισμών πράξεων κινητής υποδιαστολής ανά στοιχείο. Αυτό οδηγεί αυτομάτως σε ένα λόγο $\text{flop:byte} \Theta(1)$ σε σχέση με τον λόγο της τάξης $\Theta(n)$ (όπου n η διάσταση του πίνακα) των πυρήνων πολλαπλασιασμού πίνακα-επί-πίνακα. Στην πράξη, αυτό σημαίνει ότι για να αποφευχθεί η συμφόρηση, θα πρέπει η ιεραρχία μνήμης να μπορεί να παρέχει δεδομένα στον επεξεργαστή με συγκρίσιμη με αυτόν ταχύτητα, κάτι το οποίο δεν μπορεί να συμβεί στις σύγχρονες μικρο-αρχιτεκτονικές. Η κατάσταση επιδεινώνεται ακόμα περισσότερο στην περίπτωση των αραιών πινάκων, όπου ο πυρήνας πρέπει πρώτα να μεταφέρει την πληροφορία της θέσης των μη μηδενικών στοιχείων, προτού μπορέσει να τα προσπελάσει.

Στο Σχήμα 1.3 παρουσιάζεται η κλιμάκωση της βασικής πολυνηματικής υλοποίησης του πυρήνα SpMV και το καταναλούμενο εύρος ζώνης μνήμης σε GB/s σε ένα σύστημα συμμετρικής πρόσβασης στην μνήμη με δύο τετραπύρηνους επεξεργαστές. Παρά το γεγονός ότι ο πυρήνας SpMV παρουσιάζει ένα σημαντικό βαθμό παραλληλίας, η εκτέλεσή του αποτυγχάνει να κλιμακωθεί πέρα από τα τέσσερα νήματα, λόγω του κορεσμού του εύρους ζώνης μνήμης του συστήματος. Ένα 85% του διαθέσιμου εύρους ζώνης μνήμης καταναλώνεται ήδη από τα δύο νήματα, ενώ έχει κορεστεί πλήρως από τα τέσσερα νήματα

1. Εισαγωγή

```
1: procedure MATVEC(A::in, x::in, y::out)
2:   foreach row vector  $a_{i*}$  in A do
3:      $y_i \leftarrow a_{i*}^T x$ 
4:   end for
```

Αλγόριθμος 1.1: Αφ' υψηλού αναπαράσταση του πολλαπλασιασμού αραιού πίνακα με διάνυσμα.



Σχήμα 1.3: Επίδειξη της κλιμάκωσης του πυρήνα SpMV σε συνάρτηση με την κατανάλωση εύρους ζώνης μνήμης σε ένα σύστημα συμμετρικής πρόσβασης στην κύρια μνήμη με δύο τετραπύρηνους επεξεργαστές.

και πέρα. Η ελαχιστοποίηση του μεγέθους αναπαράστασης του αραιού πίνακα γίνεται, επομένως, ζωτικής σημασίας για την βελτιστοποίηση του πυρήνα SpMV, ειδικά σε αρχιτεκτονικές συμμετρικής πρόσβασης στην κύρια μνήμη.

Παρά το ότι είναι το πιο εμφανές, η ένταση των προσβάσεων στην κύρια μνήμη δεν είναι το μοναδικό πρόβλημα του πυρήνα SpMV. Ακόμα πιο προκλητικό είναι το γεγονός ότι η επίδοση του πυρήνα εξαρτάται σε σημαντικό βαθμό από την διάταξη των μη μηδενικών στοιχείων του πίνακα, πράγμα που μπορεί να οδηγήσει σε δραματικές διακυμάνσεις της επίδοσης. Για παράδειγμα, πίνακες με πολύ ακανόνιστη δομή μπορούν να οδηγήσουν σε σημαντικό αριθμό αστοχιών στην κρυφή μνήμη και σημαντική ανισορροπία του φόρτου εργασίας των νημάτων, καταστρέφοντας την συνολική επίδοση. Αντιθέτως, σε πίνακες με πιο κανονική διάταξη των στοιχείων τους το υπολογιστικό μέρος του πυρήνα μπορεί να εκτίθεται περισσότερο, με αποτέλεσμα να απαιτούν μία προσέγγιση λιγότερη εστιασμένη στην βελτιστοποίηση της κατανάλωσης εύρους ζώνης μνήμης. Είναι, επομένως, επιτακτική ανάγκη μία επιτυχημένη βελτιστοποίηση του πυρήνα SpMV να μπορεί να προσαρμόζεται στις ιδιαιτερότητες κάθε άγνωστου πίνακα εισόδου κατά την διάρκεια της εκτέλεσης και

αυτό ίσως αποτελεί μία από τις μεγαλύτερες προκλήσεις στην βελτιστοποίηση του συγκεκριμένου πυρήνα.

1.3.3 Η ενεργειακή απόδοση του πυρήνα SpMV

Η φύση του πυρήνα SpMV καθιστά την επίδοσή του ιδιαίτερα ευαίσθητη στον διαμοιρασμό των επιπέδων της κρυφής μνήμης από τα νήματα. Τα πολλαπλά νήματα που εκτελούν τον πυρήνα SpMV και διαμοιράζονται το ίδιο επίπεδο της κρυφής μνήμης ανταγωνίζονται για τον χώρο της, προκαλώντας μία αύξηση των αστοχιών χωρητικότητας (capacity misses), υποβαθμίζοντας έτσι σημαντικά την επίδοση του πυρήνα. Από την άλλη πλευρά, το να «απλώσει» κανείς τα νήματα σε διαφορετικές κρυφές μνήμες δεν αποτελεί ενεργειακά-φιλική εκτέλεση, παρά την βελτίωση της επίδοσης, καθώς οι κρυφές μνήμες συνεισφέρουν ένα σημαντικό μέρος της συνολικής κατανάλωσης ισχύος του επεξεργαστή [Kaxiras και Martonosi, 2008]. Αυτή η αύξηση της κατανάλωσης ισχύος μπορεί να εξισορροπηθεί με την χρήση πολλαπλών νημάτων σε χαμηλότερη συχνότητα, χωρίς να διακινδυνεύει η επίδοση του πυρήνα. Η επιλογή της κατάλληλης τοποθέτησης των νημάτων στου επεξεργαστές και της συχνότητάς τους για την επίτευξη ενός βέλτιστου συμβιβασμού μεταξύ επίδοσης και κατανάλωσης ενέργειας αποτελεί μία πρόκληση για τον πυρήνα SpMV, καθώς η επίδοσή του εξαρτάται σε σημαντικό βαθμό από την δομή του πίνακα εισόδου. Η εξέταση της δομής του πίνακα εκ των προτέρων και η χρησιμοποίηση ενός προχωρημένου σχήματος κατηγοριοποίησης μπορεί να είναι απαραίτητα για την επιλογή μίας εκτέλεσης του πυρήνα που θα επιτύχει ένα βέλτιστο συμβιβασμό επίδοσης-κατανάλωσης ενέργειας.

1.4 Συμβολή της διατριβής

Σε αυτή τη διατριβή εστιάζουμε στην βελτιστοποίηση του πυρήνα SpMV σε σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών, καθώς αποτελεί τον πιο χρονοβόρο πυρήνα των επαναληπτικών μεθόδων επίλυσης αραιών γραμμικών συστημάτων. Παρά το ότι οι preconditioners είναι αναπόσπαστο κομμάτι των επιτυχημένων επαναληπτικών μεθόδων και μπορούν να γίνουν ιδιαίτερα χρονοβόροι υπό συγκεκριμένες συνθήκες, δεν εστιάζουμε σε αυτές τις πράξεις. Στην πραγματικότητα, οι πράξεις αυτές αποτελούν άμεσες μεθόδους επίλυσης και, επομένως, δεν μπορούν να θεωρηθούν ως άμεσα συνδεδεμένες με την επαναληπτική διαδικασία επίλυσης. Αντιθέτως, ανήκουν στο ιδιαίτερα προκλητικό πεδίο των αμέσων μεθόδων επίλυσης και η βελτιστοποίησή τους μπορεί να θεωρηθεί ως ένα άλλο σημαντικό, ωστόσο διαφορετικό, υπολογιστικό πρόβλημα [Davis, 2006; Asanovic et al., 2006].

1.4.1 Εις βάθος ανάλυση της επίδοσης και μοντέλα πρόβλεψης

Η επίδοση του πυρήνα SpMV έχει απασχολήσει σημαντικά την κοινότητα και στο παρελθόν· ωστόσο, μία δυνατή ανάλυση, η οποία θα ποσοτικοποιούσε την επίδραση των εικαζόμενων προβλημάτων επίδοσης, δεν είχε εκπονηθεί. Ως αποτέλεσμα, δεν είχε υπάρξει ένα καθαρό συμπέρασμα ως προς το ποια είναι τα πιο σημαντικά προβλήματα επίδοσης του πυρήνα SpMV. Σε αυτή την διατριβή, ρίχνουμε μια διεισδυτική ματιά στα αλγοριθμικά χαρακτηριστικά του πυρήνα SpMV και πραγματοποιούμε μία ποσοτική ανάλυση της επίδοσης σε μία ευρεία γκάμα αραιών πινάκων και σύγχρονων πολυπύρηνων αρχιτεκτονικών, με σκοπό να εκτιμήσουμε την ακριβή επίδρασή τους στην επίδοση του πυρήνα. Η πραγματοποιηθείσα ανάλυση παρέχει μία σαφή κατανόηση της επίδοσης του πυρήνα SpMV και λειτουργεί ως οδηγός για αποδοτικές υλοποιήσεις του. Πράγματι, η αναγνώριση της συμφόρησης για εύρος ζώνης μνήμης ως του προβλήματος-κλειδιού του SpMV στις πολυνηματικές εκτελέσεις μας οδήγησε στην υιοθέτηση αμέσων μεθόδων συμπίεσης για την βελτιστοποίηση του πυρήνα. Επιπρόσθετα, βασιζόμενοι σε αυτή την ανάλυση, αναπτύξαμε μοντέλα επίδοσης για την πρόβλεψη του βελτίστου μπλοκ για τις μεθόδους αποθήκευσης σε μπλοκ σταθερού μεγέθους.

1.4.2 Το σχήμα αποθήκευσης αραιών πινάκων CSX

Η κύρια συμβολή της διατριβής αυτής είναι το σχήμα αποθήκευσης αραιών πινάκων Compressed Sparse eXtended (CSX). Βασισμένο στην ιδέα της άμεσης συμπίεσης των δομών δεικτοδότησης των σχημάτων αποθήκευσης αραιών πινάκων, ώστε να ελαχιστοποιηθεί το μέγεθος αναπαράστασης του πίνακα [Willcock και Lumsdaine, 2006; Kourttis et al., 2010], το σχήμα CSX ενσωματώνει την έννοια της οργάνωσης σε μπλοκ, ώστε να επιτύχει ακόμα υψηλότερο βαθμό συμπίεσης και καλύτερα υπολογιστικά χαρακτηριστικά [Karakasis et al., 2009a,b]. Η ιδέα της άμεσης συμπίεσης των δεδομένων μιας απαιτητικής σε εύρος ζώνης μνήμης εφαρμογής, όπως είναι ο πυρήνας SpMV, στοχεύει στην μείωση της πίεσης στο υποσύστημα μνήμης σε βάρος των υπολογισμών αποσυμπίεσης. Παραταύτα, εάν η ένταση των προσβάσεων στην κύρια μνήμη μιας εφαρμογής μετριάζεται από την αυξημένη διαθεσιμότητα εύρους ζώνης μνήμης, οι υπολογισμοί αποσυμπίεσης μπορεί να εκτεθούν περισσότερο στον συνολικό χρόνο εκτέλεση και γι' αυτό τον λόγο θα πρέπει να μειωθούν. Σε αυτή την περίπτωση, το υπολογιστικό μέρος του πυρήνα SpMV εκτίθεται ακόμη περισσότερο, ένα θέμα που θα πρέπει να επιλυθεί επιτυχώς, ώστε να επιτευχθεί υψηλή επίδοση.

Το CSX είναι ένα συμπαγές σχήμα αποθήκευσης αραιών πινάκων, το οποίο μπορεί να ανιχνεύσει και να κωδικοποιήσει ποικίλες κανονικότητες μη μηδενι-

κών στοιχείων του πίνακα. Ο όρος *κανονικότητα* αφορά σε οποιαδήποτε μονοδιάστατη ή δισδιάστατη ακολουθία από μη μηδενικά στοιχεία του αραιού πίνακα. Το πλήθος όλων των δυνατών κανονικοτήτων που μπορεί να ανιχνεύσει ο CSX είναι απείρως μεγάλο και εκ των προτέρων άγνωστο. Για τον λόγο αυτό, το σχήμα CSX υιοθετεί τεχνικές παραγωγής κώδικα στον χρόνο εκτέλεσης, ώστε να παρέχει υλοποίησης υψηλών επιδόσεων του πυρήνα SpMV, προσαρμοσμένες στις ιδιαιτερότητες κάθε συγκεκριμένου πίνακα. Συγκρινόμενο με άλλα σχήματα αποθήκευσης που εκμεταλλεύονται ένα τύπο κανονικοτήτων, π.χ., το σχήμα BCSR που εκμεταλλεύεται μόνο δισδιάστατες κανονικότητες [Im και Yelick, 2001], το σχήμα CSX είναι ικανό να επιτύχει σταθερή και συνεπή υψηλή επίδοση, μέσω της επιτυχούς προσαρμογής του σε μία μεγάλη ποικιλία αραιών πινάκων, ξεκινώντας από πίνακες με κανονική διάταξη των στοιχείων και φθάνοντας σε πίνακες με ακανόνιστη διάταξη. Το σχήμα CSX μπορεί να επιταχύνει τον πυρήνα SpMV περισσότερο από 50% κατά μέσο όρο σε αρχιτεκτονικές συμμετρικής πρόσβασης στην κύρια μνήμη, όπου η ένταση των προσβάσεων του πυρήνα στη μνήμη είναι αυξημένη. Οι αρχιτεκτονικές NUMA υποστηρίζονται και αυτές αποδοτικά από το σχήμα CSX, το οποίο προσαρμόζεται κατάλληλα χρησιμοποιώντας ένα λιγότερο επιθετικό σχήμα συμπίεσης, ώστε να μετριαστεί το κόστος αποσυμπίεσης· σε αυτές τις αρχιτεκτονικές, το σχήμα CSX μπορεί να παρέχει περίπου 20% βελτίωση της επίδοσης σε σχέση με τις τυπικές υλοποιήσεις του SpMV.

Ένας βασικός στόχος του σχεδιασμού του σχήματος CSX ήταν να είναι πρακτική η χρήση του εντός υπάρχοντων επαναληπτικών μεθόδων επίλυσης. Για τον λόγο αυτό, επικεντρωθήκαμε επισταμένα στην ελαχιστοποίηση του κόστους προ-επεξεργασίας του σχήματος CSX, το οποίο αποτελείται από το κόστος της εξόρυξης των κανονικοτήτων του πίνακα και το κόστος της τελικής κατασκευής του σχήματος CSX. Χρησιμοποιώντας ένα συνδυασμό στατιστικού δειγματισμού του πίνακα εισόδου, προσεκτικής διαχείρισης της μνήμης και παραλληλοποίησης, καταφέραμε να μειώσουμε το συνολικό κόστος προ-επεξεργασίας μερικές δεκάδες σειριακών πράξεων SpMV. Πράγματι, παρά το κόστος προ-επεξεργασίας του, το σχήμα CSX μπόρεσε να επιταχύνει τον πολυπλασιασμό SpMV του επιλυτή Bi-CG Stabilized του λογισμικού φυσικών προσομοιώσεων Elmer περίπου 40%, ενώ κατάφερε να προσφέρει μία βελτίωση 15% κατά μέσο όρο στον χρόνο εκτέλεσης ολόκληρου του επιλυτή.

Μία πρώιμη έκδοση του σχήματος CSX είχε παρουσιαστεί από τον Kourtis [2010] ως μία πρώτη προσπάθεια για την επίτευξη υψηλότερου βαθμού συμπίεσης μέσω της εκμετάλλευσης γραμμικών κανονικοτήτων εντός τους αραιού πίνακα (οριζοντίων, κατακορύφων, διαγωνίων και αντι-διαγωνίων). Σε αυτή την διατριβή το σχήμα CSX επεκτάθηκε σημαντικά και ουσιαστικά, ώστε να αποτελέσει μία βιώσιμη προσέγγιση υψηλής επίδοσης για την βελτιστοποίηση του πυρήνα SpMV στις σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών.

Ένα σύνολο νέων χαρακτηριστικών υλοποιήθηκε, το οποίο επέτρεψε στο σχήμα CSX να αποτελεί πλέον ένα state-of-the-art σχήμα αποθήκευσης αραιών πινάκων:

- (α') *Υποστήριξη δισδιάστατων κανονικότητων.* Οι δισδιάστατες κανονικότητες επιτρέπουν στο CSX να επιτυγχάνει βαθμούς συμπίεσης κοντά στο θεωρητικό μέγιστο, ενά παράλληλα επιδεικνύει καλύτερα υπολογιστικά χαρακτηριστικά.
- (β') *Προσαρμοζόμενο σχήμα συμπίεσης.* Το σχήμα αυτό επιτρέπει την «χαλάρωση» της συμπίεσης στις αρχιτεκτονικές NUMA, όπου το πρόβλημα συμφόρησης στο σύστημα μνήμης δεν είναι τόσο έντονο, κερδίζοντας επεξεργαστικούς κύκλους από το κόστος αποσυμπίεσης.
- (γ') *Προχωρημένη ευρετική συνάρτηση επιλογής κανονικότητων.* Η νέα ευρετική συνάρτηση επιτρέπει την αποδοτική εξισορρόπηση του ωφέλους της συμπίεσης και της επιβάρυνσης της αποσυμπίεσης ανάλογα με την υποκείμενη αρχιτεκτονική. Σε συνδυασμό με το προσαρμοζόμενο σχήμα συμπίεσης, το CSX παρέχει σημαντικά οφέλη επίδοσης και στις αρχιτεκτονικές NUMA.
- (δ') *Βελτιστοποιημένη προ-επεξεργασία.* Το κόστος προ-επεξεργασίας του πίνακα που απαιτείται για την ανίχνευση και κωδικοποίηση των κανονικότητων του CSX μειώθηκε αισθητά, επιτρέποντας στο CSX να μπορεί να χρησιμοποιηθεί «ως έχει» στο λογισμικό φυσικών προσομοιώσεων Elmer [Lyll et al., 1999–2000] και να επιταχύνει την επίδοση του επιλυτή.
- (ε') *Βελτιωμένη παραγωγή κώδικα στον χρόνο εκτέλεσης.* Η μονάδα παραγωγής κώδικα τροποποιήθηκε εκ βάθρων, ώστε να επιτρέπει την πιο άμεση και αποδοτική υλοποίηση των ρουτινών του SpMV για κάθε συγκεκριμένη κανονικότητα.
- (ς') *Υποστήριξη συμμετρικών πινάκων.* Το σχήμα CSX υποστηρίζει επιπλέον συμμετρικούς πίνακες, επιτρέποντας σημαντικό κέρδος στην επίδοση του SpMV, λόγω της σημαντικής μείωσης του μεγέθους αναπαράστασης του πίνακα.

Αποδοτική υλοποίηση για συμμετρικούς πίνακες

Προτείνουμε επίσης μία παραλλαγή του σχήματος CSX για συμμετρικούς πίνακες, η οποία είναι ικανή να επιτυγχάνει μείωση περίπου $3\times$ στην αναπαράσταση του αραιού πίνακα. Οι πολυνηματικές υλοποιήσεις του πυρήνα SpMV, ωστόσο, πρέπει να αντιμετωπίσουν μία εξάρτηση δεδομένων RAW στο διάνυσμα εξόδου. Με σκοπό να αποφευχθεί το απαγορευτικό κόστος των κλειδωμάτων, η εξάρτηση αυτή επιλύεται συνήθως με την χρήση τοπικών, ανά νήμα,

διανυσμάτων, τα οποία ανάγονται στο τελικό διάνυσμα εξόδου στο τέλος του υπολογισμού του SpMV. Ωστόσο, η φάση αυτή της αναγωγής εμπεριέχει μία σημαντική επιβάρυνση της επίδοσης, η οποία αυξάνεται γραμμικά σε σχέση με το πλήθος των νημάτων, στερώντας από το SpMV την δυνατότητα κλιμάκωσης. Επιλύουμε αυτό το πρόβλημα με τη χρήση ενός σχήματος δεικτοδότησης των τοπικών διανυσμάτων, το οποίο επιτρέπει την επιλεκτική αναγωγή μόνο των στοιχείων που «συγκρούονται», ενώ όλες οι υπόλοιπες πράξεις ανανέωσης κατευθύνονται απευθείας στο διάνυσμα εξόδου. Το σχήμα αυτό αποσυνδέει επί της ουσίας το κόστος της αναγωγής από το πλήθος των νημάτων, καθώς τα τοπικά διανύσματα γίνονται αραιότερα όσο αυξάνεται το πλήθος των νημάτων και, επομένως, το μέγεθος της δομής δεικτοδότησης μειώνεται. Ως αποτέλεσμα, η επίδοση του πυρήνα SpMV κλιμακώνεται κατά το αναμενόμενο. Το συμμετρικό σχήμα CSX σε συνδυασμό με το σχήμα αναγωγής χαμηλής επιβάρυνσης που προτείνουμε επιτρέπει την επιτάχυνση της επίδοσης του πυρήνα SpMV περισσότερο από δύο φορές.

Η συμπίεση δεδομένων πέρα από το SpMV

Η ιδέα της συμπίεσης δεδομένων που χρησιμοποιείται από το σχήμα CSX για την βελτιστοποίηση του πυρήνα SpMV έχει ευρύτερο αντίκτυπο στην σύγχρονη εποχή των πολυπύρηνων επεξεργαστών. Κάθε νέα γενιά επεξεργαστών όχι μόνο αυξάνει την απόσταση επίδοσης μεταξύ επεξεργαστή και κύριας μνήμης, αλλά αυξάνει και την απόσταση των δύο σε κατανάλωση ισχύος. Η κύρια μνήμη είναι βασικός παράγων της συνολικής κατανάλωσης ισχύος ενός υπολογιστικού συστήματος και για τον λόγο αυτό απαιτητικές σε μνήμη εφαρμογές τείνουν να καταναλώνουν περισσότερη ενέργεια απ' ό,τι εφαρμογές απαιτητικές σε κύκλους επεξεργαστή [Kamil et al., 2008]. Επομένως, άμεσες τεχνικές συμπίεσης δεδομένων είναι αρκετά πιθανό να αποτελέσουν κοινό τόπο στο μέλλον, όχι μόνο ως μέσο μείωσης της συμφόρησης στο υποσύστημα μνήμης, αλλά και ως μέσο μείωσης της κατανάλωσης ενέργειας των μελλοντικών εφαρμογών HPC.

1.4.3 Προς μία ενεργειακά αποδοτική υλοποίηση του SpMV

Στη διατριβή αυτή κάνουμε ένα πρώτο βήμα στην διερεύνηση των ισορροπιών επίδοσης-κατανάλωσης ενέργειας του πυρήνα SpMV και προτείνουμε μία μεθοδολογία για την επιλογή εκείνης της συχνότητας του επεξεργαστή και της τοποθέτησης των νημάτων που οδηγούν σε βέλτιστες ισορροπίες για τον πυρήνα SpMV. Βασιζόμενοι στην έννοια του κατά Pareto βελτίστου, δείχνουμε ότι υπάρχει ένα σύνολο βέλτιστων ισορροπιών επίδοσης-κατανάλωσης ενέργειας και ότι τα συχνά χρησιμοποιούμενα γινόμενα energy-delay είναι όντως

βέλτιστες κατά Pareto λύσεις. Ο κύριος στόχος της μεθοδολογίας μας είναι όχι μόνο να προβλέψουμε τις καλύτερες εκτελέσεις βάσει των γινομένων energy-delay, αλλά να παρέχουμε ένα σύνολο αντιπροσωπευτικών βέλτιστων ισορροπιών. Βασιζόμενοι στην παρατήρηση ότι πίνακες με παρόμοια δομικά χαρακτηριστικά έχουν παρόμοια συμπεριφορά επίδοσης και κατανάλωσης ενέργειας, χτίζουμε την μέθοδό μας πάνω σε τεχνικές ομαδοποίησης και κατασκευάζουμε ένα αντιπροσωπευτικό σύνολο βελτίστων εκτελέσεων για κάθε ομάδα. Η μέθοδος πρόβλεψης που προτείνουμε μπορεί να παρέχει σχεδόν βέλτιστα σύνολα επιλογών εκτέλεσης του πυρήνα SpMV για ένα μεγάλο εύρος αραιών πινάκων.

1.5 Δομή της διατριβής

Το υπόλοιπο της διατριβής προχωρά με μία λεπτομερή και εις βάθος ανάλυση των εμπλεκόμενων θεμάτων και της συμβολής της διατριβής.

Το Κεφάλαιο 2 αποτελεί μία ανασκόπηση των σχημάτων αποθήκευσης που έχει προταθεί στην βιβλιογραφία. Ξεκινώντας από τα συμβατικά και ευρέως χρησιμοποιούμενα σχήματα, προχωράμε μέσω των διαφόρων σχημάτων αποθήκευσης σε μπλοκ στα εστιασμένα σε συγκεκριμένες μικρο-αρχιτεκτονικές σχήματα και στα σχήματα αποθήκευσης που βασίζονται σε τεχνικές άμεσης συμπίεσης. Παρουσιάζουμε, επίσης, σχήματα αποθήκευσης συμμετρικών πινάκων και συζητούμε τις κύριες προκλήσεις του συμμετρικού πυρήνα SpMV.

Το Κεφάλαιο 3 παρουσιάζει μία εις βάθος ανάλυση της επίδοσης του πυρήνα SpMV στις σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών. Επισημαίνουμε από αλγοριθμικής απόψεως τα πιθανά προβλήματα επίδοσης και προχωρούμε στην ενδελεχή μελέτη τις επίδρασής τους στις σύγχρονες πολυπύρηνες αρχιτεκτονικές τόσο στη σειριακή όσο και στην πολυνηματική εκτέλεση. Στο κεφάλαιο αυτό, παρουσιάζουμε επιπλέον την σουίτα πινάκων, τις πειραματικές πλατφόρμες και την πειραματική μεθολογία που ακολουθήσαμε στην συνέχεια της διατριβής.

Το Κεφάλαιο 4 εστιάζει συγκεκριμένα στα σχήματα αποθήκευσης σε μπλοκ και διερευνά τις δυνατότητες βελτιστοποίησης που παρέχουν. Μελετούμε τις δυνατότητες συμπίεσης και τα υπολογιστικά χαρακτηριστικά κάθε σχήματος και εξετάζουμε την επίδρασή τους στις σύγχρονες πολυπύρηνες αρχιτεκτονικές. Βασιζόμενοι σε αυτή την ανάλυση, προτείνουμε δύο εναλλακτικά και απλά μοντέλα επίδοσης για την πρόβλεψη του βέλτιστου μπλοκ στο σχήμα BCSR.

Το Κεφάλαιο 5 παρουσιάζει με λεπτομέρεια το σχήμα αποθήκευσης Compressed Sparse eXtended. Η παρουσίαση των δομών δεδομένων και της κατασκευής του τελικού πίνακα ακολουθείται από μία ενδελεχή αξιολόγηση της

επίδοσης του σχήματος, η οποία επιβεβαιώνει την υπεροχή του CSX σε σύγκριση με άλλα εναλλακτικά σχήματα αποθήκευσης τόσο όσον αφορά την απόλυτη επίδοση όσο και την σταθερότητα της επίδοσης σε μία ποικιλία αραιών πινάκων και αρχιτεκτονικών. Το κεφάλαιο κλείνει με τα αποτελέσματα της ενσωμάτωσης του CSX στο λογισμικό φυσικών προσομοιώσεων Elmer.

Το Κεφάλαιο 6 παρουσιάζει την προσέγγισή μας στην βελτιστοποίηση του συμμετρικού πυρήνα SrMV. Ξεκινάμε με μία παρουσίαση και αξιολόγηση της επίδρασης της βελτιστοποίησης της φάσης αναγωγής που επιτρέπει την κλιμάκωση του SrMV και συνεχίζουμε με την παρουσίαση της παραλλαγής του CSX για συμμετρικούς πίνακες. Το κεφάλαιο κλείνει με την πειραματική αξιολόγηση των προτεινομένων βελτιστοποιήσεων στο πλαίσιο της επαναληπτικής μεθόδου CG.

Το Κεφάλαιο 7 ξεφεύγει από την αποκλειστικά προσανατολισμένη στην επίδοση λογική θεώρησης του πυρήνα SrMV και εισάγει την έννοια της ενεργειακής απόδοσης. Ξεκινώντας από μία εισαγωγή των βασικών αρχών της κατανάλωσης ισχύος των συγχρόνων επεξεργαστών, διερευνούμε στην συνέχεια τις ισορροπίες επίδοσης-κατανάλωσης ενέργειας του πυρήνα SrMV σε σχέση με τον πίνακα εισόδου, την συχνότητα του επεξεργαστή και την τοποθέτηση των νημάτων. Τέλος, προτείνουμε μία προσέγγιση βασισμένη σε τεχνικές μάθησης μηχανών για την πρόβλεψη των ρυθμίσεων εκτέλεσης (συχνότητα πυρήνων, τοποθέτηση νημάτων), η οποία οδηγεί σε βέλτιστους συμβιβασμούς επίδοσης-κατανάλωσης ενέργειας.

Το Κεφάλαιο 8 κλείνει αυτή την διατριβή, ανακεφαλαιώνοντας την συμβολή της και τα επιτεύγματά της, ενώ συζητεί και προτείνει μελλοντικές κατευθύνσεις έρευνας στο πεδίο της βελτιστοποίησης του πυρήνα SrMV και τον εφαρμογών HPC γενικότερα.

Αποθήκευση Αραιών Πινάκων

Η πυκνότητα ενός τυπικού αραιού πίνακα, δηλαδή ο λόγος των μη μηδενικών στοιχείων του επί του γινομένου των διαστάσεών του, είναι πολύ χαμηλότερη του 1% στις περισσότερες των περιπτώσεων. Η αποδοτική αποθήκευση ενός αραιού πίνακα σε σχέση με τον απαιτούμενο χώρο αποθήκευσης απαιτεί την αποθήκευση μόνο των μη μηδενικών στοιχείων του μαζί με κάποια τοπολογική πληροφορία, ώστε να είναι δυνατή η διάσχισή τους. Ωστόσο, η κατασκευή ενός σχήματος αποθήκευσης για την αποδοτική εκτέλεση του πυρήνα SpMV δεν είναι μία απλή διαδικασία. Η κατανομή των μη μηδενικών στοιχείων και η υποκείμενη αρχιτεκτονική υπολογιστή παίζουν σημαντικό ρόλο στην επίδοση του πυρήνα και, επομένως, ένα αποδοτικό σχήμα αποθήκευσης θα πρέπει να προσαρμόζεται με επιτυχία σε αυτές τις απαιτήσεις. Διάφορα σχήματα αποθήκευσης που ευνοούν την εκτέλεση του SpMV έχουν προταθεί, κάθε ένα με τα δικά του πλεονεκτήματα και μειονεκτήματα.

Στο κεφάλαιο αυτό, επιχειρούμε μία περιεκτική περιγραφή των πιο διαδεδομένων σχημάτων αποθήκευσης για αραιούς πίνακες και, επίσης, παρουσιάζουμε τις πιο πρόσφατες προσεγγίσεις στην βελτιστοποίηση του πυρήνα SpMV μέσω της χρήσης προχωρημένων σχημάτων αποθήκευσης. Εστιάζουμε συγκεκριμένα σε σχήματα γενικού σκοπού, δηλαδή σχήματα που μπορούν να αποθηκεύσουν αραιούς πίνακες με οποιαδήποτε διάταξη των μη μηδενικών στοιχείων τους, ενώ δεν λαμβάνουν μέριμνα για την δυναμική εισαγωγή νέων μη μηδενικών στοιχείων. Τέλος, παρόλο που παρουσιάζουμε σύντομα δύο προσεγγίσεις που συμπιέζουν τα μη μηδενικά στοιχεία του πίνακα, ασχολούμαστε κυρίως με σχήματα αποθήκευσης που εστιάζουν στην ελαχιστοποίηση του μεγέθους των δομών δεικτοδότησης των μη μηδενικών στοιχείων.

2.1 Συμβατικά σχήματα αποθήκευσης

Ο πιο απλός και άμεσος τρόπος αποθήκευσης ενός αραιού πίνακα είναι η αναπαράστασή του ως μία ακολουθία πλειάδων της μορφής (i, j, v) , όπου v η τιμή του μη μηδενικού στοιχείου και i, j είναι η αντίστοιχη γραμμή και στήλη. Το

2. Αποθήκευση Αραιών Πινάκων

$$A = \begin{pmatrix} 7.5 & 2.9 & 2.8 & 2.7 & 0 & 0 \\ 6.8 & 5.7 & 3.8 & 0 & 0 & 0 \\ 2.4 & 6.2 & 3.2 & 0 & 0 & 0 \\ 9.7 & 0 & 0 & 2.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.8 & 5.0 \\ 0 & 0 & 0 & 0 & 6.6 & 8.1 \end{pmatrix}$$

```
rowind: ( 0 0 0 0 1 1 1 2 2 2 3 3 4 4 5 5 )
colind: ( 0 1 2 3 0 1 2 0 1 2 0 3 4 5 4 5 )
values: ( 7.5 2.9 2.8 2.7 6.8 5.7 3.8 2.4 6.2 3.2 9.7 2.3 5.8 5.0 6.6 8.1 )
```

Σχήμα 2.1: Το σχήμα αποθήκευσης Coordinate.

σχήμα αυτό είναι γνωστό ως *Coordinate (COO)* [Tewarson, 1973; Rooch και Nieder, 1973; Duff και Reid, 1979; Saad, 1992]¹. Το Σχήμα 2.1 παρουσιάζει μία τυπική υλοποίηση του σχήματος COO. Οι δομές `rowind` και `colind` είναι συνήθως 32-bit ακέραιοι, ενώ οι τιμές των μη μηδενικών στοιχείων είναι αριθμοί κινητής υποδιαστολής διπλής ακρίβειας. Ο Αλγόριθμος 2.1 παρουσιάζει την υλοποίηση του πυρήνα SpMV με το συγκεκριμένο σχήμα αποθήκευσης. Το βασικό πλεονέκτημα αυτού του σχήματος είναι η ευελιξία του στην εισαγωγή και διαγραφή μη μηδενικών στοιχείων, καθότι δεν υπάρχει κάποιος περιορισμός στην σειρά αποθήκευσης των στοιχείων. Ωστόσο, το κυριώτερο μειονέκτημά του για την χρήση του στον πυρήνα SpMV είναι το μεγάλο μέγεθος αναπαράστασης ($\approx 16NNZ$ bytes) του πίνακα και η αφθονία έμμεσων, και πιθανώς ακανονίστων, αναφορών τόσο στο διάνυσμα εισόδου όσο και στο διάνυσμα εξόδου κατά την εκτέλεση του πυρήνα.

Το πιο ευρέως χρησιμοποιούμενο σχήμα αποθήκευσης για αραιούς πίνακες είναι το σχήμα *Compressed Sparse Row (CSR)* [Tinney και Walker, 1967; Rooch και Nieder, 1973; Duff και Reid, 1979; Saad, 1992]. Το σχήμα αυτό καταργεί την δομή `rowind` του σχήματος COO, η οποία αποθήκευε με άμεσο τρόπο τις γραμμές των στοιχείων, και την αντικαθιστά με ένα σύνολο δεικτών στην αρχή της κάθε γραμμής του πίνακα. Το Σχήμα 2.2 παρουσιάζει σχηματικά μία τυπική υλοποίηση του σχήματος CSR. Οι πίνακες `colind` και `values` παραμένουν οι ίδιοι όπως στο σχήμα COO, ενώ ο πίνακας `rowind` αντικαθίσταται από τον `rowptr`, ο οποίος αποθηκεύει την θέση του πρώτου στοιχείου κάθε γραμμής μέσα στους πίνακες `colind` και `values`. Για ένα $N \times M$ αραιό πίνακα, το μέγεθος του πίνακα `rowptr` είναι $N + 1$, όπου το τελευταίο στοιχείο του δείχνει

¹ Για την πληρότητα της περιγραφής μας, θα πρέπει να σημειώσουμε εδώ ότι η ονοματοδοσία των παλαιότερων σχημάτων αποθήκευσης αραιών πινάκων καθιερώθηκε από τον Saad [1992]. Παλαιότερες ερευνητικές εργασίες ή επισκοπήσεις συνήθως μόνο περιγράφουν τα διαφορετικά σχήματα, χωρίς να εισάγουν συγκεκριμένους όρους.

```

1: procedure MATVECCOO(A::in, x::in, y::out)
   A: matrix in COO format
   x: input vector
   y: output vector
2:   for i ← 0 to NNZ do
3:     yi ← rowind[i]
4:     y[yi] ← y[yi] + values[i] · x[colind[i]]
5:   end for

```

Αλγόριθμος 2.1: Υλοποίηση του πυρήνα SpMV με χρήση του σχήματος COO.

$$A = \begin{pmatrix} 7.5 & 2.9 & 2.8 & 2.7 & 0 & 0 \\ 6.8 & 5.7 & 3.8 & 0 & 0 & 0 \\ 2.4 & 6.2 & 3.2 & 0 & 0 & 0 \\ 9.7 & 0 & 0 & 2.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.8 & 5.0 \\ 0 & 0 & 0 & 0 & 6.6 & 8.1 \end{pmatrix}$$

rowptr: (0 4 7 10 12 14 16)

colind: (0 1 2 3 0 1 2 0 1 2 0 3 4 5 4 5)

values: (7.5 2.9 2.8 2.7 6.8 5.7 3.8 2.4 6.2 3.2 9.7 2.3 5.8 5.0 6.6 8.1)

Σχήμα 2.2: Το σχήμα αποθήκευσης Compressed Sparse Row.

πάντοτε στο τέλος των πινάκων colind και values. Καθώς $N \ll NNZ$ για την πλειονότητα των αραιών πινάκων, το μέγεθος αναπαράστασης του σχήματος CSR είναι αισθητά μειωμένο σε σχέση με το σχήμα COO ($\approx 12NNZ$), ενώ η κατασκευή του παραμένει επίσης εύκολη και άμεση. Αυτό αποτελεί και το κυριώτερο πλεονέκτημα του CSR, που σε μεγάλο βαθμό το έχει καθιερώσει: είναι ένα συμπαγές και εύκολο στη χρήση σχήμα αποθήκευσης. Παραταύτα, το σχήμα CSR επιβάλλει μία διάσχιση κατά γραμμές των μη μηδενικών στοιχείων του πίνακα, το οποίο υπονοεί μία λεξικογραφική ταξινόμηση των στοιχείων βάσει των συντεταγμένων τους. Η δεικτοδότηση των γραμμών από το σχήμα CSR διευκολύνει την τυχαία προσπέλαση των μη μηδενικών στοιχείων (πολυπλοκότητα $O(N)$), αλλά δυσκολεύει την εισαγωγή και διαγραφή στοιχείων. Το κατά στήλες ταίρι του CSR είναι το σχήμα *Compressed Sparse Column* (CSC). Το σχήμα CSC αποθηκεύει τα μη μηδενικά στοιχεία κατά στήλες, διατηρώντας τις γραμμές ρητά, ενώ δεικτοδοτεί τις στήλες του πίνακα.

Ο Αλγόριθμος 2.2 παρουσιάζει μία τυπική υλοποίηση του σχήματος του

πυρήνα SpMV χρησιμοποιώντας το σχήμα CSR. Τα χαρακτηριστικά επίδοσης αυτού του πυρήνα και οι δυνατότητες βελτιστοποίησής του θα συζητηθούν με λεπτομέρεια σε επόμενα κεφάλαια.

```
1: procedure MATVEC_CSR( $A::in$ ,  $x::in$ ,  $y::out$ )
    $A$ : matrix in CSR format
    $x$ : input vector
    $y$ : output vector
2:   for  $i \leftarrow 0$  to  $N$  do
3:     for  $j \leftarrow rowptr[i]$  to  $rowptr[i + 1]$  do
4:        $y[i] \leftarrow y[i] + values[j] \cdot x[colind[j]]$ 
5:     end for
6:   end for
```

Αλγόριθμος 2.2: Υλοποίηση του πυρήνα SpMV kernel με χρήση του σχήματος CSR.

2.2 Εκμετάλλευση των πυκνών δομών του αραιού πίνακα

Παρά την σχετικά συμπαγή αναπαράστασή του, το σχήμα CSR έχει πολλή πλεονάζουσα πληροφορία στην δομή `colind`, η οποία αποθηκεύει τις στήλες των μη μηδενικών στοιχείων. Τα μη μηδενικά στοιχεία των αραιών πινάκων που προκύπτουν από πραγματικές εφαρμογές παρουσιάζουν κάποια κανονικότητα στην δομή τους. Για παράδειγμα, κάποιες στοιχεία μπορεί να είναι διατεταγμένα σε οριζόντιες, κατακόρυφες ή διαγώνιες ακολουθίες, ενώ άλλα μπορεί να σχηματίζουμε μικρά δισδιάστατα πυκνά μπλοκ. Στο εξής θα ονομάζουμε κάθε μία τέτοια ακολουθία μη μηδενικών στοιχείων ως *κανονικότητα*. Καθώς η ακολουθία των μη μηδενικών στοιχείων εντός μίας κανονικότητας είναι γνωστή εξ ορισμού, θα μπορούσε κανείς να χρησιμοποιήσει μόνο μία στήλη ανά κανονικότητα για να περιγράψει όλα τα στοιχεία της, μειώνοντας επομένως το μέγεθος του πίνακα `colind`. Η τεχνική αυτή της ομαδοποίησης των γειτονικών μη μηδενικών στοιχείων και η αναπαράστασή τους με μία στήλη είναι επίσης γνωστή ως *τεχνική οργάνωσης σε μπλοκ*. Εκτός από το εμφανές πλεονέκτημα της μείωσης του μεγέθους της δομής δεικτοδότησης των στηλών του πίνακα, οι τεχνικές οργάνωσης σε μπλοκ έχουν μία σημαντική παράπλευρη ιδιότητα: παρέχουν τα μέσα και την δυνατότητα για περαιτέρω βελτιστοποίηση του υπολογιστικού τμήματος του πυρήνα SpMV [Im και Yelick, 2001; Karakasis et al., 2009b]. Ο εσωτερικός βρόχος του πυρήνα SpMV, ο οποίος υπολογίζει το σημειακό γινόμενο μεταξύ ενός διανύσματος-γραμμής και του πίνακα εισόδου (Αλγόριθμος 2.2, γραμμές 3–5) δεν προχωρά στοιχείο-στοιχείο, αλλά

κανονικότητα-κανονικότητα, εκθέτοντας περαιτέρω το υπολογιστικό μέρος του πυρήνα και επιτρέποντας συγκεκριμένες βελτιστοποιήσεις.

Διάφορα σχήματα αποθήκευσης σε μπλοκ έχουν προταθεί, κάθε ένα εκμεταλλεύόμενο διαφορετικούς τύπους κανονικοτήτων. Ωστόσο, μπορούμε να ξεχωρίσουμε δύο μεγάλες κατηγορίες μεθόδων μπλοκ: μεθόδους μπλοκ σταθερού μεγέθους και μεθόδους μπλοκ μεταβλητού μεγέθους. Στην συνέχεια, περιγράφουμε αυτές τις κατηγορίες με περισσότερη λεπτομέρεια και παρουσιάζουμε τα πιο αντιπροσωπευτικά σχήματα από κάθε κατηγορία. Παρουσιάζουμε, επίσης, σύντομα κάποιες πιο εξειδικευμένες προσεγγίσεις.

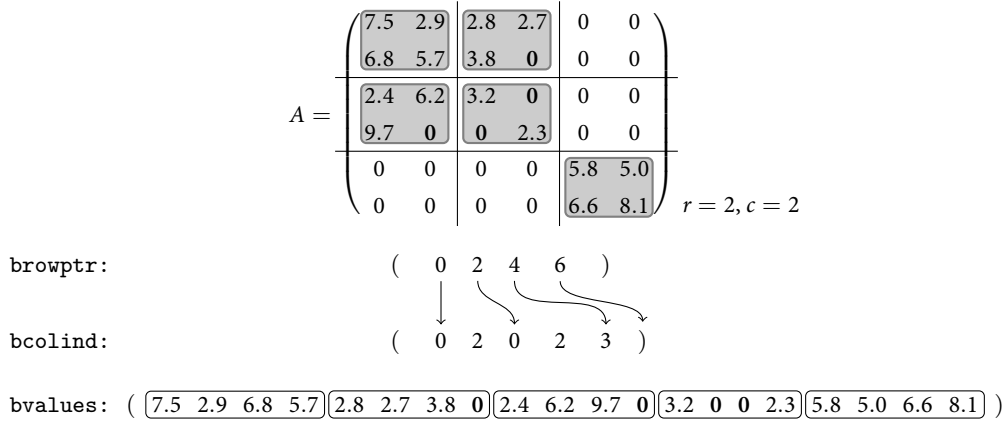
2.2.1 Σχήματα σταθερού μεγέθους μπλοκ

Τα σχήματα αποθήκευσης σταθερού μεγέθους μπλοκ εκμεταλλεύονται μονοδιάστατες ή διδιάστατες κανονικότητες, προσπαθώντας να δημιουργήσουν πλήρη μπλοκ σταθερού μεγέθους. Το μέγεθος και οι διαστάσεις των μπλοκ είναι σταθερές για ολόκληρο τον πίνακα, ενώ προστίθενται επιπλέον μηδενικά στοιχεία, ώστε να σχηματιστούν πλήρη μπλοκ. Αυτές οι μέθοδοι εμπεριέχουν ένα αρχικό βήμα προ-επεξεργασίας, κατά το οποίο ανιχνεύεται ο πίνακας για μπλοκ και στη συνέχεια λαμβάνεται απόφαση για το πιο μέγεθος μπλοκ θα χρησιμοποιηθεί τελικά. Η απόφαση αυτή βασίζεται κυρίως στην ελαχιστοποίηση των έξτρα μηδενικών στοιχείων [Im και Yelick, 2001; Im et al., 2004], ωστόσο υπάρχουν προσεγγίσεις που λαμβάνουν υπόψη τους τα υπολογιστικά χαρακτηριστικά του κώδικα που πρόκειται εκτελεστεί [Karakasis et al., 2009c].

Το πιο χαρακτηριστικό σχήμα αποθήκευσης αυτής της κατηγορίας είναι το σχήμα *Blocked Compressed Sparse Row (BCSR)* [Roach και Nieder, 1973; Saad, 1992; Im και Yelick, 2001]. Το σχήμα BCSR είναι στην ουσία η έκδοση με μπλοκ του καθιερωμένου σχήματος CSR, όπου αποθηκεύονται μπλοκ αντί για μεμονωμένα μη μηδενικά στοιχεία. Το Σχήμα 2.3 παρουσιάζει ένα παράδειγμα υλοποίησης του σχήματος BCSR. Η δομή `bcslind` αποθηκεύει την στήλη του πάνω αριστερά στοιχείου του κάθε μπλοκ, ενώ η δομή `brwptr` χωρίζει τον πίνακα κατά γραμμές-μπλοκ. Τα μη μηδενικά στοιχεία αποθηκεύονται κατά γραμμές ανά μπλοκ. Το σχήμα BCSR επιβάλλει μία αυστηρή στοίχιση στα μπλοκ του, απαιτώντας κάθε $r \times c$ μπλοκ να ξεκινά στα όρια r γραμμών και c στηλών, αντίστοιχα. Παρόλο που αυτή η στοίχιση συμβάλλει σε μια πιο γρήγορη προ-επεξεργασία του πίνακα, οδηγεί συνήθως σε αύξηση των ανώφελων επιπλέον μηδενικών στοιχείων. Παραλλαγές του BCSR που χαλαρώνουν αυτούς του περιορισμούς έχουν προταθεί από τον Vuduc και Moon [2005].

Ο Αλγόριθμος 2.3 παρουσιάζει μία τυπική υλοποίηση του πυρήνα SpMV με χρήση του σχήματος BCSR. Ο εσωτερικός διπλός βρόχος στις γραμμές 7–11 υπολογίζει το γινόμενο πίνακα-επί-διάνυσμα για ένα $r \times c$ μπλοκ. Στην πράξη, οι διαστάσεις του μπλοκ (r, c) είναι σταθερές και δεδομένες κατά τον

2. Αποθήκευση Αραιών Πινάκων



Σχήμα 2.3: Το σχήμα αποθήκευσης Blocked Compressed Sparse Row.

χρόνο μεταγλώττισης, επομένως, ο προαναφερθείς βρόχος μπορεί να βελτιστοποιηθεί σημαντικά με την χρήση κοινών τεχνικών βελτιστοποίησης των σύγχρονων μεταγλωττιστών, όπως *loop unrolling*, *vectorization* κ.λπ. Πράγματι, η βιβλιοθήκη βελτιστοποίησης αραιών πινάκων OSKI, η οποία είναι μία «έτοιμη» υλοποίηση του BCSR, υλοποιεί εξειδικευμένες εκδόσεις του πυρήνα για όλα τα μπλοκ μέχρι 8×8 , ενώ καταλήγει στην γενικευμένη υλοποίηση του Αλγορίθμου 2.3 στις περιπτώσεις μεγαλύτερων μπλοκ [Vuduc et al., 2005].

Μία κοινή κανονικότητα που εμφανίζεται σε πολλούς αραιούς πίνακες είναι ακολουθίες διαγωνίων στοιχείων, τα οποία δεν βρίσκονται απαραίτητα στην κύρια διαγώνιο του πίνακα. Δεδομένης της φύσης του, το σχήμα BCSR δεν μπορεί καθόλου να χειριστεί τέτοιου είδους κανονικότητες και εισάγει υπερβολικό αριθμό επιπρόσθετων μηδενικών στοιχείων. Μία κοινή προσέγγιση για την εφαρμογή τεχνικών μπλοκ σε αυτούς του πίνακες είναι ο διαχωρισμός του πίνακα σε ζώνες σταθερού μεγέθους και η εν συνεχεία προσπάθεια κατασκευής διαγωνίων πλήρων μπλοκ χρησιμοποιώντας επιπρόσθετα μηδενικά στοιχεία. Το σχήμα αυτό είναι γνωστό ως *Row Segmented Diagonal (RSDIAG)* ή *Blocked Compressed Sparse Diagonal (BCSD)*, αντίστοιχα με το σχήμα BCSR [Agarwal et al., 1992; Vuduc, 2003; Karakasis et al., 2009a]. Το Σχήμα 2.4 παρουσιάζει ένα παράδειγμα υλοποίησης του σχήματος RSDIAG.

Παρά την απλότητα της αναπαράστασής τους και την εύκολη υλοποίησή τους, η οποία παρέχει αρκετές δυνατότητες βελτιστοποίησης, οι μέθοδοι μπλοκ σταθερού μεγέθους πέφτουν συνήθως θύματα της τεχνικής της προθήκης μηδενικών στοιχείων που χρησιμοποιούν για την κατασκευή πλήρων μπλοκ. Υπάρχουν συχνά περιπτώσεις πινάκων χωρίς την επιθυμητή κατανομή μη μηδενικών στοιχείων, π.χ., πίνακες με πολλά διαγώνια στοιχεία στην περίπτωση του BCSR, όπου το μέγεθος αναπαράστασης του πίνακα ξεπερνά κα-

```

1: procedure MATVECBCSR(A::in, x::in, y::out, r::in, c::out)
  A: matrix in BCSR format
  x: input vector
  y: output vector
  r, c: block dimensions
2:    $i_r \leftarrow 0$ 
3:   for  $i \leftarrow 0$  to  $N$  step by  $r$  do
4:     for  $j \leftarrow \text{browptr}[i_r]$  to  $\text{browptr}[i_r + 1]$  step by  $r \cdot c$  do
5:        $j_b \leftarrow \frac{j}{r \cdot c}$ 
6:        $x_0 \leftarrow \text{bcolind}[j_b]$ 
7:       for  $k \leftarrow 0$  to  $r$  do
8:         for  $l \leftarrow 0$  to  $c$  do
9:            $y[i + k] = y[i + k] + \text{bvalues}[j + k \cdot c + l] \cdot x[x_0 + l]$ 
10:        end for
11:      end for
12:    end for
13:     $i_r \leftarrow i_r + 1$ 
14:  end for

```

Αλγόριθμος 2.3: Υλοποίηση του πυρήνα SpMV με χρήση του σχήματος BCSR.

$$A = \begin{pmatrix}
 7.5 & 2.9 & 2.8 & 2.7 & 0 & 0 \\
 6.8 & 5.7 & 3.8 & 0 & 0 & 0 \\
 2.4 & 6.2 & 3.2 & 0 & 0 & 0 \\
 9.7 & 0 & 0 & 2.3 & 0 & 0 \\
 0 & 0 & 0 & 0 & 5.8 & 5.0 \\
 0 & 0 & 0 & 0 & 6.6 & 8.1
 \end{pmatrix} \quad b = 2$$

Σχήμα 2.4: Το σχήμα αποθήκευσης Row Segmented Diagonal.

τά πολύ το μέγεθος αναπαράστασης του βασικού σχήματος CSR, οδηγώντας ακόμη και σε σημαντική επιβάρυνση της επίδοσης. Το σχήμα και ο προσανατολισμός των σχηματιζόμενων μπλοκ παίζει σημαντικό ρόλο στην τελική μείωση του μεγέθους αναπαράστασης του πίνακα. Είναι επομένως απαραίτητο να υπάρχει ένα βήμα προ-επεξεργασίας για την ανίχνευση του πίνακα και την λήψη της τελικής απόφασης ως προς το βέλτιστο μπλοκ. Η χρήση ευρετικών συναρτήσεων αποτελεί κοινή πρακτική για την επιλογή του καταλληλότερου μπλοκ για τον πίνακα εισόδου. Οι Im και Yelick [2001] εκτιμούν το πλήθος των πρόσθετων μηδενικών στοιχείων που απαιτούνται για κάθε υποψήφιο μπλοκ, χρησιμοποιώντας ομοιόμορφο τυχαίο δειγματισμό του πίνακα, και συνδέζουν αυτή τη μετρική με μία εκτίμηση της επίδοσης του υποψήφιου μπλοκ, η οποία λαμβάνεται μέσω μίας offline διαδικασίας. Οι Vuduc et al. [2002] βελτιώνουν στα σημεία αυτή την ευρετική και παρέχουν άνω και κάτω όρια της επίδοσης του BCSR, μοντελοποιώντας την συμπεριφορά της κρυφής μνήμης. Στο επόμενο κεφάλαιο, θα παρουσιάσουμε την δική μας προσέγγιση για την επιλογή του βέλτιστου μπλοκ για το σχήμα BCSR, η οποία εκτός από την εκτίμηση του τελικού μεγέθους αναπαράστασης του πίνακα, συνυπολογίζει τα χαρακτηριστικά επίδοσης των υποψηφίων μπλοκ.

Απαιτήσεις χώρου αποθήκευσης

Παρά το γεγονός ότι τα σχήματα αποθήκευσης με μπλοκ σταθερού μεγέθους μπορούν να μειώσουν σημαντικά το μέγεθος της αναπαράστασης σε περιπτώσεις πινάκων με μία σχετικά κανονική διάταξη των μη μηδενικών στοιχείων τους, οι δυνατότητες συμπίεσης αυτών των μεθόδων παραμένουν περιορισμένες. Ας υποθέσουμε ένα $N \times N$ αραιό πίνακα με NNZ μη μηδενικά στοιχεία, που μπορούν να ομαδοποιηθούν ιδανικά σε μπλοκ $r \times c$ χωρίς την προσθήκη επιπλέον μηδενικών στοιχείων ($r, c \ll N$) χρησιμοποιώντας το σχήμα BCSR. Το μέγεθος της δομής `bcslind` θα είναι τότε $4 \frac{NNZ}{r \times c} = \Theta(NNZ)$ bytes, παραμένοντας στην τάξη μεγέθους NNZ , όπως και στην περίπτωση του σχήματος CSR.

2.2.2 Σχήματα μεταβλητού μεγέθους μπλοκ

Τα σχήματα αποθήκευσης μπλοκ μεταβλητού μεγέθους αποφεύγουν την χρήση επιπροσθέτων μηδενικών στοιχείων μέσω της κατασκευής μπλοκ μεταβλητού μεγέθους. Για να το επιτύχουν αυτό, εισάγουν επιπλέον δομές δεδομένων που αποθηκεύουν είτε το μέγεθος κάθε μπλοκ είτε την αρχική στήλη και γραμμή κάθε μπλοκ. Το συνολικό μέγεθος αυτών των επιπλέον δομών δεδομένων μπορεί να διατηρηθεί σε χαμηλά επίπεδα χρησιμοποιώντας μικρούς ακεραίους των ένα ή δύο byte.

$$A = \begin{pmatrix} 7.5 & 2.9 & 2.8 & 2.7 & 0 & 0 \\ 6.8 & 5.7 & 3.8 & 0 & 0 & 0 \\ 2.4 & 6.2 & 3.2 & 0 & 0 & 0 \\ 9.7 & 0 & 0 & 2.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.8 & 5.0 \\ 0 & 0 & 0 & 0 & 6.6 & 8.1 \end{pmatrix}$$

rowptr: (0 4 7 10 12 14 16)

bcolind: (0 0 0 0 3 4 4)

bsize: (4 3 3 1 1 2 2)

bvalues: (7.5 2.9 2.8 2.7 6.8 5.7 3.8 2.4 6.2 3.2 9.7 2.3 5.8 5.0 6.6 8.1)

Σχήμα 2.5: Το σχήμα αποθήκευσης Variable Block Length storage.

Τα πιο αντιπροσωπευτικά σχήματα αποθήκευσης σε αυτή την κατηγορία είναι το σχήμα *Variable Block Length (VBL)*, το οποίο εκμεταλλεύεται μονοδιάστατα οριζόντια μπλοκ, και το σχήμα *Variable Block Row (VBR)*, το οποίο εκμεταλλεύεται δισδιάστατα μπλοκ [Saad, 1994; Pinar και Heath, 1999; Vuduc και Moon, 2005].

Το Σχήμα 2.5 παρουσιάζει μία τυπική υλοποίηση του σχήματος VBL. Οι πίνακες `values` και `rowptr` είναι οι ακριβώς οι ίδιοι όπως στο σχήμα CSR, καθώς το σχήμα VBL κατασκευάζει μόνο μονοδιάστατα οριζόντια μπλοκ. Ο πίνακας `bcolind` αποθηκεύει την στήλη του πρώτου στοιχείου κάθε μπλοκ, ενώ ο πίνακας `bsize` αποθηκεύει το μέγεθος κάθε μπλοκ. Καθώς δεν συναντώνται συχνά πολύ μεγάλα μπλοκ στους αραιούς πίνακες, η χρήση ενός byte για την αναπαράσταση του μεγέθους των μπλοκ στον πίνακα `bsize` είναι αρκετή στις περισσότερες των περιπτώσεων· σε περιπτώσεις πολύ μεγάλων μπλοκ, ωστόσο, αυτά χωρίζονται σε κομμάτια των 255 στοιχείων. Ένα χαρακτηριστικό του σχήματος VBL είναι ότι όλα τα μη μηδενικά στοιχεία του πίνακα οργανώνονται σε μπλοκ· ακόμα και τα μεμονωμένα στοιχεία σχηματίζουν εκφυλισμένα μπλοκ μεγέθους ένα. Είναι φανερό ότι για πίνακες με ακανόνιστη ή με μη οριζόντια οργάνωση των στοιχείων τους, η επιβάρυνση λόγω της δομής για τα μεγέθη των μπλοκ θα είναι σημαντική για τον VBL, μειώνοντας την δυναμικότητά του συμπίεσης του πίνακα.

Ο Αλγόριθμος 2.4 παρουσιάζει μία υλοποίηση του πυρήνα SpMV χρησιμοποιώντας το σχήμα VBL. Ο αλγόριθμος είναι πολύ όμοιος με την υλοποίηση για το σχήμα CSR, με την διαφορά ότι διατηρεί την πληροφορία του τρέχοντος μπλοκ που διασχίζεται κάθε στιγμή. Καθώς το μέγεθος των μπλοκ δεν είναι σταθερό, ο αλγόριθμος διατηρεί επίσης την πληροφορία του μεγέθους

2. Αποθήκευση Αραιών Πινάκων

του τρέχοντος μπλοκ. Αυτό προσθέτει επιπλέον πράξεις στον αλγόριθμο και αυξάνει την επιβάρυνση του εσωτερικού βρόχου, ένα πρόβλημα που γίνεται ιδιαίτερα αισθητό σε περιπτώσεις πολύ μικρών μπλοκ.

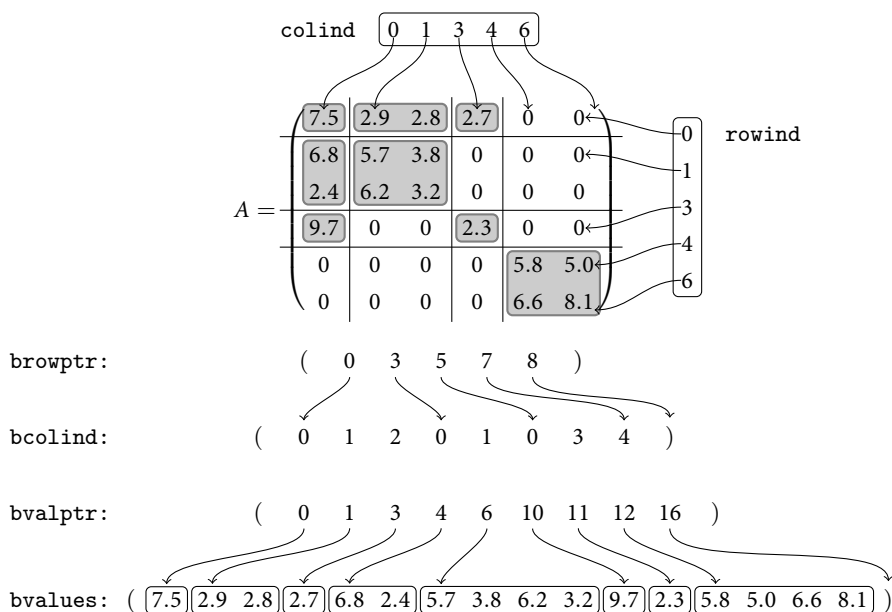
```
1: procedure MATVECVBL( $A::in, x::in, y::out$ )
    $A$ : matrix in VBL format
    $x$ : input vector
    $y$ : output vector
2:    $i_b \leftarrow 0$ 
3:   for  $i \leftarrow 0$  to  $N$  do
4:      $j \leftarrow rowptr[i]$ 
5:      $s \leftarrow bsize[i_b]$ 
6:     while  $j < rowptr[i + 1]$  do
7:        $b_0 \leftarrow bcolind[i_b]$ 
8:       for  $k \leftarrow 0$  to  $s$  do
9:          $y[i] \leftarrow y[i] + values[j + k] * x[b_0 + k]$ 
10:      end for
11:      $j \leftarrow j + s$ 
12:      $i_b \leftarrow i_b + 1$ 
13:      $s \leftarrow bsize[i_b]$ 
14:   end for
```

Αλγόριθμος 2.4: Υλοποίηση του πυρήνα SpMV με χρήση του σχήματος VBL.

Ένα βήμα παραπέρα από το VBL είναι το σχήμα VBR, το οποίο παρουσιάζεται στο Σχήμα 2.6 και μπορεί να εκμεταλλευτεί διαστάσεις κανονικότητας μεταβλητού μεγέθους. Το σχήμα VBR χωρίζει τον πίνακα σε γραμμές και στήλες μπλοκ με μεταβλητές διαστάσεις. Τα μη μηδενικά στοιχεία αποθηκεύονται στον πίνακα `bvalues` κατά γραμμές ανά μπλοκ, ενώ οι πίνακες `rowind` και `colind` αποθηκεύουν τις αρχικές γραμμές και στήλες, αντίστοιχα. Η δομή `bvalptr` σημειώνει την αρχή κάθε μπλοκ στον πίνακα `bvalues`, ενώ η δομή `bcolind` αποθηκεύει την στήλη του μπλοκ αναφερόμενη στον πίνακα `colind`. Τέλος, ο πίνακας `browptr` σημειώνει την αρχή κάθε γραμμής μπλοκ δεικτοδοτώντας τους πίνακες `bcolind` και `bvalptr`.

Παρόλο που το σχήμα VBR μπορεί να ανιχνεύσει οποιοδήποτε διαστάσεις κανονικότητες στον αραιό πίνακα, η υπολογιστική και χωρική επιβάρυνση των υποστηρικτικών δομών δεδομένων μπορεί να γίνει υπερβολική. Για μεγάλους πίνακες, ολόκληρη η πληροφορία δεικτοδότησης θα απαιτεί κατά πάσα πιθανότητα ακεραίους 32-bit, εξαφανίζοντας σε μεγάλο βαθμό το όφελος της ομαδοποίησης των μη μηδενικών στοιχείων σε μπλοκ. Συμπύση των

2.2. Εκμετάλλευση των πυκνών δομών του αραιού πίνακα



Σχήμα 2.6: Το σχήμα αποθήκευσης Variable Block Row.

δομών δεδομένων του VBR θα μπορούσε να επιτευχθεί, εάν οι πίνακες rowind και colind αποθήκευαν τις διαστάσεις των σχηματιζόμενων μπλοκ, οποία περίπτωση, η χρήση ακεραίων ενός byte θα ήταν αρκετή για αυτές τις δομές. Ακόμα και σε αυτή την περίπτωση, όμως, το σχήμα VBR έχει δύο επιπλέον δομές δεδομένων συγκρινόμενο με το πολύ απλούτερο σχήμα VBL. Επιπλέον, όλη αυτή η μετα-πληροφορία δεικτοδότησης προσθέτει υπολογιστική επιβάρυνση, καθώς απαιτούνται πολλαπλά βήματα για την προσπέλαση των τελικών μη μηδενικών στοιχείων.

Απαιτήσεις χώρου αποθήκευσης

Οι δυνατότητες συμπίεσης των σχημάτων αποθήκευσης σε μπλοκ μεταβλητού μεγέθους είναι μεγαλύτερες σε σχέση με τα σχήματα σταθερών μπλοκ, ειδικά στις περιπτώσεις σχημάτων χωρίς πολλές επιπλέον δομές δεδομένων. Θεωρώντας την περίπτωση του VBL, ας υποθέσουμε ένα $N \times N$ αραιό πίνακα με NNZ_r μη μηδενικά στοιχεία ανά γραμμή και ότι k , $k \ll NNZ_r$, μπλοκ σχηματίζονται ανά γραμμή κατά μέσο όρο. Καθότι σε ένα συνήχη αραιό πίνακα ισχύει $NNZ_r \ll N$, το k μπορεί να θεωρηθεί σταθερό. Επομένως, το μέγεθος της συμπίεσμένης δομής bcolind θα είναι $4kN$ και το απαιτούμενο μέγεθος για την αποθήκευση των μεγεθών των μπλοκ θα είναι kN (θεωρώντας αναπαράσταση μεγέθους ενός byte). Προσθέτοντας, οδηγούμαστε σε ένα μέγε-

θος $5kN = \Omega(N)$ bytes, το οποίο μας οδηγεί σε ένα συνολικό μέγεθος του πίνακα πολύ κοντά στο θεωρητικό ελάχιστο². Παρόλα αυτά, σε αραιούς πίνακες όπου τα μη μηδενικά τους στοιχεία είναι οργανωμένα σε μη οριζόντιες διατάξεις, το σχήμα VBL θα δημιουργήσει εκφυλισμένα μπλοκ μεγέθους ένα και, επομένως, ο αριθμός των μπλοκ ανά γραμμή δεν μπορεί να αποσυνδεθεί από τα συνολικά μη μηδενικά στοιχεία του πίνακα. Σε αυτή την περίπτωση, το μέγεθος της δομής `bcolind` θα είναι της τάξης του NNZ . Ωστόσο, αυτό δεν μπορεί να υπερσκιιάσει τις υψηλότερες δυνατότητες συμπίεσης των σχημάτων αποθήκευσης με μπλοκ μεταβλητού μεγέθους σε σχέση με τα αντίστοιχα των μπλοκ σταθερού μεγέθους, των οποίων οι δυνατότητες περιορίζονται κυρίως από τον μεγάλο αριθμό σχηματιζόμενων μπλοκ. Στην πράξη, το σχήμα VBL επιτυγχάνει σχεδόν πάντοτε υψηλότερους βαθμούς συμπίεσης από το σχήμα BCSR.

2.2.3 Άλλες προσεγγίσεις

Οι προσεγγίσεις που έχουμε μέχρι στιγμής συζητήσει εκμεταλλεύονται ένα μοναδικό τύπο κανονικότητας στον αραιό πίνακα, αγνοώντας τους υπολοίπους και οδηγούν σε μείωση της επίδοσης σε «ακατάλληλους» πίνακες. Μία κοινή τεχνική για την εκμετάλλευση πολλαπλών κανονικότητας είναι η διάσπαση του πίνακα εισόδου σε πολλαπλούς πίνακες, κάθε ένας από τους οποίους εκμεταλλεύεται μία διαφορετική κανονικότητα, ενώ το άθροισμά τους παράγει τον αρχικό πίνακα [Agarwal et al., 1992; Geus και Röllin, 2001]. Ο τελευταίος προσθετός είναι πάντα ένας πίνακας σε σχήμα CSR, ο οποίος περιέχει τα εναπομείναντα στοιχεία που δεν ομαδοποιήθηκαν σε κάποια κανονικότητα. Αυτά τα σχήματα αποθήκευσης αποφεύγουν την χρήση επιπρόσθετων μηδενικών στοιχείων και, επομένως, το μέγεθος αναπαράστασής του μπορεί να διατηρηθεί σε χαμηλά επίπεδα. Ο πολλαπλασιασμός πίνακα-επί-διάνυσμα με την χρήση αυτών των σχημάτων, αποτελείται από την εκτέλεση του πυρήνα `SrMV` για κάθε ένα προσθετό και την εν συνεχεία συσσώρευση των ενδιάμεσων αποτελεσμάτων στο τελικό διάνυσμα εξόδου. Ωστόσο, υπάρχουν δύο πιθανά προβλήματα επίδοσης με τα διασπώμενα σχήματα αποθήκευσης. Πρώτον, η συσσώρευση των ενδιάμεσων αποτελεσμάτων μπορεί να δημιουργήσει επιπλέον επιβάρυνση και να περιορίσει τον παραλληλισμό, δεύτερον, η μεγάλη διάσπαση των μη μηδενικών στοιχείων στους πολλούς προσθετούς αυξάνει την αραιότητα

² Το θεωρητικό ελάχιστο του μεγέθους ενός αραιού πίνακα υπολογίζεται εάν αγνοήσουμε τελείως τις δομές δεικτοδότησης, θεωρώντας, για παράδειγμα, ένα φανταστικό πίνακα, του οποίου οι θέσεις των μη μηδενικών στοιχείων μπορούν να υπολογιστούν στον χρόνο εκτέλεσης. Εφόσον για τους περισσότερους πίνακες ισχύει $N \ll NNZ$, η συμπίεση των δομών δεικτοδότησης στην τάξη του N μπορεί να οδηγήσει σε μεγέθη πολύ κοντά στο θεωρητικό ελάχιστο

των επιμέρους πινάκων και μπορεί να οδηγήσει σε μείωση της επίδοσης του πυρήνα SpMV.

Η πιο πρόσφατη προσέγγιση που χρησιμοποιεί την τεχνική της διάσπασης είναι το σχήμα αποθήκευσης *Pattern Block Row (PBR)* [Belgin et al., 2009]. Το σχήμα PBR χωρίζει τον πίνακα σε τετράγωνα μπλοκ μεγέθους μέχρι 8×8 και κωδικοποιεί την διάταξη των μη μηδενικών στοιχείων σε κάθε μπλοκ χρησιμοποιώντας ένα διάνυσμα 64-bit. Στη συνέχεια, διασπά τον πίνακα σε πολλούς προσθετέους, ένα για κάθε διαφορετική διάταξη των στοιχείων του μπλοκ· τα υπόλοιπα στοιχεία αποθηκεύονται στον τελευταίο προσθετέο σε μορφή CSR. Εφόσον η διάταξη των μη μηδενικών στοιχείων δεν είναι γνωστή εκ των προτέρων, το σχήμα PBR παράγει στον χρόνο εκτέλεσης ρουτίνες SpMV εξειδικευμένες για κάθε διάταξη των στοιχείων των μπλοκ.

Ακόμη μία ενδιαφέρουσα προσέγγιση στα σχήματα αποθήκευσης σε μπλοκ είναι το σχήμα *Compressed Sparse Block (CSB)* [Buluc et al., 2009]. Το κίνητρο πίσω από αυτό το σχήμα είναι η αποδοτική υποστήριξη αμοιβαίων των πράξεων Ax και $A^T x$, εκ των οποίων η δεύτερη είναι λιγότερο αποδοτική με τα συμβατικά σχήματα αποθήκευσης κατά γραμμές. Για τον λόγο αυτό, το σχήμα CSB χωρίζει τον πίνακα σε μεγάλα τετράγωνα αραιά μπλοκ, τα οποία αποθηκεύονται με το σχήμα αποθήκευσης *Coordinate*, χρησιμοποιώντας μικρούς ακεραίους για τις γραμμές και τις στήλες. Στον χρόνο εκτέλεσης, χρησιμοποιείται η τεχνική του παραλληλισμού εργασιών (*task parallelism*) για την χρονοδομολόγηση της εκτέλεσης των επιμέρους μπλοκ.

Συμπύκνωση των μη μηδενικών στοιχείων

Οι τιμές των μη μηδενικών στοιχείων καταναλώνουν τα $2/3$ του συνολικού μεγέθους του σχήματος CSR. Η εύκολη λύση της χρήσης αριθμών κινητής υποδιαστολής απλής ακρίβειας δεν αποτελεί επιλογή σε πραγματικές εφαρμογές, καθώς η ακρίβεια 64-bit είναι σημαντική για την σταθερότητα και την σύγκλιση των επαναληπτικών μεθόδων. Η πιο κοινή προσέγγιση που χρησιμοποιείται για την συμπίεση των μη μηδενικών στοιχείων είναι η δεικτοδότησή τους, δηλαδή, η αποθήκευση μόνο των μοναδικών τιμών και η χρήση μίας επιπλέον δομής δεικτοδότησης για την προσπέλασή τους. Η ιδέα πίσω από αυτή την τεχνική είναι ότι τα μη μηδενικά στοιχεία του πίνακα συντελεστών του συστήματος έχουν συχνά την ίδια τιμή. Δύο προσεγγίσεις που χρησιμοποιούν αυτή την ιδιότητα είναι το σχήμα αποθήκευσης *Super-Sparse (SS)* του Escudero [1984] και το σχήμα *CSR Value Indexed (CSR-VI)* των Kourtis et al. [2008b]. Το μειονέκτημα των τεχνικών δεικτοδότησης των μη μηδενικών στοιχείων είναι ότι η δομή δεικτοδότησης θα πρέπει να κατασκευάζεται εκ νέου κάθε φορά που αλλάζει ένα μη μηδενικό στοιχείο, καθιστώντας με αυτό τον τρόπο τα σχήματα αυτά λιγότερο ευέλικτα.

$$A = \begin{pmatrix} 7.5 & 2.9 & 2.8 & 2.7 & 0 & 0 \\ 6.8 & 5.7 & 3.8 & 0 & 0 & 0 \\ 2.4 & 6.2 & 3.2 & 0 & 0 & 0 \\ 9.7 & 0 & 0 & 2.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.8 & 5.0 \\ 0 & 0 & 0 & 0 & 6.6 & 8.1 \end{pmatrix}$$

$$\text{values:} \begin{pmatrix} 7.5 & 2.9 & 2.8 & 2.7 \\ 6.8 & 5.7 & 3.8 & 0 \\ 2.4 & 6.2 & 3.2 & 0 \\ 9.7 & 2.3 & 0 & 0 \\ 5.8 & 5.0 & 0 & 0 \\ 6.6 & 8.1 & 0 & 0 \end{pmatrix} \quad \text{colind:} \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & * \\ 0 & 1 & 2 & * \\ 0 & 3 & * & * \\ 4 & 5 & * & * \\ 4 & 5 & * & * \end{pmatrix}$$

Σχήμα 2.7: Το σχήμα αποθήκευσης Ellpack-Itrpack, κατάλληλο για διανυσματικούς επεξεργαστές και σύγχρονες αρχιτεκτονικές γραφικών (GPUs).

Σχήματα αποθήκευσης για συγκεκριμένες μικρο-αρχιτεκτονικές

Παρόλο που τα εξειδικευμένα σχήματα αποθήκευσης για συγκεκριμένους πίνακες ή αρχιτεκτονικές είναι εκτός του σκοπού αυτής της παρουσίασης, είναι σχετικό να παρουσιάσουμε σύντομα το σχήμα αποθήκευσης *Ellpack-Itrpack* (*ELL*, *ELLPACK* ή *ITPACK*) [Saad, 1992]. Το σχήμα αυτό κατασκευάστηκε αρχικά για τους διανυσματικούς επεξεργαστές [Orpe και Kincaid, 1987], αλλά πρόσφατα ήρθε πάλι στο προσκήνιο με την έλευση των επεξεργαστών γραφικών γενικού σκοπού. Το σχήμα *ELLPACK* αποθηκεύει ένα $N \times M$ αραιό πίνακα χρησιμοποιώντας δύο διαστάτους πίνακες (Σχήμα 2.7). Ο πίνακας *values*, διαστάσεων $N \times K$, αποθηκεύει τα μη μηδενικά στοιχεία κατά γραμμές, όπου K είναι ο μέγιστος αριθμός μη μηδενικών στοιχείων ανά γραμμή· οι γραμμές με λιγότερα μη μηδενικά στοιχεία συμπληρώνονται με μηδενικά. Ο πίνακας *colind* αποθηκεύει τις αντίστοιχες στήλες χρησιμοποιώντας άσχετες τιμές στην θέση μη υπαρκτών στοιχείων. Το πλεονέκτημα του σχήματος *ELLPACK* είναι ότι διευκολύνει τους υπολογισμούς κατά ροές (*streaming*) των διανυσματικών επεξεργασιών. Από την άλλη πλευρά, όμως, πίνακες με ακανόνιστη διάταξη των μη μηδενικών στοιχείων τους μπορούν να προσθέσουν σημαντικό αριθμό πρόσθετων μηδενικών στοιχείων. Οι Bell και Garland [2009] χρησιμοποιούν ένα συνδυασμό των σχημάτων *ELLPACK* και *COO* για την επιτάχυνση της επίδοσης του πυρήνα *Spmv* σε σύγχρονες αρχιτεκτονικές γραφικών (GPUs), ενώ οι Choi et al. [2010], εμπνεόμενοι από το σχήμα *BCSR*, επεκτείνουν το *ELLPACK* ώστε να αποθηκεύει γραμμές από μπλοκ.

Σχετικά με το σχήμα *ELLPACK* είναι, σε κάποιο βαθμό, και τα σχήματα

αποθήκευσης *Streaming CSR* (S-CSR) και *Streaming BCSR* (S-BCSR), τα οποία προτείνονται από τους Guo και Gropp [2011]. Το κίνητρο πίσω από αυτά τα σχήματα είναι η αποδοτική χρήση των μονάδων προφόρτωσης ροών δεδομένων των επεξεργαστών IBM POWER. Το σχήμα S-CSR χωρίζει τον πίνακα σε n ροές δεδομένων και αποθηκεύει τις γραμμές του με κυκλικό τρόπο έτσι, ώστε η γραμμή i να αποθηκεύεται στην ροή $i \bmod n$. Κάθε ροή γραμμών αποθηκεύεται με τη χρήση του σχήματος ELLPACK. Τόσο το σχήμα S-CSR όσο και έκδοσή του με μπλοκ, S-BCSR, έχουν το πλεονέκτημα της ταυτόχρονης εκμετάλλευσης n ανεξάρτητων ροών προφόρτωσης δεδομένων, οδηγώντας σε καλύτερη εκμετάλλευση του διαθέσιμου εύρους ζώνης μνήμης και υψηλότερη επίδοση του SpMV στην συγκεκριμένη αρχιτεκτονική.

2.3 Άμεση συμπίεση των δομών δεικτοδότησης του πίνακα

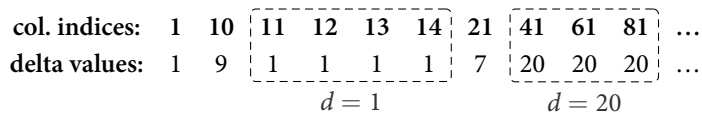
Μία εναλλακτική προσέγγιση για την ελαχιστοποίηση του μεγέθους αναπαράστασης του αραιού πίνακα είναι η άμεση συμπίεση της δομής `colind` του σχήματος CSR. Αντίθετα με τις μεθόδους αποθήκευσης σε μπλοκ, οι τεχνικές άμεσης συμπίεσης δεν εκμεταλλεύονται την δομή του πίνακα με κάποιο άμεσο τρόπο, αλλά θεωρούν τον πίνακα `colind` ως μία κοινή ακολουθία από ακεραίους 32-bit, της οποίας το μέγεθος προσπαθούν να ελαχιστοποιήσουν. Λόγω της μεγέθυνσης του προβλήματος «memory wall» στις σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών, τέτοιες προσεγγίσεις κερδίζουν συνεχώς το ενδιαφέρον πρόσφατα [Willcock και Lumsdaine, 2006; Kourtis et al., 2008b].

Η πιο διαδεδομένη προσέγγιση για την άμεση συμπίεση της δομής `colind` είναι η *διαφορική δεικτοδότηση* ή *κωδικοποίηση δέλτα* των στηλών των μη μηδενικών στοιχείων. Αντί της αποθήκευσης της πλήρους στήλης ενός μη μηδενικού στοιχείου, η τεχνική της διαφορικής δεικτοδότησης αποθηκεύει την διαφορά της στήλης του τρέχοντος στοιχείου με το προηγούμενό του (απόσταση δ) και χρησιμοποιεί την πλήρη στήλη μόνο για το πρώτο στοιχείο κάθε γραμμής [Rooch και Nieder, 1973]. Η ιδέα πίσω από την εφαρμογή της διαφορικής δεικτοδότησης σε αραιούς πίνακες είναι ότι η απόσταση δ είναι πολύ πιθανό να χωρά σε ένα μικρό ακέραιο, επιτρέποντας έτσι την σημαντική μείωση της δομής `colind`.

Ένα βήμα παραπέρα από την διαφορική δεικτοδότηση είναι *κωδικοποίηση run-length* των στηλών. Η κωδικοποίηση *run-length* συμπιέζει ακολουθίες αποστάσεων δ ομαδοποιώντας ίδιες τιμές αποστάσεων και αναπαριστώντας τις ως μία δυάδα που περιέχει την κοινή τιμή δ και το πλήθος των στοιχείων της ομάδας. Το Σχήμα 2.8 παρουσιάζει γραφικά την έννοια της κωδικοποίησης *run-length* των στηλών του αραιού πίνακα.

Παρόλο που η κωδικοποίηση δέλτα είχε συζητηθεί αρκετά παλαιότερα [Rooch

2. Αποθήκευση Αραιών Πινάκων



Σχήμα 2.8: Κωδικοποίηση run-length των στηλών του πίνακα.

και Nieder, 1973], χρησιμοποιήθηκε μόλις πρόσφατα για την βελτιστοποίηση του πυρήνα SpMV. Οι Willcock και Lumsdaine [2006] εφαρμόζουν την κωδικοποίηση δέλτα στις στήλες του πίνακα και προτείνουν το σχήμα αποθήκευσης *Delta-Coded Sparse Row (DCSR)*. Το σχήμα DCSR κωδικοποιεί την πληροφορία δεικτοδότησης του πίνακα (δομές `rowptr` και `colind`) ως μία ακολουθία από δυάδες της μορφής (*command, argument*). Το όρισμα *argument* είναι πάντοτε ένα byte, ενώ οι εντολές (όρισμα *command*, συνολικά έξι) είναι υπεύθυνες για την παραγωγή της γραμμής και της στήλης του τρέχοντος μη μηδενικού στοιχείου. Η διαδικασία της αποσυμπίεσης αποτελείται από την ανάγνωση και εφαρμογή της εντολής και, στην συνέχεια, την εκτέλεση του κατάλληλου υπολογισμού με το αντίστοιχο μη μηδενικό στοιχείο. Οι συγγραφείς προτείνουν επιπλέον μία παραλλαγή του DCSR, η οποία χρησιμοποιεί την τεχνική της κωδικοποίησης run-length για την ανίχνευση μέχρι τεσσάρων συνεχόμενων μη μηδενικών στοιχείων. Το μειονέκτημα του σχήματος DCSR, ωστόσο, είναι το μεγάλο κόστος αποσυμπίεσης, το οποίο για την απόσβεσή του και την επίτευξη υψηλής επίδοσης απαιτεί μη φορητές υλοποιήσεις.

Μία απλούστερη και πιο φορητή προσέγγιση για την κωδικοποίηση δέλτα των στηλών ενός αραιού πίνακα είναι το σχήμα αποθήκευσης *CSR Delta Units (CSR-DU)* των Kourtis et al. [2008b]. Το σχήμα CSR-DU θεωρεί τον πίνακα ως μία ακολουθία από μονάδες δέλτα, ορίζοντας τρεις τύπους μονάδων, ανάλογα με το πλήθος των byte που απαιτούνται για την αποθήκευση της απόστασης δ των στηλών (μονάδες ενός, δύο ή τεσσάρων byte). Το σχήμα CSR-DU αντικαθιστά αμφοτέρως τις δομές `rowptr` και `colind` με μία ακολουθία από byte, η οποία κωδικοποιεί τον τύπο, την αρχική στήλη και τις αποστάσεις δ κάθε μονάδας. Κατά την διάρκεια εκτέλεσης, το μονοπάτι εκτέλεσης του κώδικα του CSR-DU διαφοροποιείται ανάλογα με τον τύπο της μονάδας, ώστε να αποκωδικοποιηθούν σωστά οι αποστάσεις δ , και συνεχίζει με τις πράξεις του SpMV. Το σχήμα CSR-DU επεκτείνεται εύκολα για την υποστήριξη ακολουθιών μη μηδενικών στοιχείων σε οριζόντια διάταξη με αυθαίρετη σταθερή απόσταση δ μεταξύ τους με την χρήση της κωδικοποίησης run-length [Kourtis et al., 2010]. Με αυτή την επέκταση, το σχήμα CSR-DU μπορεί να θεωρηθεί ως μία γενίκευση του σχήματος VBL, αλλά σε μία πιο συμπαγή μορφή.

$$A = \begin{pmatrix} 7.5 & 6.8 & 2.4 & 9.7 & 0 & 0 \\ 6.8 & 5.7 & 6.2 & 0 & 0 & 0 \\ 2.4 & 6.2 & 3.2 & 0 & 0 & 0 \\ 9.7 & 0 & 0 & 2.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.8 & 6.6 \\ 0 & 0 & 0 & 0 & 6.6 & 8.1 \end{pmatrix}$$

rowptr: (0 1 3 3 4 5)

colind: (0 0 1 0 4)

values: (6.8 2.4 6.2 9.7 6.6)

dvalues: (7.5 5.7 3.2 2.3 5.8 8.1)

Σχήμα 2.9: Το σχήμα αποθήκευσης Symmetric Sparse Skyline.

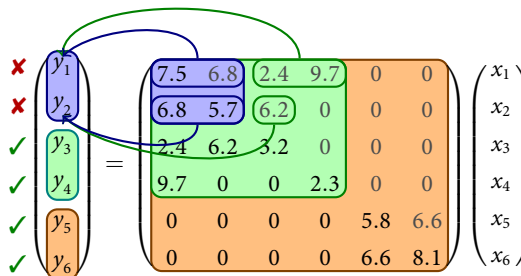
2.4 Εκμετάλλευση της συμμετρίας

Οι μέθοδοι πεπερασμένων στοιχείων συνεπάγονται συνήθως την λύση μεγάλων γραμμικών συστημάτων με αραιούς, δομημένους και συμμετρικούς πίνακες συντελεστών. Η εκμετάλλευση της συμμετρίας της δομής και των τιμών των μη μηδενικών στοιχείων μπορεί να μειώσει το μέγεθος της αναπαράστασης του πίνακα στο μισό, ελαφρύνοντας σημαντικά την πίεση στην ιεραρχία μνήμης της υποκείμενης αρχιτεκτονικής. Διάφορες μέθοδοι έχουν προταθεί στο παρελθόν για την βελτιστοποίηση του συμμετρικού πυρήνα SpMV.

Η πιο διαδεδομένη προσέγγιση για την αποθήκευση ενός αραιού πίνακα είναι το σχήμα αποθήκευσης *Symmetric Sparse Skyline* (SSS) [Eisenstat et al., 1982; Saad, 1992]. Το σχήμα SSS είναι στην ουσία η συμμετρική έκδοση του σχήματος CSR· το Σχήμα 2.9 παρουσιάζει ένα παράδειγμα υλοποίησης. Τα στοιχεία της κυρίας διαγωνίου, η οποία είναι πάντα πλήρης σε ένα καλώς ορισμένο πρόβλημα, αποθηκεύονται ξεχωριστά στον πίνακα `dvalues` array, ενώ ο κάτω τριγωνικός υποπίνακας αποθηκεύεται με την χρήση του καθιερωμένου σχήματος CSR.

Ο πυρήνας SpMV για συμμετρικούς πίνακες αποτελείται από την άπαξ διάσχιση των στοιχείων κάτω τριγωνικού υποπίνακα και την ταυτόχρονη εκτέλεση των πράξεων για τα στοιχεία τόσο του κάτω όσο και του άνω τριγωνικού τμήματος. Ο Αλγόριθμος 2.5 παρουσιάζει μία τυπική υλοποίηση του πυρήνα με το σχήμα αποθήκευσης SSS. Το κυριώτερο θέμα επίδοσης αυτού του πυρήνα είναι οι πράξεις εγγραφής στο διάνυσμα εξόδο λόγω των υπολογισμών στο άνω τριγωνικό τμήμα του πίνακα (Αλγόριθμος 2.5, γραμμή 7). Παρά το ότι οι προσβάσεις εγγραφής στο διάνυσμα εξόδου είναι κάπως ακανόνιστες,

2. Αποθήκευση Αραιών Πινάκων



Σχήμα 2.10: Η εξάρτηση RAW στο διάνυσμα εξόδου κατά την παράλληλη εκτέλεση του συμμετρικού πυρήνα SpMV. Οι διαμερίσεις των νημάτων φαίνονται με διαφορετικό χρώμα.

αυτό δεν αποτελεί σημαντικό πρόβλημα στην σειριακή εκτέλεση του πυρήνα. Ωστόσο, η παράλληλη εκτέλεση του πυρήνα είναι προβληματική, καθώς αυτές οι πράξεις εισάγουν εξαρτήσεις τύπου RAW σε συγκεκριμένα στοιχεία του διανύσματος εξόδου· το Σχήμα 2.10 παρουσιάζει γραφικά αυτά το πρόβλημα.

```

1: procedure MATVECSs(A::in, x::in, y::out)
   A: matrix in SSS format
   x: input vector
   y: output vector
2:   for i ← 0 to N do
3:     y[i] ← dvalues[i] · x[i]
4:     for j ← rowptr[i] to rowptr[i + 1] do
5:       k ← colind[j]
6:       y[i] ← y[i] + values[j] · x[k]
7:       y[k] ← y[k] + values[j] · x[i]
8:     end for
9:   end for

```

Αλγόριθμος 2.5: Υλοποίηση του πυρήνα SpMV με χρήση του συμμετρικού σχήματος αποθήκευσης SSS.

Το κόστος της προστασίας των προσβάσεων στο διάνυσμα εξόδου με την χρήση κλειδωμάτων μπορεί να γίνει απαγορευτικό και γι' αυτό τον λόγο δεν χρησιμοποιείται στην πράξη. Οι πιο διαδεδομένες προσεγγίσεις για την βελτιστοποίηση του συμμετρικού πυρήνα SpMV βασίζονται στην χρήση τοπικών, ανά νήμα, διανυσμάτων, τα οποία στο τέλος ανάγονται στο τελικό διάνυσμα εξόδου. Το κόστος της φάσης αυτής της αναγωγής μπορεί να γίνει επίσης σημαντικό, ειδικότερα καθώς αυξάνεται το πλήθος των νημάτων, και για τον λό-

γο αυτό είναι σημαντική η ελαχιστοποίηση της ποσότητας της πληροφορίας που ανταλλάσσεται μεταξύ των νημάτων. Τα «συγκρουόμενα» στοιχεία του διανύσματος εξόδου εξαρτώνται από την δομή του πίνακα και, πιο συγκεκριμένα, από το εύρος ζώνης του πίνακα. Το εύρος ζώνης ενός αραιού πίνακα είναι η μέγιστη απόσταση των μη μηδενικών του στοιχείων από την κύρια διαγώνιο· εάν το πιο απομακρυσμένο στοιχείο του κάτω τριγωνικού πίνακα βρίσκεται στην διαγώνιο l και το πιο απομακρυσμένο στοιχείο του πάνω τριγωνικού πίνακα βρίσκεται στην διαγώνιο u , τότε το εύρος ζώνης b του πίνακα ορίζεται ως $b = l + u + 1$. Είναι εύκολο να διαπιστώσει κανείς ότι όσο μεγαλύτερο είναι το εύρος ζώνης, τόσο μεγαλύτερη είναι η αλληλεπίδραση μεταξύ των νημάτων του συμμετρικού πυρήνα $SrMV$. Έχουν προταθεί διάφοροι αλγόριθμοι για την μείωση του εύρους ζώνης ενός πίνακα, βασιζόμενοι κυρίως στην μετάθεση των γραμμών και των στηλών του πίνακα [Cuthill και McKee, 1969; George, 1971; Gibbs et al., 1976; Karypis και Kumar, 1995; Çatalyürek και Aykanat, 1999]. Οι αλγόριθμοι αυτοί χρησιμοποιούν ευρετικές μεθόδους για την εύρεση της καλύτερης αναδιάταξης του πίνακα, καθώς το πρόβλημα της ελαχιστοποίησης του εύρους ζώνης του πίνακα έχει αποδειχθεί ότι είναι NP-complete [Papadimitriou, 1976].

Παρά την μείωση του εύρους ζώνης του πίνακα που επιτυγχάνεται με τις τεχνικές αναδιάταξης, η εκτέλεση των παράλληλων νημάτων του $SrMV$ δεν ανεξαρτητοποιείται πλήρως. Επίσης, υπάρχουν πίνακες με μεγάλο εύρος ζώνης, το οποίο δεν μπορεί να μειωθεί σημαντικά με οποιαδήποτε τεχνική αναδιάταξης, διατηρώντας έτσι την αλληλεπίδραση των νημάτων σε υψηλά επίπεδα. Είναι, επομένως, αναγκαίο να υπερκεραστούν αυτές οι αλληλεπιδράσεις χωρίς να μειώνεται ο παραλληλισμός του πυρήνα. Η πιο συνήθης τεχνική είναι η χρήση τοπικών διανυσμάτων ανά νήμα, τα οποία θα αναχθούν στο τελικό διάνυσμα εξόδου. Παρόλο που αυτή η προσέγγιση έχει παραλληλισμό της τάξης του $\Theta(N)$ για ένα πίνακα $N \times N$, στην πράξη η μέθοδος αυτή περιορίζεται κυρίως από την αλληλεπίδραση της τελικής φάσης της αναγωγής με την ιεραρχία μνήμης, κυρίως για μεγάλα διανύσματα, τα οποία δεν χωρούν στην κρυφή μνήμη του συστήματος. Το πρόβλημα αυτό γίνεται ακόμα πιο σημαντικό, όταν τα δεδομένα του διανύσματος θα πρέπει επιπλέον να μεταφερθούν πάνω από κάποιο δίκτυο διασύνδεσης. Οι Geus και Röllin [2001] εξετάζουν τρόπους ελαχιστοποίησης αυτού του κόστους μεταφέροντας μόνο μία μικρή περιοχή των τοπικών διανυσμάτων και επικαλύπτοντας, συγχρόνως, την επικοινωνία με χρήσιμους υπολογισμούς. Αντίστοιχη είναι και η προσέγγιση των Batista et al. [2010] με το σχήμα αποθήκευσης *Compressed Sparse Row-Column* (CSRC) που προτείνουν. Το σχήμα CSRC είναι ένα υβρίδιο των CSR και CSC για την αποθήκευση δομικά συμμετρικών πινάκων· τα μη μηδενικά στοιχεία του κάτω τριγωνικού πίνακα αποθηκεύονται κατά γραμμές, ενώ τα του άνω τριγωνικού αποθηκεύονται κατά στήλες. Οι συγγραφείς εξετάζουν ένα σύνολο

λο διαφορετικών τεχνικών, βασισμένων σε τοπικά διανύσματα, για την επίλυση της εξάρτησης δεδομένων στα στοιχεία του διανύσματος εξόδου και σκιαγραφούν μία λύση διαχωρισμού του πίνακα σε διαμερίσεις χωρίς συγκρούσεις δεδομένων, που βασίζεται σε τεχνικές χρωματισμού γράφων.

Μία ενδιαφέρουσα προσέγγιση για την παραλληλοποίηση του συμμετρικού πυρήνα SpMV είναι αυτή των Buluç et al. [2011]. Οι συγγραφείς επεκτείνουν το σχήμα αποθήκευσης CSB (βλ. Ενότητα 2.2.3) και παρέχουν μία αποδοτική υλοποίηση του παράλληλου αλγορίθμου για τον συμμετρικό πυρήνα SpMV, η οποία παρουσιάζει μεγάλο παραλληλισμό. Η βασική ιδέα είναι η αποφυγή της χρησιμοποίησης ενός στατικού σχήματος διαμερισμού, αλλά η χρήση της τεχνικής παραλληλισμού εργασιών (task parallelism) στο επίπεδο των μπλοκ του CSB. Έχοντας μειώσει το εύρος ζώνης του πίνακα χρησιμοποιώντας ένα αλγόριθμο αναδιάταξης, οι συγγραφείς χωρίζουν τον πίνακα σε τρεις διαγωνίους μπλοκ και προχωρούν στον υπολογισμό του γινομένου πίνακα-επί-διάνυσμα. Σε κάθε μία από τις τρεις διαγωνίους μπλοκ ανατίθεται ένα τοπικό διάνυσμα, στο οποίο υπολογίζεται ένα ενδιάμεσο γινόμενο πίνακα-επί-διάνυσμα σε δύο διαδοχικές παράλληλες φάσεις. Καθώς σε κάθε διαγώνιο μπλοκ έχει ανατεθεί ένα ιδιωτικό διάνυσμα, ο υπολογισμός του SpMV μπορεί να προχωρήσει παράλληλα για όλες τις διαγωνίους μπλοκ, ενώ τα υπολειπόμενα στοιχεία, δηλ., αυτά που δεν ανήκουν σε κάποια από τις διαγωνίους, χρησιμοποιούν ατομικές πράξεις για την απευθείας εγγραφή στο διάνυσμα εξόδου. Εκτός από την χρήση του παραλληλισμού εργασιών, που μπορεί να προσφέρει ένα καλύτερο καταμερισμό του φόρτου εργασίας, το κύριο πλεονέκτημα αυτής της μεθόδου είναι η αποσύνδεση των ενδιαμέσων διανυσμάτων από το πλήθος των νημάτων, σταθεροποιώντας με αυτό τον τρόπο το τελικό κόστος της αναγωγής.

Η επίδοση του πυρήνα SpMV

Ο υπολογιστικό πυρήνας του Πολλαπλασιασμού Αραιού Πίνακα με Διάνυσμα (SpMV) βρίσκεται στον πυρήνα των επαναληπτικών μεθόδων επίλυσης αραιών γραμμικών συστημάτων. Όπως συζητήθηκε στο Κεφάλαιο 1, ο πυρήνας SpMV κυριαρχεί στον συνολικό χρόνο εκτέλεσης αυτών των μεθόδων στις σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών και η αποδοτική υλοποίηση και βελτιστοποίησή του είναι καίριας σημασίας για την κοινότητα των υπολογιστικών συστημάτων υψηλών επιδόσεων. Το βασικό πρόβλημα επίδοσης του πυρήνα SpMV προέρχεται πρωτίστως από την αλγοριθμική του φύση, η οποία του επιβάλλει μία πολύ χαμηλή υπολογιστική ένταση. Δυστυχώς, αυτό δεν είναι το μόνο πρόβλημα αυτού του πυρήνα, καθώς και άλλοι παράγοντες, που σχετίζονται κυρίως με την δομή του πίνακα εισόδου, μπορούν να παίξουν καθοριστικό ρόλο.

Σε αυτό το κεφάλαιο, εξετάζουμε τα διάφορα προβλήματα επίδοσης του πυρήνα SpMV που έχουν αναφερθεί στην βιβλιογραφία και πραγματοποιούμε μία ποσοτική ανάλυσή τους, ώστε να μπορέσουμε να χαρακτηρίσουμε την σπουδαιότητα κάθε πιθανού παράγοντα περιορισμού της επίδοσης. Εξετάζουμε τόσο αρχιτεκτονικές συμμετρικής πρόσβασης στην κύρια μνήμη όσο και αρχιτεκτονικές NUMA και προτείνουμε μία απλή και αποδοτική τεχνική για την δημιουργία εκδόσεων του πυρήνα SpMV γι' αυτές τις αρχιτεκτονικές.

3.1 Η αλγοριθμική άποψη

Ο πυρήνας SpMV δημιουργεί μία ποικιλία πιθανών προβλημάτων επίδοσης, τα οποία σε πρώτη φάση θα παρουσιάσουμε και θα συζητήσουμε από μία περισσότερο θεωρητική σκοπιά. Για την πληρότητα της παρουσίασής μας, αλλά και για την διευκόλυνση του αναγνώστη στην συζήτηση που θα ακολουθήσει, επαναλαμβάνουμε εδώ τον αλγόριθμο του πυρήνα SpMV χρησιμοποιώντας το σχήμα αποθήκευσης CSR (Αλγόριθμος 3.1), ο οποίος παρουσιάστηκε λεπτομερώς στο Κεφάλαιο 2. Το σχήμα CSR είναι το πιο ευρέως διαδεδομένο σχήμα αποθήκευσης αραιών πινάκων και θα το χρησιμοποιήσουμε ως το σχή-

3. Η επίδοση του πυρήνα SpMV

μα αναφοράς καθόλη τη διάρκεια του κειμένου αυτού. Τροποποιήσαμε επίσης λίγο τον αλγόριθμο του SpMV που παρουσιάζουμε εδώ, ώστε να ταιριάζει περισσότερο σε μία πραγματική υλοποίηση, όπου τα υπολογιζόμενα γινόμενα σε μία γραμμή του πίνακα δεν προστίθενται απευθείας στο αντίστοιχο στοιχείο του διανύσματος εξόδου, αλλά σε μία τοπική μεταβλητή, η οποία διατηρείται συνήθως σε ένα καταχωρητή του επεξεργαστή.

```
1: procedure MATVEC_CSR(A::in, x::in, y::out)
   A: matrix in CSR format
   x: input vector
   y: output vector
2:   for i ← 0 to N do
3:      $y_i \leftarrow 0$ 
4:     for j ← rowptr[i] to rowptr[i + 1] do
5:        $y_i \leftarrow y_i + \text{values}[j] \cdot x[\text{colind}[j]]$ 
6:     end for
7:      $y[i] \leftarrow y_i$ 
8:   end for
```

Αλγόριθμος 3.1: Υλοποίηση του πυρήνα SpMV χρησιμοποιώντας το σχήμα CSR (βλ. Κεφάλαιο 2, Ενότητα 2.1 για μία λεπτομερή περιγραφή).

Ένα σύνολο από αλγοριθμικά χαρακτηριστικά του πυρήνα SpMV έχουν αναγνωρισθεί στην βιβλιογραφία ως πιθανά σημεία συμφόρησης. Στην συνέχεια, συζητάμε αυτά τα χαρακτηριστικά από θεωρητικής απόψεως.

Χαμηλή υπολογιστική ένταση Ο όρος *υπολογιστική ένταση*, επίσης γνωστός και ως *λόγος flops:byte*, είναι μία αλγοριθμική μετρική για τον καθορισμό του πλήθους των χρήσιμων υπολογιστικών πράξεων που πραγματοποιούνται από τον επεξεργαστή επί του πλήθους των δεδομένων που απαιτούνται για την πραγματοποίηση αυτών των πράξεων [Harris, 2005; Williams et al., 2009]. Η μετρική αυτή είναι καθαρά αλγοριθμική, στην λογική ότι δεν συνυπολογίζει την επίδραση του υλικού, όπως την αντικατάσταση γραμμών της κρυφής μνήμης, η οποία μπορεί να προκαλέσει επιπλέον κίνηση στο υποσύστημα μνήμης. Με την έλευση των πολυπύρηνων αρχιτεκτονικών, η μετρική της υπολογιστικής έντασης γίνεται ολοένα και πιο σημαντική, καθώς η διαφορά ταχύτητας μεταξύ επεξεργαστή και κύριας μνήμης συνεχώς αυξάνεται. Όσο μεγαλύτερος είναι ο λόγος flops:byte ενός υπολογιστικού πυρήνα, τόσο μεγαλύτερη είναι η δυνατότητά του καλύτερης εκμετάλλευσης του επεξεργαστή. Αντιθέτως, ένας χαμηλός λόγος flops:byte υποδηλώνει ότι η επίδοση του πυρήνα θα περιορίζε-

ται πιθανότατα από το υποσύστημα μνήμης.

Τα γινόμενα πίνακα-επί-διάνυσμα, είτε πυκνού είτε αραιού πίνακα, επιδεικνύουν ένα μάλλον χαμηλό λόγο flops:byte, λόγω της ίδιας τους της αλγοριθμικής φύσης: οι αλγόριθμοι αυτοί προχωρούν σε φάσεις ανάκτησης των δεδομένων από την μνήμη και υπολογισμού, χωρίς να επιδεικνύουν καμία τοπικότητα στις αναφορές τους στα στοιχεία του πίνακα. Θεωρώντας μία τυπική αποθήκευση χρησιμοποιώντας αριθμούς κινητής υποδιαστολής διπλής ακρίβειας (οκτώ byte) για τις τιμές των μη μηδενικών στοιχείων, ο λόγος flops:byte r_{dense} του γινομένου πίνακα-επί-διάνυσμα για ένα $N \times N$ πυκνό πίνακα αποθηκευμένο στο τυπικό σχήμα αποθήκευσης για πυκνούς πίνακες μπορεί να υπολογιστεί εύκολα ως

$$r_{dense} = \frac{2N^2}{8N^2 + 16N} = \frac{1}{4 + \frac{8}{N}} \approx 0.25 \quad (3.1)$$

Ο λόγος αυτός είναι ιδιαίτερα χαμηλός, καθώς για κάθε πράξη κινητής υποδιαστολής θα πρέπει να ανακτηθούν τέσσερα byte από το σύστημα μνήμης. Θεωρώντας μία αρχιτεκτονική 64-bit, ο λόγος αυτός υπονοεί ότι η ιεραρχία μνήμης θα πρέπει να μπορεί να παρέχει δεδομένα στον επεξεργαστή στο μισό της ταχύτητάς τους, κάτι το οποίο είναι εξαιρετικά δύσκολο να συμβεί στις σύγχρονες πολυπύρηνες αρχιτεκτονικές υψηλών επιδόσεων.

Η περίπτωση των του γινομένου αραιού πίνακα-επί-διάνυσμα γίνεται ακόμα χειρότερη, καθώς θα πρέπει να ανακτηθούν και τα μετα-δεδομένα της δεικτοδότησης, τα οποία απαιτούνται για την πρόσβαση των μη μηδενικών στοιχείων. Θεωρώντας ακεραίους των τεσσάρων byte για την δεικτοδότηση του πίνακα, ο λόγος flops:byte r_{sparse} για ένα αραιό πίνακα $N \times N$ με NNZ μη μηδενικά στοιχεία, αποθηκευμένο με το σχήμα CSR, μπορεί να υπολογιστεί ως εξής:

$$r_{sparse} = \frac{2NNZ}{\underbrace{8NNZ}_{values} + \underbrace{4NNZ}_{colind} + \underbrace{4N}_{rowptr} + \underbrace{16N}_{x+y}} = \frac{1}{6 + 10\frac{N}{NNZ}} \approx 0.167 \quad (3.2)$$

Ο λόγος αυτός χειροτερεύει περαιτέρω για πολύ αραιούς πίνακες, όπου το N είναι στην τάξη μεγέθους του NNZ και το μέγεθος του πίνακα `rowptr` γίνεται επίσης σημαντικό. Συγκριτικά, η υπολογιστική ένταση ενός πυρήνα επίλυσης διαφορικών εξισώσεων με πολυ-επίπεδα πλέγματα (multigrid stencil PDE) κυμαίνεται μεταξύ 0.33 και 0.50, του τρισδιάστατου μετασχηματισμού Fourier (3D FFT) αγγίζει το 1.64 [Williams et al., 2009], ενώ η υπολογιστική ένταση του πολλαπλασιασμού πίνακα-επί-πίνακα είναι στην τάξη μεγέθους του N . Είναι εμφανές από αυτή την ανάλυση ότι η επίδοση του πυρήνα SpMV στις σύγχρονες πολυπύρηνες αρχιτεκτονικές αναμένεται να περιορίζεται από

3. Η επίδοση του πυρήνα SpMV

ην επίδοση του υποσυστήματος μνήμης, και κυρίως από το εύρος ζώνης μνήμης [Mellor-Crummey και Garvin, 2004; Buttari et al., 2007; Williams et al., 2007; Goumas et al., 2008].

Ακανόνιστη πρόσβαση στο διάνυσμα εισόδου Μία σημαντική διαφορά μεταξύ των πυρήνων πυκνού και αραιού πίνακα-επί-διάνυσμα είναι ότι στην δεύτερη περίπτωση η πρόσβαση στο διάνυσμα εισόδου δεν είναι σειριακή, αλλά εξαρτάται από την δομή των μη μηδενικών στοιχείων του πίνακα. Αυτό μπορεί να οδηγήσει σε μία αύξηση των αστοχιών στην κρυφή μνήμη κατά την πρόσβαση στο διάνυσμα εισόδου σε πίνακες με μία μάλλον ακανόνιστη διάταξη των μη μηδενικών στοιχείων τους [Im, 2000; Geus και Röllin, 2001; Pichel et al., 2004].

Έμμεσες αναφορές στην μνήμη Η ανάγκη για την οικονομία χώρου κατά την αποθήκευση του πίνακα υπαγορεύει την χρήση επιπλέον δομών δεδομένων για την αποθήκευση της μετα-πληροφορίας της θέσης των μη μηδενικών στοιχείων στον πίνακα. Αυτές οι δομές δεδομένων όχι μόνο μειώνουν την υπολογιστική ένταση του πυρήνα (βλ. εξίσωση (3.2)), αλλά εισάγουν επιπλέον πράξεις, μη σχετιζόμενες με τον καθαυτό υπολογισμό, και προσθέτουν ανεπιθύμητες «παρεμβολές» στην χρήση τη ιεραρχίας της κρυφής μνήμης [Pinar και Heath, 1999].

Μικρές γραμμές Ο αριθμός των επαναλήψεων του εσωτερικού βρόχου του πυρήνα SpMV (Αλγόριθμος 3.1, γραμμές 4–6) εξαρτάται από το μέγεθος της αντίστοιχης γραμμής του πίνακα. Επομένως, μικρότερες γραμμές είναι αρκετά πιθανό να δημιουργούν σημαντική επιβάρυνση, η οποία μπορεί να υπερκεράσει τους χρήσιμους υπολογισμούς [White και Sadayappan, 1997; Buttari et al., 2007]. Ακόμα χειρότερα, η λύση της εκτύλιξης βρόχων (loop unrolling) δεν μπορεί να εφαρμοστεί χωρίς την εξέταση της δομής του πίνακα, καθώς το πλήθος των επαναλήψεων κάθε βρόχου είναι διαφορετικό κατά την διάρκεια της συμπίλησης.

Μία ακόμα, λιγότερο διακριτή, επίπτωση στην επίδοση του πυρήνα SpMV είναι ότι οι πολύ μικρές γραμμές επηρεάζουν αρνητικά και την υπολογιστική ένταση του πυρήνα. Ο λόγος $\frac{N}{NNZ}$ στον παρονομαστή της εξίσωσης (3.2) είναι στην πραγματικότητα το ανίστροφο του μέσου μεγέθους γραμμής του πίνακα. Αυτό, επομένως, οδηγεί σε μία σχέση μονότονα αύξουσα μεταξύ του μέσου μεγέθους γραμμής και του λόγου flops:byte, υπονοώντας ότι μικρότερες γραμμές οδηγούν σε χαμηλότερη υπολογιστική ένταση του πυρήνα, εντείνοντας με αυτό τον τρόπο τη συμφόρηση στην ιεραρχία μνήμης.

Ανισορροπία στην κατανομή του φόρτου εργασίας Ο πυρήνας SpMV μπορεί να παραλληλοποιηθεί εύκολα κατά τις γραμμές του πίνακα. Κάθε νήμα αναλαμβάνει μία ζώνη του πίνακα και προχωρά ανεξάρτητα με τον υπολογισμό του γινομένου πίνακα-επί-διάνυσμα. Ένα κατάλληλο σχήμα στατικού διαμερισμού του φόρτου εργασίας θα χώριζε τον πίνακα σε διαμερίσεις με περίπου τον ίδιο αριθμό μη μηδενικών στοιχείων, ώστε να επιτύχει μία δίκαιη κατανομή του υπολογιστικού φορτίου. Ωστόσο, για πίνακες με ακανόνιστη κατανομή μη μηδενικών στοιχείων, ακόμα και αυτό το σχήμα μπορεί να οδηγήσει σε ανισορροπίες στην κατανομή του φόρτου εργασίας, καθώς τα υπολογιστικά χαρακτηριστικά των διαμερίσεων μπορεί να είναι αρκετά διαφορετικά. Για παράδειγμα, ένα νήμα που έχει αναλάβει μία πιο πυκνή διαμέριση αναμένεται να είναι γρηγορότερο από ένα νήμα που έχει αναλάβει μία πιο αραιή, μιας και το τελευταίο μπορεί να υποφέρει από ένα χαμηλότερο λόγο `float:byte` ή από ένα αυξημένο ρυθμό αστοχιών στην κρυφή μνήμη, λόγω των ακανόνιστων προσβάσεων στο διάνυσμα εισόδου.

3.2 Προκαταρκτικά πειραμάτων

Προτού προχωρήσουμε στην ποσοτική αξιολόγηση του πυρήνα SpMV, είναι απαραίτητο να παρουσιάσουμε τους πίνακες, τις πειραματικές πλατφόρμες και την μεθοδολογία που χρησιμοποιήσαμε στην αξιολόγηση. Αφιερώνουμε μία ξεχωριστή ενότητα για αυτή την περιγραφή, καθώς χρησιμοποιούμε αυτή την πειραματική διάταξη σε ολόκληρη την διατριβή. Αν δεν δηλωθεί διαφορετικά, όλες οι αναφορές σε συγκεκριμένους πίνακες και πλατφόρμες θα επιλύονται σε αυτή την ενότητα και όλες οι περιγραφές μετρήσεων επίδοσης και μετρικών θα αναφέρονται στο υλικό που παρουσιάζεται στη συνέχεια.

3.2.1 Σουίτα πινάκων

Η σουίτα πινάκων που χρησιμοποιήθηκε στα πειράματα αποτελείται από 30 πίνακες, που επιλέχθηκαν από την συλλογή αραιών πινάκων του Πανεπιστημίου της Φλόριδα [Davis και Hu, 2011]. Η συλλογή αυτή περιλαμβάνει χιλιάδες αραιούς πίνακες που εμφανίζονται σε πραγματικές εφαρμογές από μία πληθώρα επιστημονικών πεδίων. Έχει καταστεί η καθιερωμένη πηγή αραιών πινάκων για την κοινότητα της αριθμητικής γραμμικής άλγεβρας και για την ανάπτυξη και πειραματική αξιολόγηση αλγορίθμων αραιών πινάκων. Διαλέξαμε πίνακες για την σουίτα πειραμάτων μας βασιζόμενοι σε ένα σύνολο κριτηρίων:

- *Ποικιλία*: Οι πίνακες που επιλέχθηκαν προέρχονται από μία μεγάλη ποικιλία εφαρμογών, περιλαμβανομένων προβλημάτων χωρίς κάποια υποκειμένη $2\Delta/3\Delta$ γεωμετρία. Αυτά τα προβλήματα οδηγούν συνήθως σε

3. Η επίδοση του πυρήνα SpMV

πίνακες με μια πιο ακανόνιστη δομή και περιμένουμε να τονίζουν διαφορετικές πτυχές του πυρήνα SpMV.

- *Μέγεθος*: Καθώς ο πυρήνας SpMV αποτελεί σημείο συμφόρησης στην επίλυση μεγάλων αραιών γραμμικών συστημάτων, οι πίνακες που επιλέχθηκαν είναι αρκετά μεγάλοι, ώστε να μην χωρούν στην συνολική κρυφή μνήμη ενός τυπικού high-end πολυεπεξεργαστικού συστήματος.
- *Αλγεβρικά κριτήρια*: Οι μισοί από τους επιλεγμένους πίνακες είναι συμμετρικοί και θετικά ορισμένοι. Αυτά τα χαρακτηριστικά απαιτούνται για την εκτέλεση της επαναληπτικής μεθόδου CG, την οποία χρησιμοποιούμε στην αξιολόγησή μας στα τελευταία κεφάλαια.

Ο πίνακας 3.1 παρουσιάζει με λεπτομέρεια τα χαρακτηριστά της σουίτας πινάκων μας. Τα 2/3 των πινάκων προέρχονται από προβλήματα με μία υποκείμενη γεωμετρία $2\Delta/2\Delta$, καθώς αυτά είναι που συναντώνται συχνότερα κατά την επίλυση αραιών γραμμικών συστημάτων. Εννέα πίνακες έχουν μικρές γραμμές και παρουσιάζουν πολύ χαμηλή υπολογιστική ένταση, η οποία κυμαίνεται κάτω από 1.5. Ο πιο αραιός πίνακας είναι ο Hamrle3, ο οποίος έχει μόνο τέσσερα μη μηδενικά στοιχεία ανά γραμμή κατά μέσο όρο, ενώ ο πιο πυκνός είναι ο TSOPF_RS_b2383 με 424 στοιχεία ανά γραμμή.

3.2.2 Πλατφόρμες υλικού

Οι πλατφόρμες υλικού που χρησιμοποιήσαμε για την ποσοτική μας ανάλυση αποτελούνται από δύο πολυεπεξεργαστικά συστήματα συμμετρικής πρόσβασης στη μνήμη (SMP) και ένα μη ομοιόμορφης πρόσβασης με συνεκτική κρυφή μνήμη (cc-NUMA). Τα συστήματα SMP είναι ένας δι-επεξεργαστικό τετραπύρηνων Intel Xeon E5405 (κωδική ονομασία *Harpertown*) και ένα τετραεπεξεργαστικό εξαπύρηνων Intel Xeon X7460 (κωδική ονομασία *Dunnington*). Το σύστημα NUMA είναι ένα δι-επεξεργαστικό τετραπύρηνων Intel Xeon W5580 (κωδική ονομασία *Gainestown*). Στην συνέχεια, θα αναφερόμαστε σε κάθε πλατφόρμα με την κωδική της ονομασία. Ο πίνακας 3.2 παρουσιάζει ένα κατάλογο των τεχνικών χαρακτηριστικών των πειραματικών πλατφορμών, ενώ το Σχήμα 3.1 παρουσιάζει τα μπλοκ διαγράμματά τους.

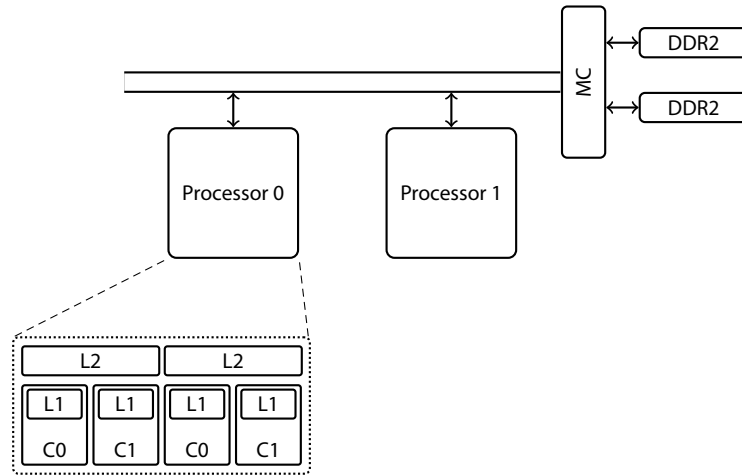
Από αρχιτεκτονικής απόψεως, το σύστημα Dunnington είναι μία επέκταση του Harpertown, όχι μόνο σε επίπεδο συστήματος, αλλά και εντός του επεξεργαστή. Ο επεξεργαστής του Dunnington περιέχει ακόμα μία διπύρηνια μονάδα, φτάνοντας τελικά τους έξι πυρήνες, και μία πολύ μεγάλη L3 κρυφή μνήμη των 16 MiB, ώστε να μπορέσει να αποσοβήσει την συμφόρηση στον εμπρόσθιο δίαυλο (front-end bus), που δημιουργείται από την πληθώρα των πυρήνων. Πράγματι, λόγω της πολύ μεγάλης κρυφής του μνήμης, το σύστημα Dunnington μπορεί να επιτύχει ρυθμούς μετάδοσης δεδομένων πολύ κοντά

3.2. Προκαταρκτικά πειραμάτων

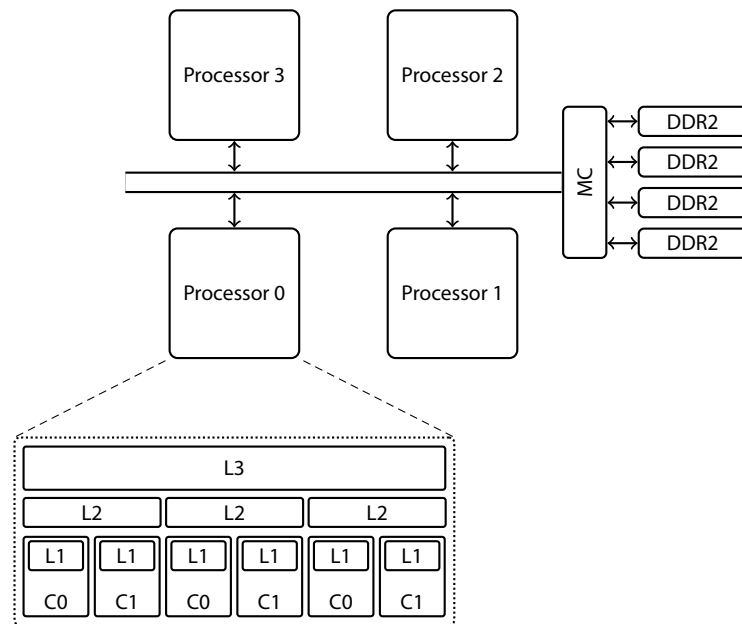
Matrix	Rows	Non-zeros	Size (MiB)	f:b ratio	P. D.	Problem	2D/3D
xenon2	157,464	3,866,688	44.85	0.156	No	Materials	Yes
ASIC_680k	682,862	3,871,773	46.91	0.129	No	Circuit Sim.	No
torso3	259,156	4,429,042	51.67	0.152	No	Other	Yes
Chebyshev4	68,121	5,377,761	61.80	0.163	No	Structural	Yes
Hamrle3	1,447,360	5,514,242	68.63	0.116	No	Circuit Sim.	No
pre2	659,033	5,959,282	70.71	0.141	No	Circuit Sim.	No
cage13	445,315	7,479,343	87.29	0.152	No	Graph	No
atmosmodj	1,270,432	8,814,880	105.72	0.135	No	C.F.D.	Yes
ohne2	181,343	11,063,545	127.30	0.162	No	Semiconductor	Yes
kkt_power	2,063,494	14,612,663	175.10	0.135	No	Optimization	No
TSOPF_RS_b2383	38,120	16,171,169	185.21	0.166	No	Power	No
Ga41As41H72	268,096	18,488,476	212.61	0.163	No	Chemistry	No
Freescall1	3,428,755	18,920,347	229.61	0.128	No	Circuit Sim.	No
rajat31	4,690,002	20,316,253	250.39	0.120	No	Circuit Sim.	No
F1	3,428,755	26,837,113	308.44	0.163	No	Structural	Yes
parabolic_fem	525,825	3,674,625	44.06	0.135	Yes	C.F.D.	Yes
offshore	259,789	4,242,673	49.54	0.151	Yes	Electromagnetics	Yes
consph	83,334	6,010,480	69.10	0.163	Yes	F.E.M.	Yes
bmw7st_1	141,347	7,339,667	84.54	0.161	Yes	Structural	Yes
G3_circuit	1,585,478	7,660,826	93.72	0.124	Yes	Circuit Sim.	No
thermal2	1,228,045	8,580,313	102.88	0.135	Yes	Thermal	Yes
m_t1	97,578	9,753,570	111.99	0.164	Yes	Structural	Yes
bmwcra_1	148,770	10,644,002	122.38	0.163	Yes	Structural	Yes
hood	220,542	10,768,436	124.08	0.161	Yes	Structural	Yes
crankseg_2	63,838	14,148,858	162.16	0.165	Yes	Structural	Yes
nd12k	36,000	14,220,946	162.88	0.166	Yes	Other	Yes
af_5_k101	503,625	17,550,675	202.77	0.159	Yes	Structural	Yes
inline_1	503,712	36,816,342	423.25	0.163	Yes	Structural	Yes
ldoor	952,203	46,522,475	536.04	0.161	Yes	Structural	Yes
boneS10	914,898	55,468,422	638.28	0.162	Yes	Model Reduction	Yes

Πίνακας 3.1: Η σουίτα πινάκων που χρησιμοποιήθηκε για την πειραματική αξιολόγηση. Για κάθε πίνακα παρουσιάζεται το πλήθος των γραμμών (όλοι οι πίνακες είναι τετραγωνικοί), το πλήθος των μη μηδενικών στοιχείων, το μέγεθός τους και η υπολογιστική ένταση (λόγος flops:byte) με το σχήμα CSR, εάν είναι θετικά ορισμένοι (P. D.), η κατηγορία προβλήματος στην οποία ανήκουν και εάν προέρχονται από κάποιο πρόβλημα με 2D/3D γεωμετρία. Για περισσότερες πληροφορίες σχετικά με συγκεκριμένες ιδιότητες, ο αναγνώστης μπορεί να ανατρέξει στην Συλλογή Αραιών Πινάκων του Πανεπιστημίου της Φλόριδα [Davis και Hu, 2011].

3. Η επίδοση του πυρήνα SPMV

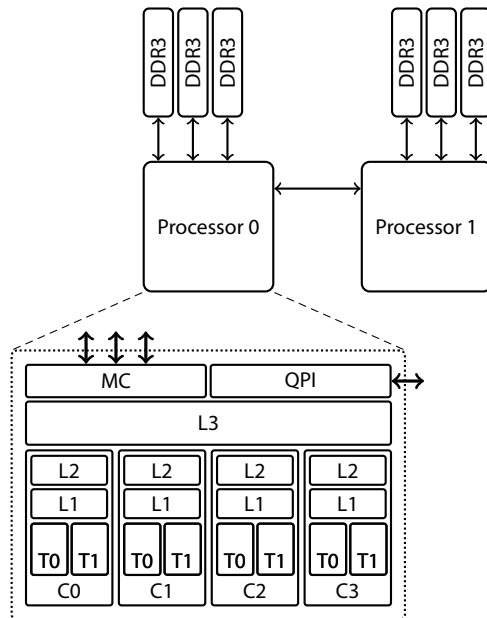


(i) Harperstown: δι-επεξεργαστικό σύστημα SMP τετραπύρηνων επεξεργαστών· δύο κανάλια μνήμης.



(ii) Dunnington: τετρα-επεξεργαστικό σύστημα SMP εξαπύρηνων επεξεργαστών· τέσσερα κανάλια μνήμης.

Σχήμα 3.1: Τα μπλοκ διαγράμματα των πολυεπεξεργαστικών συστημάτων που χρησιμοποιήθηκαν για τις πειραματικές αξιολογήσεις σε αυτή την διατριβή (συνεχίζεται στην σελ. 47).



(iii) Gainestown: δι-επεργαστικό σύστημα NUMA τετραπύρηνων επεξεργαστών· δύο νήματα υλικού ανά πυρήνα· τέσσερα κανάλια μνήμης για κάθε ενσωματωμένο ελεγκτή μνήμης.

Σχήμα 3.1 (συνέχεια): Τα μπλοκ διαγράμματα των πολυεπεξεργαστικών συστημάτων που χρησιμοποιήθηκαν για τις πειραματικές αξιολογήσεις σε αυτή την διατριβή.

στο θεωρητικό μέγιστο του εύρους ζώνης της μνήμης του. Ωστόσο, και οι τέσσερις επεξεργαστές του συστήματος χρησιμοποιούν τον κοινό διάυλο για την επικοινωνία τους με την κύρια μνήμη, αλλά και μεταξύ τους, μία διάταξη μπορεί να αποτελέσει σημείο σημαντικής συμφόρησης για πολυνηματικές εφαρμογές με μεγάλη ένταση αναφορών στην κύρια μνήμη, όπως είναι ο πυρήνας SpMV.

Το σύστημα Gainestown ξεφεύγει από την συγκεντρωτική λογική των συστημάτων SMP, μεταφέροντας τον ελεγκτή μνήμης εντός του επεξεργαστή και διαχωρίζοντας, ως αποτέλεσμα, την φυσική μνήμη σε κόμβους ανά επεξεργαστή. Κάθε ελεγκτής μνήμης μπορεί να εξυπηρετήσει μέχρι τρία κανάλια μνήμης DDR3 επιτυγχάνοντας ένας εύρος ζώνης μνήμης της τάξης των 15.5 GB/s, τρεις περίπου φορές υψηλότερο σε σύγκριση με τις αρχιτεκτονικές SMP. Η επικοινωνία μεταξύ των επεξεργαστών, όπως επίσης και η πρόσβαση στους απομακρυσμένους κόμβους μνήμης, δρομολογείται μέσω ενός εξει-

3. Η επίδοση του πυρήνα SPMV

	Harpertown	Dunnington	Gainestown
Model	Intel Xeon E5405	Intel Xeon X7460	Intel Xeon W5580
Microarchitecture	Intel Core	Intel Core	Intel Nehalem
Clock freq.	2.00 GHz	2.66 GHz	3.20 GHz
L1 cache (D/I)	32 KiB/32 KiB	32 KiB/32 KiB	32 KiB/32 KiB
L2 cache	6 MiB (per 2 cores)	3 MiB (per 2 cores)	256 KiB (per core)
L3 cache	–	16 MiB	8 MiB
Cores/Threads	4/4	6/6	4/8
Peak Front-end b/w	10.7 GB/s	8.5 GB/s	2 × 30 GB/s
Interconnection b/w	–	–	25.6 GB/s
Sustained Mem. b/w	5.8 GB/s	8.1 GB/s	2 × 15.5 GB/s
Sustained i/c b/w	–	–	9.4 GB/s
Multiprocessor Configurations			
Sockets	2	4	2
Cores/Threads	8/8	24/24	8/16

Πίνακας 3.2: Τεχνικά χαρακτηριστικά των πλατφόρμων υλικού που χρησιμοποιήθηκαν στις πειραματικές αξιολογήσεις. Τα νούμερα για το «sustained» εύρος ζώνης μνήμης έχουν ληφθεί με το μετροπρόγραμμα STREAM [McCalpin, 1995] χρησιμοποιώντας το πλήρες σύστημα. Το «sustained» εύρος ζώνης διασύνδεσης (i/c) για το σύστημα Gainestown μετρήθηκε με μία τροποποιημένη έκδοση του STREAM, όπου ολοκληρώθηκε η κίνηση για την κύρια μνήμη δρομολογήθηκε μέσω του απομακρυσμένου ελεγκτή μνήμης.

δικευμένου δικτύου διασύνδεσης (Intel QuickPath Interconnect – QPI [Kurd et al., 2008]), το οποίο επιτυγχάνει ένα εύρος ζώνης της τάξης των 9.4 GB/s, περίπου 40% λιγότερο από το διαθέσιμο εύρος ζώνης μνήμης. Συνεπώς, το δίκτυο διασύνδεσης είναι πιθανό να αποτελέσει σημείο συμφόρησης, εάν δρομολογηθεί μέσω αυτού μία σημαντική ποσότητα κίνησης προς την κύρια μνήμη, λόγω πολλών απομακρυσμένων προσβάσεων.

Ρυθμίσεις λογισμικού

Όλα τα συστήματα τρέχουν λειτουργικό σύστημα Linux 64-bit (έκδοση πυρήνα 2.6.30.5 ή μεγαλύτερη), ενώ χρησιμοποιήθηκε η συλλογή συμπιλιστών GNU (gcc, g++, κ.λπ.), έκδοση 4.6, για την συμπίληση κάθε προγράμματος, εκτός και εάν αναφέρεται διαφορετικά. Οι πολυνηματικές εκδόσεις των κωδίκων γράφηκαν με χρήση της βιβλιοθήκης χρήστη Pthreads (NPRTL, έκδοση

2.7). Τέλος, οι εκδόσεις για τις αρχιτεκτονικές NUMA χρησιμοποιούν την βιβλιοθήκη *numactl* (έκδοση 2.0.7), η οποία περικαλύπτει την χαμηλότερου επιπέδου διεπαφή κλήσεων συστήματος του πυρήνα του Linux.

3.2.3 Διαδικασίες και πολιτικές μετρήσεων

Για να εγγυηθούμε αξιόπιστες και δίκαιες μετρήσεις, κατασκευάσαμε μία κοινή υποδομή μετρήσεων για όλα τα πειράματα. Η υποδομή αυτή επικοινωνεί με τις υλοποιήσεις των σχημάτων αποθήκευσης μέσω μίας καλά ορισμένης διεπαφής για τον πολλαπλασιασμό αραιού πίνακα-επί-διάνυσμα. Μετράμε την επίδοση 128 συνεχόμενων πράξεων SpMV με τυχαία διανύσματα εισόδου. Δεν καταβάλλουμε κάποια προσπάθεια για την «μόλυνση» της κρυφής μνήμης μετά από κάθε επανάληψη έτσι, ώστε να προσομοιώσουμε καλύτερα την συμπεριφορά των επαναληπτικών μεθόδων, όπου τα δεδομένα του πίνακα και των διανυσμάτων βρίσκονται ήδη στην ιεραρχία της κρυφής μνήμης, είτε διότι μόλις παρήχθησαν είτε διότι προσπελάφησαν πρόσφατα. Επαναλαμβάνουμε κάθε πείραμα τρεις ή πέντε φορές και κρατάμε τον διάμεσο των τιμών επίδοσης.

Για να προσομοιώσουμε την τυπική περίπτωση αποθήκευσης αραιών πινάκων, χρησιμοποιούμε ακεραίους 32-bit για την δεικτοδότηση και αριθμούς κινητής υποδιαστολής διπλής ακρίβειας 64-bit για τις τιμές των μη μηδενικών στοιχείων. Για τα σχήματα που χρησιμοποιούν επιπλέον δομές δεδομένων, π.χ., το VBL, θα ανακοινώνουμε το μέγεθός τους στην οικεία συζήτηση μέσα στο κείμενο.

Οι σύγχρονοι πολυπύρρηνοι επεξεργαστές μοιράζουν τους πόρους τους (σωλήνωση, ιεραρχία κρυφής μνήμης, διεπαφή διαύλου/μνήμης) στα νήματα μιας πολυνηματικής εφαρμογής σε διαφορετικά επίπεδα. Για παράδειγμα, δύο νήματα στο σύστημα Gainestown μπορούν να μοιράζονται όλους τους πόρους του επεξεργαστή, από την σωλήνωση μέχρι τον ελεγκτή μνήμης¹, αλλά μπορεί και να μην μοιράζονται τίποτα, εάν έχουν τοποθετηθεί σε διαφορετικούς επεξεργαστές. Ανάλογα με το φορτίο της κάθε εφαρμογής, διαφορετικές αναθέσεις των νημάτων στους πυρήνες μπορεί να οδηγήσει σε σημαντικές διακυμάνσεις της επίδοσης. Είναι επομένως σημαντικό όχι μόνο να αναθέτει κανείς τα νήματα σε συγκεκριμένους πυρήνες κατά την διαδικασία των μετρήσεων, αλλά και να ορίσει μία πολιτική ανάθεσης των νημάτων στους πυρήνες. Η ανάθεση των νημάτων στην περίπτωση μας πραγματοποιείται με χρήση της κλήσης συστήματος `sched_setaffinity()` του πυρήνα του Linux, η οποία επιτρέπει την ανάθεση του καλούντος νήματος σε ένα οποιοδήποτε σύνολο

¹ Η τεχνολογία αυτή είναι γνωστή ως Ταυτόχρονος Πολυνηματισμός (Simultaneous Multithreading – SMT) [Tullsen et al., 1995] και καθιερώθηκε εμπορικά καταρχήν από την Intel στην μικροαρχιτεκτονική Netburst [Koufaty και Marr, 2003].

3. Η επίδοση του πυρήνα SpMV

από λογικούς επεξεργαστές. Για την ανάθεση των νημάτων στους πυρήνες, ορίζουμε δύο διαφορετικές πολιτικές:

Share-all: Η πολιτική αυτή αναθέτει τα νήματα στους πυρήνες έτσι, ώστε να επιτυγχάνεται ο μέγιστος διαμοιρασμός των πόρων, με την εξαίρεση των πόρων της σωλήνωσης του επεξεργαστή (χαρακτηριστικό Hyperthreading²). Για παράδειγμα, στο σύστημα Dunnington, οι πυρήνες που μοιράζονται την κρυφή μνήμη L2 θα «γемίσουν» πρώτοι, ακολουθούμενοι από τους πυρήνες που μοιράζονται την L3 και ούτω καθεξής για τους υπόλοιπους επεξεργαστές. Το πλεονέκτημα αυτής της πολιτικής είναι ότι παρέχει περισσότερη πληροφόρηση για την συμπεριφορά της επίδοσης καθώς κλιμακώνουμε ένα σύστημα προσθέτοντας περισσότερους επεξεργαστές.

Share-nothing: Αυτή είναι η ακριβώς αντίθετη πολιτική: αναθέτει τα νήματα στους πυρήνες έτσι, ώστε να επιτυγχάνεται ο ελάχιστος διαμοιρασμός πόρων. Συνεχίζοντας το παράδειγμα του συστήματος Dunnington, τα νήματα με αυτή την πολιτική πρώτα θα «απλωθούν» σε όλους τους επεξεργαστές, στην συνέχεια ανά διαφορετική κρυφή μνήμη L2, και τέλος, θα αρχίσει ο πλήρης διαμοιρασμός. Το πλεονέκτημα αυτής της πολιτικής είναι ότι χρησιμοποιεί εξαρχής την πλήρη δυναμική του συστήματος.

Στην αξιολόγηση και στα πειράματα που ακολουθούν χρησιμοποιούμε κατά σύστημα την πολιτική 'share-all', εκτός και εάν αναφέρεται διαφορετικά.

3.3 Ποσοτική αξιολόγηση

Στην Ενότητα 3.1 συζητήσαμε μία σειρά από χαρακτηριστικά του πυρήνα SpMV που μπορούν να προκαλέσουν επιβάρυνση της επίδοσης κατά την εκτέλεση του πυρήνα. Στις επόμενες ενότητες, πραγματοποιούμε μία ποσοτική ανάλυση αυτών των χαρακτηριστικών, με σκοπό να αξιολογήσουμε την πραγματική τους επίδραση στην επίδοση του πυρήνα σε ένα ευρύ φάσμα πινάκων και συγχρονων αρχιτεκτονικών υπολογιστών.

3.3.1 Σειριακή επίδοση

Για να ποσοτικοποιήσουμε την επίδραση των πιθανών προβλημάτων επίδοσης, υλοποιούμε μία σειρά από μετροπρογράμματα (microbenchmarks), κάθε ένα από τα οποία απαλείφει ένα εικαζόμενο πρόβλημα επίδοσης. Τα μετροπρογράμματα αυτά είναι στην πραγματικότητα απλοποιημένες εκδόσεις του πυρήνα SpMV για το σχήμα αποθήκευσης CSR, όπου έχουμε εξαλείψει την πηγή

² Το χαρακτηριστικό Hyperthreading δεν το χρησιμοποιούμε παρά μόνο στην διάταξη 16 νημάτων στο σύστημα Gainestown.

κάθε προβλήματος, χρησιμοποιώντας απλούς μετασχηματισμούς του κώδικα. Εφαρμόζουμε αυτούς τους μετασχηματισμούς αθροιστικά και αναμένουμε την επίδοση του πυρήνα να αυξάνεται, μέχρις ότου φτάσει την επίδοση πυρήνα πολλαπλασιασμού πυκνού πίνακα με διάνυσμα (DMV). Όσο περισσότερο αυξάνεται η επίδοση του μετασχηματισμένου κώδικα, τόσο πιο σημαντική είναι επίδραση του εικαζόμενου προβλήματος στον πραγματικό πυρήνα. Η βασική απαίτηση για όλα τα μετροπρογράμματα είναι να προσπελούν όλα τα μη μηδενικά στοιχεία και όλες τις γραμμές του αρχικού πίνακα. Δεν υπάρχει κανένας περιορισμός στην σειρά με την οποία τα στοιχεία προσπελούνται ή πόσα στοιχεία θα προσπελαθούν ανά γραμμή. Αυτό επιτρέπει την διαδοχική χαλάρωση των παραγόντων μείωσης της επίδοσης (π.χ., προσβάσεις στο διάνυσμα εισόδου, επιπλέον δομές δεδομένων κ.λπ.), διατηρώντας παράλληλα το σύνολο των μη μηδενικών στοιχείων εντός του συνόλου εργασίας (working set) του αλγορίθμου.

Ο πίνακας 3.3 παρουσιάζει μία περίληψη των μετροπρογραμμάτων που υλοποιήθηκαν και τον κύριο σκοπό του καθενός. Το μετροπρόγραμμα `nochmiss` εξαλείφει τις ακανόνιστες προσβάσεις στο διάνυσμα εισόδου, σειριοποιώντας το σχήμα προσπέλασης. Αυτό το μετροπρόγραμμα δεν μετασχηματίζει τον κώδικα του πυρήνα· απλά, σειριοποιεί τις στήλες στην δομή `colind`. Το μετροπρόγραμμα `rowwptr` εξαλείφει την χρήση της δομής `rowptr`, θεωρώντας ότι κάθε γραμμή έχει τον ίδιο αριθμό μη μηδενικών στοιχείων (ίσο με το μέσο μέγεθος των γραμμών του πίνακα), ενώ το μετροπρόγραμμα `nocolind` εξαλείφει την χρήση τη δομής `colind`, θεωρώντας σειριακή πρόσβαση στον διάνυσμα εισόδου στο πλαίσιο της κάθε γραμμής. Η διαδοχική εφαρμογή όλων αυτών των μετασχηματισμών εξαλείφει όλα τα κύρια προβλήματα επίδοσης που σχετίζονται με την σειριακή εκτέλεση του πυρήνα `SrMV`. Ο μετασχηματισμένος κώδικας μπορεί να θεωρηθεί σαν να υποθέτει ένα αραιό πίνακα με τέλειο σχήμα προσβάσεων στο διάνυσμα εισόδου και με γνωστή διάταξη των μη μηδενικών στοιχείων του, η οποία εξαλείφει την ανάγκη διατήρησης επιπλέον δομών δεδομένων. Η μόνη διαφορά αυτού του φανταστικού πίνακα από ένα πυκνό είναι ότι μπορεί να περιέχει μικρές γραμμές, καθότι το μέγεθος της γραμμής του παραμένει το ίδιο με του αρχικού πίνακα. Είναι, επομένως, αναμενόμενο η επίδοση του μετασχηματισμένου κώδικα να πλησιάζει την επίδοση του πυρήνα `DMV`· οποιεσδήποτε διαφορές μπορούν να αποδοθούν στην επιβάρυνση των υπολογισμών των βρόχων λόγω των πολύ μικρών γραμμών του αρχικού πίνακα.

Το Σχήμα 3.2 παρουσιάζει τις διαφορές επιβαρύνσεις στην επίδοση του πυρήνα `SrMV` που προκαλούνται λόγω της ειδικής φύσης του. Η πρώτη παρατήρηση είναι ότι η επιβάρυνση τη δομής `colind` είναι ιδιαίτερα σημαντική στις αρχιτεκτονικές `SMP`, ξεπερνώντας το 20% στις περισσότερες των περιπτώσεων, ενώ είναι πιο έντονη στο σύστημα `Dunnington`, κυρίως λόγω της

3. Η επίδοση του πυρήνα SpMV

Benchmark	Goal	Action
noxmiss	Irregular accesses in the input vector	Serialize access pattern in <code>colind</code>
norowptr	Effect of the <code>rowptr</code> data structure	Rows with the same number of non-zero elements with a size equal to the average row size of the original matrix
nocolind	Effect of the <code>colind</code> data structure	Serial access pattern within each matrix row

Πίνακας 3.3: Microbenchmarks for assessing the performance problems of the SpMV kernel. Η διαδοχική εφαρμογή όλων των μετροπρογραμμάτων δίνει ένα μέτρο της επίδρασης των πολύ μικρών γραμμών στην επίδοση του πυρήνα (βλ. συζήτηση στο κείμενο).

υψηλότερης ταχύτητας του ρολογιού του και του χαμηλότερου εύρους ζώνης μνήμης σε σχέση με το σύστημα Harpertown. Η επιβάρυνση της δομής `rowptr` είναι μάλλον μικρή, καθώς είναι υπεύθυνη για ένα πολύ μικρό κομμάτι του συνολικού μεγέθους της αναπαράστασης του πίνακα στις περισσότερες των περιπτώσεων. Η κατάσταση, όμως, γίνεται αρκετά ενδιαφέρουσα σχετικά με το άγνωστο σχήμα προσβάσεων στο διάνυσμα εισόδου. Στην πραγματικότητα, το σχήμα αυτό φαίνεται να είναι αρκετά κανονικό στις περισσότερες των περιπτώσεων, ειδικότερα στους πίνακες που υπόκεινται σε $2\Delta/3\Delta$ γεωμετρία (π.χ., `xenon2`, `bone510` κ.λπ.). Το κανονικό σχήμα των προσβάσεων διευκολύνει τους μηχανισμούς προφόρτωσης δεδομένων των σύγχρονων επεξεργαστών, επιτρέποντάς τους να προφορτώνουν τα σωστά δεδομένα στη κρυφή μνήμη του ανώτερου επιπέδου, μειώνοντας με αυτό τον τρόπο σημαντικό το ποσοστό των αστοχιών στην κρυφή μνήμη. Ωστόσο, υπάρχουν συγκεκριμένοι αραιοί πίνακες (π.χ., `Hamrle3`, `parabolic_fem`), οι οποίοι παρουσιάζουν ένα μάλλον ακανόνιστο σχήμα προσβάσεων. Ο υπολογισμός του SpMV παλινδρομεί συνεχώς στο διάνυσμα εισόδου, χρησιμοποιώντας μεγάλα βήματα, όχι μόνο καταστρέφοντας καταστρέφοντας με αυτό τον τρόπο την χωρική τοπικότητα των αναφορών, αλλά καθιστώντας επίσης αδύνατη την επιτυχή πρόβλεψη των προσβάσεων από τον μηχανισμό της προφόρτωσης δεδομένων. Για παράδειγμα, οι προσβάσεις για τον πίνακα `Hamrle3` αποτελούνται από μη σταθερά άλματα που κυμαίνονται μεταξύ 4.480 και 755.520 στοιχείων. Παρατηρήσαμε, επίσης, μία σημαντική συσχέτιση μεταξύ των πινάκων με πολύ μικρές γραμμές και αυτών με ακανόνιστο σχήμα προσβάσεων. Οι πίνακες με πολύ μικρές

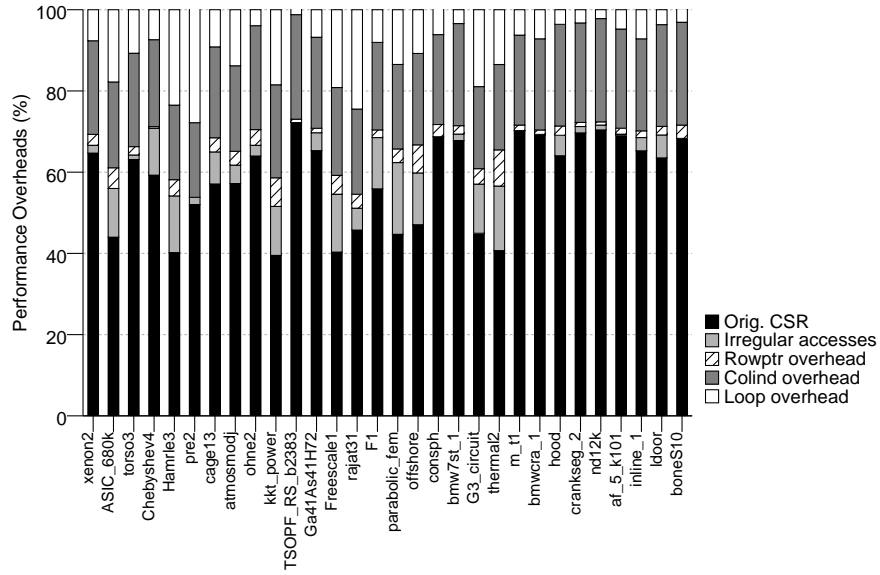
γραμμές προέρχονται συνήθως από μη δομικά (structural) προβλήματα (βλ. Πίνακα 3.1) και τα στοιχεία τους είναι σκορπισμένα σε μικρές ομάδες σε ολόκληρη την γραμμή, οδηγώντας έτσι σε ένα ακανόνιστο σχήμα προσβάσεων στο διάλυμα εισόδου. Τέλος, η επιβάρυνση βρόχου λόγω των πολύ μικρών γραμμών (4–9 στοιχεία) μπορεί να αποβεί ιδιαίτερα σημαντική σε μερικές περιπτώσεις, ξεπερνώντας ακόμη και το 20% (π.χ., Hamrle3, pre2, rajat31 κ.ά.), καθώς πολύ λίγες χρήσιμες πράξεις εκτελούνται εντός του βρόχου. Το αρνητικό με αυτές τις περιπτώσεις είναι ότι δεν μπορούν εύκολα να εφαρμοστούν κοινές τεχνικές βελτιστοποίησης, όπως η εκτύλιξη βρόχων, ώστε να μειωθεί η επιβάρυνση, καθώς το πλήθος των επαναλήψεων όχι μόνο είναι άγνωστο, αλλά μπορεί να μεταβάλλεται και μέσα στον ίδιο τον πίνακα από γραμμή σε γραμμή.

Τα αποτελέσματα των μετροπρογραμμάτων στις αρχιτεκτονικές NUMA (Σχήμα 3.2iii) παρουσιάζουν κάπως διαφορετική συμπεριφορά. Η βασική παρατήρηση είναι ότι η επιβάρυνση της δομής `colind` δεν είναι τόσο εμφανής, λόγω του επαρκούς εύρους ζώνης μνήμης που παρέχεται από τον ενσωματωμένο ελεγκτή μνήμης. Στην πραγματικότητα, οι πίνακες με σχετικά κανονική δομή τείνουν να επιτυγχάνουν περίπου 80% της επίδοσης του πυρήνα DMV. Αντιθέτως, η επιβάρυνση της δομής `rowptr` γίνεται πλέον σημαντική, ξεπερνώντας το 10% για τους πιο ακανόνιστους πίνακες. Παρά το ότι πρόκειται για μικρή δομή, τα άγνωστα όρια του βρόχου και η έμμεση αναφορά για την πρόσβασή τους, που προκαλούνται από την χρήση της δομής `rowptr`, εμποδίζουν την παραγωγή αρκετά βελτιστοποιημένου κώδικα από τον συμπιλεστή για τον εσωτερικό βρόχο. Δεδομένου του αυξημένου διαθέσιμου εύρους ζώνης μνήμης, αυτή η υπολογιστική επιβάρυνση εκτίθεται περισσότερο, ενώ και η επιβάρυνση βρόχων λόγω των πολύ μικρών γραμμών είναι πιο έντονη.

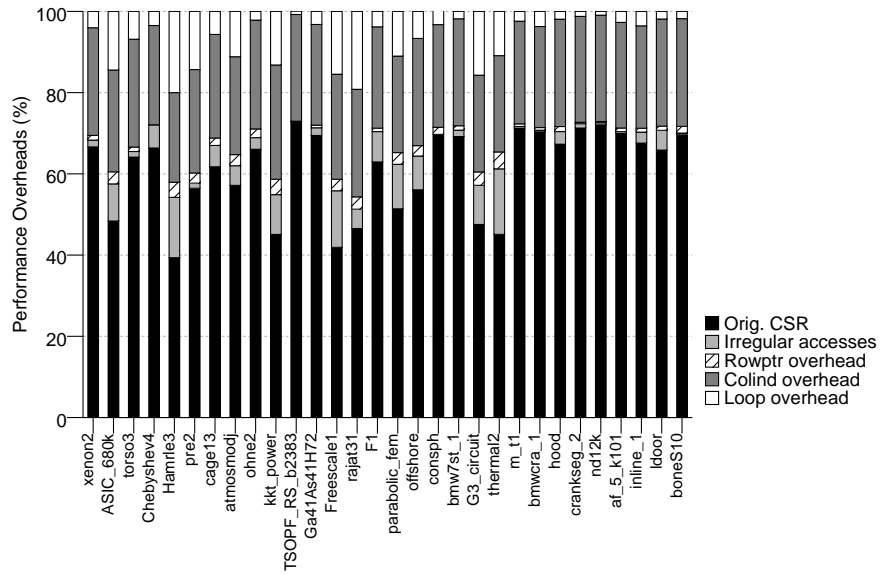
3.3.2 Πολυνηματική επίδοση

Ο πυρήνας SpMV παρουσιάζει πολύ μεγάλο παραλληλισμό [Buluc et al., 2011]. Πράγματι, εάν ο αραιός πίνακας χωριστεί κατά γραμμές και σε κάθε νήμα ανατεθεί ένα σύνολο γραμμών, όλα τα νήματα μπορούν να προχωρήσουν τελείως ανεξάρτητα χωρίς καμία ανάγκη επικοινωνίας. Η μόνη μέριμνα που πρέπει να ληφθεί αφορά στην ισορροπημένη κατανομή του φόρτου εργασία μεταξύ των νημάτων. Προφανώς, η ανάθεση σε κάθε νήμα του ίδιου αριθμού γραμμών, όπως συμβαίνει στην περίπτωση των πυκνών πινάκων, δεν αποτελεί επιλογή στην περίπτωση των αραιών πινάκων, μιας και η κατανομή των μη μηδενικών στοιχείων δεν είναι πάντοτε ομοιόμορφη. Το καλύτερο σχήμα στατικής εξισορρόπησης του φόρτου εργασίας, επομένως, είναι ο διαχωρισμός του πίνακα κατά γραμμές έτσι, ώστε σε κάθε νήμα να ανατίθεται περίπου ο ίδιος αριθμός μη μηδενικών στοιχείων. Δυστυχώς, παρά το μεγάλο παραλληλισμό και μία

3. Η επίδοση του πυρήνα SpMV

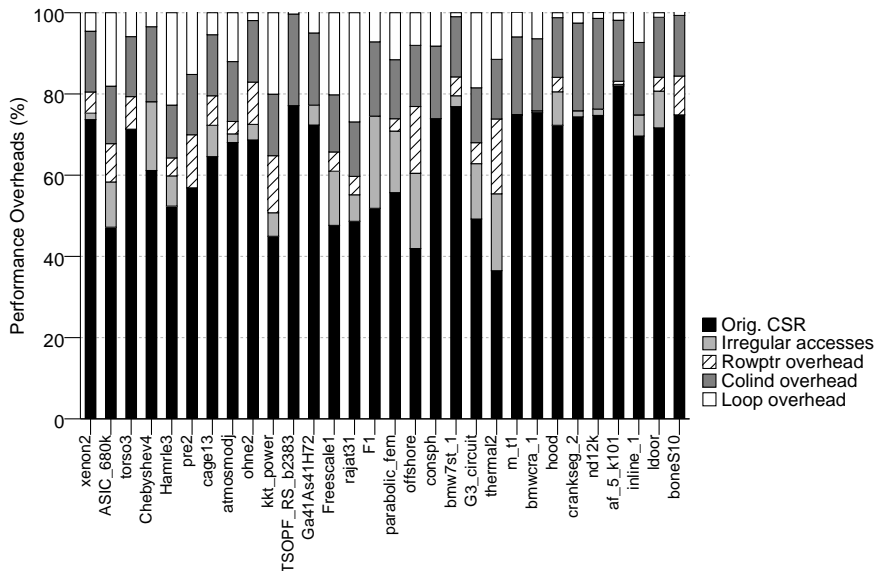


(i) Harpertown.



(ii) Dunnington.

Σχήμα 3.2: Επιβαρύνσεις της επίδοσης του πυρήνα SpMV λόγω των απαιτούμενων δομών δεδομένων και του αγνώστου σχήματος πρόσβασης στο διάνυσμα εισόδου (συνεχίζεται στη σελ. 55). [Οι ράβδοι προκύπτουν από την διαδοχική εφαρμογή των μετροπρογραμμάτων του Πίνακα 3.3: “Irregular accesses” → *noxmiss*, “Rowptr overhead” → *norowptr*, “Colind overhead” → *nocolind*, “Loop overhead” → το υπόλοιπο από το 100%].



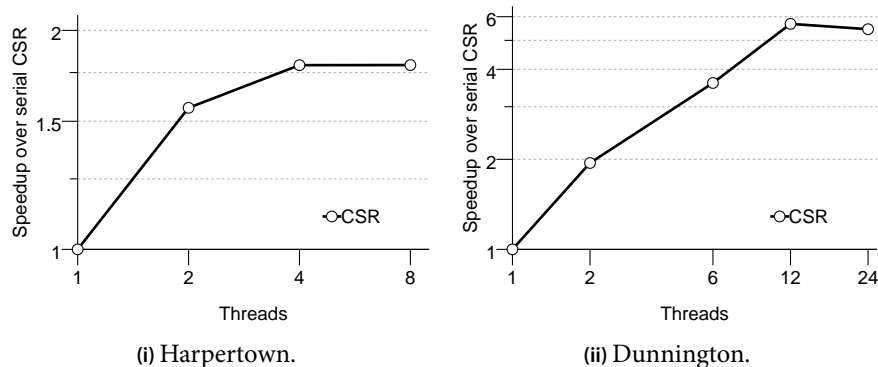
(iii) Gainestown.

Σχήμα 3.2 (συνέχεια): Επιβαρύνσεις της επίδοσης του πυρήνα S_pMV λόγω των απαιτούμενων δομών δεδομένων και του αγνώστου σχήματος πρόσβασης στο διάνυσμα εισόδου (συνεχίζεται στη σελ. 55). [Οι ράβδοι προκύπτουν από την διαδοχική εφαρμογή των μετροπρογραμμάτων του Πίνακα 3.3: “Irregular accesses” → *noxmiss*, “Rowptr overhead” → *norowptr*, “Colind overhead” → *nocolind*, “Loop overhead” → το υπόλοιπο από το 100%].

αξιοπρεπή εξισορρόπηση του φόρτου εργασίας, η παράλληλη εκτέλεση του πυρήνα S_pMV δεν κλιμακώνεται.

Το Σχήμα 3.3 δείχνει την κλιμάκωση του πυρήνα S_pMV χρησιμοποιώντας το σχήμα CSR στα συστήματα Harpertown και Dunnington, χρησιμοποιώντας την πολιτική ανάθεσης «share-nothing» (βλ. Ενότητα 3.2.3). Στο σύστημα Harpertown, ο πυρήνας S_pMV σταματά να κλιμακώνεται στους τέσσερις πυρήνες, επιτυγχάνοντας μόνο 1.8× επιτάχυνση. Η κλιμάκωση στο σύστημα Dunnington ευνοείται από την μεγάλη κρυφή μνήμη L3 κάθε επεξεργαστή, η οποία παρέχει μία συνολική κρυφή μνήμη 64 MiB στο σύστημα. Ο πυρήνας S_pMV κλιμακώνεται γραμμικά μέχρι τους δύο πυρήνες και συνεχίζει με περίπου μισό ρυθμό μέχρι τους 12 πυρήνες, όπου η κλιμάκωση σταματά απότομα και παρατηρείται μία κάμψη της επίδοσης καθώς προχωράμε στην 24-νηματική εκτέλεση. Το κύριο σημείο συμφόρησης για την πλειονότητα των πινάκων στις αρχιτεκτονικές SMP είναι το περιορισμένο εύρος ζώνης μνήμης.

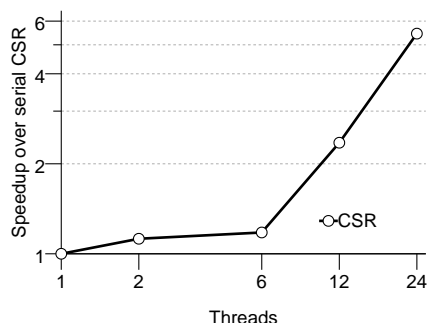
3. Η επίδοση του πυρήνα SpMV



Σχήμα 3.3: Κλιμάκωση του πυρήνα SpMV σε δύο συστήματα SMP χρησιμοποιώντας την πολιτική πλήρωσης των πυρήνων «share-nothing», η οποία μεγιστοποιεί την χρησιμοποίηση του διαθέσιμου εύρους ζώνης μνήμης.

Όλα τα νήματα ανταγωνίζονται μεταξύ τους για την πρόσβαση στον κοινό εμπρόσθιο διάυλο και, έτσι, ο αλγόριθμος σειριοποιείται εμμέσως στο σημείο αυτό, καθώς τα νήματα περιμένουν να εξυπηρετηθούν. Ο εγγενώς χαμηλός λόγος flops:byte του SpMV καθιστά αυτό το πρόβλημα αναπόφευκτο και ο πυρήνας αργά ή γρήγορα θα χτυπήσει σε ένα «τοίχο μνήμης». Όντως, το 85% του διαθέσιμου εύρους ζώνης μνήμης στο σύστημα Harperton καταναλώνεται ήδη από την δι-νηματική εκτέλεση, ενώ κορέννεται πλήρως από την τετρα-νηματική και έπειτα. Ομοίως στο σύστημα Dunnington, οι μεγάλοι πίνακες με κανονική διάταξη καταναλώνουν το 75% του διαθέσιμου εύρους ζώνης μνήμης στη εξα-νηματική εκτέλεση, ενώ στη 12-νηματική έχει πλήρως κορεστεί.

Για να διαφανεί καλύτερη η επίδραση του εύρους ζώνης μνήμης, παρουσιάζουμε στο Σχήμα 3.4 την κλιμάκωση που επιτεύχθηκε στο σύστημα Dunnington χρησιμοποιώντας την πολιτική πλήρωσης των πυρήνων «share-all». Σύμφωνα με αυτή την πολιτική, αναθέτουμε τα νήματα στους πυρήνες έτσι, ώστε οι διαθέσιμοι επεξεργαστές να γεμίζουν διαδοχικά. Για παράδειγμα, η εξα-νηματική εκτέλεση χρησιμοποιεί μόνο ένα επεξεργαστή, ενώ η 12-νηματική χρησιμοποιεί μόνο δύο. Η διαφορά της επίδοσης μεταξύ των δύο πολιτικών είναι τρομακτική μέχρι και την 12-νηματική εκτέλεση, καθώς τα νήματα ανταγωνίζονται όχι μόνο για τον χώρο στην κρυφή μνήμη (πλέον μόνο 16 MiB και 32 MiB είναι διαθέσιμα, αντίστοιχα), αλλά και για την πρόσβαση στον κοινό ελεγκτή του εμπρόσθιου διαύλου. Το εύρος ζώνης μνήμης που επιτυγχάνεται μέσω του ενός ελεγκτή FSB μετρήθηκε στα 2.6 GB/s και καταναλώνεται σχεδόν πλήρως ήδη από την σειριακή εκτέλεση, εξαλείφοντας κάθε δυνατότητα κλιμάκωσης. Μόλις ο υπολογισμός μεταφερθεί και στον δεύτερο επεξεργαστή, αμφότερα η



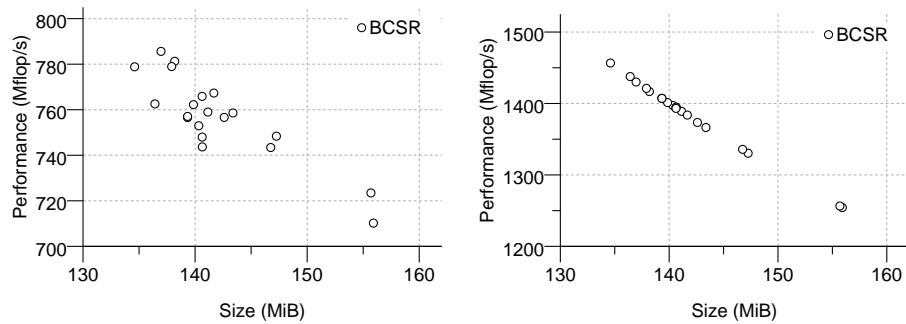
Σχήμα 3.4: Κλιμάκωση του πυρήνα SpMV στο σύστημα Dunnington χρησιμοποιώντας την πολιτική πλήρωσης των πυρήνων «share-all»: ο κορεσμός του εύρους ζώνης μνήμης είναι φανερός εντός του ενός επεξεργαστή (μέχρι τα έξι νήματα), όπου παρατηρείται σχεδόν μηδενική επιτάχυνση της επίδοσης.

συνολική κρυφή μνήμη και το εύρος ζώνης μνήμης διπλασιάζονται, το ίδιο και η επιτάχυνση της επίδοσης, η οποία συνεχίζει να αυξάνει γρήγορα μέχρι την 24-νηματική εκτέλεση.

Πρέπει να έχει ήδη φανερό από την μέχρι τώρα ανάλυση ότι η συμμόρφωση στο εύρος ζώνης μνήμης αποτελεί ένα σημαντικό θέμα κλιμάκωσης του πυρήνα SpMV στις αρχιτεκτονικές SMP. Λόγω της αλγοριθμικής του φύσης και της πολύ χαμηλής υπολογιστικής του έντασης, ο πυρήνας SpMV τείνει να κορέσει γρήγορα τους διαθέσιμους πόρους του επεξεργαστή, καθώς προστίθενται περισσότερα νήματα στον υπολογισμό. Η συμπεριφορά αυτή καταλήγει σε μία σχεδόν γραμμική σχέση μεταξύ της επίδοσης του SpMV και του μεγέθους αναπαράστασης του πίνακα, όπως φαίνεται στο Σχήμα 3.5. Σε αυτό το σχήμα φαίνεται η επίδοση του σχήματος BCSR για διαφορετικά μεγέθη μπλοκ για τον σχετικά πυκνο και κυριαρχούμενο από μπλοκ πίνακα TSOPF_RS_b2383. Παρά την εμφανή τάση στην σειριακή εκτέλεση προς χαμηλότερη επίδοση καθώς το μέγεθος αναπαράστασης αυξάνεται, η σχέση αυτή γίνεται τελείως γραμμική καθώς προσθέτουμε περισσότερα νήματα που πιέζουν το κοινό εμπρόσθιο δίαυλο. Σε επίπεδο πίνακα, η σχέση αυτή διακρίνεται ακαλύπτερα στο Σχήμα 3.6, όπου φαίνεται η σχέση της επίδοσης του SpMV για κάθε πίνακα της σουίτας με την αντίστοιχη υπολογιστική ένταση (λόγος flops:byte) για την σειριακή και πολυνηματική εκτέλεση στο σύστημα Harpertown. Η τάση προς υψηλότερη επίδοση καθώς αυξάνεται ο λόγος flops:byte, ειδικότερα στις πολυνηματικές εκτελέσεις, είναι μία σαφής ένδειξη του κορεσμού του εύρους ζώνης μνήμης από τον πυρήνα SpMV³. Στην πράξη μπορούμε να ξεχωρίσουμε εύκο-

³ Οι Williams et al. [2009] αναδεικνύουν επίσης αυτή την συμπεριφορά μέσω του μοντέλου τους

3. Η επίδοση του πυρήνα SpMV



(i) Harpertown, σειριακή εκτέλεση.

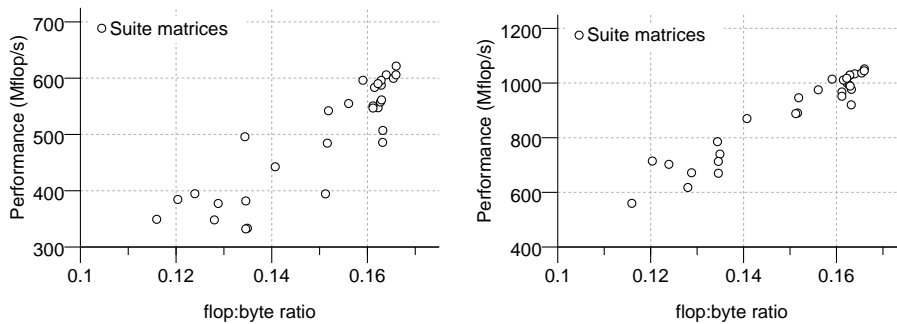
(ii) Harpertown, πολυνηματική εκτέλεση (οκτώ νήματα).

Σχήμα 3.5: Η συμμόρφωση στον διάυλο μνήμης των αρχιτεκτονικών SMP, καθώς ολοένα και περισσότερα νήματα προστίθενται στον υπολογισμό, καθιστά την επίδοση του πυρήνα SpMV ιδιαίτερα ευαίσθητη στο μέγεθος αναπαράστασης του πίνακα. Τα αποτελέσματα του σχήματος αφορούν στο σχήμα BCSR με διαφορετικά μεγέθη μπλοκ για τον πίνακα TSOPF_RS_b2383.

λα δύο διαφορετικές κατηγορίες πινάκων: τους «χαμηλών επιδόσεων» με λόγο $\text{flop:byte} < 1.5$ και τους «υψηλών επιδόσεων» με λόγο ≥ 1.5 . Η διακύμανση της επίδοσης στο κάτω άκρο του φάσματος των πινάκων είναι μεγαλύτερη, καθώς αυτοί οι πίνακες δεν υποφέρουν μόνο από επιπλέον επιβαρύνσεις (ακανόνιστες προσβάσεις, μικρές γραμμές), αλλά είναι αρκετά πιθανό να παρουσιάζουν και σημαντικές ανισορροπίες στην κατανομή του φόρτου εργασίας, λόγω της ακανόνιστης κατανομής των μη μηδενικών στοιχείων τους.

Θέματα εξισορρόπησης φόρτου εργασίας Ο βασισμένος στα μη μηδενικά στοιχεία στατικός διαχωρισμός του πίνακα αποτελεί ένα λογικό σχήμα εξισορρόπησης του φόρτου εργασίας για τον πυρήνα SpMV και μπορεί να οδηγήσει σε μία δίκαιη κατανομή του φόρτου ανάμεσα στα νήματα. Ωστόσο, όπως φαίνεται στο Σχήμα 3.7, το σχήμα αυτό δείχνει να είναι ανεπαρκές για πίνακες με αρκετά ακανόνιστη διάταξη των μη μηδενικών στοιχείων τους. Ο κύριος λόγος για αυτή την συμπεριφορά έγκειται στο γεγονός ότι αυτό το σχήμα δεν λαμβάνει υπόψη την κατανομή των μη μηδενικών στοιχείων εντός της διαμέρισης. Για παράδειγμα, δύο διαμερίσεις μπορεί να έχουν τον ίδιο αριθμό μη μηδενικών στοιχείων, αλλά διαφορετικά σχήματα προσβάσεων στο διάνυσμα εισόδου, με το ένα να είναι πιο ακανόνιστο· ο πυρήνας SpMV θα είναι, επομένως, αργός σε αυτή την διαμέριση, εμποδίζοντας την πολυνηματική εκτέλεση στο σύνολο

«Roofline», όπου η επίδοση των περιορισμένων από το εύρος ζώνης μνήμης εφαρμογών αυξάνεται αντίστοιχα με τον λόγο flop:byte .



(i) Harpertown, σειριακή εκτέλεση.

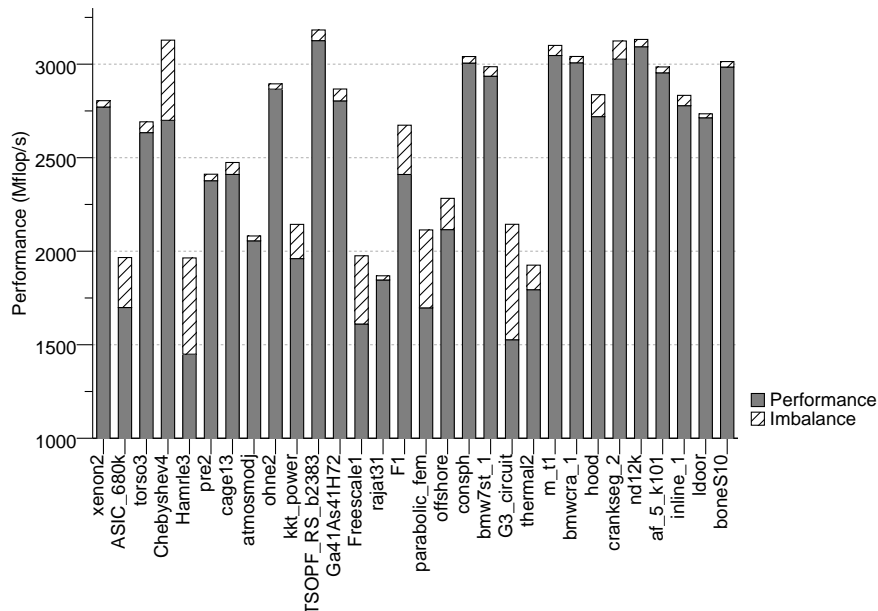
(ii) Harpertown, πολυνηματική εκτέλεση (οκτώ νήματα).

Σχήμα 3.6: Η επίδοση του πυρήνα SpMV σε σχέση με την υπολογιστική ένταση (λόγος flop:byte). Η τάση για υψηλότερη επίδοση καθώς αυξάνεται ο λόγος flop:byte γίνεται πιο εμφανής στις πολυνηματικές εκτελέσεις, αναδεικνύοντας το πρόβλημα συμφόρησης στο υποσύστημα μνήμης.

λό της. Τα πιο χαρακτηριστικά παραδείγματα αυτής της κατάστασης είναι οι πίνακες Hamrle3 και G3_circuit. Το Σχήμα 3.8 παρουσιάζει την διακύμανση του λόγου flop:byte κατά μήκος αυτών των πινάκων. Και στις δύο περιπτώσεις η κατανομή των μη μηδενικών στοιχείων είναι μάλλον ακανόνιστη με τον πίνακα Hamrle3 να είναι αρκετά αραιός στις τελευταίες διαμερίσεις. Μία πιο προσεκτική μελέτη της δομής του πίνακα Hamrle3 αποκαλύπτει ότι το πρώτο μισό του έχει αρκετά κανονική δομή, ενώ το υπόλοιπο είναι αρκετά ακανόνιστο με μεγάλες και κυμαινόμενες αποστάσεις μεταξύ των μη μηδενικών στοιχείων της γραμμής. Τα μη μηδενικά στοιχεία του πίνακα G3_circuit ακολουθούν μια πιο ομοιόμορφη κατανομή· ωστόσο, η πρώτη διαμέριση επιβαρύνεται από την πιο αραιή δομή της, οδηγώντας σε χαμηλότερης υπολογιστική ένταση και, αντίστοιχα, μεγαλύτερη επιβάρυνση βρόχων.

Οι τεχνικές ελαχιστοποίησης του εύρους ζώνης του πίνακα (βλ. Ενότητα 2.4) αποτελούν ένα μέσο για την ομογενοποίηση της κατανομής των μη μηδενικών στοιχείων ενός αραιού πίνακα. Οι τεχνικές αυτές προσπαθούν να φέρουν όλα τα μη μηδενικά στοιχεία όσο το δυνατόν πιο κοντά στην κύρια διαγώνιο, αναδιατάσσοντας τις γραμμές και τις στήλες του πίνακα. Παρά το ότι η αρχική ιδέα πίσω από αυτές τις τεχνικές ήταν η εξισορρόπηση του μεγέθους της επικοινωνίας (διανύσματα εισόδου/εξόδου) για υλοποιήσεις του πυρήνα SpMV σε περιβάλλοντα κατανεμημένης μνήμης, αποδεικνύονται αποτελεσματικές και στην περίπτωση των σύγχρονων πολυπύρηνων αρχιτεκτονικών. Η μεταφορά των μη μηδενικών στοιχείων κοντά στην κυρία διαγώνιο δημιουργεί ένα πιο κανονικό σχήμα προσβάσεων στο διάνυσμα εισόδου με μι-

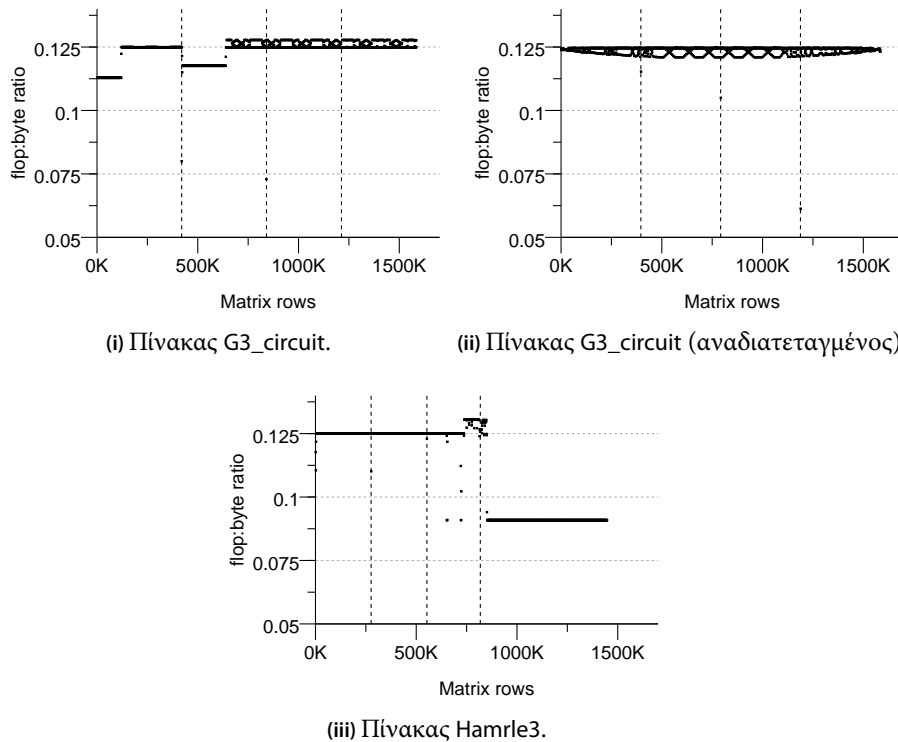
3. Η επίδοση του πυρήνα SpMV



Σχήμα 3.7: Ανισορροπία στην κατανομή του φόρτου εργασίας του πυρήνα SpMV. Οι πίνακες με ακανόνιστη κατανομή των μη μηδενικών στοιχείων τους υποφέρουν από σημαντικές ανισορροπίες, παρά το «δίκαιο» στατικό σχήμα εξισορρόπησης, που βασίζεται στο πλήθος των μη μηδενικών στοιχείων. Τα αποτελέσματα του σχήματος αφορούν την εκτέλεση σε ένα επεξεργαστή (τέσσερα νήματα) του συστήματος Gainestown.

κρότερα και πιο προβλέψιμα βήματα, ενώ ο στόχος για την εξισορρόπηση του κόστους επικοινωνίας οδηγεί σε μία πιο ομοιόμορφη κατανομή των μη μηδενικών στοιχείων του πίνακα. Το Σχήμα 3.8 παρουσιάζει την μεταβολή του λόγου flop:byte για τον πίνακα G3_circuit πριν και μετά την εφαρμογή του αλγορίθμου αναδιάταξης Reverse Cuthill-McKee (RCM) [Cuthill και McKee, 1969]⁴. η κατανομή των μη μηδενικών στοιχείων του πίνακα έχει πλέον ομογενοποιηθεί πλήρως και η ανισορροπία στην κατανομή του φόρτου εργασίας έχει εξαλειφθεί, όπως απεικονίζεται και στο Σχήμα 3.9. Μία ακόμα σημαντική πλευρά των τεχνικών αναδιάταξης που απεικονίζεται στο σχήμα αυτό είναι ότι το κέρδος στην επίδοση δεν προέρχεται μόνο από την καλύτερη εξισορρόπηση των υπολογισμών· το πιο κανονικό σχήμα προσβάσεων στο διάνυσμα εισόδου, ως αποτέλεσμα του μικρότερου εύρους ζώνης του πίνακα, παρέχει και αυτό ένα σημαντικό πλεονέκτημα στην επίδοση του πυρήνα SpMV.

⁴ Ο αλγόριθμος RCM, όπως επίσης και άλλοι αλγόριθμοι αναδιάταξης, π.χ., METIS [Karypis και Kumar, 1995], μπορούν να εφαρμοστούν μόνο σε πίνακες με συμμετρική δομή. Επομένως, δεν είναι η δυνατή η αναδιάταξη του πίνακα Hamrle3 στο παράδειγμά μας.

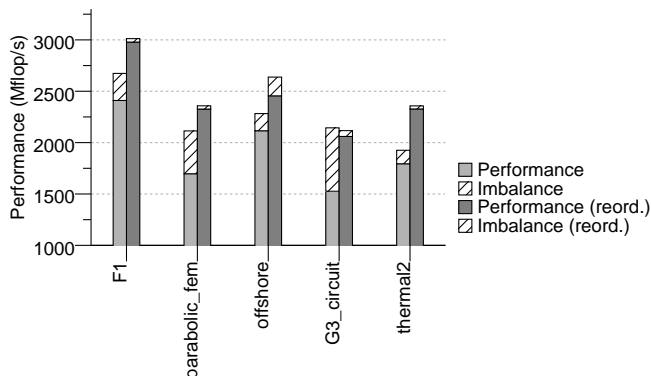


Σχήμα 3.8: Μεταβολή του πίνακα flop:byte κατά μήκος του πίνακα. Οι διαμερίσεις σημειώνονται με κατακόρυφες διακεκομμένες γραμμές. Η διαφορετική υπολογιστική ένταση ανάμεσα στις διαμερίσεις οδηγεί σε σημαντικές ανισορροπίες στην κατανομή του φόρτου εργασίας. Οι τεχνικές αναδιάταξης ομογενοποιούν την δομή του πίνακα και εξισορροπούν τους υπολογισμούς, αλλά μπορούν να εφαρμοστούν μόνο σε πίνακες με συμμετρική δομή· ο πίνακας Hamrle3 δεν έχει τέτοια δομή.

Αρχιτεκτονικές NUMA

Η συμπεριφορά του πολυνηματικού πυρήνα S_pMV στις αρχιτεκτονικές NUMA πηγάζει και σ' αυτή την περίπτωση από την υπολογιστική φύση του και από το πολύ μεγάλο σύνολο εργασίας του. Σε ένα σύστημα NUMA (Σχήμα 3.1iii), κάθε επεξεργαστής έχει τον δικό του ενσωματωμένο ελεγκτή μνήμης, ο οποίος είναι υπεύθυνος για την πρόσβαση σε μία συγκεκριμένη διαμέριση (κόμβο) της κύριας μνήμης. Όλες οι προσβάσεις στην κυρία μνήμη που προέρχονται από ένα επεξεργαστή προς τον οικείο του κόμβο εξυπηρετούνται από τον δικό του ελεγκτή μνήμης, ενώ όλες οι απομακρυσμένες προσβάσεις δρομολογούνται μέσω του ελεγκτή μνήμης ενός ομότιμου επεξεργαστή, χρησιμοποιώντας ένα

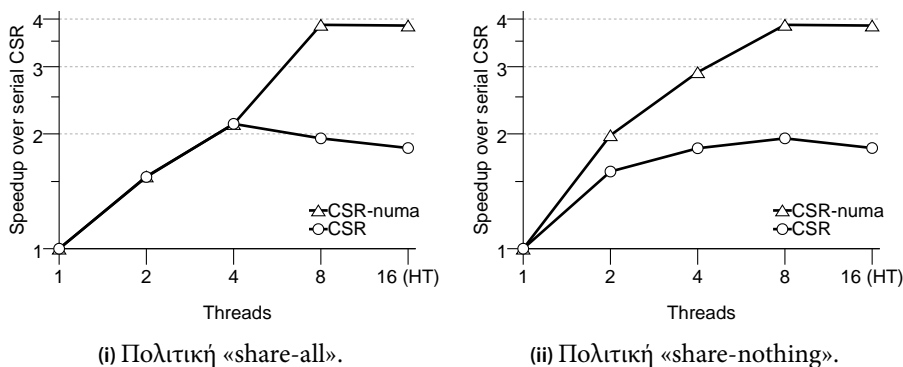
3. Η επίδοση του πυρήνα SpMV



Σχήμα 3.9: Η ωφέλεια της αναδιάταξης του πίνακα. Οι εξισορροπημένοι υπολογισμοί και η πιο κανονική δομή του πίνακα, που επιτρέπει την καλύτερη χρησιμοποίηση του διανύσματος εισόδου, οδηγούν σε μία σημαντική βελτίωση της επίδοσης του πυρήνα SpMV.

δίκτυο διασύνδεσης υψηλής ταχύτητας. Στην περίπτωση του Gainestown, το εύρος ζώνης του δικτύου διασύνδεσης είναι σημαντικά χαμηλότερο από το εύρος ζώνης μνήμης του ενσωματωμένου ελεγκτή μνήμης (βλ. Πίνακα 3.2). Επομένως, οι απομακρυσμένες προσβάσεις είναι πιθανό να κορέννουν το δίκτυο διασύνδεσης, περιορίζοντας την επιτάχυνση μιας παράλληλης εφαρμογής.

Το κλειδί για την επίτευξη υψηλής επίδοσης σε μία εφαρμογή με μεγάλη ένταση αναφορών στην μνήμη σε ένα σύστημα NUMA είναι η σωστή τοποθέτηση των δεδομένων της στους διαθέσιμους κόμβους μνήμης, ώστε να ελαχιστοποιηθούν οι απομακρυσμένες προσβάσεις και να αποφευχθούν τυχόν συμφορήσεις στο δίκτυο διασύνδεσης. Αυτό φαίνεται καλύτερα στο Σχήμα 3.10, όπου παρουσιάζεται η κλιμάκωση του πυρήνα SpMV στο σύστημα Gainestown χρησιμοποιώντας τις δύο τεχνικές πλήρωσης των πυρήνων για την απλή και την κατά-NUMA (NUMA-aware) υλοποίηση. Η τυπική υλοποίηση δεν ασχολείται με την τοποθέτηση των δεδομένων του SpMV στους κόμβους μνήμης και επαφίεται στο λειτουργικό σύστημα για την σωστή τους τοποθέτηση. Το Linux χρησιμοποιεί μία πολιτική «αντιγραφή-κατά-την-εγγραφή» (copy-on-write) για την εκχώρηση φυσικών σελίδων μνήμης και θα εκχωρήσει μία φυσική σελίδα μνήμης στον κόμβο μνήμης, στον οποίο τρέχει το νήμα που πρώτο θα γράψει στην αντίστοιχη εικονική σελίδα. Επομένως, στην τυπική υλοποίηση του SpMV, η εκχώρηση των σελίδων θα λάβει χώρα κατά την αρχικοποίηση του αλγορίθμου, όπου ο πίνακας και τα σχετικά διανύσματα κατασκευάζονται και αρχικοποιούνται. Η φάση αυτή είναι συνήθως σειριακή, επομένως, όλα τα δεδομένα του πυρήνα θα βρίσκονται σε ένα μόνο κόμβο μνήμης. Κατά συνέπεια, εάν ένα νήμα υπολογισμού του SpMV ανατεθεί σε διαφορετι-

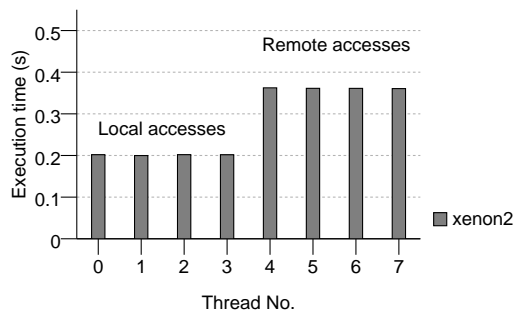


Σχήμα 3.10: Η κλιμάκωση του πυρήνα SpMV στο σύστημα NUMA Gainestown. Η επίδραση της τοποθέτησης των δεδομένων είναι πολύ σημαντική, καθώς οι απομακρυσμένες προσβάσεις στην μνήμη μπορούν εύκολα να οδηγήσουν το δίκτυο διασύνδεσης των επεξεργαστών σε κορεσμό. Το επίθεμα «-uma» δηλώνει τοποθέτηση κατά-NUMA, ενώ το «HT» στις πολυνηματικές εκτελέσεις δηλώνει την χρησιμοποίηση του χαρακτηριστικού HyperThreading.

κό κόμβο, όλες οι προσβάσεις του στην μνήμη θα πρέπει να δρομολογηθούν μέσω του δικτύου διασύνδεσης, το οποίο πιθανώς και να κορεστεί.

Χρησιμοποιώντας την πολιτική «share-all» (Σχήμα 3.10i), τα νήματα ανατίθενται σε ένα μοναδικό κόμβο μνήμης (επεξεργαστή) μέχρι την τετρα-νηματική εκτέλεση, ενώ στην οκτώ-νηματική και 16-νηματική εκτέλεση ανατίθενται και στους δύο κόμβους μνήμης. Το σημαντικό εύρος ζώνης μνήμης που προσφέρεται από τον ενσωματωμένο ελεγκτή μνήμης επιτρέπει στον πυρήνα SpMV να κλιμακώνεται καλά στην εκτέλεση στον ένα επεξεργαστή, επιτυγχάνοντας $2.1 \times$ επιτάχυνση, ξεπερνώντας κατά πολύ όχι μόνο τις εκτελέσεις σε ένα επεξεργαστή SMP, αλλά και την εκτέλεση σε ολόκληρο το σύστημα Harpertown. Η δι-νηματική εκτέλεση καταναλώνει το 70% του διαθέσιμου εύρους μνήμης, ενώ η τετρα-νηματική κορέννει τον ελεγκτή. Ξεκινώντας να χρησιμοποιεί κανείς τον δεύτερο επεξεργαστή, ωστόσο, η τοποθέτηση των δεδομένων παίζει πλέον σημαντικό ρόλο στην επίδοση του πυρήνα. Η τυπική υλοποίηση του SpMV αντιμετωπίζει μία σημαντική πτώση στην επίδοση που φθάνει το 14%, όταν χρησιμοποιείται το πλήρες σύστημα, ενώ η επίδοσης της κατά-NUMA έκδοσης συνεχίζει να κλιμακώνεται σημαντικά μέχρι την οκτώ-νηματική εκτέλεση, οπότε αμφότεροι οι ελεγκτές μνήμης έχουν κορεστεί. Μια πιο προσεκτική μελέτη των χρόνων εκτέλεσης του κάθε νήματος στην τυπική υλοποίηση του SpMV για την οκτώ-νηματική εκτέλεση (Σχήμα 3.11) αποκαλύπτει μία σημαντική ανισορροπία μεταξύ των νημάτων που ανατίθενται στον τοπι-

3. Η επίδοση του πυρήνα SPMV



Σχήμα 3.11: Ο κορεσμός του δικτύου διασύνδεσης των αρχιτεκτονικών NUMA αυξάνει τον χρόνο εκτέλεσης των νημάτων που πραγματοποιούν απομακρυσμένες προσβάσεις, οδηγώντας σε σημαντική ανισορροπία του φόρτου εργασίας. Η επίδοση του SPMV καθορίζεται, επομένως, από την το εύρος ζώνης του δικτύου διασύνδεσης. Τα αποτελέσματα στο σχήμα αφορούν στον πίνακα xenon2, αλλά είναι παρόμοια για όλους τους πίνακες με κανονική διάταξη των στοιχείων τους.

κό και στον απομακρυσμένο επεξεργαστή. Πράγματι, παρατηρείται μία διαφορά στην επίδοση της τάξης του 45%, η οποία ταιριάζει με την αντίστοιχη διαφορά μεταξύ εύρους ζώνης μνήμης και δικτύου διασύνδεσης, πράγμα που υποδηλώνει την συμφόρηση στο δίκτυο διασύνδεσης.

Το διάγραμμα κλιμάκωσης της επίδοσης για την πολιτική πλήρωσης των πυρήνων «share-nothing» (Σχήμα 3.10ii) αναδεικνύει καλύτερα την διαφορά στο εύρος ζώνης μεταξύ του ελεγκτή μνήμης και του δικτύου διασύνδεσης. Η συμπεριφορά της τυπικής υλοποίησης του σχήματος CSR θυμίζει την αντίστοιχη εκτέλεση στα συστήματα SMP. Η μισή κίνηση προς την μνήμη στις πολυνηματικές εκτελέσεις πρέπει να περάσει μέσα από το δίκτυο διασύνδεσης, 80% του οποίου έχει ήδη καταναλωθεί ήδη στην τετρα-νηματική εκτέλεση, οδηγώντας τελικά σε μία μέγιστη επιτάχυνση λιγότερη από $2\times$ στην οκτώ-νηματική εκτέλεση. Η συμπεριφορά της κατά-NUMA υλοποίησης είναι η ακριβώς αντίθετη: το υπεραρκετό εύρος ζώνης μνήμης (≈ 30 GB/s), που προσφέρουν οι δύο ενσωματωμένοι ελεγκτές μνήμης, επιτρέπει στον πυρήνα SPMV να κλιμακώσει σημαντικά την επίδοσή του μέχρι και την οκτώ-νηματική εκτέλεση, επιτυγχάνοντας ακόμα και γραμμική κλιμάκωση μέχρι τα δύο νήματα, όπου ακόμα δεν αντιμετωπίζεται κανένα πρόβλημα συμφόρησης.

Διαφανής τοποθέτηση δεδομένων σε αρχιτεκτονικές NUMA

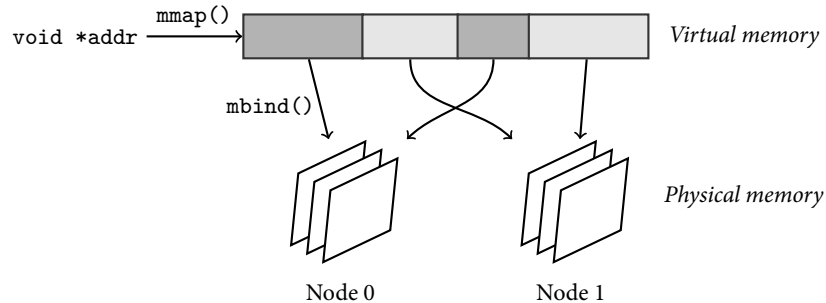
Η συζήτηση για τα αποτελέσματα της επίδοσης του πυρήνα SPMV στις αρχιτεκτονικές NUMA αποκάλυψε ότι η σωστή τοποθέτηση των δεδομένων του πυρήνα είναι καθοριστικής σημασίας για την επίτευξη υψηλής επίδοσης. Μία

τέτοια τοποθέτηση απαιτεί από το κάθε νήμα να καταχωρεί εκείνο τα δικά του δεδομένα (διαμερίσεις πίνακα και διανύσματος εξόδου, διάνυσμα εισόδου) στον οικείο του κόμβο. Ωστόσο, μια τέτοια καταχώρηση απαιτεί την τροποποίηση του αλγορίθμου SPMV. Στην ουσία, θα πρέπει ο αρχικός πίνακας να χωριστεί σε ανεξάρτητους υποπίνακες ανά νήμα και ο υπολογισμός του SPMV θα πρέπει να τροποποιηθεί, ώστε να εφαρμοστεί σε αυτούς. Αντίστοιχα, το διάνυσμα εισόδου θα πρέπει είτε να διαιρεθεί σε υποδιανύσματα είτε να καταχωρηθεί ολόκληρο σε ένα μοναδικό κόμβο μνήμης. Οι αλλαγές αυτές για την κατά-NUMA μετατροπή ενός κώδικα βασισμένου στη λογική των συστημάτων SMP απαιτεί σημαντική προγραμματιστική προσπάθεια, παρόλο που αυτό που πρέπει να αλλάξει στην πραγματικότητα είναι μόνο η τοποθέτηση των δεδομένων του αλγορίθμου.

Αντιμετωπίζουμε το πρόβλημα της σωστής τοποθέτησης των δεδομένων με την ανάπτυξη ενός εξειδικευμένου σχήματος εκχώρησης μνήμης χαμηλού επιπέδου. Το σχήμα αυτό επιτρέπει στον πυρήνα SPMV να διατηρήσει το κατά-SMP «φαίνεσθαι» με την πλήρη απόκρυψη της τοποθέτησης των δεδομένων από τον κώδικα της εφαρμογής. Για να διατηρηθεί το κατά-SMP «φαίνεσθαι», θα πρέπει να εγγυηθεί η συνεχόμενη καταχώρηση των δομών του πίνακα στην εικονική μνήμη, ενώ η κατά-NUMA τοποθέτηση μπορεί να επιτευχθεί διαφανώς ως προς το χρήστη με την αντιστοίχιση των εικονικών σελίδων στους φυσικούς φυσικούς κόμβους μνήμης (βλ. Σχήμα 3.12). Αυτό μπορεί να επιτευχθεί εύκολα στο Linux, με την χρήση των κλήσεων συστήματος που παρέχει για την διαχείριση μνήμης. Πιο συγκεκριμένα, καταχωρούμε την δομή δεδομένων συνεχόμενα στον εικονικό χώρο διευθυνσεων με μία κλήση στη συνάρτηση `mmap()` και στην συνέχεια προσδένουμε κάθε διαμέριση στον επιθυμητό κόμβο με διαδοχικές κλήσεις στην συνάρτηση `mbind()`, στρογγυλοποιώντας τα μεγέθη των διαμερίσεων στο πλησιέστερο πολλαπλάσιο του μεγέθους της σελίδας του συστήματος⁵. Ο μηχανισμός αυτός επαφίεται στο σχήμα εκχώρησης φυσικών σελίδων «copy-on-write» που χρησιμοποιεί το Linux. Η κλήση της `mmap()` δεσμεύει απλά τον απαιτούμενο χώρο στην εικονική μνήμη για το καλόν νήμα, χωρίς να εκχωρεί καμία φυσική σελίδα, ενώ η κλήση στην `mbind()` σημειώνει την δεδομένη περιοχή μνήμης, ώστε να εκχωρηθεί στον επιθυμητό φυσικό κόμβο μνήμης· αυτή η εκχώρηση δεν θα λάβει χώρα μέχρις ότου κάποιο νήμα γράψει στην περιοχή αυτή. Αυτός, ο οριζόμενος από τον χρήστη, παρεμβλητικός (interleaved) τρόπος εκχώρησης της μνήμης επιτρέπει την διαφανή μετατροπή ενός κατά-SMP κώδικα σε κατά-NUMA, με την απλή αντικατάσταση των τυπικών κλήσεων εκχώρησης μνήμης με κλήσεις στον παρεμβλητικό εκχωρητή μνήμης που προτείνουμε· ο υπόλοιπος αλγόριθμος παραμένει

⁵ Το ελάχιστο μέγεθος εκχώρησης και διαχείρισης της μνήμης στο Linux είναι η σελίδα, συνήθως 4 KiB στα συστήματα 64-bit.

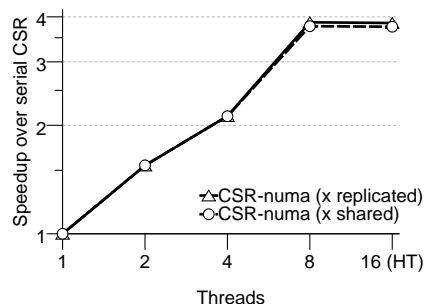
3. Η επίδοση του πυρήνα SPMV



Σχήμα 3.12: Διαφανής τοποθέτηση δεδομένων στις αρχιτεκτονικές NUMA στο Linux. Ο χώρος που απαιτείται για μία δομή δεδομένων καταχωρείται συνεχόμενα στην εικονική μνήμη, ενώ οι επιμέρους διαμερίσεις προσδίδονται στην συνέχεια σε συγκεκριμένους φυσικούς κόμβους. Ο κώδικας του χρήστη δεν έχει γνώση της τοποθέτησης των φυσικών σελίδα και «βλέπει» μόνο το συνεχόμενο εικονικό χώρο μνήμης μέσω του δείκτη `addr` κατά το αναμενόμενο.

άθικτος. Όλες οι κατά-NUMA εκδόσεις του πυρήνα SPMV που παρουσιάζουμε και συζητούμε σε αυτή την διατριβή υλοποιήθηκαν με την ελάχιστη προγραμματιστική προσπάθεια χρησιμοποιώντας την τεχνική αυτή.

Παρά το ότι τα δεδομένα του πίνακα και του διανύσματος εξόδου μπορούν να καταχωρηθούν στους φυσικούς κόμβους μνήμης με την χρήση του εκχωρητή μνήμης που μόλις παρουσιάσαμε, το διάνυσμα εισόδου θα πρέπει ιδανικά να αντιγραφεί αυτούσιο σε κάθε κόμβο μνήμης, καθώς το σχήμα προσβάσεων σε αυτή δεν είναι γνωστό εκ των προτέρων. Ωστόσο, κάτι τέτοιο δεν είναι πρακτικό υπό το πρίσμα μίας πραγματικής εφαρμογής, όπου το διάνυσμα εισόδου αλλάζει κατά την διάρκεια των επαναλήψεων του αλγορίθμου. Για τον λόγο αυτό, η τοποθέτησή του σε ένα μόνο κόμβο αποτελεί μία λογική επιλογή. Παραταύτα, όπως φαίνεται στο Σχήμα 3.13, αυτό το κοινό αντίγραφο του διανύσματος δεν αποτελεί σημείο συμφόρησης της επίδοσης. Η μέση επιβάρυνση κυμαίνεται στο 2.5% και μπορεί να φτάσει μέχρι 7% σε περιπτώσεις πιο ακανόνιστων πινάκων, π.χ., στους πίνακες `parabolic_fem`, `offshore` κ.λπ. Βάσει της προηγούμενης συζήτησης επί της επίδοσης του SPMV, η συμπεριφορά αυτή είναι αναμενόμενη, καθώς οι προσβάσεις στο διάνυσμα εισόδου δεν δημιουργούν σημαντική κίνηση προς την κύρια μνήμη για την πλειονότητα των αραιών πινάκων και, επομένως, η επίδραση του κοινού αντιγράφου στην επίδοση του πυρήνα SPMV στις αρχιτεκτονικές NUMA είναι αμελητέα.



Σχήμα 3.13: Η επίδραση του διαμοιρασμού του διανύσματος εισόδου στις αρχιτεκτονικές NUMA είναι αμελητέα, καθώς η επίδοση του πυρήνα SpMV περιορίζεται κυρίως από το εύρος ζώνης μνήμης και οι προσβάσεις στο διάνυσμα εισόδου δεν αποτελούν το κύριο πρόβλημα επίδοσης για την πλειονότητα των αραιών πινάκων.

3.4 Σχετικές εργασίες

Για αρκετά χρόνια το ακανόνιστο σχήμα προσβάσεων στο διάνυσμα εισόδου θεωρείτο ως το πιο σημαντικό πρόβλημα επίδοσης του πυρήνα SpMV. Οι Temam και Jalby [1992] πραγματοποιούν είναι ενδελεχή ανάλυση της συμπεριφοράς της κρυφής μνήμης λόγω των ακανόνιστων προσβάσεων στο διάνυσμα εισόδου και συζητούν διάφορες τεχνικές βελτιστοποίησης, όπως την αναδιάταξη του πίνακα και αποθήκευση σε μπλοκ. Υπό το ίδιο πρίσμα ο Toledo [1997] προτείνουν μία σειρά από τεχνικές βελτιστοποίησης για την βελτίωση της επίδοσης του συστήματος μνήμης στο πλαίσιο του πυρήνα SpMV, συμπεριλαμβανομένων τεχνικών αναδιάταξης, αποθήκευσης σε μπλοκ και προφόρτωσης των μη μηδενικών στοιχείων και των στηλών του πίνακα. Οι Geus και Röllin [2001] εστιάζουν στους συμμετρικούς πίνακες και προτείνουν τεχνικές σωλήνωσης στο λογισμικό (software pipelining) για την βελτιστοποίηση του παραλληλισμού σε επίπεδο εντολών του επεξεργαστή, τεχνικές αναδιάταξης του πίνακα για την αύξηση της επαναχρησιμοποίησης της κρυφής μνήμης και τεχνικές αποθήκευσης σε μπλοκ για την ελαχιστοποίηση των έμμεσων αναφορών στην μνήμη. Οι Im και Yelick [2001] εστιάζουν συγκεκριμένα στις μεθόδους αποθήκευσης σε μπλοκ σταθερού μεγέθους, ως ένα μέσο για την αύξηση της επαναχρησιμοποίησης των καταχωρητών του επεξεργαστή, και προτείνουν τεχνικές για την αυτόματη επιλογή του βέλτιστου μεγέθους μπλοκ. Στην ίδια κατεύθυνση, οι Viduc et al. [2002] μελετούν την αλληλεπίδραση του πυρήνα SpMV σε μπλοκ με την ιεραρχία της κρυφής μνήμης και εξετάζουν το όρια επίδοσης του πυρήνα, ενώ οι Pinar και Heath [1999] μελετούν τεχνικές αποθήκευσης με μπλοκ μεταβλητού μεγέθους, ως ένα εναλλακτικό τρόπο για

3. Η επίδοση του πυρήνα SpMV

την μείωση των έμμεσων αναφορών στην μνήμη. Η υπολογιστική επιβάρυνση των πολύ μικρών γραμμών τονίζεται από τους White και Sadayappan [1997], οι οποίοι την χαρακτηρίζουν ως κρισιμότερο πρόβλημα επίδοσης από το ακανόνιστο σχήμα προσβάσεων, καθώς περιορίζει σημαντικά τον παραλληλισμό σε επίπεδο εντολών του επεξεργαστή. Οι Mellor-Crummey και Garvin [2004] επιμένουν περαιτέρω σε αυτό το πρόβλημα και προτείνουν τεχνικές βελτιστοποίησης στην συμπύληση για την απόδοσή του.

Δεν είναι παρά αρκετά πρόσφατα που η συμφόρηση για εύρος ζώνης μνήμης άρχισε να θεωρείται πιο επισταμένα ως το κρισιμότερο πρόβλημα επίδοσης της εκτέλεσης του SpMV. Πράγματι, οι προηγούμενες εργασίες ακολουθούσαν μία μάλλον έμμεση προσέγγιση στο θέμα. Ένα τυπικό παράδειγμα είναι οι μέθοδοι αποθήκευσης σε μπλοκ, των οποίων η αυξημένη επίδοση απεδίδετο την μείωση των έμμεσων αναφορών στην κύρια μνήμη. Στην πραγματικότητα, όμως, αυτό δεν είναι παρά ένα παραπροϊόν του πραγματικού οφέλους, το οποίο είναι η σημαντική μείωση του μεγέθους της δομής δεικτοδότησης των στηλών. Οι Williams et al. [2007] και οι Goumas et al. [2008] είναι οι πρώτοι που τόνισαν τον σημαντικό ρόλο του εύρους ζώνης μνήμης στην εκτέλεση του πυρήνα SpMV στις σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών, ενώ οι Willcock και Lumsdaine [2006] ήταν οι πρώτοι που προσέγγισαν άμεσα το πρόβλημα αυτό προτείνοντας τεχνικές άμεσης συμπίεσης της αναπαράστασης του πίνακα.

3.5 Περίληψη

Στο κεφάλαιο αυτό παρουσιάσαμε μία εις βάθος ανάλυση της επίδοσης του πυρήνα SpMV στις σύγχρονες πολυπύρηνες αρχιτεκτονικές, αναγνωρίζοντας τα κύρια προβλήματα της επίδοσής του. Η συμπεριφορά του πυρήνα SpMV στις σύγχρονες αρχιτεκτονικές SMP και NUMA καθορίζεται από την ίδια την αλγοριθμική φύση του πυρήνα, ο οποίος παρουσιάζει ένα εξαιρετικά χαμηλό λόγο flops:byte. Η έλλειψη χρονικής και χωρικής επαναχρησιμοποίησης των δεδομένων απαιτεί από το υποσύστημα μνήμης να μπορεί να παρέχει δεδομένα σε συγκρίσιμη με τον επεξεργαστή ταχύτητα. Το μονοπάτι των δεδομένων από και προς την κύρια μνήμη υπό το πρίσμα του διαθέσιμου εύρους ζώνης μνήμης γίνεται πλέον το σημείο συμφόρησης της επίδοσης του πυρήνα, ειδικότερα στις πολυνηματικές εκτελέσεις, όπου πολλά νήματα διαμοιράζονται αυτό το μονοπάτι. Το πρόβλημα αυτό είναι πιο έντονο στις αρχιτεκτονικές SMP, όπου ολόκληρη η κίνηση της μνήμης και μεταξύ των επεξεργαστών διέρχεται μέσω ενός διαύλου με αρκετά χαμηλό εύρος ζώνης. Οι αρχιτεκτονικές NUMA, από την άλλη πλευρά, προσφέρουν σημαντικά υψηλότερο εύρος ζώνης μέσω του ενσωματωμένου τους ελεγκτή μνήμης, το οποίο μπορεί εύκολα να φιλοξενήσει

την κίνηση όλων των πυρήνων του επεξεργαστή. Το μειονέκτημα αυτού του τύπου αρχιτεκτονικών, ωστόσο, είναι ότι η κύρια μνήμη διαχωρίζεται σε πολλά φυσικούς κόμβους και η επίδοση του πυρήνα γίνεται πλέον αρκετά ευαίσθητη στην τοποθέτηση των δεδομένων στους κόμβους, καθώς οι απομακρυσμένες προσβάσεις τείνουν να κορέσουν το δίκτυο διασύνδεσης των επεξεργαστών. Η υπέρβαση αυτού του προβλήματος απαιτεί την σωστή τοποθέτηση των δεδομένων του αλγορίθμου στους κόμβους μνήμης, μία διαδικασία που επιτυγχάνεται εύκολα και διαφανώς ως προς τον κώδικα της εφαρμογής χρησιμοποιώντας των προτεινόμενο, βασισμένο στο Linux, κατά-NUMA παρεμβλητικό εκχωρητή μνήμης.

Η επίδοση του πυρήνα SpMV εξαρτάται σε σημαντικό βαθμό από την δομή των μη μηδενικών στοιχείων του πίνακα και την αραιότητά του. Μπορούμε να ξεχωρίσουμε δύο μεγάλες κατηγορίες αραιών πινάκων: (α') πίνακες με σχετικά κανονική διάταξη των μη μηδενικών στοιχείων τους και (β') πολύ αραιούς πίνακες με ακανόνιστη διάταξη των στοιχείων τους. Η πρώτη κατηγορία αποτελείται κυρίως από πίνακες που προκύπτουν από προβλήματα μερικών διαφορικών εξισώσεων (ΜΔΕ) με μία υποκείμενη 2Δ/3Δ γεωμετρία και αποτελούν την τυπική περίπτωση αραιών πινάκων που συναντώνται στην πράξη. Το κυριώτερο πρόβλημα επίδοσης αυτής της κατηγορίας είναι η συμφόρηση για εύρος ζώνης μνήμης και η επίδοσή τους είναι σχεδόν άμεσα συσχετιζόμενη με το μέγεθος της αναπαράστασης του πίνακα. Η δεύτερη κατηγορία αποτελείται από πίνακες, οι οποίοι δεν κορέννουν το εύρος ζώνης μνήμης του συστήματος, καθώς υποφέρουν από μία σειρά διαφορετικών προβλημάτων επίδοσης, συμπεριλαμβανομένων των ακανονίστων προσβάσεων στο διάνυσμα εισόδου, των επιβαρύνσεων βρόχου λόγω πολύ μικρών γραμμών και ανισορροπιών στην κατανομή του φόρτου εργασίας.

Κλείνουμε το κεφάλαιο αυτό με μία σημείωση όσον αφορά στις κατευθύνσεις βελτιστοποίησης του πυρήνα SpMV, όπως αυτές αποκαλύφθηκαν από την ανάλυση της επίδοσης που πραγματοποιήθηκε σε αυτό το κεφάλαιο. Ο κύριος στόχος μιας επιτυχούς βελτιστοποίησης του πυρήνα SpMV πρέπει να είναι η ελαχιστοποίηση του μεγέθους αναπαράστασης του πίνακα, καθώς η συμφόρηση για εύρος ζώνης μνήμης είναι το κύριο πρόβλημα επίδοσης στις σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών. Τα προβλήματα επίδοσης που σχετίζονται με ακανόνιστους πίνακες είναι δευτερευούσης σημασίας και μπορούν να αντιμετωπιστούν ανεξάρτητα από την ελαχιστοποίηση του μεγέθους αναπαράστασης του πίνακα. Για παράδειγμα, οι τεχνικές αναδιάταξης του πίνακα όχι μόνο επιλύουν επιτυχώς το θέμα της ανισορροπίας του φόρτου εργασίας των πινάκων με ακανόνιστη κατανομή των μη μηδενικών τους στοιχείων, αλλά καταφέρνουν να βελτιστοποιήσουν και το σχήμα προσβάσεων στο διάνυσμα εισόδου.

Δυνατότητες βελτιστοποίησης των μεθόδων μπλοκ

Τα σχήματα αποθήκευσης αραιών πινάκων σε μπλοκ εκμεταλλεύονται την κανονική διάταξη των μη μηδενικών στοιχείων ορισμένων πινάκων με σκοπό τον σχηματισμό μονοδιαστάτων ή δισδιάστατων πυκνών μπλοκ μέσω της ομαδοποίησης γειτονικών μη μηδενικών στοιχείων. Διατηρώντας μία στήλη ανά μπλοκ, τα σχήματα αποθήκευσης σε μπλοκ καταφέρνουν να μειώσουν δραστηρικά το μέγεθος αναπαράστασης του πίνακα και ελαφρύνουν την πίεση στο υποσύστημα μνήμης. Επιπλέον, η γνωστή και κανονική δομή των μπλοκ, ειδικότερα όσον αφορά στις μεθόδους σταθερού μεγέθους μπλοκ, επιτρέπει και την βελτιστοποίηση των υπολογισμών του πυρήνα SpMV, οι οποίοι πλέον είναι περισσότερο εκτεθειμένοι.

Στο κεφάλαιο αυτό εξετάζουμε τις δυνατότητες βελτιστοποίησης που μας παρέχει μία πληθώρα μεθόδων μπλοκ, όσον αφορά στις δυνατότητες συμπίεσης που προσφέρουν και στα υπολογιστικά χαρακτηριστικά τους. Βασιζόμενοι σε αυτή τη γνώση, προτείνουμε ένα απλό μοντέλο επίδοσης για την επιλογή του βελτίστου μπλοκ για το σχήμα BCSR, λαμβάνοντας υπόψη τόσο το μέρος των προσβάσεων στην μνήμη όσο και το υπολογιστικό μέρος του πυρήνα SpMV. Το προτεινόμενο μοντέλο είναι ικανό να διακρίνει όχι μόνο το καλύτερο μπλοκ, αλλά και την πιο αποδοτική υλοποίηση.

4.1 Η επίδραση της συμπίεσης

Στο Κεφάλαιο 2 παρουσιάσαμε με λεπτομέρεια τα πιο αντιπροσωπευτικά σχήματα αποθήκευσης σε μπλοκ για αραιούς πίνακες, συγκεκριμένα, σχήματα αποθήκευσης σε μπλοκ σταθερού ή μεταβλητού μεγέθους και διασπώμενα σχήματα. Το βασικό χαρακτηριστικό αυτών των σχημάτων είναι η συμπίεση τη δομής `colind` του CSR (Κεφάλαιο 2, Σχήμα 2.2) με την διατήρηση μίας μόνο στήλης ανά σχηματιζόμενο μπλοκ. Η βασική τους διαφορά έγκειται στον

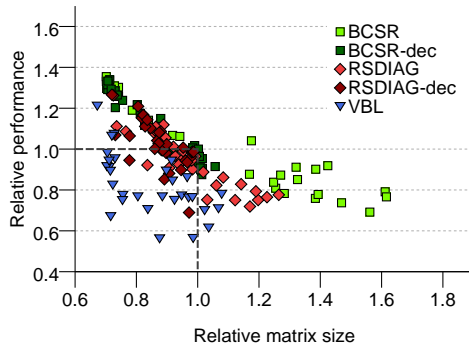
4. Δυνατότητες βελτιστοποίησης των μεθόδων μπλοκ

τρόπο σχηματισμού των μπλοκ και, όπως θα δούμε σε αυτό το κεφάλαιο, αυτό χαρακτηρίζει τις δυνατότητες συμπίεσης και τα υπολογιστικά χαρακτηριστικά τους. Οι μέθοδοι μπλοκ σταθερού μεγέθους σχηματίζουν μονοδιάστατα ή δισδιάστατα μπλοκ σταθερού μεγέθους και είτε χρησιμοποιούν επιπρόσθετα μηδενικά στοιχεία για τον σχηματισμό πλήρων μπλοκ είτε διασπούν τον αρχικό πίνακα σε υποπίνακες με πλήρη μπλοκ, αποθηκεύοντας τα εναπομείναντα στοιχεία σε μορφή CSR. Οι μέθοδοι μεταβλητού μεγέθους μπλοκ, από την άλλη πλευρά, χρησιμοποιούν μονοδιαστατα ή δισδιάστατα μπλοκ αυθαίρετου μεγέθους και χρησιμοποιούν επιπλέον δομές δεδομένων για να διατηρούν γνώση των σχηματιζόμενων μπλοκ.

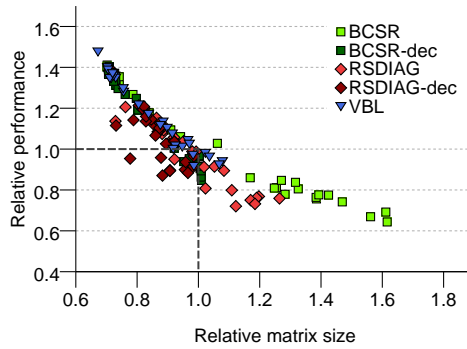
Το Σχήμα 4.1 παρουσιάζει την συσχέτιση του λόγου συμπίεσης και της βελτίωσης της επίδοσης για ένα σύνολο διαφορετικών σχημάτων μπλοκ, συμπεριλαμβανομένων σχημάτων μπλοκ σταθερού μεγέθους με χρήση επιπροσθέτων μηδενικών στοιχείων (BCSR, RSDIAG), διασπασμένων σχημάτων μπλοκ σταθερού μεγέθους (BCSR-dec, RSDIAG-dec) και σχημάτων μεταβλητού μεγέθους μπλοκ (VBL). Δείχνουμε τα αποτελέσματα για τα συστήματα Harpertown και Gainestown σε σειριακές και πολυνηματικές εκτελέσεις για τους 30 πίνακες τις σουίτας μας. Η σημειωμένη περιοχή στην κάτω αριστερή πλευρά των διαγραμμάτων υποδηλώνει πίνακες, για τους οποίους κάποιο σχήμα αποθήκευσης δεν κατάφερε να προσφέρει κάποια βελτίωση της επίδοσης, παρά την συμπίεση του μεγέθους αναπαράστασης. Η πρώτη σαφής παρατήρηση είναι η τάση προς υψηλότερες επιδόσεις καθώς μειώνεται το μέγεθος αναπαράστασης, η οποία γίνεται ακόμα πιο έντονη στις πολυνηματικές εκτελέσεις. Αυτό είναι τελείως αναμενόμενο βάσει της ποσοτικής ανάλυσης της επίδοσης του πυρήνα SpMV που έγινε στο προηγούμενο κεφάλαιο. Είναι αξιοσημείωτο ότι κανένα σχήμα αποθήκευσης με αναπαράσταση μεγαλύτερη του CSR δεν κατάφερε να επιτύχει καλύτερη επίδοση, ακόμα και στη σειριακή εκτέλεση, όπου η συμφόρηση για την πρόσβαση στη μνήμη δεν είναι τόσο έντονη. Το αντίστροφο, ωστόσο, είναι αληθές για την πλειονότητα των σχημάτων αποθήκευσης, αποκαλύπτοντας σημαντικές επιβαρύνσεις της επίδοσης, που δεν σχετίζονται με το μέγεθος της αναπαράστασης. Το πιο χαρακτηριστικό παράδειγμα είναι το σχήμα VBL, το οποίο στις σειριακές εκτελέσεις προκαλεί περισσότερο από 20% επιβάρυνση της επίδοσης, παρά την αρκετά συμπίεσμένη αναπαράστασή του. Αυτό οφείλεται στο αυξημένο υπολογιστικό κόστος της πρόσβασης των επιπλέον δομών δεδομένων, που απαιτούνται για την ανάκτηση του μεγέθους κάθε μπλοκ¹. Η κατάσταση αυτή αντιστρέφεται καθώς αυξάνεται το πλήθος

¹ Το σχήμα VBL πρέπει να ανακτήσει το μέγεθος του μπλοκ για κάθε ένα μπλοκ που συναντά. επομένως, όσο μικρότερο είναι το μπλοκ, τόσο μεγαλύτερη είναι η επιβάρυνση. Επιπλέον, συγκρινόμενο με το σχήμα BCSR, δεν είναι εύκολο για κάποιο συμπιιστή να εξάγει αρκετά βελτιστοποιημένο κώδικα για την διάσχιση του VBL, καθώς τα όρια των βρόχων (μέγεθος μπλοκ) είναι άγνωστα.

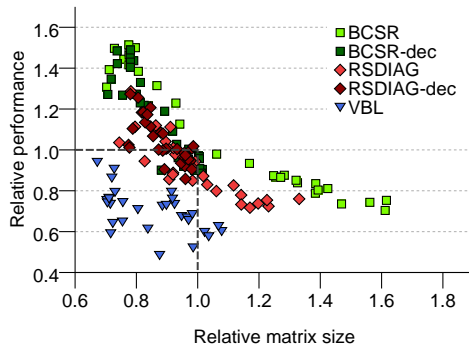
4.1. Η επίδραση της συμπίεσης



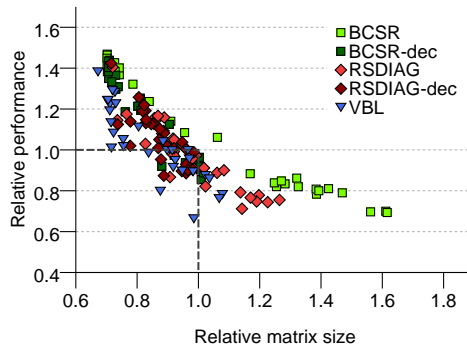
(i) Harpertown, σειριακή εκτέλεση.



(ii) Harpertown, ένας επεξεργαστής (τέσσερα νήματα).



(iii) Gainestown, σειριακή εκτέλεση



(iv) Gainestown, ένας επεξεργαστής (τέσσερα νήματα).

Σχήμα 4.1: Συσχέτιση του λόγου συμπίεσης και της βελτίωσης της επίδοσης ως προς το σχήμα CSR, που επιτυγχάνεται από διαφορετικά σχήματα αποθήκευσης σε μπλοκ σε ένα σύνολο 30 αραιών πινάκων. Τα αποτελέσματα για τις μεθόδους μπλοκ σταθερού μεγέθους αντιστοιχούν στο μπλοκ με την καλύτερη επίδοση.

των νημάτων και η συμφόρηση στην πρόσβαση στην μνήμη γίνεται φανερή: η σημαντική συμπίεση που επιτυγχάνει το σχήμα VBL ελαφρύνει τις απαιτήσεις για εύρος ζώνης μνήμης και προσφέρει μια δυναμική υψηλότερης επίδοσης.

Η περίπτωση διαγωνίων σχημάτων αποθήκευσης είναι ενδιαφέρουσα. Καταρχή, ως ήταν αναμενόμενο, η δυνατότητα συμπίεσής τους είναι χαμηλότερη από αυτή του BCSR, καθώς ανιχνεύουμε μόνο μονοδιάστατα μπλοκ. Ωστόσο, ακόμα και στις περιπτώσεις που πετυχαίνουν συγκρίσιμη συμπίεση, η επίδοσή τους είναι χαμηλότερη, ειδικά στην περίπτωση του συστήματος Gainestown. Η

4. Δυνατότητες βελτιστοποίησης των μεθόδων μπλοκ

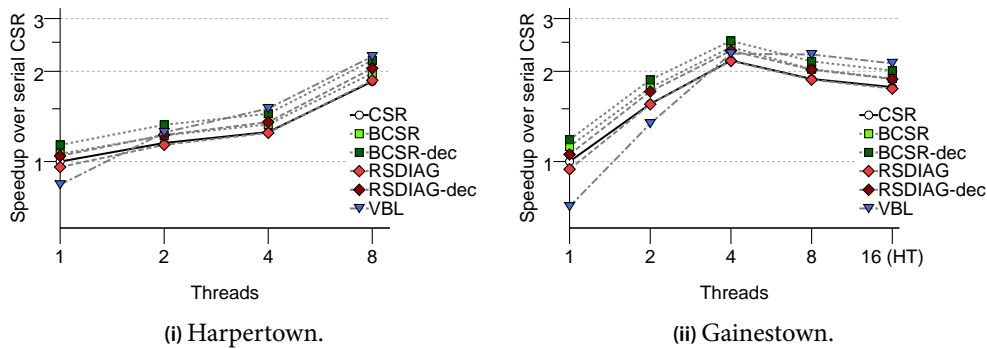
συμπεριφορά αυτή μπορεί να εξηγηθεί εάν εξετάσουμε από πιο κοντά το σχήμα προσβάσεων στην μνήμη των διαγωνίων μπλοκ: για κάθε μπλοκ μεγέθους b , ο πυρήνας θα πρέπει να διαβάσει $16b + 4$ bytes ($8b$ για τα μη μηδενικά στοιχεία, $8b$ για το διάνυσμα εισόδου και τέσσερα bytes για την στήλη του μπλοκ) και να γράψει πίσω στην μνήμη $8b$ bytes ανά γραμμή μπλοκ². Για ένα μπλοκ 4×1 , ωστόσο, ο πυρήνας του BCSR διαβάζει μόνο $8b + 12$ bytes ανά μπλοκ (χρειάζεται ένα μοναδικό στοιχείο του διανύσματος εισόδου) και γράφει πίσω πάλι $8b$ bytes ανά γραμμή μπλοκ. Η υπολογιστική ένταση των διαγωνίων μπλοκ, επομένως, είναι αρκετά χαμηλότερη, πράγμα που εξηγεί το έλλειμμα επίδοσης σε σχέση με τα μπλοκ του BCSR.

Τα διασπώμενα σχήματα επιλύουν το πρόβλημα της χρήσης επιπροσθέτων μηδενικών στοιχείων των σχημάτων αποθήκευσης σε μπλοκ σταθερού μεγέθους, με το μέγεθός τους να υπερβαίνει ελαφρώς αυτό του CSR στην χειρότερη περίπτωση, την στιγμή που η χρήση επιπροσθέτων μηδενικών στο σχήμα BCSR μπορεί να οδηγήσει μέχρι και σε 60% αύξηση του μεγέθους αναπαράστασης του πίνακα, ακόμα και για το μπλοκ με την καλύτερη επίδοση. Ωστόσο, τα διασπώμενα σχήματα θα πρέπει να πληρώσουν το κόστος των πολλαπλών πράξεων SpMV, πράγμα που οδηγεί σε ελαφρώς χαμηλότερη επίδοση σε σχέση με τις μεθόδους που κάνουν χρήση επιπροσθέτων στοιχείων, ακόμα και στις περιπτώσεις αντίστοιχης συμπίεσης. Στις περιπτώσεις που δεν επιτυγχάνεται συμπίεση, η επιβάρυνση αυτή γίνεται ακόμα πιο εμφανής και μπορεί να οδηγήσει σε περισσότερο από 10% μείωση της επίδοσης συγκριτικά με το σχήμα CSR.

Η υπεροχή του σχήματος BCSR στο θέμα των υπολογισμών σε σχέση με τα υπόλοιπα σχήματα αποθήκευσης σε μπλοκ φαίνεται καθαρά στην σειριακή εκτέλεση στο σύστημα Gainestown, όπου επιτυγχάνει την καλύτερη επίδοση για το ίδιο μέγεθος αναπαράστασης. Το πλεονέκτημα αυτό διατηρείται και στην μονο-επεξεργαστική εκτέλεση στο σύστημα Gainestown, αλλά εξαφανίζεται τελείως στο σύστημα Harpertown, λόγω της συμφόρησης στο σύστημα μνήμης. Ωστόσο, παρά την ιδιαίτερα υψηλή του επίδοση σε πίνακες που καταφέρνει σημαντική συμπίεση, οι δυνατότητες συμπίεσης του σχήματος BCSR είναι περιορισμένες· στους μισούς περίπου πίνακες της σουίτας αύξησε το μέγεθος της αναπαράστασης, επιδεινώνοντας την επίδοση του SpMV.

Η επίδραση της συμπίεσης, καθώς η συμφόρηση για εύρος ζώνης μνήμης αρχίζει να γίνεται ορατή, φαίνεται και στα διαγράμματα επιτάχυνσης του Σχήματος 4.2. Το σχήμα VBL πληρώνει το κόστος των επιπλέον πράξεων του

² Η τυπική υλοποίηση του πυρήνα SpMV για αποθήκευση σε μπλοκ εκτυλίσσει τους υπολογισμούς των μπλοκ και μεταθέτει τις εγγραφές στο διάνυσμα εξόδου στο τέλος της κάθε γραμμής μπλοκ, συσσωρεύοντας τα ενδιάμεσα αποτελέσματα σε καταχωρητές. Αυτό γίνεται για να ελαχιστοποιηθούν οι απαιτήσεις για εύρος ζώνης εγγραφής στη μνήμη, ένας πόρος που είναι περισσότερο περιορισμένος.



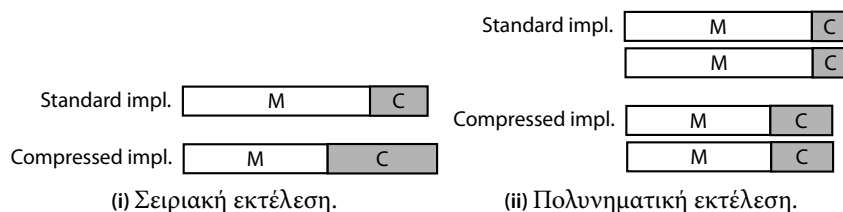
Σχήμα 4.2: Κλιμάκωση της επίδοσης των σχημάτων αποθήκευσης σε μπλοκ. Η κάμψη της επίδοσης στο σύστημα Gainestown από την οκτώ-νηματική εκτέλεση οφείλεται στο γεγονός ότι δεν χρησιμοποιείται κατά-NUMA υλοποίηση.

«αποσυμπίεσης» στις ολιγο-νηματικές εκτελέσεις, αλλά καταφέρνει να αποζημιωθεί επιτυχώς καθώς ο αριθμός των νημάτων αυξάνεται. Από την τετρανηματική εκτέλεση και ένθεν στο σύστημα Harpertown, το σχήμα VBL καταφέρνει να επιτύχει την καλύτερη επίδοση κατά μέσο όρο, ενώ στο σύστημα Gainestown (όχι κατά-NUMA υλοποιήσεις) γίνεται επικεφαλής από την οκτώ-νηματική εκτέλεση, όπου πλέον χρησιμοποιείται και ο δεύτερος επεξεργαστής και ο υπολογισμός του SpMV περιορίζεται από το δίκτυο διασύνδεσης των επεξεργαστών. Η συμπεριφορά αυτή, που απεικονίζεται γραφικά στο Σχήμα 4.3, είναι χαρακτηριστική των σχημάτων αποθήκευσης υψηλής συμπίεσης [Kourtis et al., 2008a], όπου το περιορισμένο εύρος ζώνης μνήμης στις πολυνηματικές εκτελέσεις μπορεί εύκολα να αποκρύψει το κόστος της αποσυμπίεσης. Τέλος, τα διασπώμενα σχήματα επιτυγχάνουν 5–8% καλύτερη επίδοση σε σχέση με τα αντίστοιχα σχήματα που κάνουν χρήση επιπροσθέτων μηδενικών στοιχείων, λόγω της πιο συμπαγούς τους αναπαράστασης.

4.2 Η επίδραση των υπολογισμών

Η προηγούμενη συζήτηση για τις δυνατότητες συμπίεσης των διαφόρων σχημάτων αποθήκευσης σε μπλοκ αποκάλυψε την σπουδαιότητα του υπολογιστικού τμήματος του πυρήνα SpMV στις εκτελέσεις όπου το διαθέσιμο εύρος ζώνης μνήμης δεν έχει ακόμη κορεστεί. Στην ενότητα αυτή θα εστιάσουμε περισσότερο στο υπολογιστικό μέρος του σχήματος BCSR, το οποίο είναι πιο φιλικό διακείμενο προς τους υπολογισμούς σχήμα αποθήκευσης, ώστε να αποκτήσουμε καλύτερη γνώση των ιδιαιτεροτήτων του πυρήνα SpMV.

4. Δυνατότητες βελτιστοποίησης των μεθόδων μπλοκ



Σχήμα 4.3: Η επίδραση της συμπίεσης. Τα σχήματα αποθήκευσης υψηλής συμπίεσης υποφέρουν στις σειριακές τους εκτελέσεις από το αυξημένο κόστος αποσυμπίεσης. Στις πολυνηματικές εκτελέσεις, ωστόσο, αυτό το κόστος αποκρύβεται από την συμφόρηση στο υποσύστημα μνήμης και η ωφέλεια στην επίδοση γίνεται φανερή.

4.2.1 Η τεχνική «vectorization» και το σχήμα των μπλοκ

Τα μπλοκ σταθερού μεγέθους του BCSR παρέχουν ένα σημαντικό υπολογιστικό πλεονέκτημα σε σχέση με τις μεθόδους μπλοκ μεταβλητού μεγέθους, μιας και επιτρέπουν την χρήση μιας σειράς βελτιστοποιήσεων. Όχι μόνο οι σύγχρονοι συμπιμιστές μπορούν να εξαγάγουν ιδιαίτερα βελτιστοποιημένο κώδικα για βρόχους με σταθερά όρια, αλλά και «χειροκίνητες» βελτιστοποιήσεις είναι επίσης δυνατές. Μία τέτοια βελτιστοποίηση είναι η τεχνική *vectorization*, μια τεχνική που αυξάνει τον ρυθμό διεκπεραίωσης των πράξεων κινητής υποδιαστολής εκτελώντας την ίδια εντολή σε πολλαπλά δεδομένα κινητής υποδιαστολής (*Single Instruction Multiple Data – SIMD*). Προερχόμενες από τους διανυσματικούς επεξεργαστές της δεκαετίας '70 [Russell, 1978], οι εντολές SIMD έχουν γίνει αναπόσπαστο κομμάτι του συνόλου εντολών των σύγχρονων μικροεπεξεργαστών εδώ και αρκετά χρόνια [Raman et al., 2000], ενώ πρόσφατα ήρθαν ξανά δυναμικά στο προσκήνιο με την έλευση των αρχιτεκτονικών GPU γενικού σκοπού [Lindholm et al., 2008]. Οι πειραματικές πλατφόρμες που χρησιμοποιούμε για τις αξιολογήσεις μας σε αυτή την διατριβή υποστηρίζουν ένα σύνολο εντολών SIMD (*Streaming SIMD Extensions – SSE*), οι οποίες δρουν σε ένα διαφορετικό αρχείο καταχωρητών, αποτελούμενο από 16 «ευρείς» καταχωρητές κινητής υποδιαστολής των 128-bit. Υπεύθυνη για την υλοποίηση αυτών των επεκτάσεων SIMD είναι μία ξεχωριστή αριθμητική μονάδα, συνδεδεμένη από μία εξειδικευμένη μονάδα μετατοπίσεων (*shuffle unit*), η μονάδα αυτή είναι υπεύθυνη για την μετατόπιση μεμονομένων στοιχείων του διανύσματος SIMD είτε μεταξύ διαφορετικών καταχωρητών SIMD είτε εντός του ίδιου καταχωρητή. Οι εντολές που εμπλέκουν την μονάδα μετατοπίσεων για την εκτέλεσή τους μπορεί να αντιμετωπίσουν αυξημένη καθυστέρηση και μικρό ρυθμό διεκπεραίωσης, λόγω των σχετικά σύνθετων κυκλωμάτων αυτής της μονάδας.

Οι Αλγόριθμοι 4.1 και 4.2 παρουσιάζουν τις υλοποιήσεις SIMD για τα μπλοκ 1×2 και 2×1 του BCSR, αντίστοιχα, υποδεικνύοντας τα σημεία που εμπλέκεται η μονάδα μετατοπίσεων. Ο υπολογισμός των μπλοκ 1×2 ξεκινά με την φόρτωση των δύο τιμών του διανύσματος εισόδου και των αντιστοίχων μη μηδενικών στοιχείων σε δύο καταχωρητές SIMD. Στο σημείο αυτό, η τυπική υλοποίηση θα προσέθετα τα δύο στοιχεία του καταχωρητή y_0 χρησιμοποιώντας μία εντολή *οριζόντιας πρόσθεσης* (*horizontal add*). Ωστόσο, η εντολή αυτή εμπλέκει την μονάδα μετατοπίσεων και μπορεί να είναι ιδιαίτερα ακριβή, ειδικότερα σε παλαιότερες αρχιτεκτονικές· για τον λόγο αυτό, καθυστερούμε την εκτέλεσή της μέχρι το τέλος της γραμμής μπλοκ, οπότε και το τελικό αποτέλεσμα πρέπει να γραφεί στο διάνυσμα εξόδου. Ο υπολογισμός των μπλοκ 2×1 ξεκινά με την φόρτωση των δύο μη μηδενικών στοιχείων στο καταχωρητή SIMD y_0 , ενώ το μοναδικό στοιχείο του διανύσματος εισόδου αποθηκεύεται στα 64 χαμηλότερα bit του καταχωρητή y_1 . Στη συνέχεια, η τιμή αυτή αντιγράφεται (χρησιμοποιώντας την μονάδα μετατόπισης) στα ανώτερα 64 bit, ώστε ο υπολογισμός του SpMV να προχωρήσει με λογική «vectorization». Οι υλοποιήσεις για αυτούς τους δύο τύπους μπλοκ μπορούν να επεκταθούν εύκολα για μεγαλύτερα μονοδιάστατα και δισδιάστατα μπλοκ.

Φόρτωση διανυσμάτων SIMD Υπάρχουν δύο τύποι εντολών για την φόρτωση πλήρων διανυσμάτων SIMD: *ευθυγραμμισμένες* και *μη ευθυγραμμισμένες φορτώσεις*. Μία ευθυγραμμισμένη φόρτωση SIMD είναι το ίδιο γρήγορη με μία κανονική φόρτωση, αλλά απαιτεί η διεύθυνση φόρτωσης να είναι ευθυγραμμισμένη στα 16 byte. Ο περιορισμός αυτός αίρεται για τις μη ευθυγραμμισμένες φορτώσεις SIMD, αλλά τότε χρησιμοποιείται η μονάδα μετατοπίσεων για την σωστή ευθυγράμμιση των στοιχείων του διανύσματος μέσα στον καταχωρητή SIMD. Ως αποτέλεσμα, οι μη ευθυγραμμισμένες φορτώσεις αντιμετωπίζουν μία επιβάρυνση επίδοσης. Λόγω των αυστηρών απαιτήσεων ευθυγράμμισης, οι ευθυγραμμισμένες φορτώσεις SIMD δεν μπορούν να χρησιμοποιηθούν για μπλοκ με διαστάσεις περιττού αριθμού, π.χ., για τα μπλοκ 3×1 , 5×1 κ.ο.κ., καθώς τα επιμέρους μπλοκ δεν μπορούν να ευθυγραμμιστούν σωστά. Αντιθέτως, η σωστή ευθυγράμμιση μπορεί να επιτευχθεί εύκολα για τα μπλοκ με διαστάσεις ζυγού αριθμού.

Η τεχνική «vectorization» για τα μπλοκ του BCSR εισήγαγε ένα σύνολο νέων παραγόντων που επηρεάζουν την επίδοση του SpMV. Τα Σχήματα 4.5 και 4.4 παρουσιάζουν την επίδραση της τεχνικής αυτής στα συστήματα Harpertown και Gainestown, αντίστοιχα, για την σειριακή και την πολυνηματική εκτέλεση. Παρουσιάζουμε μόνο τα αποτελέσματα για του 15 συμμετρικούς πίνακες της σουίτας μας έτσι, ώστε τα συμμετρικά μπλοκ (π.χ., τα 1×2 και 2×1) να οδηγούν στο ίδιο μέγεθος αναπαράστασης. Η επίδραση της τεχνικής

4. Δυνατότητες βελτιστοποίησης των μεθόδων μπλοκ

```

1: procedure MATVECBCSR1X2( $A::in$ ,  $x::in$ ,  $y::out$ )
    $A$ : matrix in BCSR format
    $x$ : input vector
    $y$ : output vector
2:   for  $i \leftarrow 0$  to  $N$  do
3:      $y_0 \leftarrow 0$ 
4:     for  $j \leftarrow browptr[i]$  to  $browptr[i + 1]$  step by 2 do
5:        $j_b \leftarrow \frac{j}{2}$ 
6:        $x_c \leftarrow bcolind[j_b]$ 
7:        $x_0 \leftarrow x[x_c]$ 
8:        $v_0 \leftarrow bvalues[j]$ 
9:        $y_0 \leftarrow y_0 + v_0 \cdot x_0$ 
10:    end for
11:     $y_0 \leftarrow y_0 \oplus y_0$  ▷ Shuffle unit
12:     $y[i] \leftarrow lower_{64}(y_0)$  ▷ Shuffle unit
13:  end for

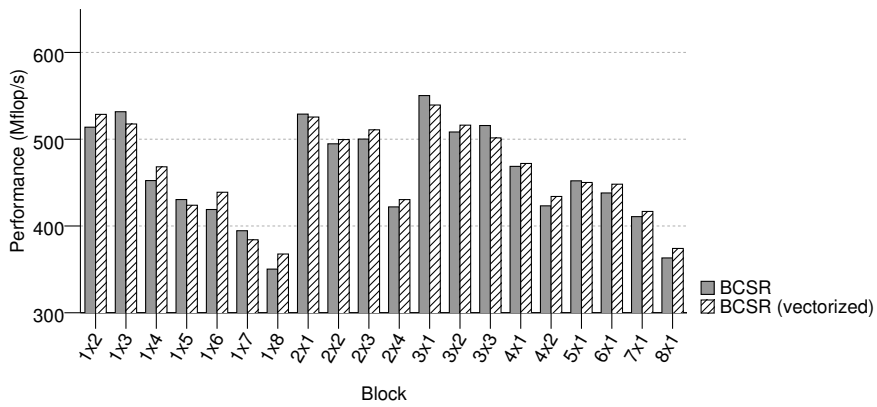
```

Αλγόριθμος 4.1: Υλοποίηση SIMD του πυρήνα BCSR για μπλοκ 1×2 . Οι διανυσματικοί καταχωρητές σημειώνονται με έντονη γραμματοσειρά· ο τελεστής \oplus πραγματοποιεί την πρόσθεση των επιμέρους στοιχείων του καταχωρητή (οριζόντια πρόσθεση), ενώ ο τελεστής $lower_{64}$ επιστρέφει τα 64 κατώτερα bit ενός διανυσματικού καταχωρητή.

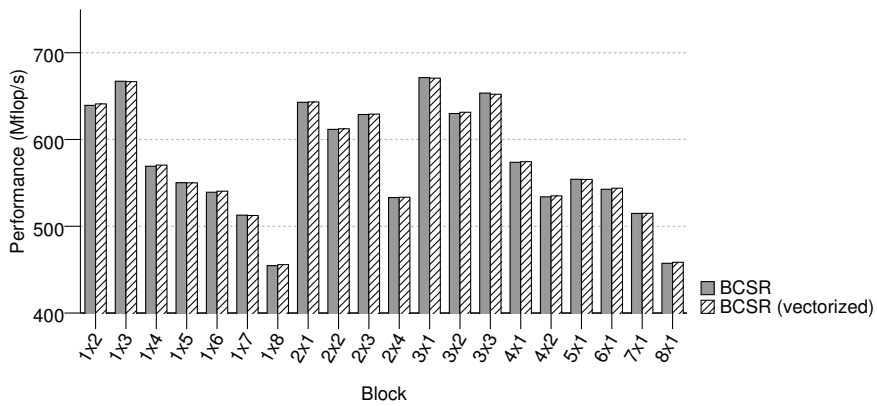
«vectorization» στο Harperton είναι αμυδρά, φθάνοντας μία μέγιστη βελτίωση της τάξης του 5% για το μπλοκ 1×8 . Το χαμηλό εύρος ζώνης μνήμης των αρχιτεκτονικών SMP δεν αφήνει πολλά περιθώρια για υπολογιστικές βελτιστοποιήσεις, όπως είναι η τεχνική «vectorization». Επιπλέον, οι υλοποιήσεις SIMD για οποιοδήποτε μπλοκ κάνουν χρήση της μονάδας μετατοπίσεων και δεν μπορούν να αποκρύψουν την συνακόλουθη επιβάρυνση, μιας και τα μπλοκ είναι αρκετά μικρά και η μονάδα αυτή δεν είναι ιδιαίτερα βελτιστοποιημένη σε αυτή την αρχιτεκτονική. Η επιβάρυνση αυτή γίνεται ακόμα πιο εμφανής στις περιπτώσεις που χρησιμοποιούνται μη ευθυγραμμισμένες φορτώσεις, όπου η τεχνική «vectorization» υποβαθμίζει την επίδοση³. Στο σύστημα Gainestown, από την άλλη πλευρά, το όφελος της τεχνικής «vectorization» είναι σημαντικό, επιτυγχάνοντας βελτίωση 58% κατά μέσο όρο για το μπλοκ 1×8 . Η συμπεριφορά αυτή οφείλεται κυρίως στο επαρκές εύρος ζώνης μνήμης και στην βελτιστοποιημένη μονάδα μετατοπίσεων της μικροαρχιτεκτονικής Nehalem [Int, 2009; Fog, 2012].

³ Η επίδοση των μπλοκ 3×1 , 5×1 κ.λπ. δεν είναι υποβαθμισμένη, διότι η μη ευθυγραμμισμένη εγγραφή που πραγματοποιούν δεν βρίσκεται το κρίσιμο μονοπάτι.

4.2. Η επίδραση των υπολογισμών



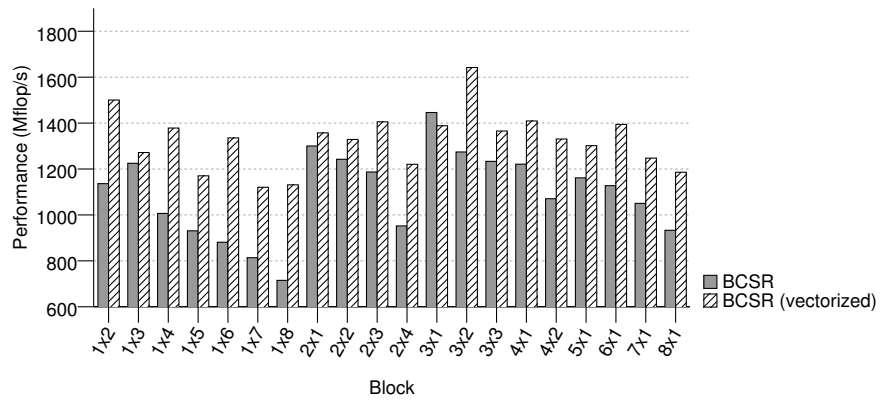
(i) Σειριακή εκτέλεση.



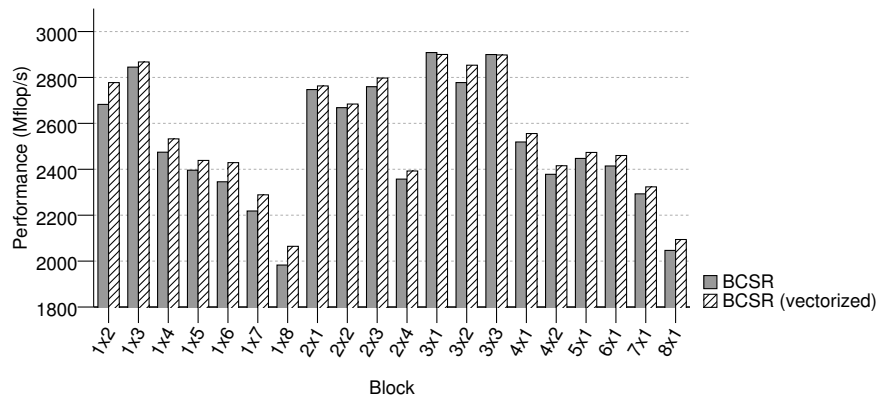
(ii) Πολυνηματική εκτέλεση (ένας επεξεργαστής, τέσσερα νήματα).

Σχήμα 4.4: Η επίδραση της τεχνικής «vectorization» στο σχήμα BCSR στην πλατφόρμα SMP Harpertown.

4. Δυνατότητες βελτιστοποίησης των μεθόδων μπλοκ



(i) Σειριακή εκτέλεση.



(ii) Πολυνηματική εκτέλεση (ένας επεξεργαστής, τέσσερα νήματα).

Σχήμα 4.5: Η επίδραση της τεχνικής «vectorization» στο σχήμα BCSR στην πλατφόρμα NUMA Gainestown.

```

1: procedure MATVECBCSR2X1( $A::in, x::in, y::out$ )
    $A$ : matrix in BCSR format
    $x$ : input vector
    $y$ : output vector
2:    $i_r \leftarrow 0$ 
3:   for  $i \leftarrow 0$  to  $N$  step by 2 do
4:      $y_0 \leftarrow 0$ 
5:     for  $j \leftarrow browptr[i_r]$  to  $browptr[i_r + 1]$  step by 2 do
6:        $j_b \leftarrow \frac{j}{2}$ 
7:        $x_c \leftarrow bcolind[j_b]$ 
8:        $lower_{64}(x_0) \leftarrow x[x_c]$  ▷ Shuffle unit
9:        $upper_{64}(x_0) \leftarrow lower_{64}(x_0)$  ▷ Shuffle unit
10:       $v_0 \leftarrow bvalues[j]$ 
11:       $y_0 \leftarrow y_0 + v_0 \cdot x_0$ 
12:    end for
13:     $y[i] \leftarrow y_0$ 
14:     $i_r \leftarrow i_r + 1$ 
15:  end for

```

Αλγόριθμος 4.2: Υλοποίηση SIMD του πυρήνα BCSR για μπλοκ 2×1 . Οι διανυσματικοί καταχωρητές σημειώνονται με έντονη γραμματοσειρά· οι τελεστές $lower_{64}$ και $upper_{64}$ επιστρέφουν τα 64 κατώτερα ή ανώτερα bits ενός διανυσματικού καταχωρητή, αντίστοιχα.

Μία πιο προσεκτική ματιά στα αποτελέσματα από το σύστημα Gainestown (Σχήμα 4.5) αποκαλύπτει κάποια σημαντικά χαρακτηριστικά των μπλοκ του BCSR. Καταρχήν, τα μπλοκ με κατακόρυφο προσανατολισμό (στην τυπική έκδοση) επιτυγχάνουν υψηλότερη επίδοση σε σχέση με αντίστοιχα οριζόντια, παρόλο που αμφότερα οδηγούν στο ίδιο μέγεθος αναπαράστασης. Τα μπλοκ με οριζόντιο προσανατολισμό θυμίζουν την περίπτωση των διαγωνίων μπλοκ (Ενότητα 4.1), καθώς για κάθε μπλοκ ο πυρήνας πρέπει να φορτώσει μία σειρά από στοιχεία του διανύσματος εισόδου, μειώνοντας την αριθμητική ένταση. Οι διαφορές επίδοσης αμβλύνονται στην έκδοση με την τεχνική «vectorization» λόγω της υψηλής επιβάρυνσης των πράξεων της μονάδας μετατοπίσεων, η οποία εμπλέκεται συχνότερα στα μπλοκ με κατακόρυφη διάταξη. Η δεύτερη παρατήρηση είναι ότι η επίδοση των εκδόσεων SIMD εξαρτάται λιγότερο από το μέγεθος αναπαράστασης του πίνακα, λόγω των διαφόρων επιβαρύνσεων που εισάγονται από τις αυστηρές απαιτήσεις των υλοποιήσεων SIMD. Για παράδειγμα, τα μονοδιάστατα μπλοκ με διαστάσεις περιττού αριθμού παρουσιάζουν χειρότερη επίδοση από τα αντίστοιχα διαστάσεων ζυγού αριθμού, παρά

4. Δυνατότητες βελτιστοποίησης των μεθόδων μπλοκ

την μικρότερη αναπαράστασή τους· ομοίως, η επίδοση των μπλοκ 3×2 είναι 17% υψηλότερη από αυτή των 2×3 . Η επιλογή, επομένως, του βελτίστου μπλοκ για το BCSR δεν είναι μία απλή διαδικασία μιας θα πρέπει να συνυπολογιστεί και το υπολογιστικό μέρος του πυρήνα.

Κλείνοντας την συζήτηση για την τεχνική «vectorization» στα μπλοκ του BCSR, θα πρέπει να επισημάνουμε την περίπτωση των πολυνηματικών εκτελέσεων τόσο στο σύστημα Harpertown όσο και στο Gainestown. Ο ανταγωνισμός για χρήση του διαθέσιμου εύρους ζώνης μνήμης δεν αφήνει σχεδόν καθόλου χώρο για υπολογιστικές βελτιστοποιήσεις, όπως είναι η τεχνική «vectorization». Στο σύστημα Harpetown, το οποίο υποφέρει περισσότερο από το πρόβλημα τη συμφόρησης στη μνήμη, το όφελος της τεχνικής «vectorization» είναι μηδαμινό, ενώ στο σύστημα Gainestown πέφτει στο 5%. Σε αυτές τις περιπτώσεις το κριτήριο του μεγέθους αναπαράστασης του πίνακα φαίνεται ως μάλλον ασφαλές για την πρόβλεψη της επίδοσης του πυρήνα SpMV.

4.3 Πρόβλεψη του βελτίστου μεγέθους μπλοκ

Η συζήτηση για τα χαρακτηριστικά επίδοσης των μπλοκ του BCSR ήρε την ανάγκη για τον δημιουργία ενός μοντέλου επίδοσης του πυρήνα SpMV, το οποίο θα μπορεί να μας οδηγήσει στην επιλογή της σωστής βελτιστοποίησης. Το μοντέλο αυτό θα πρέπει να λαμβάνει υπόψη του και το υπολογιστικό μέρος του πυρήνα, καθώς αυτό μπορεί να επηρεάσει σε σημαντικό βαθμό την συνολική επίδοση. Στην ενότητα αυτή, προτείνουμε και αξιολογούμε δύο απλά μοντέλα επίδοσης για το σχήμα BCSR συγκεκριμένα, τα μοντέλα MEMCOMP και OVERLAP. Τα μοντέλα αυτά λαμβάνουν υπ' όψη και υπολογιστικό μέρος του πυρήνα και «αποφασίζουν» για το βέλτιστο μπλοκ και την βέλτιστη υλοποίηση (τυπική ή με την τεχνική «vectorization»). Εστιάζουμε συγκεκριμένα στην σειριακή επίδοση, καθώς εκεί το υπολογιστικό μέρος του πυρήνα είναι περισσότερο σημαντικό, καθιστώντας την επιλογή της βέλτιστης λύσης δυσκολότερη.

4.3.1 Το μοντέλο Mem

Το μοντέλο MEM είναι το πιο άμεσο και εύκολο μοντέλο επίδοσης για τον πυρήνα SpMV. Προτεταγμένο από τους Gropp et al. [1999], το μοντέλο αυτό βασίζεται αποκλειστικά στο σύνολο εργασίας του αλγορίμου (μέγεθος αναπαράστασης του πίνακα μεγέθη διανυσμάτων εισόδου/εξόδου) και, επομένως, δεν εστιάζει αυστηρά σε κάποιο συγκεκριμένο σχήμα αποθήκευσης. Βασισμένα στην φύση του πυρήνα SpMV, το μοντέλο αυτό θεωρεί ότι το μοναδικό πρόβλημα επίδοσης είναι το διαθέσιμο εύρος ζώνης μνήμης του υποκειμένου συστήματος. Δεδομένου, επομένως, ενός συστήματος με εύρος ζώνης μνή-

μης B και ενός πίνακα με σύνολο εργασίας S , ο χρόνος εκτέλεσης του πυρήνα SpMV σύμφωνα με το μοντέλο MEM θα είναι

$$t_{\text{MEM}} = \frac{S}{B} \quad (4.1)$$

Το διαθέσιμο εύρος ζώνης μνήμης ενός συστήματος μπορεί να ανακτηθεί μέσω μετροπρογραμμάτων, όπως είναι το STREAM [McCalpin, 1995], το BenchIT [Juckeland et al., 2004] κ.ά.

4.3.2 Το μοντέλο Sparsity

Οι Im et al. [2004] και οι Buttari et al. [2007] εστιάζουν συγκεκριμένα στο σχήμα BCSR και προτείνουν ένα μοντέλο επίδοσης για την επιλογή του καλύτερου μπλοκ. Το μοντέλο αυτό, που χρησιμοποιείται και από την βιβλιοθήκη βελτιστοποίησης αραιών πινάκων OSKI [Vuduc et al., 2005], λαμβάνει υπόψη του τόσο το μέρος προσβάσεων στη μνήμη όσο και το υπολογιστικό, αλλά με κάπως έμμεσο τρόπο. Σύμφωνα με το μοντέλο SPARSITY, μία εκτίμηση του χρόνου εκτέλεσης για ένα πίνακα A μπορεί να υπολογιστεί ως εξής:

$$t_A \propto \frac{\text{fill}_A(r, c)}{\text{perf}_A(r, c)} \quad (4.2)$$

Η παράμετρος $\text{fill}_A(r, c)$ είναι ο λόγος ένταξης μηδενικών στοιχείων (fill-in ratio) για το μπλοκ $r \times c$ και υποδηλώνει το μέγεθος της προσθήκης μηδενικών στοιχείων που απαιτείται για την κατασκευή πλήρων μπλοκ. Ο λόγος ένταξης αποτελεί μία ένδειξη τόσο της υπολογιστικής επιβάρυνσης των επιπροσθέτων μηδενικών στοιχείων (περιττές πράξεις κινητής υποδιαστολής) όσο και του τελικού μεγέθους αναπαράστασης του πίνακα. Η παράμετρος $\text{perf}_A(r, c)$ είναι η πραγματική επίδοση του πυρήνα SpMV για το μπλοκ $r \times c$. Η παράμετρος αυτή εξαρτάται από την μικροαρχιτεκτονική και «καταγράφει» την διακύμανση της επίδοσης των διαφορετικών μπλοκ που οδηγούν στον ίδιο λόγο ένταξης μηδενικών στοιχείων, δηλαδή, στο ίδιο μέγεθος αναπαράστασης. Καθώς ο ακριβής υπολογισμός της παραμέτρου $\text{perf}_A(r, c)$ για ένα οποιοδήποτε πίνακα A δεν είναι δυνατός, οι τιμές της προέρχονται από την ανάλυση της εκτέλεσης του SpMV σε ένα μεγάλο πυκνό πίνακα, του οποίου το μέγεθος ξεπερνάει το μέγεθος της υψηλότερου επιπέδου κρυφής μνήμης για το υπό εξέταση σύστημα. Επομένως, η εκτίμηση του χρόνου εκτέλεση από το μοντέλο SPARSITY για ένα μπλοκ $r \times c$ υπολογίζεται πλέον ως εξής:

$$t_{\text{SPARSITY}}^{r,c} = \text{fill}_A(r, c) t_{\text{dense}}^{r,c} \quad (4.3)$$

Ο λόγος ένταξης μηδενικών στοιχείων είναι ο λόγος των στοιχείων του BCSR, συμπεριλαμβανομένων και των επιπροσθέτων μηδενικών, επί των μη μηδενικών στοιχείων του αρχικού πίνακα. Ο υπολογισμός του λόγου ένταξης για

κάθε σχετικό μπλοκ μπορεί να είναι αρκετά χρονοβόρος⁴. Για τον λόγο αυτό, στην πράξη, χρησιμοποιείται μία εκτίμηση του λόγου ένταξης, η οποία υπολογίζεται για ένα τμήμα του αρχικού πίνακα, το οποίο λαμβάνεται με δειγματισμό.

Το μοντέλο SPARSITY μπορεί να εφαρμοστεί και για το σχήμα CSR, όπου $\text{fill}_A(r, c) = 1$ και $t_{dense}^{r,c}$ είναι ο χρόνος εκτέλεσης του SpMV για ένα πυκνό πίνακα αποθηκευμένο στο σχήμα CSR.

4.3.3 Το μοντέλο MemComp

Το μοντέλο MEMCOMP είναι μία γενίκευση του μοντέλου MEM, η οποία λαμβάνει επίσης υπόψη της και το υπολογιστικό μέρος του πυρήνα. Θεωρεί τον υπολογισμό του SpMV αποτελούμενο από δύο φάσεις, το μέρος των προσβάσεων στη μνήμη και το υπολογιστικό. Το μέρος των προσβάσεων στη μνήμη υπολογίζεται από το μοντέλο MEM (σχέση (4.1)), ενώ το υπολογιστικό μέρος υπολογίζεται ξεχωριστά χρησιμοποιώντας μία εκτίμηση του χρόνου εκτέλεσης ενός μοναδικού μπλοκ. Θεωρώντας ένα αραιό πίνακα με μέγεθος αναπαράστασης S στο σχήμα BCSR, χρησιμοποιώντας συνολικά N_b $r \times c$ μπλοκ, ο χρόνος εκτέλεσης του πυρήνα SpMV, σύμφωνα με το μοντέλο MEMCOMP, υπολογίζεται ως εξής:

$$t_{\text{MEMCOMP}}^{r,c} = \frac{S}{B} + N_b^{r,c} \cdot t_b^{r,c} \quad (4.4)$$

Ο πρώτος όρος της σχέσης αντιστοιχεί στο μέρος των προσβάσεων στην μνήμη και B είναι το διαθέσιμο εύρος ζώνης μνήμης του υποκειμένου συστήματος. Ο δεύτερος όρος είναι το υπολογιστικό τμήμα και αντιστοιχεί στην εκτέλεση όλων των N_b μπλοκ. Ο χρόνος εκτέλεσης t_b ενός μοναδικού μπλοκ λαμβάνεται από την ανάλυση της εκτέλεσης ενός πολύ μικρού πυκνού πίνακα, που χωρά στην κρυφή μνήμη L1 του συστήματος, για κάθε ένα από όλα τα υπό θεώρηση $r \times c$ μπλοκ.

Το μοντέλο MEMCOMP μπορεί να εφαρμοστεί και στα διασπώμενα σχήματα, όπου η σχέση (4.4) υπολογίζεται για κάθε ένα υποπίνακα. Η περίπτωση του CSR μεταχειρίζεται ως ένα εκφυλισμένο BCSR με μπλοκ 1×1 . επομένως, $N_b^{r,c} = \text{NNZ}$ και $t_b^{r,c}$ είναι ο χρόνος εκτέλεσης του προς ανάλυση μικρού πυκνού πίνακα σε μορφή CSR.

4.3.4 Το μοντέλο Overlap

Το μοντέλο MEMCOMP ασχολείται με το μέρος των προσβάσεων στην μνήμη και το υπολογιστικό μέρος του πυρήνα SpMV ξεχωριστά, ως να ήταν δύο

⁴ Το κόστος της μετατροπής ενός αραιού πίνακα σε μορφή BCSR είναι μία τάξη μεγέθους υψηλότερο από μία πράξη SpMV.

εντελώς διαφορετικές φάσεις. Ωστόσο, σύμφωνα με την ανάλυση που παρουσιάστηκε στο Κεφάλαιο 3, οι υπολογισμοί του πυρήνα SpMV δεν είναι αρκετά εκτεθειμένοι, λόγω της μεγάλης έντασης προσβάσεων στην μνήμη, η οποία τείνει να καταναλώνει πλήρως το διαθέσιμο εύρος ζώνης μνήμης. Το μοντέλο OVERLAP υποθέτει μία αλληλοεπικάλυψη των φάσεων της πρόσβασης στη κύρια μνήμη και της υπολογιστικής και μόνο ένα τμήμα του υπολογιστικού μέρους γίνεται τελικά εμφανές στο συνολικό χρόνο εκτέλεση του SpMV. Η συμπεριφορά αυτή είναι δυνατή στους σύγχρονους μικροεπεξεργαστές, ακόμα και στην περίπτωση της σειριακής εκτέλεσης, χάρη στο μηχανισμό της *προφόρτωσης δεδομένων*. Ο μηχανισμός της προφόρτωσης δεδομένων προφορτώνει στη κρυφή μνήμη του επεξεργαστή δεδομένα, τα οποία είναι πιθανό να χρησιμοποιηθούν από σύντομα στο μέλλον, βασιζόμενος στο τρέχον σχήμα προσβάσεων στη μνήμη. Ο μηχανισμός αυτός όχι μόνο αποκρύβει την καθυστέρηση της πρόσβασης στη κύρια μνήμη, αλλά επιτρέπει σε εφαρμογές χωρίς χρονική τοπικότητα των αναφορών, όπως ο πυρήνας SpMV, να χρησιμοποιούν πλήρως το διαθέσιμο εύρος ζώνης μνήμης του συστήματος [Goumas et al., 2009].

Το μοντέλο OVERLAP προσαρμόζει τον όρο του υπολογιστικού μέρους του μοντέλου MEMCOMP, πολλαπλασιάζοντάς τον με ένα συντελεστή, ονόματι *Συντελεστής Μη-Επικάλυψης (Non-Overlapping Factor – NOF)*, οποίος υποδηλώνει το ποσοστό των υπολογισμών του SpMV που δεν εκτίθενται στο συνολικό χρόνο εκτέλεσης. Ο συντελεστής NOF ορίζεται για κάθε μπλοκ $r \times c$ ως εξής:

$$NOF^{r,c} = \frac{t_{\text{real}}^{r,c} - t_{\text{MEM}}^{r,c}}{N_b^{r,c} \cdot t_b^{r,c}} \quad (4.5)$$

Ο συντελεστής αυτός προκύπτει από την ανάλυση της εκτέλεσης ενός μεγάλου πυκνού πίνακα, του οποίου το μέγεθος ξεπερνά το μέγεθος της υψηλότερου επιπέδου κρυφής μνήμης του υπό εξέταση συστήματος. Μπορεί να θεωρηθεί ως ένας συντελεστής διόρθωσης του υπολογιστικού μέρους του μοντέλου MEMCOMP, καθώς στην πραγματικότητα συγκρίνει τον πραγματικό χρόνο εκτέλεσης των υπολογισμών (αριθμητής) με τον θεωρητικό χρόνο εκτέλεσης του μοντέλου MEMCOMP (παρονομαστής). Δεδομένου του συντελεστή NOF, ο χρόνος εκτέλεσης του SpMV, σύμφωνα με το μοντέλο OVERLAP, υπολογίζεται από την ακόλουθη σχέση:

$$t_{\text{OVERLAP}}^{r,c} = \frac{S}{B} + NOF^{r,c} \cdot N_b^{r,c} \cdot t_b^{r,c} \quad (4.6)$$

Όπως το MEMCOMP, έτσι και το μοντέλο OVERLAP μπορεί να εφαρμοστεί τόσο σε διασπώμενα σχήματα αποθήκευσης όσο και στο CSR, θεωρώντας το ως την εκφυλισμένη μορφή του BCSR με μπλοκ 1×1 .

4.3.5 Αξιολόγηση της ακρίβειας των μοντέλων

Ο κύριος στόχος των μοντέλων επίδοσης που παρουσιάστηκαν στο κεφάλαιο αυτό είναι η επιλογή του βελτίστου μπλοκ για το σχήμα BCSR. Ωστόσο, με την εξαίρεση του μοντέλου SPARSITY, το υπόλοιπα τρία μοντέλα, συγκεκριμένα το MEM και τα προτεινόμενα MEMCOMP και OVERLAP, ακολουθούν μία πιο σαφή προσέγγιση για την επιπλέον εκτίμηση του χρόνου εκτέλεσης του πυρήνα. Θα μπορούσαμε, επομένως, να διακρίνουμε δύο μετρικές για την αξιολόγηση της ακρίβειας πρόβλεψης των θεωρουμένων μοντέλων:

Ακρίβεια επιλογής: Η μετρική αυτή καθορίζει πόσο ακριβές είναι το μοντέλο επίδοσης στην επιλογή του βελτίστου μπλοκ για ένα συγκεκριμένο πίνακα. Ποσοτικά, η ποιότητα της πρόβλεψης μπορεί να αξιολογηθεί από το πόσο κοντά στην επίδοση του βελτίστου μπλοκ είναι η επίδοση του προτεινομένου από το μοντέλο μπλοκ.

Ακρίβεια πρόβλεψης χρόνου εκτέλεσης: Η μετρική αυτή καθορίζει την ακρίβεια του θεωρουμένου μοντέλου στην πρόβλεψη του απολύτου χρόνου εκτέλεσης του πυρήνα SpMV.

Υπάρχει μία μονόδρομη σχέση μεταξύ των δύο μετρικών πρόβλεψης που περιγράφηκαν. Εάν ένα μοντέλο επίδοσης είναι αρκετά ακριβές στην πρόβλεψη του απολύτου χρόνου εκτέλεσης του πυρήνα SpMV για διαφορετικά μπλοκ, τότε θα παρουσιάζει και υψηλή ακρίβεια επιλογής. Το αντίστροφο, ωστόσο, δεν είναι αληθές: η ακριβής επιλογή του μπλοκ εκτέλεσης δεν υπαγορεύει και την ακριβή πρόβλεψη του χρόνου εκτέλεσης, καθώς για την επιλογή αυτή απαιτείται απλά μία καλή διάταξη των υποψηφίων μπλοκ.

Η κύρια επιδίωξη αυτού του κεφαλαίου είναι η επίτευξη υψηλής ακρίβειας επιλογής, καθώς η επιλογή της καλύτερης βελτιστοποίησης για τον πυρήνα SpMV είναι αυτό που μας αφορά. Το μοντέλα επίδοσης θα πρέπει να μπορούν να επιλέξουν την καλύτερη επιλογή μεταξύ του απλού σχήματος CSR και του σχήματος BCSR, είτε στην απλή είτε στην SIMD εκδοχή του. Προφανώς, το μοντέλο MEM δεν μπορεί να διαφοροποιήσει την απλή και την SIMD εκδοχή του BCSR, καθώς ασχολείται μόνο με το μέρος των προσβάσεων στην κύρια μνήμη. Για τον λόγο αυτό, επιλέγουμε εμπειρικά την απλή έκδοση για τα συστήματα SMP και την SIMD για τα συστήματα NUMA. Το Σχήμα 4.6 παρουσιάζει γραφικά την ακρίβεια επιλογής των τεσσάρων υπό θεώρηση μοντέλων επίδοσης στα συστήματα Harpertown και Gainestown, ενώ ο Πίνακας 4.1 συνοψίζει τα αποτελέσματα της ακρίβειας πρόβλεψης. Η ανταγωνιστικότητα του μοντέλου MEM είναι εμφανής στο σύστημα Harpertown, όπου επιτυγχάνει την σωστή πρόβλεψη στους 27 από τους 30 πίνακες (90%) της σουίτας μας. Αυτό είναι λίγο-πολύ αναμενόμενο, μιας και ο πυρήνας SpMV περιορίζεται από το εύρος ζώνης μνήμης στο σύστημα Harpertown. Το μοντέλο MEMCOMP, από την άλλη, υπερεκτιμά την επίδραση των υπολογισμών σε αυτό το σύστημα, με

4.3. Πρόβλεψη του βελτίστου μεγέθους μπλοκ

	Harpertown			Gainestown		
	<i>Correct</i>	<i>Dist.</i>	<i>C.I.</i>	<i>Correct</i>	<i>Dist.</i>	<i>C.I.</i>
MEM	27/30	1.81%	±2.05%	10/30	8.50%	±1.86%
SPARSITY	26/30	3.33%	±4.54%	26/30	5.89%	±4.90%
MEMCOMP	22/30	5.61%	±2.88%	28/30	3.63%	±0.34%
OVERLAP	26/30	3.53%	±4.54%	26/30	5.89%	±4.90%

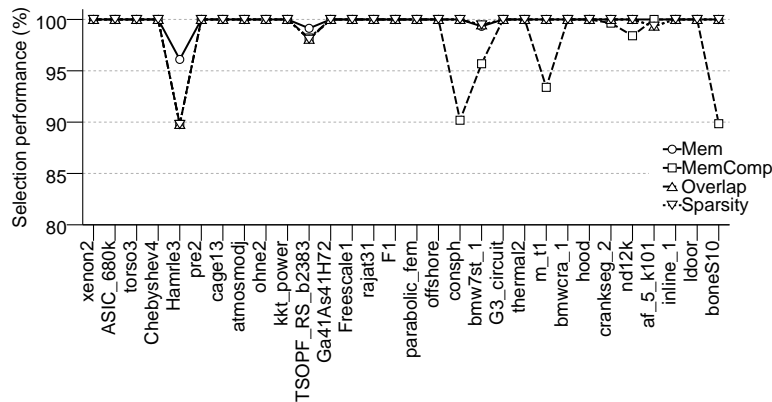
Πίνακας 4.1: Ακρίβεια επιλογής των θεωρουμένων μοντέλων επίδοσης. Παρουσιάζονται το πλήθος των σωστών προβλέψεων, η μέση απόσταση των αποτυχημένων προβλέψεων από την βέλτιστη επίδοση (*Dist.*) και ένα διάστημα εμπιστοσύνης 95% (*C.I.*). Στις περιπτώσεις λίγων αποτυχημένων προβλέψεων, τα διαστήματα εμπιστοσύνης μπορεί να υπερβούν τη μέση απόσταση από τη βέλτιστη πρόβλεψη· αυτό οφείλεται στο πολύ μικρό δείγμα αποτυχημένων προβλέψεων.

αποτέλεσμα να χάνει την σωστή πρόβλεψη σε οκτώ περιπτώσεις. Τα μοντέλα SPARSITY και OVERLAP είναι προσαρμόζονται σαφώς καλύτερα και επιτυγχάνουν 87% ακρίβεια (26 σωστές προβλέψεις). Στην περίπτωση λάθως προβλέψεων, όλα τα μοντέλα, πλην του MEMCOMP, βρίσκονται εντός του 5% της βελτίστου επίδοσης, πράγμα που υποδηλώνει μία υψηλή ποιότητα πρόβλεψης. Το μοντέλο MEMCOMP πέφτει θύμα της υπερεκτίμησης των υπολογισμών και η προβλέψεις του μπορούν να οδηγήσουν σε μέχρι 10% μείωση της επίδοσης.

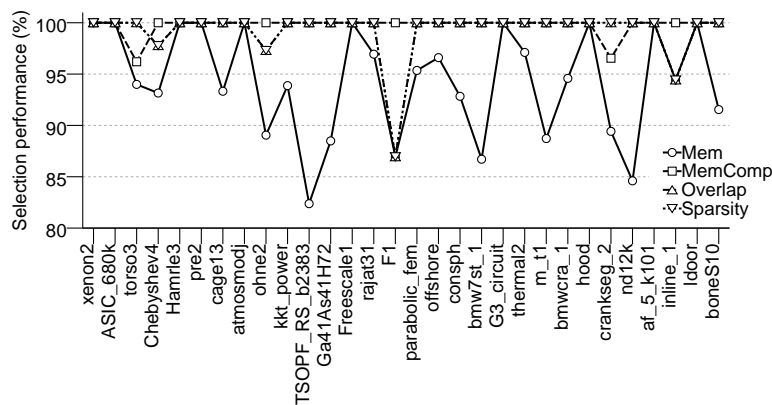
Η κατάσταση αντιστρέφεται πλήρως στο σύστημα Gainestown. Οι υπολογισμοί παίζουν αρκετά σημαντικό ρόλο σε αυτή την περίπτωση και το μοντέλο MEMCOMP αριστεύει, επιτυγχάνοντας σωστές προβλέψεις για το σύνολο σχεδόν των πινάκων (28 σωστές προβλέψεις, 93%). Η επίδοση του μοντέλου MEM είναι απογοητευτική, επιτυγχάνοντας μόνο 10 σωστές προβλέψεις (33%). Αυτό είναι αναμενόμενο, σύμφωνα με την προηγούμενη ανάλυση επίδοσης, καθώς η επίδοση του SpMV στο σύστημα Gainestown δεν είναι σχετίζεται άμεσα με το μέγεθος της αναπαράστασης του πίνακα. Τα μοντέλα SPARSITY και OVERLAP παρουσιάζουν μία μάλλον σταθερή εικόνα, παρέχοντας σωστές προβλέψεις σε 26 πίνακες (87%). Ωστόσο, το κόστος της λανθασμένης πρόβλεψης είναι αυξημένο στο σύστημα Gainestown, οδηγώντας σε μία μείωση της επίδοσης κατά 5.9% για τα μοντέλα SPARSITY και OVERLAP και 8.5% για το μοντέλο MEM.

Παρά το ότι ο κύριος στόχος των μοντέλων επίδοσης που παρουσιάστηκαν σε αυτό το κεφάλαιο είναι η σωστή πρόβλεψη του βελτίστου μπλοκ για το σχήμα BCSR, τα μοντέλα MEM, MEMCOMP και OVERLAP προέρχονται από μία αφαιρετική θεώρηση του χρόνου εκτέλεσης του πυρήνα SpMV. Για τον λόγο αυτό, παρουσιάζουμε στον Πίνακα 4.2 κάποια στατιστικά για την ακρί-

4. Δυνατότητες βελτιστοποίησης των μεθόδων μπλοκ



(i) Harpertown.



(ii) Gainestown.

Σχήμα 4.6: Ακρίβεια επιλογής των θεωρουμένων μοντέλων επίδοσης. Το μοντέλο MEM επιτυγχάνει υψηλή ακρίβεια επιλογής στο σύστημα Harpertown, όπου η συμφώρηση για εύρος ζώνης μνήμης είναι εντονότερη, αλλά αποτυγχάνει στις προβλέψεις του στο σύστημα Gainestown, όπου το υπολογιστικό μέρος του πυρήνα παίζει σημαντικό ρόλο στην συνολική επίδοση του SpMV. Το αντίστροφο συμβαίνει με το μοντέλο MEMCOMP, ενώ τα OVERLAP και SPARSITY εξισορροπούν καλύτερα το μέρος των προσβάσεων στη μνήμη με το υπολογιστικό μέρος, επιτυγχάνοντας καλές προβλέψεις σε αμφότερες των περιπτώσεων.

4.3. Πρόβλεψη του βελτίστου μεγέθους μπλοκ

	Harpertown		Gainestown	
	<i>Dist. (R)</i>	<i>Dist. (A)</i>	<i>Dist. (R)</i>	<i>Dist. (A)</i>
MEM	12.94 ± 2.71%	18.22 ± 3.78%	43.05 ± 3.35%	45.32 ± 2.85%
MEMCOMP	28.41 ± 3.93%	23.13 ± 4.36%	23.98 ± 3.35%	19.76 ± 3.75%
OVERLAP	5.28 ± 1.99%	12.04 ± 4.20%	9.06 ± 5.71%	15.81 ± 5.12%

Πίνακας 4.2: Ακρίβεια πρόβλεψης χρόνου εκτέλεσης των θεωρουμένων μοντέλων επίδοσης του SpMV. Παρουσιάζεται η μέση απόσταση μεταξύ της προβλεφθείσας και της πραγματικής επίδοσης για πίνακες με κανονική διάταξη (*Dist. (R)*, 15 πίνακες) και ολόκληρη την σουίτα πινάκων (*Dist. (A)*). Παρουσιάζονται, επίσης, τα αντίστοιχα διαστήματα εμπιστοσύνης 95%.

βεια πρόβλεψης του χρόνου εκτέλεσης. Καθώς τα προτεινόμενα μοντέλα θεωρούν την επίδοση του πυρήνα SpMV περιοριζόμενη κατά κύριο λόγο από το εύρος ζώνης μνήμης, παρουσιάζουμε τα αποτελέσματα για τους πίνακες με σχετικά υψηλό λόγο *flor:byte* (≥ 1.6), καθώς αυτοί δεν υποφέρουν από παράπλευρα προβλήματα, όπως επιβάρυνση βρόχων, ακανόνιστες προσβάσεις στη μνήμη κ.ο.κ. (βλ. Κεφάλαιο 3 για μία λεπτομερή ανάλυση) και αποτελούν την τυπική περίπτωση πινάκων των εφαρμογών που κάνουν χρήση του πυρήνα SpMV. Το μοντέλο OVERLAP επιτυγχάνει τις πιο ακριβείς προβλέψεις για τον χρόνο εκτέλεσης, οδηγώντας σε μία διαφορά στην απόλυτη επίδοση της τάξης του 5.3% στο σύστημα Harpertown για τους πίνακες με κανονική διάταξη των στοιχείων τους. Στο σύστημα Gainestown, η διαφορά αυτή αυξάνεται, αλλά παραμένει χαμηλότερη του 10%. Εάν θεωρηθούν και οι πίνακες με ακανόνιστη διάταξη των στοιχείων τους, η ακρίβεια πρόβλεψης χειροτερεύει, φθάνοντας σε μία απόσταση 12% στο σύστημα Harpertown και 15.8% στο σύστημα Gainestown. Τα μοντέλα MEM και MEMCOMP είναι λιγότερο ακριβή, με το πρώτο να ταιριάζει περισσότερο στο σύστημα Harpertown και το δεύτερο στο σύστημα Gainestown, ωστόσο, η ακρίβειά τους είναι αρκετά χαμηλότερη του OVERLAP. Θα πρέπει να σημειώσουμε εδώ ότι η ακρίβεια του μοντέλου MEMCOMP φαίνεται να ευνοείται στους πιο ακανόνιστους πίνακες. Λόγω του ακανονίστου σχήματος προσβάσεων στη μνήμη, οι πίνακες αυτοί τείνουν να υπο-εκμεταλλεύονται το διαθέσιμο εύρος ζώνης μνήμης, με αποτέλεσμα οι υπολογισμοί τους να είναι περισσότερο εκτεθειμένοι, επιτρέποντας στο μοντέλο MEMCOMP να επιτύχει καλύτερες προβλέψεις.

Τέλος, το Σχήμα 4.7 παρουσιάζει τον προβλεφθέντα χρόνο εκτέλεσης από τα τρία θεωρούμενα μοντέλα επίδοσης για τους πίνακες με κανονική διάταξη των στοιχείων τους. Το μοντέλο MEM τείνει να υπερεκτιμά την επίδοση του SpMV, καθώς θεωρεί μόνο το μέρος των προσβάσεων στην μνήμη. Από την

άλλη, το μοντέλο MEMCOMP υποεκτιμά την επίδοση, καθώς αποσυνδέει τελείως τα μέρη των υπολογισμών και των προσβάσεων στην μνήμη. Το μοντέλο OVERLAP, ωστόσο, τείνει να εξισορροπήσει τα δύο μέρη και επιτυγχάνει μία καλύτερη προσέγγιση της επίδοσης του SpMV. Η γραμμή της επίδοσής του για τις δύο αρχιτεκτονικές αναδεικνύει την καλύτερη δυνατότητα προσαρμογής του: στο σύστημα Harpertown, βρίσκεται πιο κοντά στην γραμμή του μοντέλου MEM, αποδεικνύοντας ότι το μέρος των προσβάσεων στην μνήμη είναι πιο σημαντικό σε αυτή την αρχιτεκτονική· το αντίστροφο συμβαίνει στο σύστημα Gainestown, όπου η γραμμή του OVERLAP είναι πλησιέστερα στην γραμμή του MEMCOMP.

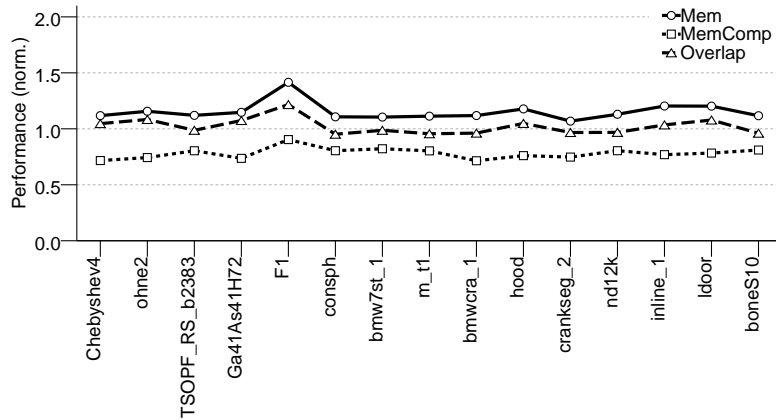
4.3.6 Επεκτάσεις

Τα μοντέλα επίδοσης που παρουσιάστηκαν σε αυτό το κεφάλαιο εστιάζουν στην επίδοση της σειριακής εκτέλεσης του πυρήνα SpMV. Η προηγούμενή μας ανάλυση της επίδοσης του πυρήνα SpMV αποκάλυψε ότι το μέρος των προσβάσεων στη μνήμη γίνεται ολοένα και πιο σημαντικό στις πολυνηματικές εκτελέσεις, προσδίδοντας μία σχεδόν γραμμική συσχέτιση μεταξύ του μεγέθους αναπαράστασης του πίνακα και της επίδοσης του SpMV. Για τον λόγο αυτό, αναμένουμε ότι το μοντέλο MEM θα προβλέπει με ακρίβεια όχι μόνο τα βέλτιστα μπλοκ, αλλά και τον απόλυτο χρόνο εκτέλεσης, υπό την προϋπόθεση ότι δεν υπάρχουν άλλοι ανασταλτικοί επί της επίδοσης παράγοντες (π.χ., ανισορροπία φόρτου εργασίας). Όσον αφορά στο μοντέλο OVERLAP, ο σωστός καθορισμός του συντελεστή μη επικάλυψης (NOF) θα του επιτρέψει να φθάσει την ακρίβεια του μοντέλου MEM, καθώς αποδείχθηκε ιδιαίτερα προσαρμόσιμο. Ομοίως, και το μοντέλο SPARSITY αποδείχθηκε ιδιαίτερα προσαρμόσιμο. Το μοντέλο MEMCOMP, από την άλλη, είναι περισσότερο μονολιθικό και αναμένουμε να είναι λιγότερο ακριβές για μία πολυνηματική εκτέλεση.

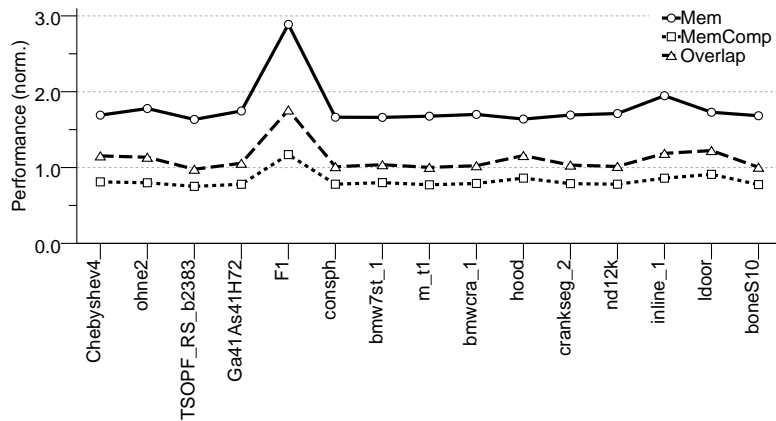
Η επίτευξη μίας απολύτου πρόβλεψης της επίδοσης του πυρήνα SpMV δεν είναι μία εύκολη διαδικασία. Όλα τα θεωρηθέντα μοντέλα κάνουν την παραδοχή ότι το μόνο πρόβλημα επίδοσης του πυρήνα είναι η έλλειψη εύρους ζώνης μνήμης, το οποίο είναι η τυπική περίπτωση για την πλειονότητα των πινάκων που εμφανίζονται στους συνήθεις υπολογισμούς. Ωστόσο, ένα πιο γενικό μοντέλο επίδοσης θα πρέπει να λαμβάνει υπόψη του και παράμετρους όπως ο λόγος flop:byte (ή ο μέσος αριθμός μη μηδενικών στοιχείων ανά γραμμή⁵) και θέματα ανισορροπίας του φόρτου εργασίας σε πολυνηματικές εκτελέσεις. Το μοντέλο OVERLAP απέδειξε μία ιδιαίτερη ικανότητα προσαρμογής και πιστεύ-

⁵ Οι Buttari et al. [2007] παρουσιάζουν ένα μοντέλο επίδοσης που λαμβάνει υπόψη του το μέσο αριθμό μη μηδενικών στοιχείων ανά γραμμή, αλλά δεν εξετάζουν πολυνηματικές εκτελέσεις και περιορίζονται σε λίγους πίνακες.

4.3. Πρόβλεψη του βελτίστου μεγέθους μπλοκ



(i) Harpertown.



(ii) Gainestown.

Σχήμα 4.7: Ακρίβεια πρόβλεψης χρόνου εκτέλεσης των θεωρουμένων μοντέλων επίδοσης για πίνακες με κανονική διάταξη των στοιχείων τους (λόγος flops:byte ≥ 1.6). Το μοντέλο MEM τείνει να υπερεκτιμά την επίδοση, ενώ το μοντέλο MEMCOMP την υποεκτιμά. Το μοντέλο OVERLAP καταφέρνει την εξισορρόπηση των μερών των προσβάσεων στη μνήμη και των υπολογισμών, επιτυγχάνοντας υψηλή ακρίβεια. Ο πίνακας F1 μπορεί να θεωρηθεί ως εξαίρεση, καθότι παρουσιάζει μία μάλλον ακανόνιστη δομή, παρά τον υψηλό του λόγο flops:byte.

ουμε ότι θα μπορεί να «συλλάβει» και αυτές τις ακραίες περιπτώσεις με την κατάλληλη προσαρμογή των παραμέτρων του.

4.4 Περίληψη

Στο κεφάλαιο αυτό, συζητήσαμε και αξιολογήσαμε τις δυνατότητες βελτιστοποίησης που παρέχουν τα διάφορα σχήματα αποθήκευσης σε μπλοκ. Δείξαμε ότι οι σειριακές εκτελέσεις δεν κορέννουν το εύρος ζώνης μνήμης του συστήματος και αφήνουν χώρο για υπολογιστικές βελτιστοποιήσεις, κυρίως στις αρχιτεκτονικές NUMA. Πράγματι, η χρήση των επεκτάσεων SIMD των συγχρόνων επεξεργαστών επέτρεψε την επιτάχυνση της επίδοσης του SpMV περισσότερο από 50% για κάποια μπλοκ του BCSR. Με κίνητρο αυτή την συμπεριφορά, προτείναμε και αξιολογήσαμε δύο μοντέλα επίδοσης (συγκεκριμένα, το MEMCOMP και το OVERLAP) για την επιλογή του βελτίστου μπλοκ του BCSR· τα μοντέλα αυτά λαμβάνουν υπόψη τους αμφότερα τα μέρη προσβάσεων στη μνήμη και υπολογισμών του πυρήνα. Επιπροσθέτως, το μοντέλο OVERLAP μπόρεσε να προσαρμοστεί επιτυχώς στις διαφορετικές αρχιτεκτονικές και να προβλέψει με αρκετή ακρίβεια την απόλυτη επίδοση του SpMV στους πίνακες με κανονική διάταξη των στοιχείων τους.

Στις πολυνηματικές εκτελέσεις, ο κορεσμός του εύρους ζώνης μνήμης εξαλείφει κάθε όφελος από οποιαδήποτε υπολογιστική βελτιστοποίηση. Η δυνατότητα συμπίεσης ενός σχήματος αποθήκευσης γίνεται πλέον ζωτικής σημασίας και τα σχήματα που οδηγούν στην μικρότερη αναπαράσταση του πίνακα παρουσιάζουν ένα υψηλό δυναμικό επίδοσης. Ένα βασικό συμπέρασμα της ανάλυσης αυτού του κεφαλαίου είναι η ύπαρξη μίας σαφούς διεγκυστίνας μεταξύ της υψηλής ικανότητας συμπίεσης και του υπολογιστικού πλεονεκτήματος ενός σχήματος αποθήκευσης. Τα σχήματα μεγαλύτερης συμπίεσης, όπως το VBL, πρέπει να πληρώσουν ένα υψηλό κόστος αποσυμπίεσης στη σειριακή εκτέλεση (π.χ., πρόσβαση στις πρόσθετες δομές δεδομένων, χρονοβόρες πράξεις κ.ο.κ.), αλλά αποζημιώνονται στις πολυνηματικές εκτελέσεις. Σε αυτές τις περιπτώσεις, το πλεονέκτημα επίδοσης που έχουν είναι εμφανές, καθώς ο χρόνος εκτέλεσης εξαρτάται αποκλειστικά από το διαθέσιμο εύρος ζώνης μνήμης.

Το Σχήμα Αποθήκευσης Compressed Sparse eXtended

Η ανάλυση της επίδοσης του πυρήνα SpMV και των σχημάτων αποθήκευσης σε μπλοκ, που πραγματοποιήσαμε στα προηγούμενα κεφάλαια, ανέδειξε τη συμφόρηση για το εύρος ζώνης της κύριας μνήμης ως το κυριώτερο πρόβλημα επίδοσης του πυρήνα SpMV στις σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών. Καθώς ο αριθμός των νημάτων του SpMV αυξάνεται, το εύρος ζώνης μνήμης κορέννεται γρήγορα, μην αφήνοντας καθόλου χώρο για υπολογιστικές βελτιστοποιήσεις. Η συμπεριφορά αυτή είναι ιδιαίτερα εμφανής στις αρχιτεκτονικές SMP, όπου η μοναδική παράμετρος επίδοσης στις πολυνηματικές εκτελέσεις είναι το μέγεθος της αναπαράστασης του πίνακα· όσο μικρότερο είναι το μέγεθος της αναπαράστασης, τόσο υψηλότερη είναι η επίδοση του πυρήνα SpMV. Λόγω της εκ των προτέρων άγνωστης κατανομής των μη μηδενικών στοιχείων ενός αραιού πίνακα, ένα σχήμα αποθήκευσης υψηλής συμπίεσης θα πρέπει να εναπόκειται σε ειδικές δομές δεδομένων και πράξεις αποσυμπίεσης, ώστε να αποφεύγει την χρήση επιπροσθέτων μηδενικών στοιχείων. Οι πράξεις αυτές πμορούν να εισάγουν σημαντική υποοριστική επιβάρυνση, η οποία, ωστόσο, αποκρύβεται πλήρως στις πολυνηματικές εκτελέσεις, όπου η συμφόρηση στο εύρος ζώνης μνήμης αφήνει αρκετό χώρο για χρονοβόρες υπολογιστικές πράξεις.

Εμπνεόμενοι από αυτή την εγγενή συμπεριφορά του SpMV, προτείνουμε στο κεφάλαιο αυτό το σχήμα *Compressed Sparse eXtended* (CSX), ένα σχήμα αποθήκευσης αραιών πινάκων υψηλής συμπίεσης, το οποίο μπορεί να ανιχνεύει και να κωδικοποιεί ταυτόχρονα μία πληθώρα κανονικοτήτων στον πίνακα. Το σχήμα CSX χρησιμοποιεί άμεσες τεχνικές συμπίεσης, όπως είναι οι κωδικοποιήσεις delta και run-length, ώστε να συμπιέσει αποδοτικά την δομή `colind` του αρχικού CSR. Σε συνδυασμό με την χρήση μετασχηματισμών του πίνακα, το σχήμα CSX μπορεί να ανιχνεύει μία ευρεία γκάμα κανονικοτήτων μη μηδενικών στοιχείων, συμπεριλαμβανομένων οριζοντίων, κατακορύφων, διαγωνίων και δισδιαστάτων κανονικοτήτων. Το χαρακτηριστικό αυτό όχι μόνο

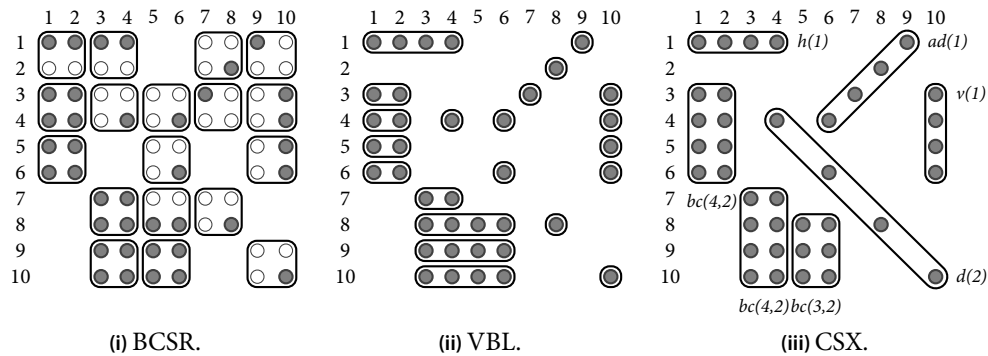
αυξάνει τις δυνατότητες συμπίεσης, αλλά επιτρέπει, επιπλέον, στο σχήμα CSX να προσαρμόζεται επιτυχώς στις ιδιαιτερότητες κάθε πίνακα χωρίς να θυσιάζεται η επίδοση. Πράγματι, το σχήμα CSX μπορεί να επιταχύνει την επίδοση του SpMV περισσότερο από 60% κατά μέσο όρο σε συστήματα SMP και περίπου 20% σε αρχιτεκτονικές NUMA. Επιπλέον, η ικανότητά του να προσαρμόζεται στην δομή των μη μηδενικών στοιχείων κάθε πίνακα παρέχει σημαντική σταθερότητα στην επίδοσης, ένα χαρακτηριστικό που λείπει από την πλειονότητα των εναλλακτικών του CSR σχημάτων.

Το CSX είναι ένα σχετικά σύνθετο σχήμα αποθήκευσης. Στο κεφάλαιο αυτό, περιγράφουμε με λεπτομέρεια κάθε άποψή του και αξιολογούμε την επίδοσή του στην υπολογιστική μας πλατφόρμα. Λαμβάνουμε ιδιαίτερα υπόψη το κόστος προεπεξεργασίας του πίνακα και αξιολογούμε την συνολική επίδραση του CSX στην επίδοση ενός προσομοιωτή φυσικών φαινομένων.

5.1 Η ανάγκη ενός ενιαίου σχήματος αποθήκευσης

Οι αραιοί πίνακες που προκύπτουν από την διακριτοποίηση των μερικών διαφορικών εξισώσεων έχουν συνήθως τα μη μηδενικά τους στοιχεία διατεταγμένα σε κανονικότητες, που είτε εκτείνονται κατά την μία διάσταση (π.χ., οριζόντια, κατακόρυφα, διαγώνια) είτε σχηματίζουν δισδιάστατα μπλοκ. Η ακριβής μορφή αυτών των κανονικοτήτων εξαρτάται κυρίως από την πεδίο της υποκειμένου εφαρμογής και από τυχόν προεπεξεργασία του αρχικού πίνακα (π.χ., τεχνικές ελαχιστοποίησης του εύρους ζώνης του) που τροποποιεί την αρχική κατανομή των μη μηδενικών στοιχείων. Όπως συζητήθηκε στο Κεφάλαιο 2, η εκμετάλλευση αυτών των κανονικοτήτων μπορεί να οδηγήσει σε μείωση του μεγέθους αναπαράστασης του πίνακα. Ωστόσο, ο περιορισμός ενός σχήματος αποθήκευσης στην ανίχνευση ενός μόνο τύπου κανονικοτήτων μπορεί να έχει φθίνουσα απόδοση σε περιπτώσεις που οι προς ανίχνευση κανονικότητες δεν υπάρχουν σε επαρκείς ποσότητες εντός του αραιού πίνακα, οδηγώντας ακόμα και σε αύξηση του μεγέθους του πίνακα σε σύγκριση με το σχήμα CSR. Ένα παράδειγμα φαίνεται στο Σχήμα 5.1, όπου παρουσιάζεται ένας πίνακας με μία πληθώρα κανονικοτήτων· προσπαθώντας κανείς να κατασκευάσει μπλοκ σταθερού μεγέθους (BCSR) καταλήγει στην χρήση υπερβολικού αριθμού μηδενικών στοιχείων, ενώ το σχήμα VBL εισάγει σημαντική επιβάρυνση στις διαγώνιες κανονικότητες. Άλλες εναλλακτικές, όπως τα διασπώμενα σχήματα, τα οποία χωρίζουν τον πίνακα σε υποπίνακες, καθένας εκ των οποίων αποθηκεύει μία διαφορετική κανονικότητα, οδηγούν σε επιπλέον επιβαρύνσεις της επίδοσης, λόγω των πολλαπλών πράξεων SpMV (βλ. Κεφάλαιο 4, Ενότητα 4.1). Η ανάγκη ενός ενιαίου σχήματος αποθήκευσης, το οποίο θα μπορεί να ενσωματώσει μία πληθώρα διαφορετικών κανονικοτήτων, αποτελεί την πιο βιώσιμη

5.1. Η ανάγκη ενός ενιαίου σχήματος αποθήκευσης



Σχήμα 5.1: Το πλεονέκτημα της ανίχνευσης πολλαπλών κανονικοτήτων εντός του αραιού πίνακα (οι γκρι κουκκίδες αποτελούν μη μηδενικά στοιχεία, οι λευκές επιπρόσθετα μηδενικά). Η χρήση επιπροσθέτων μηδενικών στοιχείων στα σχήματα αποθήκευσης σε μπλοκ σταθερού μεγέθους (BCSR) μπορεί να γίνει υπερβολική για πίνακες με ακανόνιστη δομή των μη μηδενικών τους στοιχείων, ενώ η εκμετάλλευση ενός είδους κανονικοτήτων (VBL) δεν μπορεί να εκμεταλλευτεί κανονικότητες σε διαφορετικές διευθύνσεις. Το προτεινόμενο σχήμα CSX μπορεί να ανιχνεύει πληθώρα κανονικοτήτων χωρίς την χρήση επιπροσθέτων μηδενικών στοιχείων. Οι σημειώσεις δίπλα ή κάτωθεν των ανιχνευθέντων από του CSX κανονικοτήτων υποδηλώνει την ακριβή τους πραγμάτωση (βλ. κείμενο) [Υπόμνημα κανονικοτήτων CSX – $h(X)$: οριζόντια, $v(X)$: κατακόρυφη, $d(X)$: διαγώνια, $ad(X)$: αντι-διαγώνια, $bc(X, X)$: μπλοκ κατά στήλες· οι αριθμοί εντός παρενθέσεων είναι οι αποστάσεις δέλτα ή οι διαστάσεις του μπλοκ· οι αριθμοί στο όνομα των μονάδων δέλτα υποδηλώνουν το πλήθος των bit της αναπαράστασής τους.]

λύση για ένα εναλλακτικό σχήμα αποθήκευσης υψηλών επιδόσεων. Ο Πίνακας 5.1 συνοψίζει τα βασικά χαρακτηριστικά των πιο σημαντικών εναλλακτικών του CSR σχημάτων αποθήκευσης.

Το προτεινόμενο σχήμα αποθήκευσης Compressed Sparse eXtended (CSX) είναι ένα σχήμα αποθήκευσης υψηλής συμπίεση που ενσωματώνει πέντε διαφορετικούς τύπους κανονικοτήτων και μπορεί εύκολα να επεκταθεί προς υποστήριξη περισσότερων. Κατά την σχεδίαση του CSX θέσαμε ένα σύνολο στόχων για το νέο σχήμα:

- (α') Ο κύριος στόχος του θα πρέπει να είναι η ελαχιστοποίηση του μεγέθους αναπαράστασης του πίνακα, καθώς η επίδοση του πυρήνα SpMV περιορίζεται από το εύρος ζώνης μνήμης.
- (β') Η κάλυψη ενός σημαντικού εύρους κανονικοτήτων εντός του αραιού πίνακα, συμπεριλαμβανομένων οριζοντίων, διαγωνίων και δισδιάστατων κανονικοτήτων.

5. Το Σχήμα Αποθήκευσης COMPRESSED SPARSE EXTENDED

Format	Substructures	Padding	Loop overheads	Decompression overheads
CSR	None	No	No	No
BCSR	fixed 2-D	Yes	No	No
RSDIAG	fixed diagonal	Yes	No	No
Decomposed	fixed 1-D	No	Yes	No
VBL	variable horiz.	No	No	Yes
CSX	variable 1-D, 2-D	No	No	Yes

Πίνακας 5.1: Τα κύρια χαρακτηριστικά των σημαντικότερων εναλλακτικών του CSR.

- (γ') Η σταθερή συμπεριφορά επίδοσης σε διαφορετικούς πίνακες και σε αρχιτεκτονικές συμμετρικής και μη ομοιόμορφης πρόσβασης στην κύρια μνήμη.
- (δ') Η δυνατότητα επέκτασης και προσαρμογής του, υπό την έννοια της υποστήριξης νέων κανονικοτήτων ή της υλοποίησης επεκτάσεων των υπάρχόντων.

Για την επίτευξη των δύο πρώτων στόχων, χρησιμοποιούμε επιθετική τεχνική συμπίεσης για το σχήμα CSX. Πιστεύουμε ότι οι τεχνικές συμπίεσης δεδομένων θα διαδραματίσουν ακόμα σημαντικότερο ρόλο στις μελλοντικές πολυπύρηνες αρχιτεκτονικές, ως ένα μέσο όχι μόνο για την ελαχιστοποίηση του κόστους επικοινωνίας στα διαφορετικά επίπεδα (επεξεργαστής-μνήμη, επεξεργαστής-επεξεργαστής κ.λπ.), αλλά και για την αύξηση της ενεργειακής τους απόδοσης. Το σχήμα CSX «κτίζει» επί του σχήματος CSR-DU [Kourtis et al., 2008b] (βλ. Κεφάλαιο 2, Ενότητα 2.3), προσθέτοντας την κωδικοποίηση `run-length` στην διαφορική δεικτοδότηση (κωδικοποίηση δέλτα) των στηλών που πραγματοποιεί ήδη το CSR-DU, ώστε να καταστεί εφικτή η ανίχνευση ακολουθιών μη μηδενικών στοιχείων, είτε συνεχόμενων είτε χωριζομένων με κάποια σταθερή απόσταση δ . Για την ανίχνευση μη οριζοντίων κανονικοτήτων, χρησιμοποιούμε την έννοια των μετασχηματισμών συντεταγμένων, ώστε να μετασχηματίσουμε τα στοιχεία του πίνακα κατά την επιθυμητή διάταξη διάσχισης (π.χ., οριζόντια, κατακόρυφη, κατά μπλοκ κ.ο.κ.) και στην συνέχεια επαναχρησιμοποιούμε την ίδια ακριβώς μέθοδο ανίχνευσης που χρησιμοποιούμε για τις οριζόντες κανονικότητες. Η τεχνική αυτή ικανοποιεί, επίσης, τον στόχο της δυνατότητας επέκτασης, καθώς αρκεί ο ορισμός ενός μετασχηματισμού '1-1' για την ανίχνευση οποιασδήποτε κανονικότητας εντός του πίνακα. Ο τρίτος στόχος διασφαλίζεται στα συστήματα SMP από την ιδιαίτερα συμπίεσμένη αναπαράσταση του σχήματος CSX, η οποία οδηγεί μέγιστους λόγους συμπίεσης. Στα συστήματα NUMA, λαμβάνουμε ιδιαίτερη μνεία για την βελτιστοποίηση των υπολογισμών μέσω της παραγωγής βελτιστοποιημένου κώδικα στο

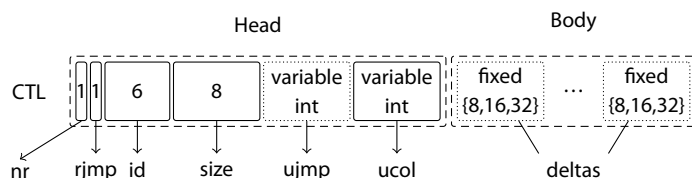
χρόνο εκτέλεσης και της ελάφρυνσης του σχήματος συμπίεσης. Η παραγωγή κώδικα κατά τον χρόνο εκτέλεσης όχι μόνο επιτρέπει την υψηλών επιδόσεων υλοποίηση του SpMV, αλλά διευκολύνει και το έργο της ομοιόμορφης υποστήριξης των πολλαπλών κανονικοτήτων.

5.2 Οι δομές δεδομένων του CSX

Το σχήμα CSX αντικαθιστά αμφοτέρους τους πίνακες `rowptr` και `colind` του σχήματος CSR με ένα μοναδικό πίνακα `byte`, ονομαζόμενο `ct1`, που περιέχει ολόκληρη την απαιτούμενη πληροφορία (Σχήμα 5.2). Αντίστοιχα με το σχήμα CSR-DU, το σχήμα CSX διαχωρίζει τον πίνακα σε μονάδες. Στην ορολογία του CSX, μία *μονάδα* αντιπροσωπεύει μία ακολουθία μη μηδενικών στοιχείων, είτε αυτά αποτελούν μία κανονικότητα (*μονάδα κανονικότητας*) είτε μία ακολουθία διαφορετικών αποστάσεων των στηλών, που αναπαρίστανται από τον ίδιο αριθμό `byte` (*μονάδα δέλτα*). Μία μονάδα του CSX απαρτίζεται από δύο μέρη: την *κεφαλή* και το *σώμα*. Η κεφαλή αποτελείται από ένα περιγραφητή της μονάδας, μεγέθους δύο `byte`, και την αρχική της στήλη (πεδίο `ucol`), κωδικοποιημένη σε ακέραιο μεταβλητού μήκους. Ο περιγραφητής των δύο `byte` αποθηκεύει σε 6-bit το αναγνωριστικό (ID) της μονάδας και το μέγεθος της σε πλήθος μη μηδενικών στοιχείων. Το `bit nr` υποδηλώνει την έναρξη καινούργιας γραμμής. Τέλος, το μέρος του σώματος υπάρχει μόνο στις μονάδες δέλτα και αποθηκεύει τις διαφορετικές αποστάσεις των στηλών σε σταθερού μήκους ακεραίους.

Το `bit rjmp` και το πεδίο `ujmp` επιτελούν ειδικό σκοπό. Εφόσον το CSX μπορεί να κωδικοποιήσει μη οριζόντιες κανονικότητες, είναι πιθανό κάποια κανονικότητα να έχει ομαδοποιήσει όλα τα στοιχεία επομένων γραμμών. Για παράδειγμα, η αντι-διαγώνια κανονικότητα του Σχήματος 5.1iii, που ξεκινά από το στοιχείο (1, 9), περιέχει και το μοναδικό στοιχείο της δεύτερης γραμμής, αφήνοντάς την κενή. Οι γραμμές αυτές θα πρέπει να αγνοηθούν κατά τον υπολογισμό του SpMV. Για τον λόγο αυτό, το `bit rjmp` υποδηλώνει την ύπαρξη κενών γραμμών, ενώ το πεδίο `ujmp` αποθηκεύει το πλήθος των προς αγνόηση κενών γραμμών σε ένα ακέραιο μεταβλητού μήκους.

Όπως και το CSR, το σχήμα CSX αποθηκεύει τα μη μηδενικά στοιχεία του πίνακα σε ένα πίνακα `values`, αλλά πλέον κατά γραμμές ανά κανονικότητα. Για παράδειγμα, οι κανονικότητες του Σχήματος 5.1iii θα αποθηκευθούν με την ακόλουθη σειρά: `h(1)`, `ad(1)`, `bc(4,2)`, `v(1)`, `d(2)`, `bc(4,2)` και `bc(3,2)`. Το Σχήμα 5.3 παρουσιάζει την ακριβή αναπαράσταση της δομής `ct1` για τον αραιό πίνακα του παραδείγματος του Σχήματος 5.1iii.



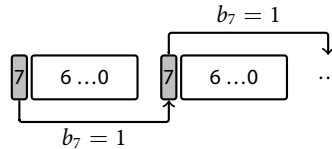
Σχήμα 5.2: Ο πίνακας byte *ctl* που χρησιμοποιείται από το σχήμα CSX για την κωδικοποίηση της τοπολογικής πληροφορίας των μη μηδενικών στοιχείων του αραιού πίνακα. Τα προαιρετικά πεδία σημειώνονται με στικτή γραμμή.

	<i>nr</i>	<i>rjmp</i>	<i>id</i>	<i>size</i>	<i>ucol</i>	-----
<i>h</i> (1)	1	0	0	4	0	
<i>ad</i> (1)	0	0	1	4	5	
<i>bc</i> (4,2)	1	1	2	8	1	0
<i>v</i> (1)	0	0	3	4	9	
<i>d</i> (2)	1	0	4	4	3	
<i>bc</i> (4,2)	1	1	2	8	2	2
<i>bc</i> (3,2)	1	0	5	6	2	

Σχήμα 5.3: Η προκύπτουσα ακολουθία byte της δομής *ctl* για τον πίνακα του Σχήματος 5.1iii. Το πρόσθετο πεδίο *ujmp* για την υπερπήδηση γραμμών, παρόν μόνο όταν έχει τεθεί το bit *rjmp*, είναι σκιασμένο. Οι διαφορικές αποστάσεις των στηλών υπολογίζονται μεταξύ των άνω αριστερά στοιχείων των κανονικοτήτων, με την εξαίρεση των οριζοντίων, από των οποίων οι αποστάσεις υπολογίζονται από του δεξιότερου στοιχείου τους (βλ. κωδικοποίηση του *ucol* της κανονικότητας *ad*(1)). Το συνολικό μέγεθος αναπαράστασης της πληροφορίας των γραμμών και των στηλών είναι μόνο 23 byte, την στιγμή που οι πίνακες *colind* και *rowptr* του CSR απαιτούν 216 byte.

Κωδικοποίηση ακεραίων μεταβλητού μήκους στο CSX

Σε μία προσπάθεια να παραμείνει όσο το δυνατόν συμπαγέστερο, το σχήμα CSX χρησιμοποιεί ακεραίους μεταβλητού μήκους για την κωδικοποίηση των αρχικών στηλών των μονάδων του και των κενών γραμμών. Με αυτή την κωδικοποίηση, ένας οποιοσδήποτε ακεραίος αποθηκεύεται σε τμήματα των 7-bit, διατηρώντας το πιο σημαντικό bit (bit 7) κάθε byte ως σύνδεσμο μεταξύ των διαφορετικών τμημάτων (Σχήμα 5.4). Το όφελος στο συνολικό μέγεθος της αναπαράστασης του πίνακα από την χρήση των ακεραίων μεταβλητού μήκους είναι 2–3% και έχει μία σχεδόν άμεση επίδραση στην επίδοση του πυρήνα SpMV στα συστήματα συμμετρικής πρόσβασης στη μνήμη, όπου ο κορεσμός του εύρους ζώνης μνήμης αφήνει αρκετό χώρο για το πρόσθετο υπολογιστικό φορτίο της διαδικασίας αποκωδικοποίησης.



Σχήμα 5.4: Η κωδικοποίηση ακεραίων μεταβλητού μήκους που χρησιμοποιείται από το σχήμα CSX. Μόνο τα επτά κατώτερα bit χρησιμοποιούνται για την αποθήκευση ενός ακεραίου· αριθμοί μεγαλύτεροι του 127 αποθηκεύονται σε τμήματα των 7 bit με το όγδοο bit να εκτελεί χρέη δείκτη συνέχισης.

5.3 Ανίχνευση και κωδικοποίηση κανονικότητων

5.3.1 Εξόρυξη κανονικότητων από τον πίνακα

Το σχήμα CSX μπορεί να ανιχνεύσει μία ευρεία γκάμα μη οριζοντίων κανονικότητων με την χρήση μετασχηματισμών επί των συντεταγμένων των μη μηδενικών στοιχείων του πίνακα. Για την διευκόλυνση της διαδικασίας ανίχνευσης, το σχήμα CSX χρησιμοποιεί μία εσωτερική αναπαράσταση του αραιού πίνακα, η οποία είναι ένα υβρίδιο του CSR και μίας γενικευμένης έκδοσης του σχήματος COO. Αντί για απλά μη μηδενικά στοιχεία, η εσωτερική αναπαράσταση του CSX αποθηκεύει *γενικευμένα στοιχεία*: ένα γενικευμένο στοιχείο είναι είτε ένα μοναδικό μη μηδενικό στοιχείο είτε μία κανονικότητα. Αντίστοιχα με το σχήμα COO, ένα γενικευμένο στοιχείο αποθηκεύεται ως μία πλειάδα της μορφής (i, j, v, t) , λεξιγραφικά διατεταγμένη ως προς το (i, j) . Στην περίπτωση μιας κανονικότητας, το (i, j) αναπαριστά τις συντεταγμένες του πρώτου στοιχείου της κανονικότητας, v είναι ένας πίνακας των στοιχείων της και t ο τύπος της. Διατηρούμε, επίσης, τους δείκτες γραμμών του CSR για γρήγορη πρόσβασης στα στοιχεία της κάθε γραμμής. Η εσωτερική αναπαράσταση κατασκευάζεται άπαξ κατά την διάρκεια της «φόρτωσης» του αρχικού πίνακα, είτε από μορφή CSR είτε από τον δίσκο.

Το CSX ανιχνεύει κανονικότητες εφαρμόζοντας την *κωδικοποίηση run-length* στις στήλες του πίνακα. Η κωδικοποίηση run-length υπολογίζει αρχικά τις διαφορικές αποστάσεις των στηλών και στην συνέχεια σχηματίζει ομάδες, ονομαζόμενες *διαδρομές (runs)*, με την ίδια διαφορική απόσταση (Σχήμα 5.5, Αλγόριθμος 5.1). Κάθε διαδρομή αναγνωρίζεται από την κοινή της διαφορική απόσταση και το μήκος της. Διαδρομές με μήκος μεγαλύτερο ή ίσο του δύο σχηματίζουμε μία κανονικότητα. Ωστόσο, επιβάλλουμε ένα περιορισμό στο ελάχιστο μέγεθος μιας διαδρομής (τεθειμένη στο τέσσερα), ώστε να αποφύγουμε τυχόν διόγκωση πολύ μικρών διαδρομών, οι οποίες αποτελούν μάλλον επιβάρυνση παρά ωφέλεια. Υπάρχει ένα ακόμα λεπτό σημείο κατά την απεικόνιση των ανιχνευμένων διαδρομών σε μονάδες CSX: όλες οι διαδρομές –εκτός των εκκινούντων από την αρχή μιας γραμμής– αγνοούν το πρώτο στοιχείο της

col. indices:	1	10	11	12	13	14	21	41	61	81	...
delta values:	1	9	1	1	1	1	7	20	20	20	...
			$d = 1$					$d = 20$			

Σχήμα 5.5: Παράδειγμα της κωδικοποίησης run-length των στηλών. Παρατηρείστε το παραλειπόμενο στοιχείο αριστερά της κάθε διαδρομής. Αυτό το εγγενές ζήτημα της κωδικοποίησης run-length επιλύεται στην υλοποίηση του CSX, επιτρέποντας ακόμα μεγαλύτερο βαθμό συμπίεσης του πίνακα εισόδου.

πραγματικής κανονικότητας. Για παράδειγμα, στο Σχήμα 5.5, το μήκος των πραγματικών μονάδων είναι 5 και 4, αντί των ανιχνευμένων διαδρομών 4 και 3, οι οποίες αγνοούν τις στήλες 10 και 21, αντίστοιχα. Αυτό μπορεί αποδειχθεί ιδιαίτερα κρίσιμο κατά την ανίχνευση 2-Δ κανονικοτήτων, όπου ισχύουν επιπλέον περιορισμοί ευθυγράμμισης. Στην πραγματική μας υλοποίηση, διορθώνουμε αυτό το θέμα και, επιπλέον, διασπώμε τις μεγάλες διαδρομές σε τμήματα των 255 στοιχείων έτσι, ώστε να χωρούν στο ενός byte πεδίο του μεγέθους μιας μονάδας του CSX.

Η διαδικασία της ανίχνευσης του συνόλου του πίνακα για ένα συγκεκριμένο τύπο κανονικότητας φαίνεται στον Αλγόριθμο 5.2. Πιο συγκεκριμένα, διασχίζουμε τις γραμμές του πίνακα και για κάθε γραμμή συλλέγουμε όλες τις στήλες, οι οποίες δεν αποτελούν ακόμα μέρος οποιασδήποτε κανονικότητας, και εφαρμόζουμε την κωδικοποίηση run-length. Για όλες τις ανιχνευόμενες πραγματώσεις της κανονικότητας (διαφορετικές τιμές d), διατηρούμε στατιστικά (πλήθος μονάδων και πλήθος μη μηδενικών στοιχείων), για να μας καθοδηγήσουν στην τελική επιλογή των προς κωδικοποίηση κανονικοτήτων στο σχήμα CSX.

Ανίχνευση μη οριζοντίων κανονικοτήτων

Κατά την προηγούμενη συζήτησή μας επί της διαδικασίας ανίχνευσης, δεν αναφερθήκαμε συγκεκριμένα σε οριζόντιες κανονικότητες. Πράγματι, η διαδικασία ανίχνευσης «βλέπει» μόνο αριθμούς στηλών και η μόνη απαίτηση είναι οι στήλες αυτές να είναι ταξινομημένες. Η ανίχνευση οριζοντίων κανονικοτήτων είναι, επομένως, άμεση, καθώς τα στοιχεία του πίνακα είναι εξ αρχής διατεταγμένα κατά γραμμές (οριζόντια διάταξη) και λεξικογραφικά ταξινομημένα. Για την ανίχνευση μη οριζοντίων κανονικοτήτων, αρκεί ο μετασχηματισμός των στοιχείων του πίνακα στην επιθυμητή φορά διάσχισης, η λεξικογραφική ταξινόμησή τους και η χρήση τη συνάρτησης DETECTSUBSTR(), που περιγράφηκε στον Αλγόριθμο 5.2. Η περίπτωση των μονοδιαστάτων, μη οριζοντίων κανονικοτήτων είναι αρκετά εύκολη και ο Πίνακας 5.2 παρουσιάζει τους ακριβείς μετασχηματισμούς που χρησιμοποιούνται για την ανίχνευσή τους. Η πε-

```

1: function RLENCODE(colind::in)
   colind: sorted column indices

2:   deltas  $\leftarrow$  DELTAENCODE(colind)
3:   d  $\leftarrow$  deltas[0] ▷ current delta value
4:   l  $\leftarrow$  1 ▷ current sequence length
5:   rle  $\leftarrow$  {(d, l)} ▷ set of column index runs
6:   for i  $\leftarrow$  1 to N do
7:     if deltas[i] = d then
8:       l  $\leftarrow$  l + 1
9:     else
10:      if l  $\geq$  min_run_length then
11:        rle  $\leftarrow$  rle  $\cup$  (d, l)
12:      d  $\leftarrow$  deltas[i]
13:      l  $\leftarrow$  1
14:   end for
15:   return rle

```

Αλγόριθμος 5.1: Κωδικοποίηση run-length των στηλών. Αυτή είναι η απλή υλοποίηση χωρίς την διόρθωση του CSX (βλ. κείμενο). Η συνάρτηση DELTAENCODE() πραγματοποιεί την κωδικοποίηση δέλτα (διαφορική δεικτοδότηση) των στηλών και επιστρέφει μία ακολουθία από τιμές δ .

ρίπτωση των 2- Δ κανονικότητας, ωστόσο, είναι κάπως περισσότερη σύνθετη, καθώς απαιτείται η «σειριοποίηση» των συντεταγμένων των στοιχείων.

Δυστυχώς, μία απλή καμπύλη πλήρωσης χώρου (space-filling curve) για την διάσχιση των μη μηδενικών στοιχείων δεν αποτελεί επιλογή για ένα αραιό πίνακα για δύο βασικούς λόγους: πρώτον, μία τέτοια καμπύλη θα απαιτούσε γραμμικό χώρο δεικτοδότησης τη τάξης του $\Theta(N^2)$, κάτι το οποίο υπερβαίνει σαφώς το $\Theta(NNZ)$ του χώρου δεικτοδότησης των στηλών του αρχικού πίνακα και, δεύτερον, θα υπονοούσε την χρήση μπλοκ σταθερού μεγέθους και αυστηρά ευθυγραμμισμένων, μία επιλογή που εκ των προτέρων θέλαμε να αποφύγουμε. Η προσέγγιση που ακολουθούμε για τις 2- Δ κανονικότητες στο CSX απεικονίζεται στο Σχήμα 5.6. Χωρίζουμε τον πίνακα σε ζώνες σταθερού πλάτους, ευθυγραμμισμένων είτε κατά γραμμές είτε κατά στήλες, και εφαρμόζουμε τον μετασχηματισμό μιας καμπύλης πλήρωσης χώρου εντός της κάθε ζώνης, περνώντας την προκύπτουσα ακολουθία στην συνάρτηση DETECTSUBSTR(). Ωστόσο, κατά την διαδικασία της ανίχνευσης χρειάζεται προσοχή, καθώς δεν αποτελούν έγκυρα μπλοκ όλες οι ανιχνεύσιμες μονάδες.

```

1: procedure DETECTSUBSTR(matrix::in, stats::inout)
   matrix: CSX's internal repr., lexicographically sorted
   stats: substructure statistics

2:   colind  $\leftarrow$   $\emptyset$  ▷ Column indices to encode
3:   foreach row in matrix do
4:     foreach generic element  $e(i, j, v, t)$  in row do
5:       if  $e.t = \text{NONE}$  then ▷  $e$  is not a substructure
6:         colind  $\leftarrow$  colind  $\cup$   $e.j$ 
7:         continue
8:       enc  $\leftarrow$  RLENCODE(colind)
9:       COLLECTSTATS(stats, enc) ▷ Collect statistics from this encoding
10:      colind  $\leftarrow$   $\emptyset$ 
11:     end for
12:   enc  $\leftarrow$  RLENCODE(colind)
13:   COLLECTSTATS(stats, enc) ▷ Collect statistics from this encoding
14:   colind  $\leftarrow$   $\emptyset$ 
15: end for

```

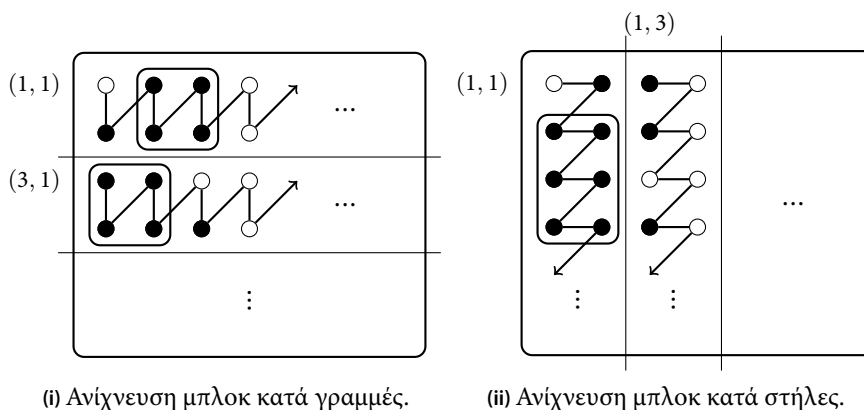
Αλγόριθμος 5.2: Ανίχνευση κανονικοτήτων στην εσωτερική αναπαράσταση του CSX. Η συνάρτηση COLLECTSTATS() συλλέγει και διατηρεί τα στατιστικά κάθε τύπου κανονικότητας και των πραγματώσεών του. Η συνειλεγμένη στη μεταβλητή *stats* πληροφορία θα χρησιμοποιηθεί αργότερα προς καθοδήγηση της επιλογής της καταλληλότερης προς κωδικοποίηση κανονικότητας.

για να είναι μία μονάδα έγκυρη, θα πρέπει η αρχική της στήλη (στον μετασχηματισμένο χώρο) να είναι πολλαπλάσιο του πλάτους της ζώνης. Με την χρήση αυτής της τεχνικής, επιτυγχάνουμε την ανίχνευση «χαλαρά» ευθυγραμμισμένων, μεταβλητού μεγέθους μπλοκ, αυξάνοντας περαιτέρω την δυναμική συμπίεσης του CSX. Υποστηρίζουμε δύο κατηγορίες κανονικοτήτων μπλοκ, τα κατά γραμμές και τα κατά στήλες ευθυγραμμισμένα μπλοκ, και κάθε κατηγορία ορίζει επτά διαφορετικούς τύπους κανονικοτήτων για διαφορετικά πλάτη ζωνών ($r, c \in [2, 8]$ στον Πίνακα 5.2). Συνολικά, η χρήση των μετασχηματισμών επιτρέπει στο CSX την ανώδυνη και αρραγή υποστήριξη 18 διαφορετικών τύπων κανονικοτήτων, ενώ επιτρέπει και την επέκτασή του σε άλλες οικογένειες κανονικοτήτων, π.χ., διαγώνια μπλοκ.

Το Σχήμα 5.7 παρουσιάζει τις κανονικότητες που ανίχνευσε το CSX στην σουίτα πινάκων μας και επιδεικνύει την χαρακτηριστική ικανότητα του CSX στην ανίχνευση πληθώρας κανονικοτήτων εντός ενός αραιού πίνακα. Είναι

Substructure type	Transformation
Horizontal	$(i', j') = (i, j)$
Vertical	$(i', j') = (j, i)$
Diagonal	$(i', j') = (N + j - i, \min(i, j))$
Anti-diagonal	$(i', j') = \begin{cases} (i + j - 1, i), & i + j - 1 \geq N \\ (i + j - 1, N + 1 - j), & i + j - 1 < N \end{cases}$
Block (row aligned)	$(i', j') = (\lfloor \frac{i-1}{r} \rfloor + 1, \text{mod}(i-1, r) + r(j-1) + 1)$
Block (column aligned)	$(i', j') = (\lfloor \frac{j-1}{c} \rfloor + 1, c(i-1) + \text{mod}(j-1, c))$

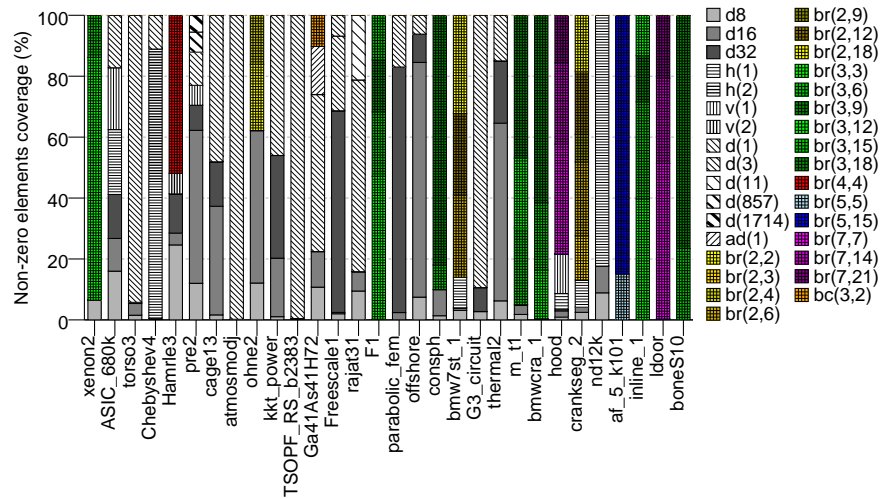
Πίνακας 5.2: Οι εφαρμοζόμενοι από του CSX μετασχηματισμοί συντεταγμένων επί των στοιχείων του πίνακα για την ανίχνευση μη οριζοντίων κανονικότητων (θεωρείται μοναδιαία βάση δεικτοδότησης).



Σχήμα 5.6: Ανίχνευση 2-Δ κανονικότητων στο σχήμα CSX (οι μαύρες κουκκίδες υποδηλώνουν μη μηδενικά στοιχεία). Ο πίνακας χωρίζεται στα όρια γραμμών ή στηλών και ένας μετασχηματισμός πλήρωσης χώρου εφαρμόζεται εντός κάθε τμήματος. Η απλή κωδικοποίηση run-length της προκύπτουσας ακολουθίας αρκεί για την ανίχνευση μπλοκ μεταβλητού μεγέθους, μη αυστηρώς ευθυγραμμισμένων (χρειάζεται μόνο μέριμνα για την αποκοπή των περιττών στοιχείων στην αρχή και το τέλος της κάθε ακολουθίας).

πολύ ενδιαφέρον ότι το CSX μπόρεσε να ανιχνεύσει ακόμα και «περίεργες» κανονικότητες, όπως οι διαγώνιες με αποστάσεις δ 857 και 1714 στον πίνακα `pre2`. Σε πίνακες που κυριαρχούνται από πυκνά μπλοκ, το σχήμα CSX μπόρεσε να ανιχνεύσει μεγάλα μπλοκ σε σημαντικές ποσότητες, όπως φαίνεται από τις περιπτώσεις των πινάκων `inline_1` και `af_k_101`. Η ανίχνευση μεγάλων μπλοκ δεν έχει μόνο θετικό αντίκτυπο στον συνολικό βαθμό συμπίεσης, αλλά παρέχει και ένα σημαντικό πλεονέκτημα επίδοσης κατά τον υπολογισμό του SpMV. Τέλος, ο *χάρτης κανονικότητων* ενός αραιού πίνακα μπορεί να γίνει ιδιαίτερα

5. Το Σχήμα Αποθήκευσης COMPRESSED SPARSE EXTENDED



Σχήμα 5.7: Οι κανονικότητες που ανιχνεύθηκαν και κωδικοποιήθηκαν από το CSX σε ένα ευρύ σύνολο αραιών πινάκων. [Υπόμνημα κανονικοτήτων – dXX : μονάδες δέλτα, $h(X)$: οριζόντιες, $v(X)$: κατακόρυφες, $d(X)$: διαγώνιες, $ad(X)$: αντι-διαγώνιες, $br(X,X)$: μπλοκ κατά γραμμές, $bc(X,X)$: μπλοκ κατά στήλες· οι αριθμοί εντός παρενθέσεων είναι οι αποστάσεις δέλτα ή οι διαστάσεις του μπλοκ· οι αριθμοί στο όνομα των μονάδων δέλτα υποδηλώνουν το πλήθος των bit της αναπαράστασής τους.]

αποκαλυπτικός για συγκεκριμένες συμπεριφορές επίδοσης του πυρήνα SpMV. Για παράδειγμα, οι πίνακες Freescale1, parabolic_fem και offshore κυριαρχούνται από μονάδες δέλτα με πολύ μεγάλες αποστάσεις δ , υπονοώντας ότι είναι αρκετά αραιοί και σχετικά ακανόνιστοι. Πράγματα, η επίδοσή τους είναι ιδιαίτερα αργή χρησιμοποιώντας οποιοδήποτε σχήμα αποθήκευσης, υποφέροντας από ακανόνιστες προσβάσεις στο διάνυσμα εισόδου και ανισορροπίες στην κατανομή του φόρτου εργασίας.

Τύποι κανονικοτήτων και πραγματώσεις τους

Στο CSX κάνουμε μία διάκριση μεταξύ του τύπου μιας κανονικότητας και της πραγμάτωσής της. Στην ορολογία του CSX, η πραγμάτωση μιας κανονικότητας υποδηλώνει το πώς ακριβώς εμφανίζεται μία κανονικότητα εντός του αραιού πίνακα. Για παράδειγμα, οι οριζόντιες κανονικότητες με $d = 1$ και $d = 20$ του Σχήματος 5.5 ανήκουν στον ίδιο τύπο κανονικοτήτων (οριζόντια), αλλά αποτελούν διαφορετικές πραγμάτωσής της. Αντίστοιχα, τα μπλοκ 2×10 και 2×20 είναι πραγμάτωσεις του κατά γραμμές τύπου μπλοκ με $r = 2$. Ένας τύπος κανονικότητας, επομένως, μπορεί να έχει απείρως πολλές πραγ-

ματώσεις σε ένα αραιό πίνακα, πράγμα που προσθέτει εξαιρετική ευελιξία στο σχήμα CSX, επιτρέποντάς του την ανίχνευση σχεδόν οποιουδήποτε σχηματισμού μη μηδενικών στοιχείων εντός του πίνακα. Το πεδίο `id` των 6 bit στη δομή `ct1` του CSX (Σχήμα 5.2) αναφέρεται στην συγκεκριμένη πραγμάτωση της κανονικότητας που της κωδικοποιούμενης μονάδας· δεν αναφέρεται στο γενικό τύπο της κανονικότητας. Αυτό έχει το μειονέκτημα του περιορισμού του συνολικού αριθμού πραγμάτων κανονικότητας σε ένα αραιό πίνακα στις 64, στην πράξη, όμως, μόνο 4–5 πραγμάτων επιλέγονται προς κωδικοποίηση (βλ. Σχήμα 5.7).

5.3.2 Επιλογή κανονικότητας προς τελική κωδικοποίηση

Η πλήρης διαδικασία της ανίχνευσης και της επιλογής των κανονικότητας προς την τελική κωδικοποίηση περιγράφεται στον Αλγόριθμο 5.3, ο οποίος αποτελεί ένα τυπικό αλγόριθμο τοπικής βελτιστοποίησης. Πιο συγκεκριμένα, για κάθε διαθέσιμο τύπο κανονικότητας, μετασχηματίζουμε τον πίνακα κατά την αντίστοιχη διάσχιση και τον διατρέχουμε, συλλέγοντας στατιστικά για τον υπό εξέταση τύπο κανονικότητας. Έχοντας συλλέξει στατιστικά για όλους τους διαθέσιμους τύπους κανονικότητας, απορρίπτουμε τις πραγματώσεις που δεν υπερβαίνουν ένα συγκεκριμένο όριο στην κωδικοποίηση των μη μηδενικών στοιχείων (στην τρέχουσα υλοποίηση, αυτό έχει τεθεί στο 5% του συνολικού αριθμού μη μηδενικών στοιχείων) και, στην συνέχεια, επιλέγουμε την καταλληλότερη κανονικότητα για την κωδικοποίηση του πίνακα (`SELECTTYPE()`). Ο αλγόριθμος συνεχίζει με την κωδικοποίηση του πίνακα με τον επιλεγμένο τύπο κανονικότητας (`ENCODESUBSTR()`), επαναλαμβάνοντας την ίδια διαδικασία μέχρις ότου να μην μπορεί να επιλεγεί κανείς επιπλέον τύπος.

Το κριτήριο για την επιλογή των προς κωδικοποίηση κανονικότητας είναι μία γρήγορη εκτίμηση της επιτευχόμενης μείωσης του μεγέθους της δομής `colind` του αρχικού σχήματος CSR. Θεωρώντας την χρήση ενός μοναδικού και πλήρους αριθμού στήλης ανά κανονικότητα (αγνοούμε τις μονάδες δέλτα), το μέγεθος τη δομής `ct1` θα είναι

$$S_{ctl} = \overbrace{N_{units}}^{\text{encoded}} + \overbrace{NNZ - NNZ_{enc}}^{\text{unencoded}} \quad (5.1)$$

όπου N_{units} είναι ο συνολικός αριθμός των κωδικοποιημένων μονάδων κανονικότητας και NNZ_{enc} είναι το πλήθος των μη μηδενικών στοιχείων που κωδικοποιήθηκαν από αυτόν τον τύπο κανονικότητας. Επομένως, το όφελος επί της δομής `colind` του CSR θα είναι περίπου

$$G = NNZ - S_{ctl} = NNZ_{enc} - N_{units} \quad (5.2)$$

```

1: procedure ENCODEMATRIX(matrix::inout)
   matrix: CSX's internal matrix in row-wise order

2:   repeat
3:     stats  $\leftarrow$   $\emptyset$ 
4:     for all available substructure types t do
5:       TRANSFORM(matrix, t)
6:       LEXSORT(matrix)
7:       DETECTSUBSTR(matrix, stats)
8:       TRANSFORM-1(matrix, t)
9:     end for
10:    FILTERSTATS(stats)  $\triangleright$  Filter out instantiations that encode less
                           than 5% of the non-zero elements
11:    s  $\leftarrow$  SELECTTYPE(stats)
12:    if s  $\neq$  NONE then
13:      TRANSFORM(matrix, s)
14:      LEXSORT(matrix)
15:      ENCODESUBSTR(matrix)  $\triangleright$  Encode the selected substructure
16:    until s = NONE

```

Αλγόριθμος 5.3: Ανίχνευση, επιλογή και κωδικοποίηση των κανονικοτήτων στο σχήμα CSX. Η διαδικασία χωρίζεται σε δύο φάσεις: (α) συλλογή στατιστικών και (β) επιλογή και κωδικοποίηση. Κάθε φορά που ο πίνακας μετασχηματίζεται (TRANSFORM()) κατά μία συγκεκριμένη διάσχιση (βλ. Πίνακα 5.2), απαιτείται η λεξικογραφική ταξινόμηση (LEXSORT()) των μη μηδενικών του στοιχείων. Η συνάρτηση TRANSFORM⁻¹() μετασχηματίζει αντιστρόφως τον πίνακα κατά την αρχική, οριζόντια διάσχιση.

το οποίο αποτελεί την μετρική, που θα πρέπει να μεγιστοποιήσει ένας τύπος κανονικοτήτων, ώστε να επιλεγεί προς κωδικοποίηση.

5.3.3 Κατασκευή των δομών δεδομένων του CSX

Το τελικό βήμα στην κατασκευή του πίνακα CSX είναι η κατασκευή των συγκεκριμένων δομών δεδομένων του CSX, ήτοι των πινάκων *ct1* και *values* (βλ. Ενότητα 5.2). Η διαδικασία, που απεικονίζεται στον Αλγόριθμο 5.4, είναι αρκετά απλή. Διατρέχουμε την κωδικοποιημένη εσωτερική αναπαράσταση του πίνακα και για κάθε συναντώμενη μονάδα, είτε κανονικότητας είτε δέλτα, «γεννάμε» τα αντίστοιχα πεδία της δομής *ct1*. Οι μονάδες δέλτα κατασκευάζονται από τα εναπομείναντα στοιχεία, που δεν ανήκουν σε κάποια συγκε-

κριμένη κανονικότητα. Τα στοιχεία αυτά συλλέγονται κατά την διάσχιση της εσωτερικής αναπαραστάσης και «αποτίθενται» στους πίνακες `ct1` και `values`, είτε κατά την εμφάνιση μιας νέας κανονικότητας είτε κατά την έναρξη μιας νέας γραμμής. Κατά την διαδικασία κατασκευής, καταγράφουμε και το πλήθος των κενών γραμμών (αποσιωπάται από τον Αλγόριθμο 5.4 για λόγους απλότητας), ώστε να θέσουμε καταλλήλως τα πεδία `rjmp` και `ujmp` της δομής `ct1`, ενώ με την έναρξη κάθε μη κενής γραμμής θέτουμε το αντίστοιχο `bit` για την επόμενη κωδικοποιημένη μονάδα.

Υπεύθυνες για την κατασκευή των πεδίων της δομής `ct1` και της «γέννησης» των μη μηδενικών στοιχείων από τις μονάδες της εσωτερικής αναπαραστάσης του CSX είναι το σύνολο των συναρτήσεων `EMIT*()`. Η συνάρτηση `EMITVALUES()` είναι αρκετά απλή: απλά αντιγράφει τα μη μηδενικά στοιχεία του/των γενικευμένου/ων στοιχείου/ων, που λαμβάνει ως όρισμα, στον πίνακα `values` του τελικού πίνακα CSX. Οι συναρτήσεις `EMITSUBSTRUCTURE()` και `EMITDEL-TAUNIT()` επιτελούν αντίστοιχες λειτουργίες: αμφότερες γεμίζουν τα πεδία `id`, `size` και `uol` του πίνακα `byte ct1` (βλ. Σχήμα 5.2), ενώ η `EMITDELTAUNIT()` «γεννά» τις τιμές δέλτα των κωδικοποιημένων μονάδων δέλτα. Οι συναρτήσεις αυτές φροντίζουν, επίσης, για την ανάθεση μοναδικών ID σε κάθε νέα πραγμάτωση κανονικότητας ή μονάδα δέλτα που συναντούν.

5.4 Παραγωγή του κώδικα SpMV

Ο απείρως μεγάλος και εκ των προτέρων άγνωστος αριθμός των πραγματώσεων των κανονικοτήτων ενός πίνακα υπαγορεύει την χρήση παραγωγής κώδικα στον χρόνο εκτέλεσης για την δημιουργία κώδικα SpMV συγκεκριμένου ανά κανονικότητα. Κατασκευάσαμε ένα νέο σύστημα συμπίλησης Just-In-Time (JIT) για το σχήμα CSX, βασισμένο στη Clang και στο LLVM [Lattner και Adve, 2004; Lattner, 2011]. Το LLVM είναι μία χαμηλού επιπέδου υποδομή για την δημιουργία συμπιλιτών, που παρέχει μία συλλογή σπονδυλωτών και επαναχρησιμοποιούμενων τεχνολογιών των απαραίτητων εργαλείων συμπίλησης. Η Clang αποτελεί το εμπρόσθιο τμήμα της υποδομής LLVM για την υποστήριξη γλωσσών προγραμματισμού, που ανήκουν στην οικογένεια της C. Ο βασικός στόχος της είναι η συντακτική ανάλυση τυπικού πηγαίου κώδικα C/C++, η μετατροπή του στην ενδιάμεση αναπαράσταση του LLVM (Intermediate Representation – IR) και η μεταφορά του στο οπίσθιο τμήμα του LLVM, που είναι υπεύθυνο για την παραγωγή του τελικού εντοπίου κώδικα. Το κυριώτερο πλεονέκτημα της Clang και του LLVM είναι ότι παρέχουν μία πολύ πλούσια προγραμματιστική διεπαφή, που επιτρέπει την δημιουργία αποδοτικού κώδικα JIT και την εύκολη ενσωμάτωσή του σε μία υπάρχουσα εφαρμογή.

```

1: function CONSTRUCTCSX(matrix::in)
   matrix: CSX's internal encoded matrix representation

2:   ctl  $\leftarrow \emptyset$  ▷ CSX's control byte sequence
3:   values  $\leftarrow \emptyset$  ▷ CSX's non-zero values
4:   stray  $\leftarrow \emptyset$  ▷ Stray elements, not in substructures
5:   foreach row in matrix do
6:     ctl  $\leftarrow$  ctl  $\oplus$  EMITNEWROW() ▷ Set new row bit
7:     foreach generic element e(i, j, v, t) in row do
8:       if e.t  $\neq$  NONE then
9:         EMITDELTAS(stray, ctl, values) ▷ Dump and empty
stray elements
10:        ctl  $\leftarrow$  ctl  $\oplus$  EMITSUBSTRUCTURE(e)
11:        values  $\leftarrow$  values  $\oplus$  EMITVALUES(e)
12:      else
13:        stray  $\leftarrow$  stray  $\cup$  e
14:      end for
15:      EMITDELTAS(stray, ctl, values) ▷ Dump and empty stray elements
16:    end for
17:    EMITDELTAS(stray, ctl, values) ▷ Dump and empty stray elements
18:    return (ctl, values)

```

Αλγόριθμος 5.4: Κατασκευή των δομών δεδομένων του CSX. Ο τελικός πίνακας κατασκευάζεται αθροιστικά κατά την διάσχιση της κωδικοποιημένης εσωτερικής αναπαράστασης. Οι συναρτήσεις EMIT*() είναι υπεύθυνες για την δημιουργία των συγκεκριμένων πεδίων της δομής *ctl* και των μη μηδενικών στοιχείων των κωδικοποιημένων κανονικοτήτων. Τα εναπομείναντα στοιχεία σχηματίζουν μονάδες δέλτα (βλ. Αλγόριθμο 5.5 για περισσότερες πληροφορίες επί της συνάρτησης EMITDELTAS()). Ο τελεστής \oplus υποδηλώνει την συνένωση ακολουθιών. Ο χειρισμός των κενών γραμμών δεν παρουσιάζεται για λόγους απλότητας.

Στο CSX, παράγουμε απλό κώδικα C για τις ανά κανονικότητα ρουτίνες του SpMV και χρησιμοποιούμε την Clang για την παραγωγή βελτιστοποιημένου ενδιάμεσου κώδικα LLVM. Η επιβάρυνση της συντακτικής ανάλυσης του προκύπτοντος αρχείου πηγαίου κώδικα (μία μονάδα συμπίλησης, λιγότερες των 150 γραμμών) και της παραγωγής της βελτιστοποιημένης ενδιάμεσης αναπαράστασης του LLVM είναι αμελητέα συγκριτικά με τον χρόνο προεπεξεργασίας του πίνακα, πράγμα που δεν δικαιολογεί τον επιπρόσθετο κόπο της διατήρησης μίας βάσης κώδικα στην ενδιάμεση αναπαράσταση του LLVM ή

```

1: procedure EMITDELTA(elems::inout, ctl::inout, values::inout)
   elems: set of generic elements
   ctl: CSX's ctl structure
   values: CSX's non-zero values

2:   if elems  $\neq$   $\emptyset$  then
3:     ctl  $\leftarrow$  ctl  $\oplus$  EMITDELTAUNIT(elems)
4:     values  $\leftarrow$  values  $\oplus$  EMITVALUES(elems)
5:     elems  $\leftarrow$   $\emptyset$ 

```

Αλγόριθμος 5.5: Βοηθητική συνάρτηση για την παραγωγή των μονάδων δέλτα.

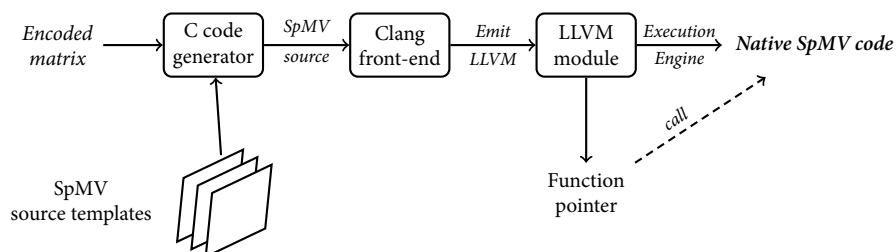
ακόμα και της χρήσης μίας κρυφής επί του δίσκου μνήμης για την αποθήκευση προσωμπλισμένων ρουτινών SpMV. Μία εναλλακτική στρατηγική θα ήταν η απευθείας παραγωγή του ενδιάμεσου κώδικα του LLVM, μέσω της αντίστοιχης διεπαφής. Ωστόσο, αυτό αποτελεί επίπονη και επιρρεπή σε λάθη διαδικασία (ακόμα και σε επίπεδο επίδοσης), καθώς όχι απαιτείται η καλή κατανόηση της ενδιάμεσης αναπαράστασης, αλλά και ο ίδιος ο κώδικας του SpMV «συσκοτίζεται» από τις σύνθετες κλήσεις που απαιτούνται για την κατασκευή της ενδιάμεσης αναπαράστασης.

Το Σχήμα 5.8 παρουσιάζει την οργάνωση της συμπίλησης JIT στο CSX. Το σχήμα CSX διατηρεί ένα κατάλογο προτύπων πηγαίου κώδικα σε C, τα οποία ορίζουν ένα σύνολο επωνύμων κενών θέσεων κειμένου (text hooks), που πρόκειται να συμπληρωθούν κατά την διάρκεια του χρόνου εκτέλεσης. Για κάθε τύπο κανονικότητας διατηρείται ένα πρότυπο πηγαίου κώδικα για τον υπολογισμό του SpMV εντός της κανονικότητας¹, ενώ υπάρχει και ένα «υπερκείμενο» πρότυπο για τον πλήρη υπολογισμό του πυρήνα SpMV με το σχήμα CSX. Μετά την επιλογή των προς κωδικοποίηση κανονικοτήτων και της δημιουργίας των αντιστοίχων δομών του CSX, επιλέγουμε τα κατάλληλα πρότυπα, παράγουμε τον αντίστοιχο κώδικα SpMV σε C και τον προωθούμε στο εμπρόσθιο τμήμα της Clang. Προγραμματίζουμε το τμήμα αυτό, ώστε να παραγάγει μία βελτιστοποιημένη μονάδα ενδιάμεσου κώδικα του LLVM (LLVM IR module), από το οποίο λαμβάνουμε ένα κοινό δείκτη στη συνάρτηση του πυρήνα SpMV και το δεσπόσιο μέρος του LLVM αναλαμβάνει την παραγωγή του τελικού εντοπίου κώδικα για το μηχάνημα εκτέλεσης.

Ο πυρήνας SpMV που εκτελούμε για το σχήμα CSX φαίνεται στον Αλγόριθμο 5.6. Καταρχήν, πρέπει να σημειώσουμε ότι στο CSX είναι απαραίτητος ο ρητός μηδενισμός των στοιχείων του διανύσματος εξόδου (γραμμές 4–5), ένα

¹ Για τις κανονικότητες τύπου μπλοκ διατηρούμε μόνο δύο γενικά πρότυπα: ένα για τα κατά γραμμές μπλοκ και ένα για τα κατά στήλες.

5. Το Σχήμα Αποθήκευσης COMPRESSED SPARSE EXTENDED



Σχήμα 5.8: Η υποδομή της Just-In-Time συμπίλησης του σχήματος CSX.

παράπλευρο πρόβλημα που προκαλείται από την χρήση μη οριζοντίων κανονικοτήτων. Ωστόσο, η επίδραση αυτής της πράξης είναι πολύ μικρή, ειδικά στις πολυνηματικές εκτελέσεις, όπου η αρχικοποίηση πραγματοποιείται παράλληλα. Ο αλγόριθμος διασχίζει την δομή `ct1` και αποκωδικοποιεί ένα πεδίο τη φορά. Εάν ανιχνευθεί νέα γραμμή, αποθηκεύουμε το υπολογισμένο σημειακό γινόμενο (yr) στο διάνυσμα εξόδου και μετατοπίζουμε την τρέχουσα θέση (y_{curr}) με τον κώδικα στο `__NEW_ROW_HOOK()`. Εάν ο πίνακας δεν περιέχει κενές γραμμές, ο παραγώμενος κώδικας απλά αυξάνει την y_{curr} κατά ένα. Σε αντίθετη περίπτωση, ελέγχεται το `bit r_jump` και εάν υπάρχει κενή γραμμή, αυξάνεται η y_{curr} κατά το πλήθος των κενών γραμμών που περιέχεται στον οικείο ακέραιο μεταβλητού μήκους. Ο αλγόριθμος συνεχίζει με την αποκωδικοποίηση του μεταβλητού μήκους ακεραίου, που περιέχει την διαφορική απόσταση της τρέχουσας στήλης από την προηγούμενή της (`DECODECOLUMN()`), και υπολογίζει τον ακριβή αριθμό της στήλης εκκίνησης (x_{curr}) της τρέχουσας μονάδας. Στην συνέχεια, ανακτά το ID της μονάδας και εκτελεί τον οικείο κώδικα του SpMV στο `__BODY_HOOK()`. Το `__BODY_HOOK()` αντικαθίσταται από μία εντολή `switch` της C με μεταβλητή επιλογής την `id`. Οι ρουτίνες SpMV για τις συγκεκριμένες μονάδες του CSX είναι υπεύθυνες για την σωστή μετατόπιση της τρέχουσας θέσης στη δομή `ct1` και της τρέχουσας στήλης, καθώς για την ανανέωση του τοπικού συσσωρευτή yr , εάν αυτό είναι αναγκαίο.

Εφόσον κωδικοποιούμε συγκεκριμένες πραγματώσεις κανονικοτήτων στο CSX, γνωρίζουμε κατά την διάρκεια της συμπίλησης της ακριβής απόσταση δ των μονοδιαστάτων κανονικοτήτων και τις ακριβείς διαστάσεις των μπλοκ των 2-Δ. Αυτό επιτρέπει την παραγωγή αποδοτικού κώδικα με σταθερές τιμές δ και διαστάσεων μπλοκ, εξισώνοντας έτσι το υπολογιστικό πλεονέκτημα των μεθόδων σταθερού μεγέθους μπλοκ, όπως είναι το σχήμα BCSR. Επιπλέον, εάν κωδικοποιείται μόνο μόνο μία πραγμάτωση κανονικότητας, τότε παραλείπουμε τελείως την εντολή `switch`.

```

1: procedure CSXSPMV(ctl::in, values::in, x::in, y::out)
2:    $y_{curr} \leftarrow y$  ▷ Current position in y vector
3:    $x_{curr} \leftarrow x$  ▷ Current position in x vector
4:   for  $i \leftarrow 0$  to  $N$  do ▷ We must zero-out the output vector
5:      $y_{curr}[i] \leftarrow 0$ 
6:   end for
7:    $yr \leftarrow 0$  ▷ Local accumulator
8:   repeat
9:      $flags \leftarrow *ctl$ 
10:     $size \leftarrow *(ctl + 1)$ 
11:     $ctl \leftarrow ctl + 2$ 
12:    if TESTBIT( $flags$ , 7) then ▷ Check if nr bit is set
13:       $*y_{curr} \leftarrow *y_{curr} + yr$ 
14:       $yr \leftarrow 0$ 
15:      __NEW_ROW_HOOK() ▷ Advances  $y_{curr}$ 
16:       $x_{curr} \leftarrow x$ 
17:       $x_{curr} \leftarrow x_{curr} + \text{DECODECOLUMN}(ctl)$ 
18:       $id \leftarrow \text{GETID}(flags)$  ▷ Retrieve the ID of the next unit
19:      __BODY_HOOK() ▷ Unit-specific SpMV code
20:    until  $ctl$  ends

```

Αλγόριθμος 5.6: Το πρότυπο εκτέλεσης του πυρήνα SpMV με το σχήμα CSX. Τα «hooks» συμπληρώνονται κατά τον χρόνο εκτέλεσης. Ο τελεστής «*» είναι τελεστής απο-αναφοράς (dereference).

Παραλληλοποίηση

Το CSX ακολουθεί το σχήμα παραλληλοποίησης τυ CSR. Κατά την κατασκευή της εσωτερικής αναπαράστασης του CSX, χωρίζουμε τον πίνακα εισόδου κατά γραμμές με περίπου τον ίδιο αριθμό μη μηδενικών στοιχείων ανά διαμέριση. Από το σημείο αυτό και έπειτα, οι φάσεις της ανίχνευσης και κωδικοποίησης τους CSX προχωρούν ανεξάρτητα, παράγοντας διαφορετικούς υποπίνακες CSX ανά διαμέριση. Ο πυρήνας SpMV στον Αλγόριθμο 5.6 αλλάζει μόνο στη γραμμή 2, όπου αρχικοποιούμε την y_{curr} στην αρχή της κάθε διαμέρισης και στη γραμμή 4, όπου η διάσχιση συμβαίνει μόνο εντός των ορίων της αντίστοιχης διαμέρισης.

5.5 Αντιμετώπιση του κόστους προεπεξεργασίας

Ο χρόνος προεπεξεργασίας του CSX οριοθετείται από τις πολλαπλές κλήσεις στην συνάρτηση LEXSORT(), η οποία ταξινομεί λεξικογραφικά τα μετασημα-

τισμένα στοιχεία του πίνακα, κατά την διάρκεια της ανίχνευσης των κανονικοτήτων (Αλγόριθμος 5.3, γραμμές 4–9). Αντιμετωπίζουμε το θέμα αυτό με ένα συνδυασμό μερικής ταξινόμησης και δειγματισμού. Πιο συγκεκριμένα, αντί να διατρέχουμε ολόκληρο τον πίνακα, των χωρίζουμε σε σταθερού μεγέθους παράθυρα προεπεξεργασίας, βασιζόμενα στο πλήθος των μη μηδενικών στοιχείων, και διατρέχουμε κάθε παράθυρο ξεχωριστά. Η τροποποίηση αυτή μειώνει αυτομάτως την ασυμπτωτική πολυπλοκότητα της φάσης ανίχνευσης από $\Theta(NNZ \lg NNZ)$ σε $\Theta(NNZ)$, εις βάρος των κανονικοτήτων που διασχίζουν τα όρια των παραθύρων, οι οποίες δεν ανιχνεύονται. Για την περαιτέρω μείωση του κόστους προεπεξεργασίας, εξετάζουμε μόνο ένα συγκεκριμένο αριθμό παραθύρων, ομοιόμορφα κατανεμημένων σε ολόκληρο τον πίνακα, καλύπτοντας ένα μικρό υποσύνολο των μη μηδενικών στοιχείων του πίνακα. Στα πειράματά μας, με δειγματισμό περίπου 1% των μη μηδενικών στοιχείων του πίνακα με χρήση 48 δειγμάτων, ο πυρήνας SpMV πέτυχε την ίδια επίδοση με την περίπτωση της πλήρους προεπεξεργασίας, αλλά σε σχεδόν μία τάξη μεγέθους χαμηλότερο χρόνο προεπεξεργασίας.

Έχοντας ελαχιστοποιήσει την φάση ανίχνευσης των κανονικοτήτων, ο χρόνος προεπεξεργασίας οριοθετείται πλέον από την φάση της τελικής κωδικοποίησης του πίνακα (Αλγόριθμος 5.3, γραμμές 13–15), ή οποία παραμένει της τάξης του $\Theta(NNZ \lg NNZ)$. Στη φάση αυτή, δυστυχώς, ο διαχωρισμός και, ακόμα περισσότερο, ο δειγματισμός του πίνακα δεν μπορούν να εφαρμοστούν, καθώς, σε κάθε περίπτωση, θα πρέπει να κωδικοποιηθεί ολόκληρος ο πίνακας. Ωστόσο, η επίπτωση της φάσης αυτή στο συνολικό κόστος προεπεξεργασίας δεν είναι τόσο σημαντικοί για δύο κυρίως λόγους: στις περισσότερες των περιπτώσεων, μόνο ένας μικρός αριθμός τύπων κανονικοτήτων κωδικοποιείται εν τέλει, και το πλήρες κόστος της ταξινόμησης θα «πληρωθεί» μόνο κατά την κωδικοποίηση του πρώτου τύπου κανονικότητας, καθώς σε κάθε επόμενο ταξινομούνται μόνο τα μη κωδικοποιημένα στοιχεία.

Σε συνέχεια της χρήσης του δειγματισμού και των παραθύρων προεπεξεργασίας, μειώνουμε επιπλέον το κόστος προεπεξεργασίας παραλληλοποιώντας το σύνολο της διαδικασίας. Η παραλληλοποίηση είναι απλή: μετά την φόρτωση του αρχικού πίνακα CSR, δημιουργούμε ένα νέο νήμα για κάθε διαμέριση, το οποίο προχωρά ανεξάρτητα με την ανίχνευση και την κατασκευή του τελικού πίνακα CSX.

Τέλος, κατά την διάρκεια της προεπεξεργασίας, λαμβάνουμε ιδιαίτερη μέριμνα στην διαχείριση της μνήμης, ώστε να αποφεύγουμε πλήρως επανακαταχωρήσεις μνήμης (reallocations), οι οποίες δύνανται να οδηγήσουν σε αντιγραφές πολύ μεγάλων περιοχών μνήμης. Για τον λόγο αυτό, προσπαθούμε είτε να συμπεράνουμε το μέγεθος κάθε καταχώρησης μνήμης για οποιαδήποτε δομή δεδομένων εξ αρχής είτε, στην περίπτωση που αυτό είναι αδύνατο, να είμαστε αρκετά γενναιόδωροι με την αρχική καταχώρηση, από την οποία απο-

δεσμεύουμε στην συνέχεια τον περιττό χώρο. Η βελτιστοποίηση αυτή οδήγησε σε σημαντική επιτάχυνση της φάσης προεπεξεργασίας, της τάξης του $2 \times$ με $3 \times$. Το συνολικό κόστος προεπεξεργασίας του CSX για την ανίχνευση όλων των υποστηριζόμενων τύπων κανονικοτήτων κυμαίνεται στην τάξη των 100 σειριακών εκτελέσεων του SpMV με το σχήμα CSR, καθιστώντας το βιώσιμο ακόμα και για online επεξεργασία του πίνακα εισόδου.

5.6 Μεταφορά σε αρχιτεκτονικές NUMA

Όπως συζητήθηκε στο Κεφάλαιο 3, ο βασικός παράγοντας για την επίτευξη υψηλής επίδοσης του πυρήνα SpMV στις αρχιτεκτονικές NUMA είναι η σωστή τοποθέτηση των δεδομένων του στους κόμβους μνήμης του συστήματος. Τα δεδομένα που κάθε νήμα προσπελαύνει θα πρέπει να βρίσκονται στον τοπικό του κόμβο, ώστε να αποφευχθεί ο κορεσμός του δικτύου διασύνδεσης και, επίσης, η αυξημένα καθυστέρηση που προκαλείται από τα πολλαπλά βήματα μίας απομακρυσμένης πρόσβασης. Ένα νήμα του πολυνηματικού πυρήνα SpMV προσπελαύνει τα δεδομένα της δικής του διαμέρισης του πίνακα, τα αντίστοιχα τμήματα του διανύσματος εξόδου και αυθαίρετα τμήματα του διανύσματος εισόδου. Η σωστή τοποθέτηση των διαμερίσεων του CSX είναι πολύ απλή: εφόσον η κατασκευή του πίνακα CSX λαμβάνει χώρα ανεξάρτητα σε κάθε διαμέριση, αρκεί να διασφαλίσουμε ότι οι δομές δεδομένων του CSX θα καταχωρηθούν στο σωστό κόμβο, κάνοντας χρήση κατά-NUMA εκχωρητών μνήμης, π.χ., με κλήσεις στη `numa_allloc_onnode()` της βιβλιοθήκης `numactl` του Linux. Για το διάνυσμα εξόδου, χρησιμοποιούμε τον χαμηλού επιπέδου παρεμβλητικό κατά-NUMA εκχωρητή που περιγράψαμε στο Κεφάλαιο 3, με σκοπό να τοποθετήσουμε σωστά τα διαφορετικά τμήματα του διανύσματος εξόδου. Τέλος, χάρη της πειραματικής αξιολόγησής μας, τοποθετούμε ένα αντίγραφο του διανύσματος εισόδου σε κάθε κόμβο μνήμης. Η διάταξη αυτή παρέχει ελαφρώς καλύτερη ισορροπία των προσβάσεων στην κύρια μνήμη και εκθέτει περισσότερο το υπολογιστικό μέρος του πυρήνα. Παρόλο που σε πρακτικές εφαρμογές μία τέτοια επιλογή δεν είναι λογική, η επίδραση της χρήσης ενός κοινού αντιγράφου του διανύσματος εισόδου στην επίδοση του πυρήνα είναι κατά μέσο όρο αμελητέα (βλ. Κεφάλαιο 3, Σχήμα 3.13).

5.6.1 Βελτιστοποίηση των υπολογισμών

Οι σύγχρονες αρχιτεκτονικές NUMA έχουν μία σημαντική παρενέργεια στην εκτέλεση του πυρήνα SpMV: το αυξημένο εύρος ζώνης μνήμης κατά την πρόσβαση στον τοπικό κόμβο μνήμης εκθέτει περισσότερο το υπολογιστικό μέρος του πυρήνα, το οποίο πλέον δεν είναι αμελητέο. Ο CSX εκτελεί κάποιες αρκετά βαρείς υπολογισμούς αποσυμπίεσης κατά την αποκωδικοποίηση των ακεραίων

ων μεταβλητού μήκους (βλ. Σχήμα 5.4), που χρησιμοποιούνται για την αποθήκευση των διαφορικών αποστάσεων των στηλών και του πλήθους των κενών γραμμών. Η περίπτωση των κενών γραμμών δεν είναι ιδιαίτερης ανησυχίας, καθώς δεν παράγουμε καθόλου τον κώδικα αποκωδικοποίησης για πίνακες χωρίς κενές γραμμές, ενώ, σε περιπτώσεις που υπάρχουν, η αποκωδικοποίησή τους δεν βρίσκεται στο κρίσιμο μονοπάτι της εκτέλεσης του πυρήνα. Αυτό που παραμένει στο κρίσιμο μονοπάτι, ωστόσο, είναι η αποκωδικοποίηση των διαφορικών αποστάσεων των στηλών πριν από τον υπολογισμό του SpMV για κάθε κανονικότητα (Αλγόριθμος 5.6, γραμμή 17). Η αποκωδικοποίηση είναι σχεδόν μηδενικού κόστους όταν η διαφορική απόσταση είναι μικρότερη του 128, αλλά χρησιμοποιούνται απαιτητικές πράξεις bit, εάν είναι μεγαλύτερη. Δυστυχώς, η κωδικοποίηση κανονικοτήτων (και όχι μονάδων δέλτα²) οδηγεί στην σημαντική αύξηση του αριθμού των αποτελουμένων από πολλά byte διαφορικών αποστάσεων, των οποίων η αποκωδικοποίηση μπορεί να επηρεάσει αρνητικά την συνολική επίδοση του SpMV. Για τον λόγο αυτό, αντικαθιστούμε την διαφορική απόσταση των στηλών με τον πλήρη αριθμό της αρχικής στήλης κάθε κανονικότητας, αποθηκευμένης σε ένα τύπικό ακέραιο 32-bit. Η βελτιστοποίηση αυτή μειώνει τον λόγο συμπίεσης 2–3%, αλλά το όφελος στην επίδοση μπορεί να ξεπεράσει το 15% σε ορισμένους πίνακες.

Η αύξηση των μονάδων δέλτα κατά την κωδικοποίηση πολλαπλών κανονικοτήτων μπορεί να υπερκεράσει το όποιο όφελος από την συμπίεση του πίνακα στις αρχιτεκτονικές NUMA, λόγω του αυξημένου κόστους της εντολής switch (αδυναμία πρόβλεψης της διακλάδωσης). Για τον λόγο αυτό, αναθεωρούμε ως ακολούθως την ευρετική συνάρτηση κέρδους (σχέση (5.2)) για τις αρχιτεκτονικές NUMA, ώστε να συμπεριλάβει και μία εκτίμηση του συνολικού αριθμού των εντολών switch που εκτελούνται:

$$G = NNZ_{enc} - N_{units} - N_{switch} \quad (5.3)$$

όπου $N_{switch} = N_{units} + N_{deltas}$ και N_{deltas} το συνολικό πλήθος των μονάδων δέλτα. Η ευρετική αυτή εξισορροπεί καλύτερα την διεκυστίδα μεταξύ των μερών του πυρήνα των προσβάσεων στη μνήμη και των υπολογισμών, τιμώντας τις κωδικοποιήσεις που οδηγούν σε μεγάλο υπολογιστικό κόστος. Με την αναθεωρημένη ευρετική συνάρτηση, παρατηρήσαμε βελτίωση της επίδοσης πλέον του 20% σε ορισμένους πίνακες στις κατά-NUMA πολυνηματικές εκτελέσεις.

² Η περίπτωση των μονάδων δέλτα δεν είναι κρίσιμη, διότι κωδικοποιούμε ολόκληρη κάθε γραμμή του πίνακα ως μία μοναδική μονάδα δέλτα.

5.7 Αξιολόγηση της επίδοσης του CSX

Αξιολογούμε την επίδοση του σχήματος αποθήκευσης CSX χρησιμοποιώντας την σουίτα πινάκων και τα υπολογιστικά συστήματα που παρουσιάστηκαν με λεπτομέρεια στο Κεφάλαιο 3, Ενότητα 3.2. Ως εναλλακτικά σχήματα αποθήκευσης, εκτός του CSR, χρησιμοποιούμε τα BCSR και VBL, καθώς αυτά αποτελούν τα πιο εδραιωμένα παραδείγματα εναλλακτικών του CSR σχημάτων και τους καλύτερους αντιπροσώπους των σχημάτων αποθήκευσης σε μπλοκ σταθερού και μεταβλητού μεγέθους, αντιστοίχως. Υλοποιήσαμε βελτιστοποιημένες εκδόσεις για όλα αυτά τα σχήματα, συμπεριλαμβανομένων κατά-NUMA υλοποιήσεων. Συγκεκριμένα για το σχήμα BCSR, υλοποιήσαμε βελτιστοποιημένες ρουτίνες του SpMV για κάθε ξεχωριστό μπλοκ (μονοδιάστατο ή διδιάστατο) μέχρι μεγέθους οκτώ, συμπεριλαμβανομένου και του μπλοκ 3×3 . Τα εκτιθέμενα αποτελέσματα για το σχήμα BCSR αντιστοιχούν στο μπλοκ με την καλύτερη επίδοση, το οποίο επελέγη ύστερα από εξαντλητική αναζήτηση και τον 20 υπό εξέταση μπλοκ. Στην πράξη, όπου το πιθανότερο είναι να χρησιμοποιείται κάποια ευρετική συνάρτηση (βλ. Κεφάλαιο 4, Ενότητα 4.3), η πραγματική επίδοση του BCSR είναι πιθανό να είναι χαμηλότερη. Για την παραλληλοποίηση του πυρήνα SpMV, χρησιμοποιούμε ένα στατικό σχήμα διαμέρισης του πίνακα κατά γραμμές βάσει των μη μηδενικών του στοιχείων. Συγκεκριμένα για το σχήμα BCSR, ο διαχωρισμός του πίνακα γίνεται μετά την κατασκευή του σχήματος, ώστε να επιτευχθεί καλύτερη εξισορρόπηση του φόρτου εργασίας. Χρησιμοποιούμε ακεραίους 32-bit για τις δομές δεικτοδότησης και αριθμούς κινητής υποδιαστολής διπλής ακρίβειας για τα μη μηδενικά στοιχεία του πίνακα. Στην περίπτωση του VBL, χρησιμοποιούμε ένα byte για τα πεδία του μήκους των μπλοκ.

Τέλος, χρησιμοποιήσαμε το LLVM 2.9 για την συμπύληση των ρουτινών του SpMV για όλα τα υπό εξέταση σχήματα αποθήκευσης, ώστε να επιτύχουμε μία δίκαιη σύγκριση. Θα πρέπει να σημειώσουμε εδώ, ότι, παρά την περί του αντιθέτου προσδοκία μας, το LLVM 2.9 πέτυχε 5% καλύτερη επίδοση σε σχέση με τον GCC 4.5 για όλα τα εναλλακτικά θεωρούμενα σχήματα αποθήκευσης³. Για την παραλληλοποίηση τόσο του SpMV όσο και της φάσης προεπεξεργασίας του CSX, χρησιμοποιήσαμε την βιβλιοθήκη Pthreads library (NPTL 2.7) και αναθέσαμε τα νήματα σε συγκεκριμένους λογικούς επεξεργαστές χρησιμοποιώντας τις κατάλληλες κλήσεις συστήματος του Linux. Για την ανάθεση των νημάτων σε λογικούς επεξεργαστές ακολουθούμε συστηματικά την πολιτική «share-all» (βλ. Κεφάλαιο 3, Ενότητα 3.2.3 για περισσότερες πληροφορίες επί εναλλακτικών πολιτικών πλήρωσης των πυρήνων).

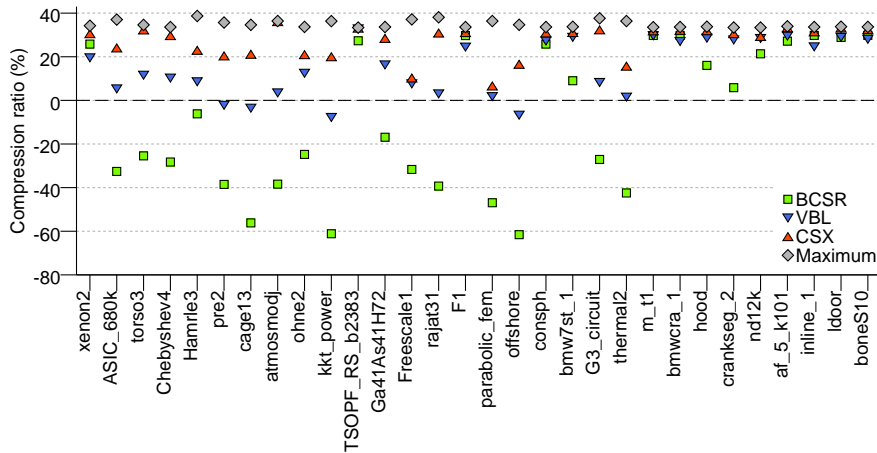
³ Το σχήμα CSX χρησιμοποιεί εξ ορισμού το LLVM.

5.7.1 Δυνατότητες συμπίεσης του CSX

Η δυνατότητα συμπίεσης του CSX εννοείται όχι μόνο από την ικανότητά του να ανιχνεύει πολλούς τύπους κανονικοτήτων, επιτρέποντάς του να διατηρεί μία μόνο στήλη ανά κανονικότητα, αλλά και από την ιδιαίτερα συμπαγή αναπαράσταση των μετα-δεδομένων των μη μηδενικών στοιχείων. Η πληροφορία των στηλών είναι περαιτέρω συμπυκνωμένη με την χρήση της διαφορικής δεικτοδότησης και των ακεραίων μεταβλητού μήκους, ενώ η πληροφορία των γραμμών αναπαρίσταται πλέον με ένα μόνο bit ανά γραμμή. Αυτό επιτρέπει στο σχήμα CSX να συμπιέζει την αναπαράσταση ακόμα και πινάκων με ακανόνιστη διάταξη των στοιχείων τους, όπου καμία κανονικότητα δεν κωδικοποιείται. Αυτό είναι ένα σημαντικό πλεονέκτημα του CSX έναντι των λοιπών εναλλακτικών σχημάτων αποθήκευσης, και ειδικότερα στα συστήματα SMP.

Το Σχήμα 5.9 παρουσιάζει τον μέγιστο βαθμό συμπίεσης που πέτυχε το σχήμα CSX σε σύγκριση με τα σχήματα VBL και BCSR. Παρουσιάζεται, επίσης, και ο μέγιστος βαθμός συμπίεσης (υποθέτωντας αποθήκευση μόνο των μη μηδενικών στοιχείων). Το σχήμα CSX επιτυγχάνει πάντοτε τον καλύτερο βαθμό συμπίεσης ανάμεσα στα υπό εξέταση σχήματα αποθήκευσης, ενώ η ικανότητά του να ανιχνεύει και να κωδικοποιεί μία πληθώρα διαφορετικών κανονικοτήτων του επιτρέπει να προσαρμόζεται με επιτυχία στις ιδιαιτερότητες κάθε πίνακα. Για παράδειγμα, είναι ικανό να ξεπεράσει τον βαθμό συμπίεσης του BCSR σε πίνακες με πολλά πυκνά μπλοκ, όπως είναι οι `xenon2`, `TSOPF_RS_b2383`, `consph`, `m_t1` κ.ά., και τον βαθμό συμπίεσης του VBL σε πίνακες με λιγότερο κανονική διάταξη, όπως είναι οι `Freescale1` και `offshore`. Σε πίνακες που κυριαρχούνται από στοιχεία σε διαγώνια διάταξη, π.χ., `torso3`, `cage13`, `atmosmodj` κ.ά., η υπεροχή του CSX είναι αδιαμφισβήτητη: επιτυγχάνει βαθμό συμπίεσης μεγαλύτερο του 20%, την στιγμή που το σχήμα BCSR αυξάνει το μέγεθος της αναπαράστασης και το VBL παραμένει πολύ κοντά στο μέγεθος του αρχικού CSR. Το πιο χαρακτηριστικό παράδειγμα είναι ο πίνακας `atmosmodj`, όπου το 99.8% των στοιχείων του κωδικοποιείται σε διαγώνιες κανονικότητες από το CSX, αγγίζοντας την μέγιστη δυνατή συμπίεση (36.4%)⁴. Την ίδια στιγμή, το σχήμα VBL επιτυγχάνει μία ελάχιστη συμπίεση της τάξης του 3.4%, ενώ το σχήμα BCSR αυξάνει το μέγεθος της αναπαράστασης περίπου 60%!

⁴ Η ακριβής συμπίεση του CSX είναι 36.1%.



Σχήμα 5.9: Η δυνατότητα συμπίεσης του CSX. Το σχήμα CSX καταφέρνει να προσαρμοστεί επιτυχώς στην δομή του πίνακα και προσφέρει τον καλύτερο βαθμό συμπίεσης ανάμεσα στα εναλλακτικά υπό θεώρηση σχήματα αποθήκευσης. Ο μέγιστος βαθμός συμπίεσης υπολογίζεται θεωρώντας την αποθήκευση μόνο των μη μηδενικών στοιχείων, δηλαδή ένα μέγεθος αναπαράστασης των 8NNZ bytes (αριθμοί κινητής υποδιαστολής διπλής ακρίβειας).

5.7.2 Η επίδοση του CSX

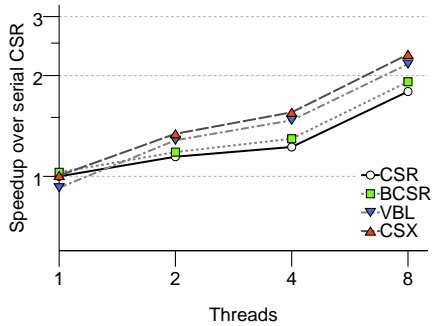
Αρχιτεκτονικές SMP

Οι μεγάλες δυνατότητες συμπίεσης του CSX τού παρέχουν μία μεγάλη δυναμική επίδοσης σε πίνακες, προερχομένους από προβλήματα ΜΔΕ, όπου το κυριώτερο πρόβλημα επίδοσης, ειδικά στα συστήματα SMP, είναι η έλλειψη επαρκούς εύρους ζώνης μνήμης. Το Σχήμα 5.10 παρουσιάζει την επιτάχυνση της επίδοσης που πέτυχαν τα υπό θεώρηση σχήματα αποθήκευσης στα δύο συστήματα συμμετρικής πρόσβασης στη μνήμη, Harpertown και Dunnington. Η επίδραση της συμπίεσης κυριαρχεί σε αυτές τις αρχιτεκτονικές, ειδικότερα στις πολυνηματικές εκτελέσεις. Τα σχήματα CSX και VBL ηγούνται της επίδοσης στο σύστημα Harpertown με μία βελτίωση της επίδοσης επί του CSR της τάξης του 26.4% και 18.5% κατά μέσο όρο στην οκτώ-νηματική εκτέλεση, ενώ το BCSR κερδίζει ένα ελάχιστο 4.1%. Αντίστοιχη είναι η εικόνα και στη περίπτωση της 24-νηματικής εκτέλεσης στο Dunnington: το σχήμα CSX προσφέρει 61% βελτίωση της επίδοσης επί τους CSR κατά μέσο όρο, το VBL ακολουθεί αρκετά πιο πίσω με μία 28.8% βελτίωση, ενώ το BCSR καταφέρνει να προσφέρει βελτίωση της τάξης του 6.3% μόνο. Το σχήμα BCSR υποφέρει από την εγγενώς χαμηλή ικανότητά του συμπίεσης και από την χρήση επιπροσθέτων μηδενι-

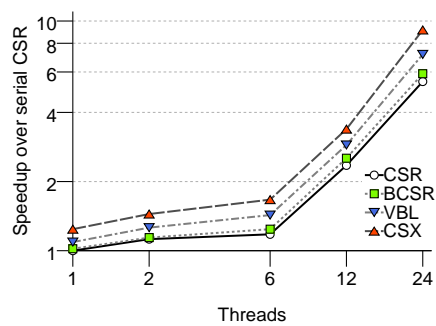
κών στοιχείων για την κατασκευή πλήρων μπλοκ. Μπορεί να χαρακτηριστεί ως ένα σχήμα αποθήκευσης «όλα-ή-τίποτα», καθώς στους μισούς πίνακες της σουίτας μας, μείωσε την επίδοση του SpMV περισσότερο από 30% κατά μέσο όρο, ενώ στους υπολοίπους προσέφερε μία βελτίωση της επίδοσης περισσότερη του 25%. Η υψηλότερη ικανότητα συμπίεσης των VBL και CSX φαίνεται καθαρά στα διαγράμματα επιτάχυνσης της επίδοσης στις αρχιτεκτονικές SMP. Η ικανότητα του CSX να ανιχνεύει και να κωδικοποιεί πολλούς τύπους κανονικοτήτων, και ειδικότερα μπλοκ, αποτελεί ένα σημαντικό πλεονέκτημα του CSX έναντι του VBL, όχι μόνο υπό όρους απόλυτης συμπίεσης, αλλά και υπό όρους υπολογιστικής επίδοσης του SpMV, καθώς το σχήμα CSX διατηρεί το υπολογιστικό πλεονέκτημα των μπλοκ σταθερού μεγέθους του BCSR, χάρη στην παραγωγή εξειδικευμένου κώδικα στον χρόνο εκτέλεσης, χωρίς την επιβάρυνση της χρήσης επιπροσθέτων μηδενικών στοιχείων. Πράγματι, όπως το σχήμα BCSR, έτσι και το CSX, επιτυγχάνει αντίστοιχη με το CSR επίδοση στην σειριακή εκτέλεση στο Harpertown, παρά την πρόσθετη επιβάρυνση του μηδενισμού των στοιχείων του διανύσματος εξόδου σε κάθε κλήση του πυρήνα. Στο σύστημα Dunnington, το σχήμα CSX ξεκινά με ένα σημαντικό πλεονέκτημα επίδοσης επί του CSR της τάξης του 16.1% ήδη από την σειριακή εκτέλεση. Θα πρέπει να σχολιάσουμε εδώ την απότομη αύξηση της επίδοσης στο σύστημα Dunnington στα 12 και 24 νήματα· αυτό οφείλεται στην γραμμική αύξηση των διαθέσιμων πόρων εύρους ζώνης μήμης καθώς προσθέτουμε περισσότερους επεξεργαστές (π.χ., ελεγκτές εμπροσθίου διαύλου) στον υπολογισμό του SpMV, λόγω της στρατηγικής «share-all» πλήρωσης των πυρήνων (βλ. Κεφάλαιο 3, Ενότητα 3.3.2).

Η χρησιμοποίηση της πολιτικής πλήρωσης πυρήνων «share-nothing», η οποία επιτρέπει την εξ αρχής αξιοποίηση του μεγίστου εύρους ζώνης μήμης (Σχήμα 5.11), δεν αλλάζει την γενική εικόνα· η κυριαρχία του CSX στα συστήματα SMP είναι καθαρή. Επιτυγχάνει 40% βελτίωση της επίδοσης επί του CSR κατά μέσο όρο στην 12-νηματική εκτέλεση, ενώ το VBL βρίσκεται στο 24% και το BCSR σχεδόν στο 5%. Ένα σημαντικό χαρακτηριστικό του CSX στο σύστημα Dunnington είναι συνεχίζει να βελτιώνει την επίδοσή του καθώς αυξάνεται το πλήθος των νημάτων, την στιγμή που όλα τα υπόλοιπα σχήματα αντιμετωπίζουν μία κάμψη της επίδοσής τους, που φθάνει το 4% στην περίπτωση του CSR. Στο σύστημα Harpertown, ο κοινός δίαυλος έχει ήδη κορεστεί από την τετρα-νηματική εκτέλεση, ενώ στο Dunnington η επιτάχυνση σταματά στη 12-νηματική εκτέλεση. Η σημαντική επιτάχυνση που εμφανίζουν όλα τα σχήματα στην εξα-νηματική και στην 12-νηματική εκτέλεση στο σύστημα Dunnington οφείλεται στην πολύ μεγάλη, 64 MiB, διαθέσιμη συνολική κρυφή μνήμη L3. Έξι πίνακες της σουίτας μας χωρούν άνετα στην διαθέσιμη κρυφή μνήμη ήδη στη μορφή CSR, ενώ τρεις επιπλέον είναι ελάχιστα μεγαλύτεροι. Το σύστημα, τέλος, κορέννεται πλήρως στην 24-νηματική εκτέλεση, όπου όλα τα σχήματα,

5.7. Αξιολόγηση της επίδοσης του CSX

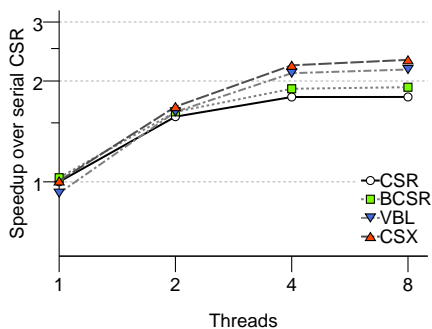


(i) Harpertown.

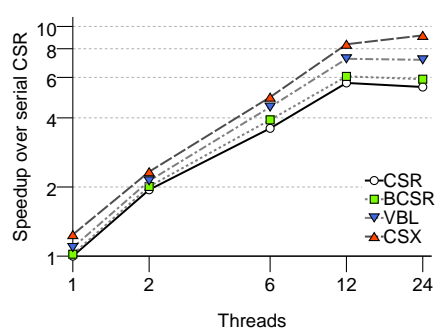


(ii) Dunnington.

Σχήμα 5.10: Επιτάχυνση του CSX σε συστήματα SMP. Οι υψηλές δυνατότητες συμπίεσης του CSX τού επιτρέπουν την σημαντική βελτίωση της επίδοσης του SpMV, υπερβαίνοντας όλα τα άλλα εναλλακτικά του CSR σχήματα. Παρουσιάζεται η μέση επιτάχυνση, χρησιμοποιώντας την πολιτική πλήρωσης των πυρήνων «share-all».



(i) Harpertown.



(ii) Dunnington.

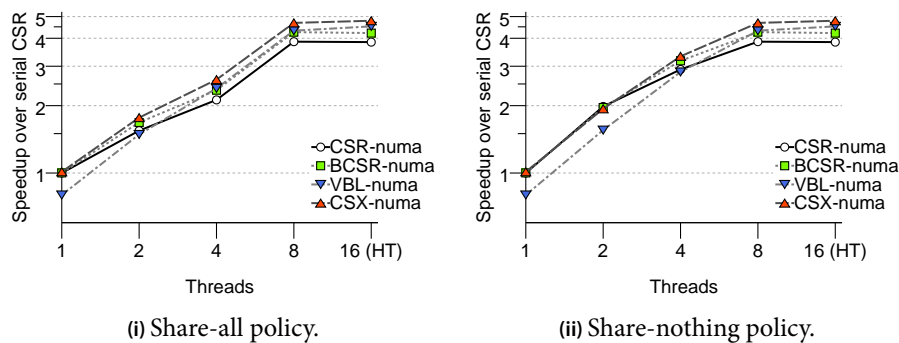
Σχήμα 5.11: Η επιτάχυνση του CSX σε συστήματα SMP χρησιμοποιώντας την πολιτική πλήρωσης πυρήνων «share-nothing», με σκοπό την εξαρχής χρησιμοποίηση των μεγίστων δυνατών πόρων εύρους ζώνης μνήμης. Παρουσιάζεται η μέση επιτάχυνση.

πλην του CSX, αντιμετωπίζουν μία μείωση της επίδοσής τους.

Αρχιτεκτονικές NUMA

Στις αρχιτεκτονικές NUMA, όπου το διαθέσιμο εύρος ζώνης μνήμης είναι σημαντικά υψηλότερο, το τοπίο της επίδοσης αλλάζει, αλλά το σχήμα CSX πα-

5. Το Σχήμα Αποθήκευσης COMPRESSED SPARSE EXTENDED



Σχήμα 5.12: Επιτάχυνση του CSX στις αρχιτεκτονικές NUMA. Παρουσιάζοντας μέσοι όροι χρησιμοποιώντας τις δύο πολιτικές πλήρωσης πυρήνων. Ακόμα και με την πολιτική «share-nothing», το σχήμα CSX καταφέρνει να παραμείνει πολύ κοντά στα BCSR και CSR μέχρι την δι-νηματική εκτέλεση, προτού περάσει επικεφαλής στις εκτελέσεις με περισσότερα νήματα.

ραμένει αυτό με την καλύτερη επίδοση σε όλες τις πολυνηματικές εκτελέσεις. Το Σχήμα 5.12i παρουσιάζει την επιτάχυνση της επίδοσης του SpMV για τα θεωρούμενα σχήματα αποθήκευσης στο σύστημα Gainestown χρησιμοποιώντας κατά-NUMA τοποθέτηση των δεδομένων τους. Η πρώτη βασική παρατήρηση είναι ότι το σχήμα CSR είναι πλέον ιδιαίτερα ανταγωνιστικό, καθώς η συμφόρηση στην κύρια μνήμη δεν είναι τόσο έντονη όσο προηγουμένως μέχρι την 16-νηματική εκτέλεση, όπου ενεργοποιείται το χαρακτηριστικό HyperThreading. Το σχήμα CSX επιτυγχάνει 17.5% βελτίωση της επίδοσης επί του CSR, όταν χρησιμοποιούνται οκτώ νήματα και αυξάνει την απόσταση στο 20.7% στην 16-νηματική εκτέλεση, όπου η συμφόρηση για τους κοινούς πόρους γίνεται εμφανής. Οι αντίστοιχες τιμές για το VBL είναι 8.4% και 13.9% για τις προαναφερθείσες εκτελέσεις. Η περίπτωση του BCSR είναι ενδιαφέρουσα, καθώς επιτυγχάνει επίδοση αντίστοιχη του VBL, κυρίως λόγω του επαρκούς εύρους ζώνης μνήμης, το οποίο εκθέτει καλύτερα το υπολογιστικό του πλεονέκτημα. Ενώ η επίδοση του VBL είναι χαμηλότερη του CSR μέχρι την δι-νηματική εκτέλεση, φθάνοντας μάλιστα σε μία μείωση της τάξης του 23.5% στην σειριακή εκτέλεση, το CSX αντιμετωπίζει μία ελάχιστη πτώση της επίδοσης της τάξης του 2.8% μόνο στην σειριακή εκτέλεση. Από την δι-νηματική εκτέλεση, ωστόσο, το σχήμα CSX είναι ήδη επικεφαλής των υπολοίπων σχημάτων με 11.4% βελτίωση επί του CSR, η οποία φθάνει στο 20.7% όταν το σύστημα χρησιμοποιείται πλήρως.

Για να εξετάσουμε την συμπεριφορά του σχήματος CSX σε εκτελέσεις, όπου το εύρος ζώνης μνήμης δεν αποτελεί πρόβλημα, εκτελέσαμε πειράματα στο

σύστημα Gainestown χρησιμοποιώντας την πολιτική πλήρωσης πυρήνων «share-nothing» (βλ. Σχήμα 5.12ii). Στις εκτελέσεις αυτές, το αυξημένο εύρος ζώνης μνήμης μέχρι και τα τέσσερα νήματα ευνοεί τα απλούστερα σχήματα CSR και BCSR, τα οποία επιτυγχάνουν αντίστοιχες επιδόσεις. Το σχήμα VBL πέφτει θύμα της αυξημένης υπολογιστικής επιβάρυνσης που δημιουργούν οι πρόσθετες δομές δεδομένων του, η οποία δεν υπερβαίνεται παρά από την οκτω-νηματική εκτέλεση. Στον αντίποδα, η επίδοση του CSX παραμένει πολύ κοντά στην επίδοση των CSR και BCSR στις δύο πρώτες πολυνηματικές εκτελέσεις, παρά το αυξημένο κόστος διατήρησης των δομών του και αποσυμπίεσης, αντιμετωπίζοντας μία ελάχιστη μείωση της επίδοσης στην δι-νηματική εκτέλεση της τάξης του 3.6%. Από την τετρα-νηματική εκτέλεση, ωστόσο, το CSX αρχίζει να κτίζει μία σημαντική διαφορά της τάξης του 11.3% επί του CSR, η οποία διευρύνεται στο 17.5% και 20.7% στην οκτω-νηματική και στην 16-νηματική εκτέλεση, αντιστοίχως.

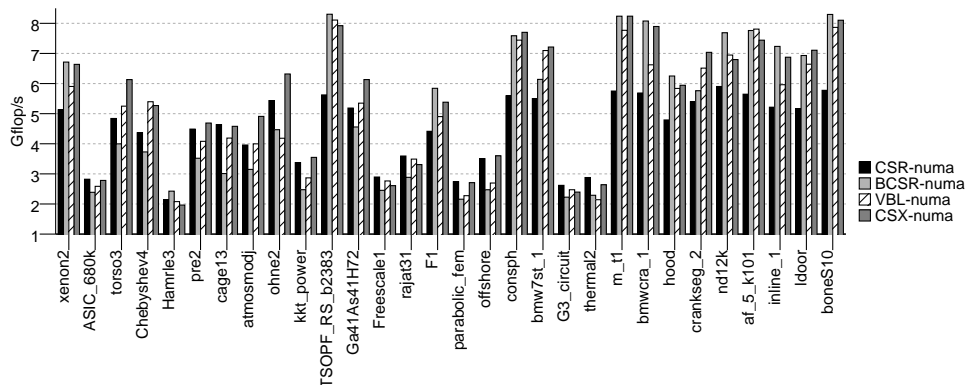
Τα αποτελέσματα της επιτάχυνσης της επίδοσης στις αρχιτεκτονικές NUMA ένα σημαντικό πλεονέκτημα σταθερότητας επίδοσης του CSX, καθώς ακόμη και σε εκτελέσεις όπου το εύρος ζώνης μνήμης δεν αποτελεί πρόβλημα, επιτυγχάνει επίδοση αντίστοιχη των CSR και BCSR. Η συμπεριφορά αυτή μπορεί να αναδειχθεί καλύτερα εξετάζοντας τα αποτελέσματα επίδοσης ανά πίνακα για οκτώ νήματα στο σύστημα Gainestown, που παρουσιάζονται στο Σχήμα 5.13. Στη εκτέλεση αυτή, οι απαιτήσεις σε εύρος ζώνης μνήμης είναι σημαντικές, αλλά το υποσύστημα μνήμης δεν έχει ακόμα κορεστεί. Παρατηρώντας την επίδοση ανά πίνακα, μπορεί κανείς να διακρίνει δύο κατηγορίες πινάκων:

(α') πίνακες χαμηλής επίδοσης (≈ 3 Gflop/s σε μορφή CSR) και

(β') πίνακες υψηλής επίδοσης (≈ 5 Gflop/s σε μορφή CSR).

Η πρώτη κατηγορία αποτελείται από πίνακες με πολύ μικρές γραμμές και με ακανόνιστη διάταξη των μη μηδενικών τους στοιχείων· σύμφωνα με την συζήτηση στο Κεφάλαιο 3, η επίδοση του SpMV σε αυτές τις περιπτώσεις εμποδίζεται από ένα σύνολο παραγόντων, ασχέτων με τον κορεσμό του εύρους ζώνης μνήμης, όπως είναι το ακανόνιστο σχήμα προσβάσεων στο διάνυσμα εισόδου, η αυξημένη επιβάρυνση βρόχων και ανισορροπίες στην κατανομή του φόρτου εργασίας. Η δεύτερη κατηγορία, η οποία είναι και η πιο συνήθης στις εφαρμογές του SpMV, αποτελείται από πίνακες με σχετικά κανονική διάταξη των στοιχείων τους και προκύπτουν συνήθως από την διακριτοποίηση των ΜΔΕ· το κυριότερο πρόβλημα στην περίπτωση αυτή είναι η συμφόρηση για εύρος ζώνης μνήμης. Το σχήμα CSX επιτυγχάνει σημαντική βελτίωση της επίδοσης σε πίνακες της δεύτερης κατηγορίας, πλησιάζοντας το άνω όριο της επίδοσης σε αυτή την εκτέλεση, οριζόμενο από την επίδοση του πολλαπλασιασμού πυκνού πίνακα με διάνυσμα (8.5 Gflop/s). Στην κατηγορία των πινάκων χαμηλής επίδοσης, το σχήμα CSX παρουσιάζει μία μάλλον σταθερή

5. Το Σχήμα Αποθήκευσης COMPRESSED SPARSE EXTENDED



Σχήμα 5.13: Ανά πίνακα επίδοση στο σύστημα Gainestown (οκτώ νήματα, κατά-NUMA). Το σχήμα CSX προσφέρει σταθερά και με συνέπεια υψηλή επίδοση από τους πιο ακανόνιστους στους πιο κανονικούς πίνακες, την στιγμή που τα υπόλοιπα εναλλακτικά του CSR σχήματα ακολουθούν μία προσέγγιση «όλα-ή-τίποτα».

συμπεριφορά, επιτυγχάνοντας ή και υπερβαίνοντας ακόμη την επίδοση του CSR. Παρόλο που τα σχήματα VBL και BCSR εμφανίζουν αντίστοιχα υψηλή επίδοση στους πιο κανονικούς πίνακες, με το σχήμα BCSR να επιτυγχάνει επίδοση αντίστοιχη του CSX σε πίνακες με πολλά μπλοκ (xenon2, consph, m_t1, bmwera_1, inline_1), η κατάσταση αντιστρέφεται στους ακανόνιστους πίνακες, όπου αμφότερα τα BCSR και VBL τείνουν μειώσουν σημαντικά την επίδοση, λόγω της σημαντικής επιβάρυνσης προκαλούν στο μέγεθός του πίνακα και στους επιμέρους υπολογισμούς. Το πλεονέκτημα του CSX είναι επίσης εμφανές στους πίνακες που διακατέχονται από διαγώνιες κανονικότητες (π.χ., torso3, cage13, atmosmodj), τις οποίες τα υπόλοιπα εναλλακτικά σχήματα δεν μπορούν να εκμεταλλευτούν αποδοτικά.

Σταθερότητα επίδοσης

Η σταθερότητα της επίδοσης του CSX παρουσιάζεται ποσοτικά στον Πίνακα 5.3, όπου παρέχονται μερικά στατιστικά στοιχεία για την οκτω-νηματική, κατά-NUMA εκτέλεση στο σύστημα Gainestown, ένα σύστημα που ευνοεί περισσότερο τα φιλικά προς τους υπολογισμούς σχήματα αποθήκευσης⁵. Η ανταγωνιστικότητα των CSR και BCSR είναι εμφανής από τον πίνακα αυτό, καθώς καταφέρνουν να «κερδίσουν» επτά και εννέα πίνακες, αντίστοιχα. Παραταύτα, ακόμα και σε αυτή την, όχι τόσο ευνοϊκή, διάταξη εκτέλεσης, το

⁵ Στα συστήματα Harperton και Dunnington, η κυριαρχία του CSX είναι σχεδόν καθολική, παρέχοντας την καλύτερη επίδοση σε 26 από τους 30 πίνακες της σουίτας μας.

5.7. Αξιολόγηση της επίδοσης του CSX

	CSR	BCSR	VBL	CSX
Best perf.	7	9	2	12
Differences from best				
Average	21.15%	20.13%	12.03%	5.79%
95% C.I.	±3.53%	±4.99%	±2.94%	±2.16%
<i>Minimum</i>	2.60%	0.01%	1.62%	1.13%
<i>Maximum</i>	32.27%	35.86%	33.79%	19.04%

Πίνακας 5.3: Σταθερότητα της επίδοσης των υπό θεώρηση σχημάτων αποθήκευσης σε σχέση με το CSX (Gainestown, οκτώ νήματα, κατά-NUMA).

σχήμα CSX επιτυγχάνει την απολύτως καλύτερη επίδοση στην πλειονότητα των πινάκων (12), την στιγμή που το VBL περιορίζεται σε δύο μόνο πίνακες. Η σημαντικότερη πληροφορία του πίνακα αυτού, ωστόσο, είναι η διαφορά της επίδοσης από την απολύτως υψηλότερη. Πιο συγκεκριμένα, για κάθε ένα από τα υπό θεώρηση σχήματα, παρουσιάζουμε την μέση διαφορά της επίδοσής του από την απολύτως υψηλότερη για όλους του πίνακες που δεν κατάφερε να «κερδίσει»· παρουσιάζουμε επιπλέον ένα 95% διάστημα εμπιστοσύνης. Οι μετρικές αυτές επιβεβαιώνουν την κυριαρχία του CSX τόσο στο επίπεδο της συνολικής επίδοσης του πυρήνα SrMV, όσο και στο επίπεδο της προσαρμογής στα διαφορετικά χαρακτηριστικά του κάθε πίνακα. Παρά την επίτευξη της απόλυτης καλύτερης επίδοσης σε λιγότερους από τους μισούς πίνακες, το σχήμα CSX καταφέρνει να παραμείνει πολύ κοντά (5.8%) στην μέγιστη επίδοση. Το σχήμα VBL ακολουθεί αρκετά πιο πίσω με μία απόσταση 12% από την καλύτερη επίδοση, ενώ τα BCSR και CSR έρχονται τελευταία με αποστάσεις 20.1% και 21.2%, αντιστοίχως. Η σταθερότητα του CSX είναι επίσης ανώτερη σε σχέση με υπόλοιπα θεωρούμενα σχήματα, όπως φαίνεται από τα εικονιζόμενα διαστήματα εμπιστοσύνης, ενώ το σχήμα BCSR, όπως αναμενόταν, παρουσιάζει την μικρότερη σταθερότητα με μία διακύμανση της τάξης του 5%. Το σχήμα CSX μπορεί, επομένως, να θεωρηθεί ως ένα σχήμα αποθήκευσης υψηλών επιδόσεων, καθώς όχι μόνο επιτυγχάνει την υψηλότερη επίδοση στην πλειονότητα των πινάκων τόσο σε αρχιτεκτονικές SMP όσο και NUMA, αλλά καταφέρνει να παραμείνει πολύ κοντά στην απολύτως υψηλότερη επίδοση στους υπόλοιπους πίνακες, συμπεριλαμβανομένων εξαιρετικών περιπτώσεων, όπως είναι οι πολύ ακανόνιστοι πίνακες, παρουσιάζοντας μία σπουδαία σταθερότητα στην επίδοσή του συγκριτικά με άλλα εναλλακτικά σχήματα αποθήκευσης.

5. Το Σχήμα Αποθήκευσης COMPRESSED SPARSE EXTENDED

	<i>1 core</i>	<i>1 socket</i>	<i>4 sockets</i>
DMV	558 Mflop/s	755 Mflop/s	10095 Mflop/s
CSX	547 Mflop/s	751 Mflop/s	9922 Mflop/s
CSR	404 Mflop/s	473 Mflop/s	5509 Mflop/s

(α) Dunnington.

	<i>1 core</i>	<i>1 socket</i>	<i>2 sockets</i>
DMV	1827 Mflop/s	4950 Mflop/s	8562 Mflop/s
CSX	1766 Mflop/s	4872 Mflop/s	8442 Mflop/s
CSR	1526 Mflop/s	3117 Mflop/s	5658 Mflop/s

(β) Gainestown.

Πίνακας 5.4: Επίδοση του CSX έναντι του DMV. Παρά το γεγονός ότι είναι ένα γενικό σχήμα αποθήκευσης, το CSX ανταγωνίζεται την επίδοση του πολλαπλασιασμού πυκνού πίνακα με διάλυμα.

Επίδοση σε πίνακες ειδικής δομής

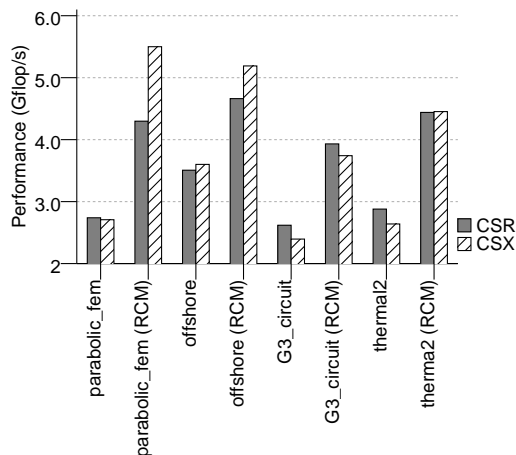
Παρά το ότι η δομή του πίνακα είναι αδιάφορη για το σχήμα CSX, εντούτοις, αυτό μπορεί να την αναγνωρίζει με επιτυχία και να προσφέρει επίδοση συγκρίσιμη εξειδικευμένων σχημάτων σε πίνακες με ειδική δομή, π.χ., διαγώνιους, κατά ζώνες κ.ά. Ως απόδειξη, συγκρίνουμε την επίδοση του SpMV στο σχήμα CSX με την επίδοση του πυρήνα πολλαπλασιασμού πυκνού πίνακα με διάλυμα (DMV) σε ένα πυκνό πίνακα 2000×2000 . Ο Πίνακας 5.4 παρουσιάζει τα αποτελέσματα της επίδοσης αμοτέρων των πυρήνων για τα συστήματα Dunnington και Gainestown. Παρά το γεγονός ότι πρόκειται για γενικό σχήμα αποθήκευσης, είναι φανερό ότι το σχήμα CSX, χωρίς καμία προηγούμενη γνώση, μπορεί να επιτυγχάνει την επίδοση εξειδικευμένων σχημάτων αποθήκευσης (στην προκειμένη περίπτωση, του σχήματος αποθήκευσης πυκνού πίνακα). Αυτό κατέστη δυνατό, διότι το σχήμα CSX ανίχνευσε και κωδικοποίησε μία μόνο κανονικότητα (οριζόντια, χωρίς μονάδες δέλτα) και το σύστημά μας παραγωγής κώδικα απάλειψε πλήρως την εντολή switch, εισάγοντας μία απλή, inline κλήση στον εξειδικευμένο κώδικα της κανονικότητας. Η μικρή επιβάρυνση που παρατηρείται προκαλείται λόγω του ορίου των 255 στοιχείων ανά κανονικότητα και της διαδικασίας διατήρησης των δομών του CSX. Η συμπεριφορά του CSX αναμένεται να είναι αντίστοιχη και στην περίπτωση πινάκων κατά ζώνες, όπου το σχήμα CSX θα ανιχνεύσει μόνο κανονικότητες `diag(1)`.

Η ικανότητα του CSX να προσαρμόζεται αυτόματα στην δομή του πίνακα προσφέρει εξαιρετική ευελιξία, καθώς μπορεί να παράσχει σχεδόν βέλτιστη

επίδοση για μία πληθώρα διαφορετικών πινάκων χωρίς οποιαδήποτε προηγούμενη γνώση για την δομή των μη μηδενικών τους στοιχείων. Τα αποτελέσματα αυτά τονίζουν την δυνατότητα προσαρμογής του CSX, το οποίο, επιπλέον, ανταγωνίζεται ή και υπερβαίνει την επίδοση του CSR σε ακανόνιστους πίνακες.

Η επίδραση της αναδιάταξης του πίνακα

Ορισμένοι πίνακες της σουίτας μας ξεφεύγουν από την τυπική λογική της περιορισμένης από το εύρος ζώνης μνήμης συμπεριφοράς, και αντιμετωπίζουν σημαντικές επιβαρύνσεις στην επίδοσή τους, προκαλούμενες κυρίως λόγω της ακανόνιστης διάταξης των μη μηδενικών τους στοιχείων (βλ. Σχήμα 5.13). Όπως συζητήθηκε στο Κεφάλαιο 2 (Ενότητα 2.4), μία κοινή τεχνική για την «ομογενοποίηση» της δομής ενός αραιού πίνακα είναι η ελαχιστοποίηση του εύρους ζώνης του [Cuthill και McKee, 1969; Karypis και Kumar, 1995; Çatalyürek και Aykanat, 1999]. Τέτοιες τεχνικές μπορούν να εφαρμοστούν σε δομικά συμμετρικούς πίνακες και χρησιμοποιούν αναδιατάξεις των γραμμών και των στηλών τους, με σκοπό να φέρουν τα μη μηδενικά στοιχεία όσο το δυνατόν πλησιέστερα στην κύρια διαγώνιο. Μία επιτυχής αναδιάταξη του πίνακα θα μετατρέψει το σχήμα των προσβάσεων στο διάνυσμα εισόδου σε πιο κανονική μορφή και θα ομογενοποιήσει τον λόγο `flor:byte` σε ολόκληρο τον πίνακα, επιτρέποντας, επομένως, υψηλότερα επίπεδα επίδοσης. Εφαρμόσαμε τον αλγόριθμο αναδιάταξης Reverse Cuthill-McKee (RCM) [Cuthill και McKee, 1969] στους συμμετρικούς πίνακες χαμηλής επίδοσης, συγκεκριμένα στους `parabolic_fem`, `offshore`, `G3_circuit` and `thermal2`. Το Σχήμα 5.14 παρουσιάζει την επίδοση των CSR και CSX στους κανονικούς και στους αναδιατεταγμένους πίνακες στο σύστημα Gainestown στην κατά-NUMA οκτώ-νηματική εκτέλεση. Το κέρδος στην επίδοση είναι σημαντικό για αμφότερα τα σχήματα, ωστόσο, το σχήμα CSX ωφελείται περισσότερο σε όλους τους θεωρούμενους πίνακες. Στους `parabolic_fem` και `offshore`, το σχήμα CSX επιτυγχάνει 28% και 11% βελτίωση της επίδοσης επί του CSR, αντίστοιχα, την στιγμή που πριν την αναδιάταξη η επίδοση και των δύο σχημάτων ήταν αντίστοιχη. Παρόμοια συμπεριφορά παρατηρείται και στους πίνακες `G3_circuit` και `thermal2`, όπου το σχήμα CSX κατάφερα να μειώσει την διαφορά επίδοσής του σε σχέση με το CSR. Ένα σημαντικό συμπέρασμα που προκύπτει από το σχήμα αυτό είναι ότι η τεχνική της αναδιάταξης του πίνακα, αλλά και άλλες αντίστοιχες τεχνικές βελτιστοποίησης της πρόσβασης στο διάνυσμα εισόδου, δεν επηρεάζουν τις τεχνικές βελτιστοποίησης της χρήσης του εύρους ζώνης μνήμης του CSX. Επιπλέον, οι τεχνικές αυτές φαίνεται να ευνοούν το CSX περισσότερο, καθώς δημιουργούν τις προϋποθέσεις για την ανίχνευσης περισσότερων κανονικοτήτων και, αντίστοιχα, μεγαλύτερης συμπίεσης.

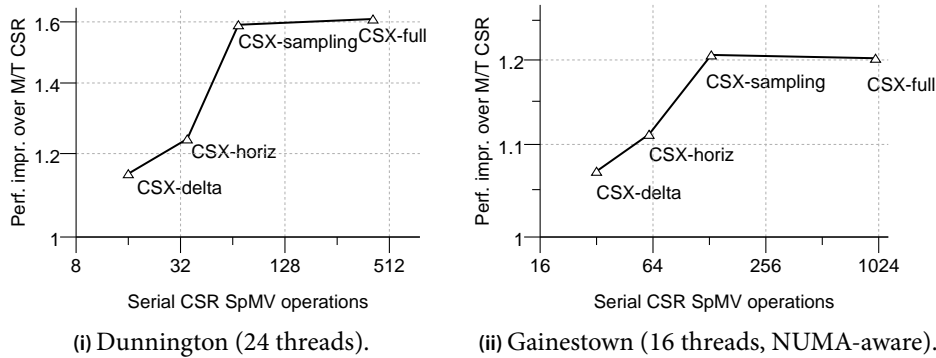


Σχήμα 5.14: Η επίδοση των CSX και CSR σε πίνακες αναδιατεταγμένους κατά RCM. Οι τεχνικές αναδιάταξης του πίνακα εννοούν περισσότερο το σχήμα CSX, καθώς η ανακατανομή των μη μηδενικών στοιχείων επιτρέπει την ανίχνευση περισσότερων κανονικοτήτων. Παρουσιάζονται τα αποτελέσματα της οκτω-νηματικής, κατά-NUMA εκτέλεσης στο σύστημα Gainestown.

5.7.3 Το κόστος προεπεξεργασίας του CSX

Όπως συζητήθηκε στην Ενότητα 5.5, το κόστος προεπεξεργασίας του CSX καθορίζεται από την ανίχνευση και κωδικοποίηση των μη οριζοντίων κανονικοτήτων, οι οποίες απαιτούν την λεξικογραφική ταξινόμηση των στοιχείων του πίνακα. Ένας απλός τρόπος μείωσης του κόστους προεπεξεργασίας είναι ο προγραμματισμός του CSX⁶ να ανιχνεύει μόνο μονάδες δέλτα ή οριζόντιες κανονικότητες. Ωστόσο, αυτός ο τρόπος μείωσης της προεπεξεργασίας του πίνακα έχει επίδραση στην συνολική επίδοση του CSX, όπως φαίνεται στο Σχήμα 5.15. Για την εκμετάλλευση της πλήρους δυναμικής του, θα πρέπει το σχήμα CSX να προγραμματιστεί για την ανίχνευση όσο το δυνατόν περισσότερων κανονικοτήτων. Σε αυτή την περίπτωση, όμως, το κόστος προεπεξεργασίας ανέρχεται σε μερικές εκατοντάδες σειριακές πράξεις του SpMV στο σχήμα CSR. Αν και όχι παράλογο ακόμα και για online προεπεξεργασία, το κόστος αυτό είναι σημαντικό και μπορεί να αναιρέσει μεσοπρόθεσμα το όφελος του CSX στην επίδοση του SpMV. Για τον λόγο αυτό, χρησιμοποιούμε στατιστικό δειγματισμό του πίνακα εισόδου, όπως περιγράφηκε με λεπτομέρεια στην Ενότητα 5.5. Η χρήση δειγματισμού μπορεί να μειώσει το κόστος προεπεξερ-

⁶ Το σχήμα CSX μπορεί να ρυθμιστεί ώστε να ανιχνεύει μόνο συγκεκριμένους τύπους κανονικοτήτων ή ακόμα και να κωδικοποιεί συγκεκριμένες πραγματώσεις κανονικοτήτων.



Σχήμα 5.15: Το κόστος προεπεξεργασίας του CSX. Ο στατιστικός δειγματισμός του πίνακα οδηγεί σε πτώση του κόστους προεπεξεργασίας του πίνακα κατά μία τάξη μεγέθους, καθιστώντας το σχήμα CSX μία βιώσιμη εναλλακτική ακόμα και για online προεπεξεργασία του πίνακα.

γασίας μία τάξη μεγέθους με μηδαμινή επίδραση στην συνολική επίδοση του CSX. Ένα κρίριο σημείο κατά τον δειγματισμό του πίνακα είναι η χρήση πολλών δειγμάτων καταναμημένων σε ολόκληρο το πίνακα, ώστε να αποκτηθεί μία συνολική ματιά της κατανομής των μη μηδενικών στοιχείων και να αποφευχθεί μία υπερεκτίμηση ή υποεκτίμηση σχετικά με την ύπαρξη συγκεκριμένων κανονικοτήτων. Στην περίπτωση μας, χρησιμοποιήσαμε 48 δείγματα επί του 1% των μη μηδενικών στοιχείων του πίνακα.

Το Σχήμα 5.15 παρουσιάζει το κόστος προεπεξεργασίας του CSX, μετρημένο σε πλήθος σειριακών εκτελέσεων του SpMV με το σχήμα CSR, σε συνάρτηση με την επιτευχθείσα επί του πολυνηματικού CSR βελτίωση της επίδοσης στα συστήματα Dunnington και Gainestown με 24 και 16 νήματα, αντίστοιχα. Το κόστος προεπεξεργασίας για την ανίχνευση μόνο μονάδων δέλτα (CSX-delta) ή οριζοντίων κανονικοτήτων (CSX-horiz) στο σύστημα Dunnington είναι αρκετά μικρό και κυμαίνεται μεταξύ 16 και 35 σειριακών πράξεων SpMV, αλλά επιτυγχάνεται μόνο 14.8% έως 23.9% βελτίωση της επίδοσης. Η χρήση δειγματισμού κατά την ανίχνευση όλων των υποστηριζομένων τύπων κανονικοτήτων (CSX-sampling) αυξάνει το κόστος προεπεξεργασίας στις 69 σειριακές πράξεις SpMV—μία προσιτή τιμή για on-line προεπεξεργασία του πίνακα—παρέχοντας 59% βελτίωση της επίδοσης. Η ενεργοποίηση της πλήρους προεπεξεργασίας (μη χρήση παραθύρων και δειγματισμού) αυξάνει κατακόρυφα το κόστος, το οποίο υπερβαίνει τις 400 σειριακές πράξεις SpMV, για να προσφέρει μία ελάχιστη επιπλέον βελτίωση της επίδοσης της τάξης του 3.1%.

Παρόμοια είναι η εικόνα στο σύστημα Gainestown, όπου ο δειγματισμός οδηγεί ακόμα και σε υψηλότερη επίδοση κατά μέσο όρο. Αυτό συμβαίνει, διότι

η επίδοση του SpMV στο σύστημα Gainestown δεν είναι ευθέως συνδεδεμένη με το μέγεθος της αναπαράστασης του πίνακα, όπως συμβαίνει στα συστήματα SMP, επομένως, κάποιες αποφάσεις κωδικοποίησης που λαμβάνονται λόγω του δειγματισμού μπορούν να οδηγήσουν σε καλύτερα υπολογιστικά χαρακτηριστικά, παρά το μεγαλύτερο μέγεθος αναπαράστασης. Μία ακόμα διαφορά μεταξύ των συστημάτων SMP και NUMA, όπως αποκαλύπτεται από το Σχήμα 5.15, είναι ότι το κόστος προεπεξεργασίας είναι περισσότερο εκτεθειμένο στις αρχιτεκτονικές NUMA και κυμαίνεται μεταξύ 32 και 985 σειριακών πράξεων SpMV, την στιγμή που στο σύστημα Dunnington το κόστος αυτό βρίσκεται μεταξύ 16 και 492 πράξεων.

5.8 Ενσωμάτωση του CSX σε λογισμικό multiphysics

Ως περαιτέρω αξιολόγηση της δυναμικής του CSX στην βελτιστοποίηση του SpMV στο πλαίσιο εφαρμογών προσομείωσης φυσικών φαινομένων (multiphysics), ενσωματώσαμε το CSX στο λογισμικό προσομοιώσεων Elmer [Lyly et al., 1999–2000]. Το Elmer χρησιμοποιεί επαναληπτικές μεθόδους Krylov για την επίλυση μεγάλων προβλημάτων, κάνοντας χρήση της μεθόδου Bi-Conjugate Gradient Stabilized (BiCGStab) [van der Vorst, 1992] με preconditioning για την επίλυση των προκυπτόντων γραμμικών συστημάτων. Το Elmer υποστηρίζει παραλληλισμό ανάμεσα σε πολλούς επεξεργαστικούς κόμβους χρησιμοποιώντας MPI, αλλά η εκτέλεσή του παραμένει σειριακή εντός του κάθε κόμβου. Για τον λόγο αυτό, υλοποιήσαμε και μία πολυνηματική έκδοση του CSR για το Elmer, ώστε να επιτύχουμε μία δίκαιη σύγκριση με το σχήμα CSX.

Η διαδικασία της ενσωμάτωσης ήταν σχετικά απλή, καθώς το Elmer υποστηρίζει ήδη την χρήση εξωτερικών βιβλιοθηκών για τον πυρήνα SpMV, μέσω μιας καλώς ορισμένης διεπαφής πολλαπλασιασμού· ωστόσο, επεκτείναμε ελαφρώς την διεπαφή, ώστε να υποστηρίζεται η έννοια της βελτιστοποίησης της αναπαράστασης του πίνακα (βλ. Σχήμα 5.16). Το Elmer περνά στην εξωτερική βιβλιοθήκη τις δομές του CSR και τα διανύσματα εισόδου και εξόδου, καθώς και μία γενική λαβή (handle), η οποία θα συνδεθεί με τον βελτιστοποιημένο πίνακα, είτε αυτόματα (π.χ., κατά την πρώτη κλήση) είτε κατ' απαίτηση (σημαία `reinit`). Όταν μία κλήση πολλαπλασιασμού πίνακα-επί-διάνυσμα ανακατευθύνεται στην βιβλιοθήκη του CSX, εάν η λαβή `*tuned` είναι έγκυρη ή έχει τεθεί η σημαία `reinit` flag, κατασκευάζουμε τον πίνακα CSX, τον αναθέτουμε στη λαβή `*tuned` και προχωράμε με τον πολλαπλασιασμό. Σε αντίθετη περίπτωση, πραγματοποιούμε απλά τον πολλαπλασιασμό με την παρασχεθείσα λαβή, αγνοώντας τις υπόλοιπες παραμέτρους του CSR. Η διαδικασία κατασκευής του πίνακα CSX από τον πίνακα εισόδου σε μορφή CSR πραγματοποιείται σε δύο βήματα:

```
void matvec_(void **tuned, void *n,
            void *rowptr, void *colind, void *values,
            void *u, void *v,
            int reinit);
```

Σχήμα 5.16: Η ανανεωμένη διεπαφή πολλαπλασιασμού πίνακα-επί-διάνυσμα στο Elmer. Τα νέα ορίσματα σημειώνονται με πλάγια γραμματοσειρά, ενώ τα u και v είναι τα διανύσματα εισόδου και εξόδου, αντίστοιχα. Εάν το $*tuned$ είναι NULL ή η $reinit$ έχει τεθεί, ο πίνακας CSX κατασκευάζεται από τις παραμέτρους του CSR και ανατίθεται στην μεταβλητή $*tuned$. Εάν η σημαία $reinit$ δεν τεθεί κατά την διάρκεια ενός υπολογισμού, ο πίνακας CSX κατασκευάζεται άπαξ κατά την πρώτη κλήση.

- (α') *Μετατροπή του CSR στην εσωτερική αναπαράσταση του CSX.* Η αναπαράσταση αυτή χρησιμοποιείται για την προεπεξεργασία του πίνακα (βλ. Ενότητα 5.3.1).
- (β') *Ανίχνευση και κωδικοποίηση κανονικοτήτων.* Αυτή είναι η φάση της κυρίως προεπεξεργασίας του CSX, όπου ο πίνακας ανιχνεύεται για την ύπαρξη κανονικοτήτων και κατασκευάζεται το τελικό σχήμα CSX.

Για την αξιολόγηση της ενσωμάτωσης του CSX, επιλέξαμε πέντε ενδεικτικά προβλήματα από την σουίτα δοκιμών του Elmer. Αυξήσαμε κατάλληλα το μέγεθος του προβλήματος (αυξάνοντας την πυκνότητα του πλέγματος), ώστε να προκύπτουν αρκετά μεγάλοι πίνακες που να υπερβαίνουν το όριο των 576 MiB⁷. Ο Πίνακας 5.5 παρουσιάζει τις λεπτομέρειες των επιλεχθέντων προβλημάτων αναφέρεται, επίσης, το ποσοστό επί του συνολικού χρόνου εκτέλεσης του επιλυτή⁸ που δαπανάται στην ρουτίνα υπολογισμού του SpMV.

Το Σχήμα 5.17 παρουσιάζει την διάρθρωση του συνολικού χρόνου εκτέλεσης που δαπανήθηκε εντός της βιβλιοθήκης του CSX σε σύνολο 1000 επαναλήψεων της μεθόδου Bi-CG. Η πρώτη παρατήρηση είναι ότι το κόστος της μετατροπής στην εσωτερική αναπαράσταση του CSX είναι αμελητέο. Πράγματι, το βήμα αυτό αποτελείται από μία μεγάλη καταχώρηση μνήμης για τις δομές δεδομένων της εσωτερικής αναπαράστασης και μία διάσχιση του σε μορφή CSR πίνακα εισόδου. Το υπόλοιπο κόστος προεπεξεργασίας αποκρύβεται καθώς προσθέτουμε νήματα στον υπολογισμό του SpMV ή όταν ο ίδιος ο υπολογισμός είναι αφεαυτού χρονοβόρος, όπως στην περίπτωση του προβλήματος vortex3d. Το συνολικό κόστος προεπεξεργασίας του CSX αποσβένεται πλήρως ύστερα από 224–300 επαναλήψεις της μεθόδου με χρήση του πολυνηματικής πυρήνα CSR. Αν και το κόστος αυτό είναι λογικό για μεγάλες και σύνθετες

⁷ Αυτό είναι το μέγεθος της συνολικής κρυφής μνήμης των συμμετεχόντων κόμβων της συστοιχίας που χρησιμοποιήθηκε για τον υπολογισμό.

⁸ Ο χρόνος κατασκευής του αρχικού πίνακα δεν περιλαμβάνεται.

Problem name	Equations involved	SpMV exec. time (%)
fluxsolver	Heat + Flux	57.4
HeatControl	Heat	57.5
PoissonDG	Poisson + Discontinuous Galerkin	62.0
shell	Reissner-Mindlin	83.0
vortex3d	Navier-Stokes + Vorticity	92.3

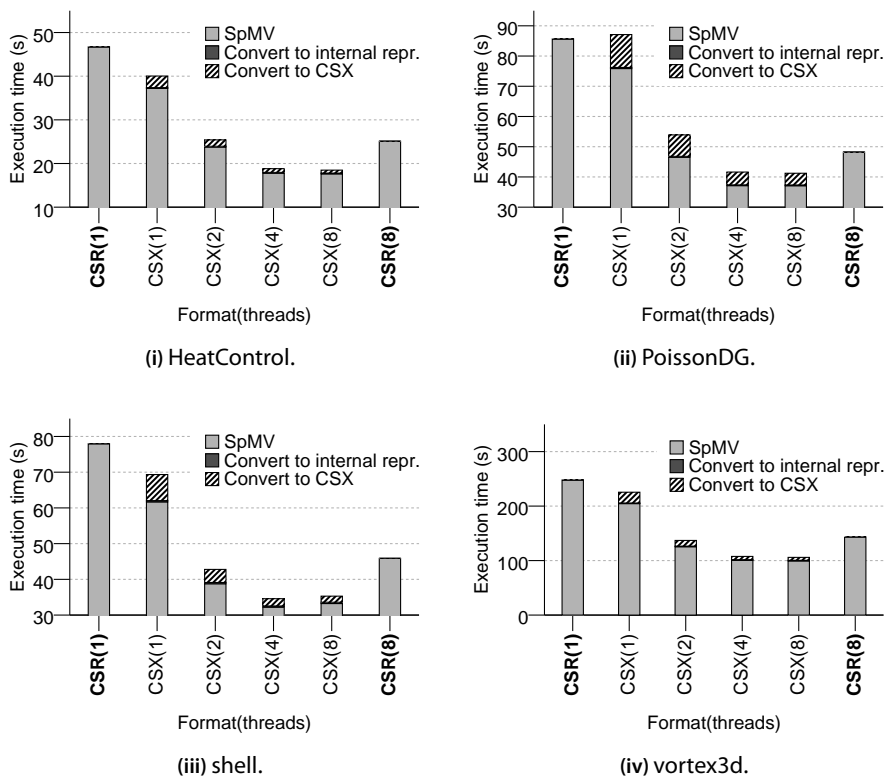
Πίνακας 5.5: Τα προβλήματα αναφοράς που χρησιμοποιήθηκαν για την αξιολόγηση της ενσωμάτωσης του CSX στο λογισμικό φυσικών προσομοιώσεων Elmer. Τα προβλήματα επελέγησαν από την σουίτα δοκιμών του Elmer και μεγεθύνθηκαν σημαντικά, ώστε να οδηγήσουν σε αρκετά μεγάλες αναπαραστάσεις πίνακα (μεγαλύτερες των 576 MiB). Τα ποσοστά του χρόνου εκτέλεσης αναφέρονται επί του συνολικού χρόνου εκτέλεσης του επιλυτή, συμπεριλαμβανομένου και ενός διαγωνίου preconditioner.

προσομοιώσεις, οι οποίες μπορεί να απαιτούν χιλιάδες επαναλήψεων, η προεπεξεργασία του CSX θα μπορούσε να πραγματοποιείται offline, ώστε να μειωθεί περαιτέρω αυτό το κόστος εκκίνησης.

Παρά το κόστος προεπεξεργασίας του, το σχήμα CSX κατάφερε να επιταχύνει σημαντικό όχι μόνο το τμήμα του SpMV του λογισμικού Elmer, αλλά και τον συνολικό χρόνο εκτέλεσης του επιλυτή (Σχήμα 5.18). Πιο συγκεκριμένα, το σχήμα CSX μπόρεσε να παράσχει 37% κατά μέσο όρο βελτίωση της επίδοσης του τμήματος SpMV σε μία μεικτή συστοιχία 24 κόμβων διεπεξεργαστικών SMP τετραπύρηνων συστημάτων Intel Xeon E5405/E5335 (Harperstown/Clovertown) ύστερα από 1000 επαναλήψεις της μεθόδου Bi-CG. Η συνολική επίδοση του επιλυτή βελτιώθηκε αξιοσημείωτα κατά 14.8%, παρά το μεγάλο κόστος preconditioning σε τρία από τα προβλήματα αναφορά (βλ. Πίνακα 5.5). Πιστεύουμε ότι η βελτίωση αυτή μπορεί να γίνει ακόμα πιο σημαντική, εάν και άλλα σημεία του επιλυτή (π.χ., ο preconditioner) παραλληλοποιηθούν.

5.9 Περίληψη

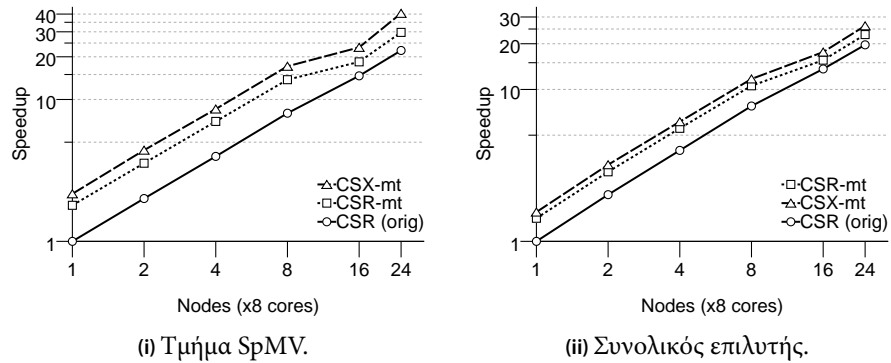
Στο κεφάλαιο αυτό παρουσιάσαμε, συζητήσαμε και αξιολογήσαμε σε λεπτομέρεια το σχήμα αποθήκευσης Compressed Sparse eXtended (CSX), μία καινούργια προσέγγιση στην αποθήκευση αραιών πινάκων, με κύριο στόχο την επίτευξη υψηλής επίδοσης στις σύγχρονες πολυπύρνες αρχιτεκτονικές υπολογιστών. Το σχήμα CSX μπορεί να ανιχνεύσει πληθώρα διαφορετικών κανονικοτήτων μη μηδενικών στοιχείων του πίνακα, π.χ., οριζόντιες, διαγώνιες, δισδιάστατα μπλοκ, τις οποίες στην συνέχεια κωδικοποιεί σε μία ενιαία, πολύ



Σχήμα 5.17: Διάρθρωση του χρόνου εκτέλεσης της βιβλιοθήκης του CSX σε σύνολο 1000 επαναλήψεων της μεθόδου Bi-CG. Το κόστος μετατροπής είναι αμελητέο, ενώ η προεπεξεργασία του CSX αποκρύβεται με την αύξηση των νημάτων υπολογισμού. Το κόστος εκκίνησης του CSX αποσβένεται πλήρως ύστερα από λιγότερες των 300 επαναλήψεων.

συμπαγή, αναπαράσταση του πίνακα. Το χαρακτηριστικό αυτό επιτρέπει στο σχήμα CSX να προσαρμόζεται με επιτυχία στις ιδιαιτερότητες κάθε πίνακα και να επιτυγχάνει υψηλά επίπεδα συμπίεσης, προσφέροντας σταθερή και υψηλή επίδοση από των πολύ ακανονίστων πινάκων μέχρι των πινάκων με ειδική δομή. Το σχήμα CSX λαμβάνει ειδική μέριμνα για τις αρχιτεκτονικές NUMA ευνοώντας το υπολογιστικό τμήμα του πυρήνα, το οποίο παίζει σημαντικότερο ρόλο σε αυτές τις περιπτώσεις. Στην πράξη, το σχήμα CSX υπερκεραίνει τα υπόλοιπα εναλλακτικά του CSR σχήματα αποθήκευσης σε όρους απόλυτης επίδοσης, προσφέροντας οφέλη μεγαλύτερα του 60% και 20% σε αρχιτεκτονικές SMP και NUMA, αντίστοιχα. Επιπλέον, είναι μάλλον σημαντικό ότι οι βελτιώσεις αυτές δεν έρχονται σε βάρος του χρόνου προεπεξεργασίας του πίνακα, πράγμα που επιτρέπει στο CSX να παράσχει σημαντική βελτίωση της

5. Το Σχήμα Αποθήκευσης COMPRESSED SPARSE EXTENDED



Σχήμα 5.18: Επιτάχυνση του λογισμικού φυσικών προσομοιώσεων Elmer με την χρήση του σχήματος CSX (συμπεριλαμβάνεται ο χρόνος προεπεξεργασίας) σε σύνολο 1000 επαναλήψεων της μεθόδου επίλυσης. Το γόνατο στο σημείο των 16 κόμβων στα διαγράμματα επιτάχυνσης οφείλεται στο χαμηλότερο εύρος ζώνης μνήμης των κόμβων Clovertown, οι οποίοι αρχίζουν να συμμετέχουν στον υπολογισμό από το σημείο αυτό.

επίδοσης, παρά το αρχικό του κόστος εκκίνησης, και στο πλαίσιο πραγματικών λογισμικών φυσικών προσομοιώσεων.

Εκμετάλλευση συμμετρίας στους αραιούς πίνακες

Συμμετρικοί αραιοί πίνακες εμφανίζονται συχνά στην επίλυση αραιών γραμμικών συστημάτων και υπάρχουν συγκεκριμένες επαναληπτικές μέθοδοι για την επίλυση τέτοιων συμμετρικών προβλημάτων. Καθόσον το βασικό πρόβλημα επίδοσης του πυρήνα $SrMV$ είναι ο κορεσμός του εύρους ζώνης του συστήματος, είναι αρκετά ελκυστικό να μειώσει κανείς σχεδόν στο μισό το μέγεθος αναπαράστασης ενός συμμετρικού πίνακα, μην αποθηκεύοντας τα μη μηδενικά στοιχεία του άνω ή κάτω τριγωνικού υποπίνακα. Ενώ κάτι τέτοιο είναι ωφέλιμο στη σειριακή εκτέλεση, γίνεται προβληματικό στις πολυνηματικές εκτελέσεις. Η τυπική κατά γραμμές διανομή του αραιού πίνακα στα νήματα εισάγει μία ανεπιθύμητη κατάσταση ανταγωνισμού (race condition) σε στοιχεία του διανύσματος εξόδου. Ο προτεινόμενος τρόπος επίλυσης αυτής της εξάρτησης, αποφεύγοντας το απαγορευτικό κόστος των κλειδωμάτων, είναι η χρήση τοπικών, ανά νήμα, διανυσμάτων για την πραγματοποίηση των τοπικών υπολογισμών του $SrMV$ · η μέθοδος αυτή, ωστόσο, εισάγει ένα επιπλέον βήμα αναγωγής στον υπολογισμό του συμμετρικού $SrMV$, το οποίο εντείνει περαιτέρω την πίεση στο υποσύστημα μνήμης και εξαλείφει το όφελος από την μείωση του μεγέθους αναπαράστασης του πίνακα.

Στο κεφάλαιο αυτό, προτείνουμε ένα σχήμα δεικτοδότησης των τοπικών διανυσμάτων του συμμετρικού πυρήνα $SrMV$, το οποίο ελαχιστοποιεί την κίνηση στο υποσύστημα μνήμης κατά την φάση αναγωγής του πυρήνα, επιτρέποντας την κλιμάκωση του πυρήνα $SrMV$. Επεκτείνουμε, επίσης, το σχήμα CSX , ώστε να υποστηρίζει αποδοτικά συμμετρικούς πίνακες. Σε συνδυασμό με το σχήμα δεικτοδότησης των τοπικών διανυσμάτων, το συμμετρικό CSX είναι ικανό να παράσχει περισσότερη από $2\times$ βελτίωση της επίδοσης επί του τυπικού σχήματος CSR και επιτυγχάνει την επιτάχυνση της μεθόδου CG περίπου κατά 80%.

6.1 Ο συμμετρικός πυρήνας SpMV

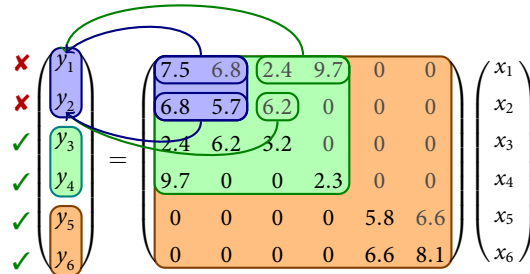
Το πιο ευρέως διαδεδομένο σχήμα αποθήκευσης συμμετρικών αραιών πινάκων είναι το Symmetric Sparse Skyline (SSS), μία παραλλαγή του CSR, η οποία αποθηκεύει την κύρια διαγώνιο και τα στοιχεία του κάτω τριγωνικού υποπίνακα. Πιο συγκεκριμένα, το σχήμα SSS αποθηκεύει τα διαγώνια στοιχεία ξεχωριστά και τον κάτω τριγωνικό υποπίνακα στην κλασσική μορφή του CSR (βλ. Κεφάλαιο 2, Ενότητα 2.4 για περισσότερες πληροφορίες). Θεωρώντας τέσσερα byte για τους αριθμούς των δεικτών και τιμές κινητής υποδιαστολής διπλής ακρίβειας των οκτώ byte για τα μη μηδενικά στοιχεία, το μέγεθος αναπαράστασης του σχήματος SSS μπορεί να υπολογιστεί ως εξής:

$$S_{SSS} = 4 \underbrace{\frac{NNZ - N}{2}}_{\text{colind}} + 8 \underbrace{\frac{NNZ - N}{2}}_{\text{lower values}} + 4 \underbrace{(N + 1)}_{\text{rowptr}} + \underbrace{8N}_{\text{diagonal}} \quad (6.1)$$

$$= 6(NNZ + N) + 4 \quad (6.2)$$

Για την τυπική περίπτωση αραιών πινάκων, όπου $NNZ \gg N$, το μέγεθος του πίνακα στη μορφή SSS είναι σχεδόν του μισό του CSR (περίπου $12NNZ$). Δεδομένης της αλγοριθμικής φύσης του πυρήνα SpMV, αυτή η σημαντική μείωση του μεγέθους αναπαράστασης του πίνακα αναμένεται να προσφέρει αντίστοιχα σημαντική βελτίωση της επίδοσης σε μία σειριακή εκτέλεση του πυρήνα. Ωστόσο, η πολυνηματική εκτέλεση του πυρήνα καθίσταται προβληματική. Ο διαχωρισμός του πίνακα κατά γραμμές εισάγει μία εξάρτηση RAW σε συγκεκριμένα στοιχεία του διανύσματος εξόδου, λόγω των πράξεων ανανέωσης τιμών του SpMV στον άνω τριγωνικό υποπίνακα (βλ. Σχήμα 6.1). Ο συνήθης τρόπος για την αποφυγή αυτών των εξαρτήσεων, χωρίς την χρήση κλειδωμάτων, των οποίων το κόστος είναι απαγορευτικό, είναι η χρήση τοπικών, ανά νήμα, διανυσμάτων για την αποθήκευση των ενδιάμεσων αποτελεσμάτων του πολλαπλασιασμού (Αλγόριθμος 6.1). Η εκτέλεση του SpMV χωρίζεται πλέον σε δύο παραλληλοποιήσιμες φάσεις: την φάση του πολλαπλασιασμού και την φάση της αναγωγής. Η φάση του πολλαπλασιασμού (Αλγόριθμος 6.1, γραμμές 2–11) εκτελεί τον καθαυτό υπολογισμό του SpMV, εγγράφωντας τα ενδιάμεσα αποτελέσματα στα τοπικά διανύσματα, ενώ η φάση της αναγωγής (Αλγόριθμος 6.1, γραμμές 12–16) ανάγει τα τοπικά διανύσματα στο τελικό διάνυσμα εξόδου.

Η πράξη της αναγωγής των τοπικών διανυσμάτων έχει αντίστοιχη του SpMV αλγοριθμική φύση και αναμένεται να περιορίζεται και αυτή από το διαθέσιμο εύρος ζώνης μνήμης, παρά το υψηλό επίπεδο παραλληλίας της, της τάξης του $\Theta(N)$. Η φάση αυτή εισάγει σημαντική επιβάρυνση του συνόλου εργασίας του SpMV, η οποία εξαρτάται από το πλήθος των χρησιμοποιούμενων νημάτων. Υποθέτοντας τιμές διπλής ακρίβειας των οκτώ byte για τα τοπικά



Σχήμα 6.1: Η εξάρτηση RAW στο διάνυσμα εξόδου κατά την παράλληλη εκτέλεση του συμμετρικού πυρήνα SpMV (αναπαραγωγή του σχήματος από το Κεφάλαιο 2).

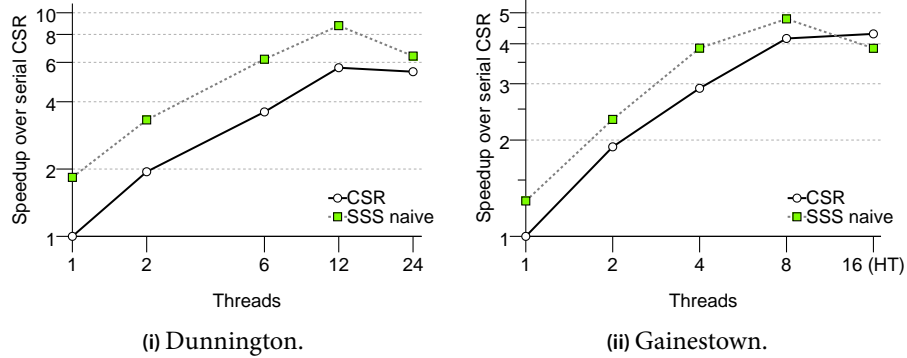
```

1: procedure MATVECSSSMT(A::in, x::in, y::out, p::in, start::in, end::in)
  A: matrix in SSS format
  x: input vector
  y: output vector
  p: participating threads
  start, end: start/end of thread partitions
2:   for i ← 0 to p do in parallel
3:     for r ← start[i] to end[i] do
4:       yi[r] ← dvalues[r] · x[r]
5:       for j ← rowptr[i] to rowptr[i + 1] do
6:         c ← colind[j]
7:         yi[r] ← yi[r] + values[j] · x[c]
8:         yi[c] ← yi[c] + values[j] · x[r]
9:       end for
10:    end for
11:  end for
12:  for r ← 0 to N do in parallel
13:    for i ← 0 to p do
14:      y[r] ← y[r] + yi[r]
15:    end for
16:  end for

```

Αλγόριθμος 6.1: Παράλληλη υλοποίηση του συμμετρικού πυρήνα SpMV με χρήση του σχήματος αποθήκευσης SSS.

6. Εκμετάλλευση συμμετρίας στους αραιούς πίνακες



Σχήμα 6.2: Επιτάχυνση της απλοϊκής υλοποίησης του συμμετρικού SpMV. Η δυνατότητα κλιμάκωσης του συμμετρικού πυρήνα SpMV με χρήση τοπικών διανυσμάτων είναι περιορίζεται από το τελικό βήμα της αναγωγής, του οποίου η επιβάρυνση αυξάνεται γραμμικά με το πλήθος των νημάτων υπολογισμού, οδηγώντας στον πλήρη κορεσμό του διαθέσιμου εύρους ζώνης μνήμης.

διανύσματα και p συμμετέχοντα στον υπολογισμό νήματα, το σύνολο εργασιών της φάσης αναγωγής είναι

$$ws_{\text{red}} = 8pN \quad (6.3)$$

Το βασικό χαρακτηριστικό της επιβάρυνσης αυτής είναι ότι αυξάνει γραμμικά ως προς το πλήθος των χρησιμοποιούμενων νημάτων και αναμένεται, αργά ή γρήγορα, να οδηγήσει σε κορεσμό το εύρος ζώνης μνήμης πέρα από ένα συγκεκριμένο πλήθος νημάτων, περιορίζοντας σημαντικά την δυνατότητα κλιμάκωσης του συμμετρικού πυρήνα SpMV. Η συμπεριφορά αυτή απεικονίζεται στο Σχήμα 6.2. Ενώ το όφελος στην επίδοση του συμμετρικού πυρήνα SSS κυμαίνεται μεταξύ 30% και 80% στην σειριακή εκτέλεση, το κέρδος αυτό συνεχώς μειώνεται καθώς αυξάνεται ο αριθμός των νημάτων, μέχρις ότου εξαλειφθεί τελείως (Dunnington, 24 νήματα, Gainestown, 8 νήματα) ή ακόμα και ανατραπεί (Gainestown, 16 νήματα), οδηγώντας σε χαμηλότερη του βασικού CSR επίδοση.

6.2 Ελαχιστοποίηση του κόστους αναγωγής

Η προηγούμενη συζήτηση επί του συμμετρικού πυρήνα SpMV αποκάλυψε το κύριο πρόβλημα επίδοσης της αυξημένης επιβάρυνσης λόγω του σημαντικού φόρτου εργασίας της φάσης της αναγωγής. Δεδομένης της αλγοριθμικής της

φύσης, είναι κρίσιμο να μειώσει κανείς τον φόρτο αυτό, ώστε να ελαφρυνθεί το υποσύστημα μνήμης και, εφόσον είναι δυνατόν, να αποσυνδεθεί η επιβάρυνση αυτή από το πλήθος των νημάτων, επιτρέποντας με αυτό τον τρόπο την κλιμάκωση του πυρήνα. Παρουσιάζουμε εδώ δύο προσεγγίσεις προς αυτή την κατεύθυνση, βασιζόμενοι στην παρατήρηση ότι αρκεί συγκεκριμένα μέρη του διανύσματος εξόδου να διατηρούνται τοπικά, καθώς δεν συγκρούονται όλες οι προσβάσεις. Επομένως, η κίνηση προς τη μνήμη λόγω της φάσης αναγωγής μπορεί να μειωθεί δραστικά.

6.2.1 Ενεργό εύρος τοπικών διανυσμάτων

Η μέθοδος του ενεργού εύρους διανυσμάτων (effective ranges method), που έχει προταθεί από τους Batista et al. [2010], ανανεώνει μόνο μία συγκεκριμένη περιοχή του κάθε τοπικού διανύσματος και ανακατευθύνει τις υπόλοιπες ανανεώσεις απευθείας στο διάνυσμα εξόδου. Σύμφωνα με τον αλγόριθμο του συμμετρικού SrMV (Αλγόριθμος 6.1), στο νήμα i ανατίθενται οι υπολογισμοί του υποπίνακα σε μορφή SSS μεταξύ των γραμμών $start[i]$ και $end[i]$. Εφόσον αποθηκεύουμε τον κατά τριγωνικό υποπίνακα, δεν υπάρχει περίπτωση το νήμα i να προσπελάσει τα στοιχεία κάτω από όριο της γραμμής $end[i]$. Επομένως, η παράλειψη αυτής της περιοχής κατά την φάση της αναγωγής είναι ασφαλής. Επιπλέον, το νήμα i μπορεί να προσπελάσει απευθείας τα στοιχεία του διανύσματος εξόδου μεταξύ των ορίων $start[i]$ και $end[i]$ χωρίς να καταστρέφεται η συνέπεια των υπολογισμών του SrMV, ακριβώς όπως και στην περίπτωση του μη συμμετρικού πυρήνα. Οι υπολογισμοί του SrMV επί της υπόλοιπης περιοχής, από την πρώτη γραμμή μέχρι την $start[i]$, μπορούν να συγκρουστούν με τις ανανεώσεις του διανύσματος εξόδου και, επομένως, θα πρέπει να αναφέρονται το τοπικό διάνυσμα· η περιοχή αυτή ονομάζεται *ενεργή περιοχή* του τοπικού διανύσματος (Σχήμα 6.3iii) και θα πρέπει να ανανεωθεί κατά την φάση της αναγωγής. Θεωρώντας, χωρίς απώλεια της γενικότητας, ότι κάθε νήμα αναλαμβάνει τον ίδιο αριθμό γραμμών, το σύνολο εργασίας της φάσης αναγωγής για την μέθοδο αυτή, χρησιμοποιώντας p νήματα, μπορεί να υπολογιστεί ως ακολούθως:

$$ws_{\text{eff}} \approx 8 \frac{p(p-1)}{2} \cdot \frac{N}{p} = 4(p-1)N \quad (6.4)$$

Η επιβάρυνση της αναγωγής υποδιπλασιάστηκε σε σχέση με την απλοϊκή έκδοση, ωστόσο το κυρίως πρόβλημα της συμμετρικής εκτέλεσης του SrMV παραμένει: η επιβάρυνση της φάσης αναγωγής συνεχίζει να αυξάνει γραμμικά σε σχέση με το πλήθος των νημάτων.

Οι Geus και Röllin [2001] παρουσιάζουν μία αντίστοιχη προσέγγιση για αραιούς πίνακες χωρίς ζώνες (non-banded) με σκοπό την ελαχιστοποίηση της επικοινωνίας των διεργασιών της κατανεμημένης εκτέλεσης του SrMV. Εξει-

δικεύουν περαιτέρω την μέθοδό τους για πίνακες κατά ζώνες, μειώνοντας το ενεργό εύρος των επικοινωνουμένων δεδομένων (δηλαδή, των τοπικών διανυσμάτων) στο ήμισυ του εύρους ζώνης του πίνακα. Η προσέγγιση αυτή μπορεί να γενικευτεί εύκολα για οποιοδήποτε κανονικό πίνακα, καταλήγοντας στην μέθοδο που περιγράφηκε στην ενότητα αυτή σε περιπτώσεις πινάκων με μεγάλο εύρος ζώνης. Εντούτοις, ακόμα και αυτή η προσέγγιση δεν αποσυνδέει το κόστος της αναγωγής από το πλήθος των νημάτων, ειδικά για πίνακες με μεγάλο εύρος ζώνης.

6.2.2 Δεικτοδότηση τοπικών διανυσμάτων

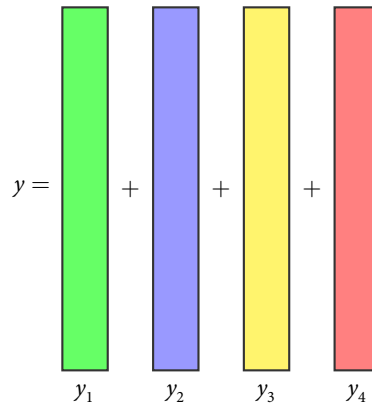
Η δική μας προσέγγιση στην ελαχιστοποίηση του κόστους αναγωγής βασίζεται στην παρατήρηση ότι οι ενεργές περιοχές των τοπικών διανυσμάτων είναι αρκετά αραιές, δηλαδή, πολύ λίγα στοιχεία των ενεργών περιοχών ανανεώνονται τελικά κατά τον υπολογισμό του SpMV. Μία ακόμα ενδιαφέρουσα ιδιότητα των ενεργών περιοχών, όπως φαίνεται και στο Σχήμα 6.4, είναι η συνεχής πτώση της πυκνότητάς τους καθώς προστίθενται επιπλέον νήματα στον υπολογισμό του SpMV, φθάνοντας ένα μέσο όρο της τάξης του 10.7% στα 24 νήματα του συστήματος Dunnington και 2.6% στα 256 νήματα. Εμπνεόμενοι από αυτή την συμπεριφορά, προτείνουμε ένα σχήμα δεικτοδότησης των μη μηδενικών στοιχείων των τοπικών διανυσμάτων, ώστε να ανανεώνονται μόνο τα συγκρουόμενα στοιχεία κατά την φάση της αναγωγής, μειώνοντας το φόρτο εργασίας της. Για κάθε μη μηδενικό στοιχείο των ενεργών περιοχών των τοπικών διανυσμάτων (Σχήμα 6.3iv), διατηρούμε ένα ζεύγος τιμών (*vid*, *idx*), όπου *vid* είναι ένα μοναδικό αναγνωριστικό (ID) του τοπικού διανύσματος και *idx* είναι ο αριθμός του μη μηδενικού στοιχείου εντός του οικείου τοπικού διανύσματος. Το μέγεθος του *idx* είναι ίδιο με το πρωτογενές μέγεθος των δομών δεικτοδότησης του πίνακα, δηλαδή, τέσσερα byte, ενώ το μέγεθος του *vid* μπορεί να μεταβάλεται ανάλογα με τον μέγιστο αναμενόμενο αριθμό νημάτων. Στην υλοποίησή μας, χρησιμοποιούμε –αρκετά γενναιόδωρα– τέσσερα byte για το πεδίο *vid* field, αλλά δύο ή ακόμα και ένα byte θα ήταν αρκετό για τις σύγχρονες πολυπύρηνες αρχιτεκτονικές.

Το σύνολο εργασίας και η αντίστοιχη επιβάρυνση της φάσης αναγωγής χρησιμοποιώντας το σχήμα δεικτοδότησης εξαρτάται από την πυκνότητα d των ενεργών περιοχών των τοπικών διανυσμάτων. Πιο συγκεκριμένα, θεωρώντας, χωρίς απώλεια της γενικότητας, ότι οι N γραμμές του πίνακα έχουν εξίσου διαμοιρασθεί στα p συμμετέχοντα στον υπολογισμό νήματα, η επιβά-

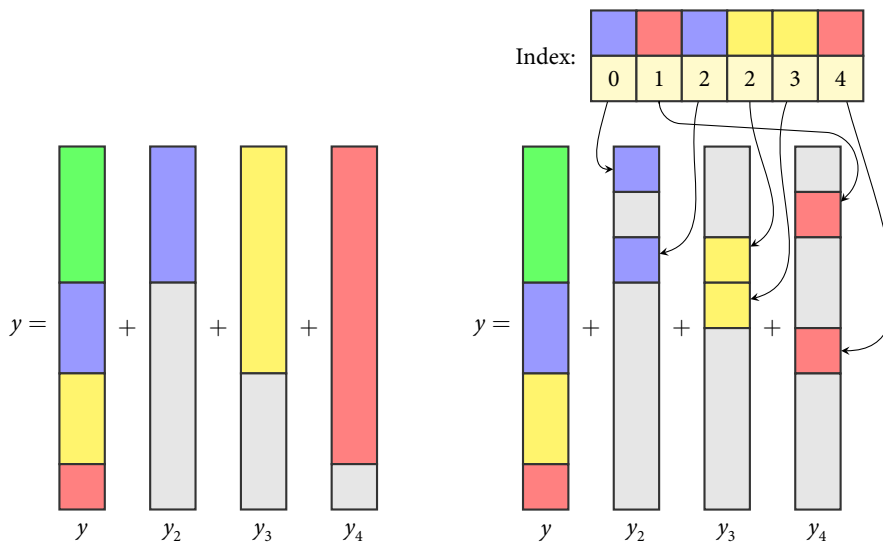
6.2. Ελαχιστοποίηση του κόστους αναγωγής

$$\begin{pmatrix} 2.7 & 0.5 & 0 & 0 & 1.2 & 0 & 0 & 0 \\ 0.5 & 5.6 & 6.6 & 0 & 0 & 0 & 0 & 3.4 \\ 0 & 6.6 & 9.4 & 5.4 & 0 & 4.1 & 0 & 0 \\ 0 & 0 & 5.4 & 0.7 & 9.8 & 7.2 & 0 & 0 \\ 1.2 & 0 & 0 & 9.8 & 2.4 & 0 & 0 & 3.3 \\ 0 & 0 & 4.1 & 7.2 & 0 & 7.8 & 4.7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.7 & 9.8 & 0 \\ 0 & 3.4 & 0 & 0 & 3.3 & 0 & 0 & 4.1 \end{pmatrix}$$

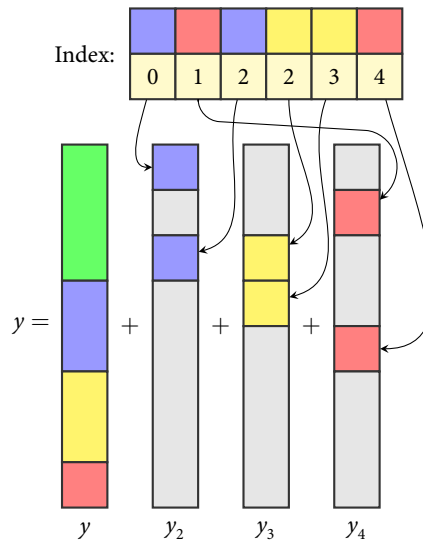
(i) Πίνακας εισόδου.



(ii) Απλοϊκή αναγωγή.



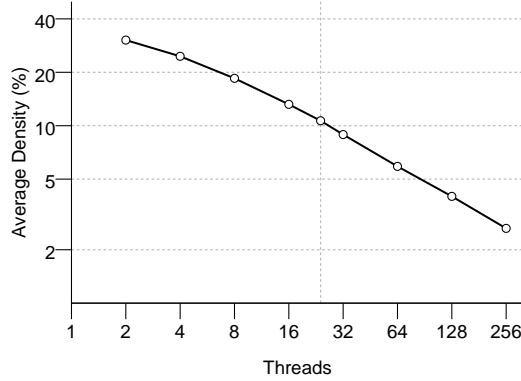
(iii) Αναγωγή ενεργού εύρους.



(iv) Δεικτοδοτημένη αναγωγή.

Σχήμα 6.3: Μέθοδοι τοπικών διανυσμάτων για την φάση αναγωγής του συμμετρικού πυρήνα $SrMV$. Παράδειγμα υλοποίησης του συμμετρικού $SrMV$ για $p = 4$ νήματα. Η απλοϊκή μέθοδος (6.3ii) χρησιμοποιεί απλά τέσσερα τοπικά διανύσματα, τα οποία στη συνέχεια αναγάγει στο τελικό διάνυσμα εξόδου. Η μέθοδος του ενεργού εύρους διανυσμάτων (6.3iii) χρησιμοποιεί $p - 1$ τοπικά διανύσματα γράφοντας μόνο τις πιθανώς συγκρουόμενες περιοχές. Το προτεινόμενο σχήμα δεικτοδότησης (6.3iv) χρησιμοποιεί $p - 1$ τοπικά διανύσματα και μία δομή δεικτοδότησης που υποδεικνύει μόνο τα πραγματικώς συγκρουόμενα στοιχεία.

6. Εκμετάλλευση συμμετρίας στους αραιούς πίνακες



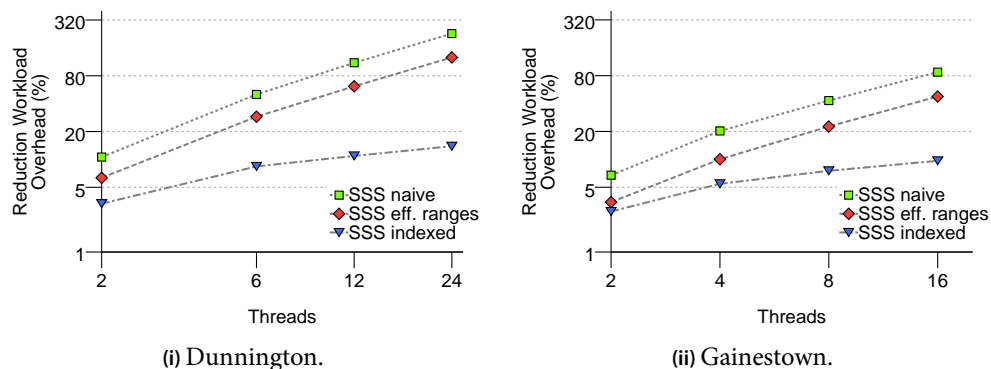
Σχήμα 6.4: Η πυκνότητα των ενεργών περιοχών των τοπικών διανυσμάτων. Τα τοπικά διανύσματα γίνονται ολοένα και πιο αραιά με την αύξηση του πλήθους των νημάτων, φθάνοντας ία πυκνότητα της τάξης του 2.7% στα 256 νήματα. Η κατακόρυφη γραμμή σημειώνει την πυκνότητα στα 24 νήματα του συστήματος Dunnington.

ρυνση του συνόλου εργασίας είναι

$$ws_{idx} = ws_{eff}d + 8 \underbrace{\frac{(p-1)Nd}{2}}_{\text{index size}} \quad (6.5)$$

$$\approx 8(p-1)Nd \quad (6.6)$$

Παρόλο που, θεωρητικά, η δεικτοδότηση των τοπικών διανυσμάτων δεν αποσυνδέει το κόστος της φάσης αναγωγής από το πλήθος των νημάτων, στην πράξη, η επίδραση της αύξησης του πλήθους του είναι σημαντικά εξασθενημένη. Αυτό οφείλεται στην αντίστροφη συσχέτιση του πλήθους των διανυσμάτων με την πυκνότητα των ενεργών περιοχών, η οποία τελικά τείνει να σταθεροποιήσει το κόστος της αναγωγής με την αύξηση των νημάτων, όπως παρουσιάζεται και γραφικά στο Σχήμα 6.5. Η επιβάρυνση της αναγωγής τόσο της απλοϊκής μεθόδου όσο και της μεθόδου του ενεργού εύρους αυξάνει γραμμικά με το πλήθος των νημάτων, υπερβαίνοντας σημαντικά την φάση του πολλαπλασιασμού στα 24 νήματα στο σύστημα Dunnington. Από την άλλη πλευρά, η επιβάρυνση του προτεινόμενου σχήματος δεικτοδότησης αυξάνει με πολύ αργότερο ρυθμό και τείνει να σταθεροποιηθεί περίπου στο 15% στα 24 νήματα εκτέλεσης. Πράγματι, με μία μέση πυκνότητα 11% σε αυτό τον αριθμό νημάτων, η επιβάρυνση του σχήματος δεικτοδότησης είναι περισσότερο από τέσσερις φορές μικρότερη σε σχέση με την περίπτωση της μεθόδου ενεργού εύρους. Στο σύστημα Gainestown, η επιβάρυνση της αναγωγής είναι μικρότερη, μιας και η συμφόρηση για εύρος ζώνης μνήμης δεν είναι τόσο έντονη,



Σχήμα 6.5: Η επιβάρυνση της φάσης αναγωγής (επί της σειριακής υλοποίησης του SSS) για τις τρεις θεωρούμενες μεθόδους τοπικών διανυσμάτων. Η επιβάρυνση της αναγωγής αυξάνει γραμμικά με το πλήθος των νημάτων για την απλοϊκή μέθοδο και για την μέθοδο ενεργού εύρους, υπερβαίνοντας τον χρόνο εκτέλεσης του κυρίως πολλαπλασιασμού. Αντίθετα, η επιβάρυνση της προτεινόμενης δεικτοδότησης των τοπικών διανυσμάτων τείνει να σταθεροποιηθεί με την αύξηση των νημάτων, προκαλώντας μία μάλλον μικρή επιβάρυνση της τάξης του 15%.

ωστόσο η ίδια συμπεριφορά παρατηρείται σε γενικές γραμμές και σε αυτή την αρχιτεκτονική.

Παραλληλοποίηση Η παραλληλοποίηση του φάσης αναγωγής στην περίπτωση της μεθόδου δεικτοδότησης βασίζεται επί της δομής δεικτοδότησης των τοπικών διανυσμάτων, καθώς αυτή καθορίζει τις πραγματικές πράξεις αναγωγής. Πιο συγκεκριμένα, αρχικά ταξινομούμε τους αριθμούς των δεικτών κατά αύξουσα σειρά βάσει του πεδίου *idx* και στην συνέχεια διαμοιράζουμε εξίσου την δομή δεικτοδότησης στα συμμετέχοντα νήματα. Ο μόνος περιορισμός στην διαδικασία διανομής είναι ο μη διαμοιρασμός οποιασδήποτε τιμής *idx* μεταξύ δύο νημάτων, ώστε να εξασφαλιστεί η ανεξαρτησία των πράξεων ανανέωσης στο τελικό διάνυσμα εξόδου.

6.2.3 Εναλλακτικές μέθοδοι

Η φάση της αναγωγής του συμμετρικού πυρήνα SpMV δεν μπορεί να αποφευχθεί, εκτός αν χρησιμοποιηθούν ατομικές πράξεις επί των στοιχείων του διανύσματος εξόδου. Οι Buluç et al. [2011] ακολουθούν μία ενδιαφέρουσα προσέγγιση υιοθετώντας μία υβριδική μέθοδο που χρησιμοποιεί ένα σταθερό αριθμό πράξεων αναγωγής και κάποιες μεμονωμένες ατομικές πράξεις επί ορισμένων

μη στοιχείων του διανύσματος εξόδου. Η βασική ιδέα της μεθόδου βασίζεται στην οργάνωση του πίνακα σε μεγάλες διαγωνίους μπλοκ (τρεις στην προτεινόμενη υλοποίηση), οι οποίες είναι αποθηκευμένες με το αποδοτικό σχήμα αποθήκευσης σε μπλοκ CSB (βλ. Κεφάλαιο 2, Ενότητα 2.2.3). Ο υπολογισμός του SpMV προχωρά ανεξάρτητα μεταξύ των τριών διαγωνίων μπλοκ¹ και γράφει τα ενδιάμεσα αποτελέσματα σε αποκλειστικά τοπικά διανύσματα, τα οποία εν τέλει θα αναχθούν στο τελικό διάνυσμα. Οι υπολογισμοί του SpMV επί μη μηδενικών στοιχείων εκτός των διαγωνίων μπλοκ, προχωρούν ανεξάρτητα και γράφουν απευθείας επί του διανύσματος εξόδου με χρήση ατομικών πράξεων. Το βασικό πλεονέκτημα αυτής της μεθόδου είναι η χρησιμοποίηση σταθερού αριθμού τοπικών διανυσμάτων, αποσυνδέοντας έτσι το κόστος της αναγωγής από το πλήθος των νημάτων. Ωστόσο, η επιβάρυνση της αναγωγής ακόμα και τριών τοπικών διανυσμάτων για πολύ μεγάλους και αραιούς πίνακες, όπου τα τοπικά διανύσματα είναι πιθανό να μην χωρούν στην συνολική κρυφή μνήμη του συστήματος, μπορεί να είναι ιδιαίτερα σημαντική. Επιπλέον, το κόστος των ατομικών πράξεων για τα μεμονωμένα μη μηδενικά στοιχεία μπορεί να καταστεί σημαντικό για πίνακες με μεγάλο εύρος ζώνης.

6.3 Το σχήμα CSX για συμμετρικούς πίνακες

Έχοντας βελτιστοποιήσει την φάση αναγωγής του συμμετρικού πυρήνα SpMV, προχωράμε ένα βήμα παραπέρα και βελτιστοποιούμε και την φάση του πολλαπλασιασμού, επεκτείνοντας το σχήμα υψηλής συμπίεσης CSX (βλ. Κεφάλαιο 5) προς υποστήριξη συμμετρικών πινάκων. Η βελτιστοποίηση αυτή είναι ανεξάρτητη της βελτιστοποίησης της φάσης αναγωγής, που συζητήθηκε στις προηγούμενες ενότητες. Αντίστοιχα με το σχήμα SSS, η συμμετρική έκδοση του CSX, στο εξής CSX-Sym, αποθηκεύει ξεχωριστά την κύρια διαγώνιο του πίνακα στον μονοδιάστατο πίνακα `diagValues` και τον κάτω τριγωνικό υποπίνακα στην τυπική μορφή του CSX. Όλες οι κανονικότητες που ανιχνεύονται στον κάτω τριγωνικό υποπίνακα έχουν ένα συμμετρικό ταίρι στον άνω τριγωνικό· για παράδειγμα, κάθε οριζόντια κανονικότητα στον κάτω τριγωνικό που ξεκινά από το στοιχείο (i, j) υποδηλώνει, επίσης, την ύπαρξη μία κατακόρυφης κανονικότητας στον άνω τριγωνικό που ξεκινά από την θέση (j, i) κ.ο.κ. Ο παραγόμενος κώδικας για το σχήμα CSX-Sym φροντίζει και για τον υπολογισμό της συμμετρικής κανονικότητας. Ο μόνος περιορισμός που επιβάλλουμε κατά την ανίχνευση των συμμετρικών κανονικοτήτων είναι ότι όλες οι πράξεις τους επί των διανυσμάτων εξόδου θα πρέπει είτε να γράφουν απευθείας στο τελικό διάνυσμα εξόδου είτε στο τοπικό διάνυσμα. Με άλλα λόγια, εάν η διαμέριση

¹ Ο υπολογισμός εντός κάθε διαγωνίου είναι επίσης παράλληλος και προχωρά σε δύο ξεχωριστές φάσεις.

$$A = \begin{pmatrix} 2.7 & 0.5 & 3.1 & 0 & 1.2 & 0 & 0 & 0 \\ 0.5 & 5.6 & 6.6 & 0 & 0 & 9.8 & 0 & 0 \\ 3.1 & 6.6 & 9.4 & 5.4 & 0 & 4.1 & 0 & 0 \\ 0 & 0 & 5.4 & 0.7 & 0 & 7.2 & 0 & 0 \\ 1.2 & 0 & 0 & 0 & 2.4 & 1.9 & 4.6 & 3.3 \\ 0 & 9.8 & 4.1 & 7.2 & 1.9 & 7.8 & 4.7 & 3.4 \\ 0 & 0 & 0 & 0 & 4.6 & 4.7 & 9.8 & 0 \\ 0 & 0 & 0 & 0 & 3.3 & 3.4 & 0 & 4.1 \end{pmatrix}$$

Σχήμα 6.6: Κωδικοποιούμενες υπό του CSX-Sym κανονικότητες σε ένα αραιό πίνακα 8×8 . Οι στικτές γραμμές υποδηλώνουν τις διαμερίσεις των νημάτων· τα σκιασμένα στοιχεία του άνω τριγωνικού υποπίνακα δεν αποθηκεύονται. Παρατηρήστε ότι η οριζόντια κανονικότητα της γραμμής 6 δεν κωδικοποιείται, καθώς η συμμετρική της κατακόρυφη κανονικότητα διασχίζει το όριο των διαμερίσεων των νημάτων.

του νήματος i ξεκινά στη γραμμή $start[i]$, οι συμμετρικές κανονικότητες που διασχίζουν αυτό το όριο δεν κωδικοποιούνται. Το Σχήμα 6.6 παρουσιάζει ένα παράδειγμα ανιχνευθέντων από το CSX-Sym κανονικοτήτων· παρατηρείστε ότι η οριζόντια κανονικότητα που ξεκινά στην γραμμή 6 δεν κωδικοποιείται, καθώς τα στοιχεία της συμμετρικής της διασχίζουν το σύνορο μεταξύ του τοπικού και του τελικού διανύσματος εξόδου. Παρά την μικρή μείωση της δυναμικής συμπίεσης του CSX-Sym, ο περιορισμός αυτός οδηγεί σε πιο αποδοτικό κώδικα, καθώς, σε αντίθετη περίπτωση, θα απαιτούνταν έλεγχοι ανά στοιχείο για την εξακρίβωση του διανύσματος προς το οποίο θα έπρεπε να ανακατευθυνθεί η πράξη ανανέωσης.

Δυνατότητες συμπίεσης Χρησιμοποιώντας όλη την τέχνη του CSX στη ανίχνευση και κωδικοποίηση κανονικοτήτων εντός ενός αραιού πίνακα, το σχήμα CSX-Sym έχει την δυνατότητα να συμπιέζει σημαντικά την αναπαράσταση του συμμετρικού πίνακα, επιτυγχάνοντας βαθμούς συμπίεσης πολύ κοντά στο θεωρητικό μέγιστο (67%), την στιγμή που το σχήμα SSS περιορίζεται σε λιγότερο από 50% στις περισσότερες των περιπτώσεων. Ο Πίνακας 6.1 τον βαθμό συμπίεσης αμφοτέρων των σχημάτων για τους 15 συμμετρικούς πίνακες της σουίτας μας.

6.4 Αξιολόγηση της επίδοσης

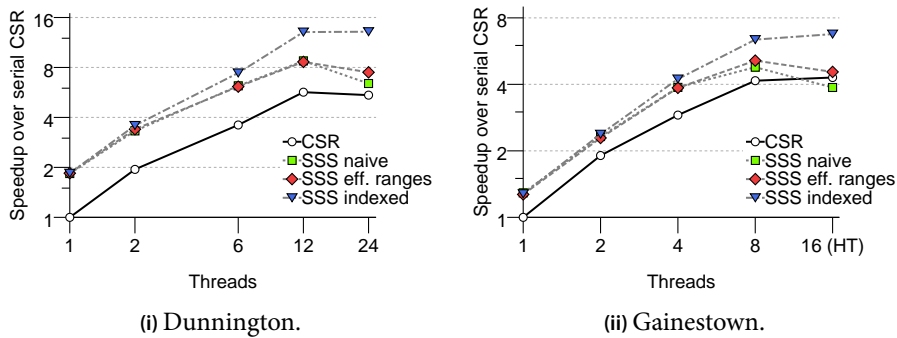
Στα επόμενα, αξιολογούμε την επίδοση των τριών μεθόδων τοπικών διανυσμάτων, συγκεκριμένα, της απλοϊκής μεθόδου, της μεθόδου ενεργού εύρους

6. Εκμετάλλευση συμμετρίας στους αραιούς πίνακες

Matrix	CSR Size (MiB)	C.R. (SSS)	C.R. (CSX-Sym)	C.R. (Max.)
parabolic_fem	44.06	45.4%	49.6%	63.6%
offshore	49.54	48.0%	56.1%	65.3%
conspH	69.10	49.5%	63.9%	66.4%
bmw7st_1	84.54	49.3%	64.4%	66.2%
G3_circuit	93.72	43.5%	60.2%	62.4%
thermal2	102.88	45.4%	53.4%	63.6%
m_t1	111.99	49.7%	65.3%	66.4%
bmwcra_1	122.38	49.5%	65.1%	66.4%
hood	124.08	49.3%	64.4%	66.2%
crankseg_2	162.16	49.8%	64.9%	66.6%
nd12k	162.88	49.9%	64.9%	66.6%
af_5_k101	202.77	49.1%	63.9%	66.0%
inline_1	423.25	49.5%	64.7%	66.4%
ldoor	536.04	49.3%	64.5%	66.2%
boneS10	638.28	49.5%	65.1%	66.3%

Πίνακας 6.1: Βαθμός συμπίεσης (C.R.) του συμμετρικού σχήματος CSX. Το συμμετρικό CSX μπορεί να επιτύχει σχεδόν βέλτιστο βαθμό συμπίεσης για συμμετρικούς αραιούς πίνακες.

και της μεθόδου δεικτοδότησης των τοπικών διανυσμάτων. Αξιολογούμε, επίσης, την επίδραση της παραλλαγής CSX-Sym του CSX επί της εκτέλεσης του συμμετρικού πυρήνα SpMV. Για την αξιολόγηση, χρησιμοποιούμε 15 συμμετρικούς και θετικά ορισμένους πίνακες της σουίτας μας και εστιάζουμε στα συστήματα Dunnington και Gainestown, ως αντιπροσωπευτικά των αρχιτεκτονικών SMP και NUMA, αντίστοιχα. Χρησιμοποιούμε την τεχνική πλήρωσης πυρήνα «share-nothing» σε αυτή την αξιολόγηση, ώστε να αναδείξουμε καθαρότερα το πρόβλημα της συμφόρησης εύρος ζώνης μνήμης κατά την φάση αναγωγής του συμμετρικού πυρήνα SpMV. Η παραλλαγή CSX-Sym ρυθμίζεται με ακριβώς τον ίδιο τρόπο όπως και το σχήμα CSX (βλ. Κεφάλαιο 5, Ενότητα 5.7). Για την ανάδειξη της πλήρους δυναμικής του κατά την σύγκρισή του με τα υπόλοιπα συμμετρικά σχήματα αποθήκευσης, παρουσιάζουμε τα αποτελέσματα επίδοσης με χρήση της πλήρους προεπεξεργασίας. Ωστόσο, κατά την αξιολόγηση της μεθόδου CG, χρησιμοποιούμε τον στατιστικό δειγματισμό όπως περιγράφηκε στο Κεφάλαιο 5, Ενότητα 5.7.3, και συμπεριλαμβανόμε τον χρόνο προεπεξεργασίας στην παρουσίαση των αποτελεσμάτων. Για περισσότερες πληροφορίες επί της σουίτας πινακων, των υπολογιστικών συστημάτων και των διαδικασιών πειραματισμού, ο αναγνώστης μπορεί να ανατρέξει στο Κεφάλαιο 3, Ενότητα 3.2.



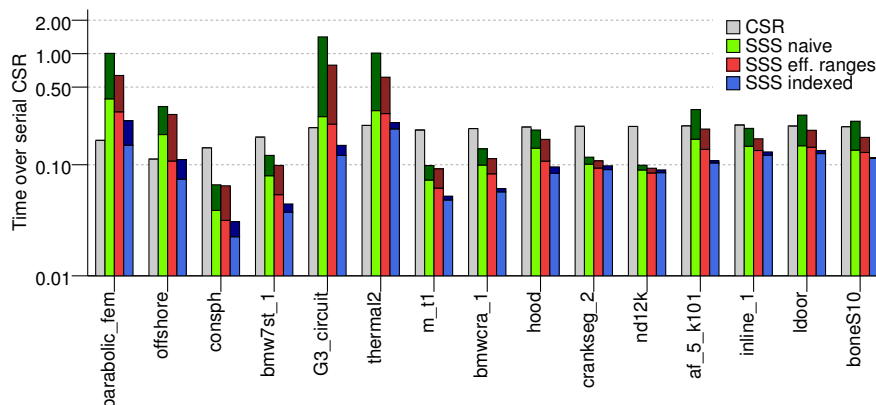
Σχήμα 6.7: Επιτάχυνση του συμμετρικού SpMV με τις διαφορετικές μεθόδους τοπικών διανυσμάτων. Η μέθοδος δεικτοδότησης είναι πιο αποδοτική επιτρέποντας την κλιμάκωση του SpMV.

6.4.1 Μέθοδοι τοπικών διανυσμάτων

Η βελτίωση της επίδοσης που επιτυγχάνεται από τις διαφορετικές θεωρούμενες μεθόδους τοπικών διανυσμάτων παρουσιάζεται στα διαγράμματα επιτάχυνσης του Σχήματος 6.7. Όλες οι μέθοδοι ξεκινούν με μία σημαντική βελτίωση της επίδοσης στην σειριακή εκτέλεση, ειδικά στο σύστημα Dunnington, όμως η απολοϊκή μέθοδος και η μέθοδος ενεργού εύρους κλιμακώνουν την επίδοσή τους με μικρότερο ρυθμό συγκριτικά με το βασικό σχήμα CSR. Η προσφερόμενη από αυτές τις μεθόδους βελτίωση της επίδοσης βαίνει συνεχώς μειούμενη καθώς αυξάνεται ο αριθμός των νημάτων και εξαλείφεται τελείως μόλις κορεστεί το εύρος ζώνης μνήμης. Λόγω του μειωμένου σύνολου εργασίας της, η μέθοδος ενεργού εύρους παρουσιάζει ελαφρώς καλύτερη συμπεριφορά σε αυτές τις περιπτώσεις, αλλά εξακολουθεί να μην επιτρέπει την κλιμάκωση του συμμετρικού SpMV. Στο σύστημα Gainestown, οι επίδοσεις της απλοϊκής μεθόδου και της μεθόδου ενεργού εύρους βρίσκονται πολύ κοντά σε αυτή του βασικού CSR στην οκτώ-νηματική εκτέλεση, ενώ επιδεινώνονται σημαντικά στην 16-νηματική.

Το όφελος επίδοσης της προτεινόμενης μεθόδου της δεικτοδότησης τοπικών διανυσμάτων είναι εμφανές σε αμφότερες τις αρχιτεκτονικές. Η σημαντικά μειωμένη επιβάρυνση της φάσης αναγωγής, η οποία παραμένει σχεδόν σταθερή σε σχέση με το πλήθος των νημάτων, επιτρέπει την κλιμάκωση της επίδοσης του συμμετρικού πυρήνα SpMV με τον ρυθμό της βασικής υλοποίησης του CSR, χωρίς να υποβαθμίζεται η βελτίωση της επίδοσης στις περιπτώσεις κορεσμού του εύρους ζώνης μνήμης. Πιο συγκεκριμένα, η προτεινόμενη μέθοδος δεικτοδότησης επιτυγχάνει 83.9% κέρδος επίδοσης επί της καλύτερης εκτέλεσης στο σύστημα Dunnington (12 νήματα) και 44% βελτίωση στο

6. Εκμετάλλευση συμμετρίας στους αραιούς πίνακες



Σχήμα 6.8: Διάρθρωση του χρόνου εκτέλεσης του συμμετρικού SpMV στα 24 νήματα του Dunnington. Η επιβάρυνση της αναγωγής (σκοτεινές περιοχές) μειώνεται σημαντικά με την χρήση της δεικτοδότησης των τοπικών διανυσμάτων.

σύστημα Gainestown. Συνολικά, ο συμμετρικός πυρήνας SSS με χρήση της μεθόδου δεικτοδότησης των τοπικών διανυσμάτων μπόρεσε να παράσχει μία βελτίωση επί της βασικής υλοποίησης του CSR της τάξης του $2\times$ στο σύστημα Dunnington και $1.5\times$ στο Gainestown στις πολυνηματικές εκτελέσεις. Αυτό αποτελεί ένα μάλλον σημαντικό επίτευγμα, καθώς η προτεινόμενη τεχνική επιτρέπει την αποδοτική εκμετάλλευση της συμμετρικής δομής ορισμένων πινάκων, που αλλιώς θα παρέμενε ανεκμετάλλευτη.

Με σκοπό την καλύτερη κατανόηση της σπουδαιότητας της φάσης αναγωγής στην εκτέλεση του συμμετρικού πυρήνα SpMV, παρουσιάζουμε στο Σχήμα 6.8 την διάρθρωση της εκτέλεσης όλων των θεωρουμένων μεθόδων αναγωγής για την 24-νηματική εκτέλεση στο Dunnington. Είναι φανερό ότι η προτεινόμενη μέθοδος της δεικτοδότησης των τοπικών διανυσμάτων μειώνει σημαντικά την επιβάρυνση της φάσης αναγωγής, διατηρώντας την σε πολύ χαμηλά επίπεδα. Η προτεινόμενη μέθοδος αναγωγής έχει και μία σημαντική παράπλευρη επίδραση: μειώνεται ο χρόνος του πολλαπλασιασμού. Αυτό οφείλεται κυρίως στην χαμηλότερη παρεμβολή στην χρήση της κρυφής μνήμης, λόγω του μειωμένου συνόλου εργασίας της δικής μας μεθόδου αναγωγής. Το μεγάλο σύνολο εργασίας των υπολοίπων μεθόδων είναι πιθανό να εκδιώξει χρήσιμα δεδομένα από την κρυφή μνήμη, προκαλώντας μία επιπλέον επιβάρυνση στην φάση του πολλαπλασιασμού κατά την διάρκεια της επόμενης επάνληψης.

Τέλος, είναι σημαντικό να ξεχωρίσουμε τις τέσσερις περιπτώσεις πινάκων (parabolic_fem, offshore, G3_circuit, thermal2) όπου η επίδοση του σχήματος

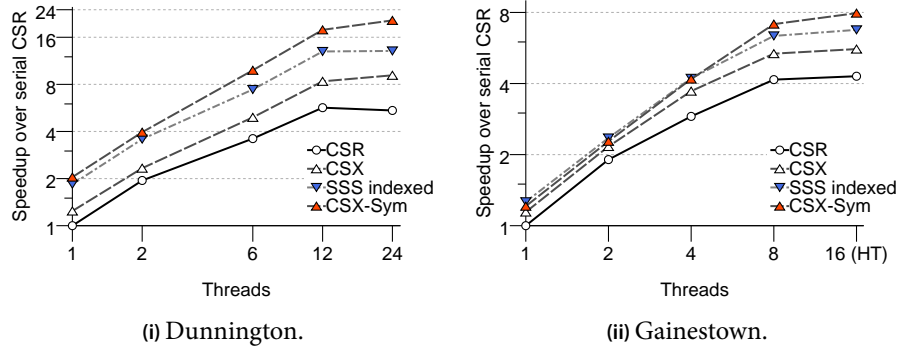
CSR είναι αντίστοιχη ή υπερβαίνει την επίδοση της μεθόδου δεικτοδότησης. Οι πίνακες αυτοί έχουν μεγάλο εύρος ζώνης, με πολλά τους στοιχεία να βρίσκονται σε μεγάλες αποστάσεις από την κύρια διαγώνιο, προκαλώντας με αυτό τον τρόπο σημαντική κίνηση προς την κύρια μνήμη. Ωστόσο, η μέθοδος της δεικτοδότησης των τοπικών διανυσμάτων φαίνεται να χειρίζεται αποδοτικά ακόμα και αυτές τις περιπτώσεις, παρουσιάζοντας μία μάλλον μικρή επιβάρυνση, την στιγμή που η εκτέλεση των υπολοίπων μεθόδων κυριαρχείται από το κόστος της αναγωγής.

6.4.2 Συμμετρικό CSX

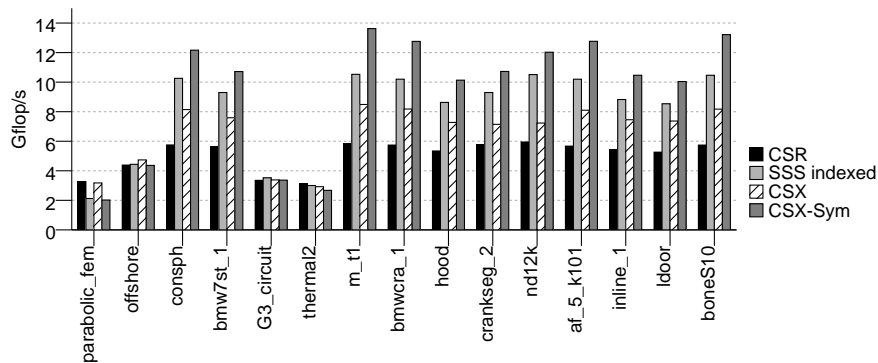
Η επίδοση της παραλλαγής CSX-Sym και του βελτιστοποιημένου σχήματος SSS (μέθοδος δεικτοδότησης τοπικών διανυσμάτων) σε σύγκριση με τις υλοποιήσεις των μη συμμετρικών CSR και CSX παρουσιάζεται στο Σχήμα 6.9. Χάρη στην ιδιαίτερα συμπεσμένα αναπαράστασή του, το σχήμα CSX-Sym επιτρέπει μία περαιτέρω βελτίωση της επίδοσης του πυρήνα κατά 43.4% επί του βελτιστοποιημένου σχήματος SSS στο σύστημα Dunnington, μία αρχιτεκτονική που επηρεάζεται άμεσα από το μέγεθος αναπαράστασης του πίνακα. Στο σύστημα Gainestown, όπου το εύρος ζώνης μνήμης είναι επαρκές, η διαφορά επίδοσης κλείνει στο 10% κατά μέσο, ωστόσο το CSX-Sym μπορεί ακόμα να προσφέρει όφελος στην επίδοση. Οι υλοποιήσεις των μη συμμετρικών CSX και CSR βρίσκονται αρκετά χαμηλότερα, ειδικά στην περίπτωση του Dunnington, όπου η συμφόρηση στη μνήμη είναι πιο εμφανής. Το CSX-Sym πέτυχε την εντυπωσιακή (για τα δεδομένα των υλοποιήσεων του SpMV) επιτάχυνση της τάξης του $21\times$ στο σύστημα Dunnington στα 24 νήματα και περίπου $8\times$ στο σύστημα Gainestown στην 16-νηματική εκτέλεση. Αυτή η αξιοσημείωτη συμπεριφορά οφείλεται στη χαμηλή επιβάρυνση του μεθόδου δεικτοδότησης των τοπικών διανυσμάτων, η οποία επιτρέπει την ανάδειξη του οφέλους από την προχωρημένη συμπίεση του σχήματος CSX και στην περίπτωση του συμμετρικού πυρήνα.

Προς την περαιτέρω κατανόηση της επίδοσης του σχήματος CSX-Sym, το Σχήμα 6.10 παρουσιάζει την ανά πίνακα επίδοση του συμμετρικού SpMV για τα τέσσερα θεωρούμενα σχήματα αποθήκευσης (CSR, SSS, CSX, CSX-Sym) στην 16-νηματική εκτέλεση στο σύστημα Gainestown. Το CSX-Sym επιτυγχάνει την απολύτως καλύτερη επίδοση, υπερβαίνοντας τα 10 Gflop/s, στους 11 από τους 15 πίνακες. Οι υπόλοιποι τέσσερις πίνακες πρόκειται για ακραίες περιπτώσεις με πολύ μεγάλο εύρος ζώνης, όπου κανένα συμμετρικό σχήμα δεν κατάφερε να βελτιώσει την επίδοση του CSR. Οι πίνακες μεγάλου εύρους ζώνης έχουν τα μη μηδενικά τους στοιχεία κατανομημένα κατά μήκος ολόκληρου του πίνακα, οδηγώντας σε μία μάλλον μικρή συχνότητα εμφάνισης κανονικοτήτων. Παραταύτα, με την εξαίρεση του `parabolic_fem`, ο οποίος έχει

6. Εκμετάλλευση συμμετρίας στους αραιούς πίνακες



Σχήμα 6.9: Επιτάχυνση του συμμετρικού SpMV με το σχήμα CSX-Sym. Το σχήμα CSX-Sym βελτιώνει περαιτέρω την επίδοση του συμμετρικού πυρήνα SpMV, ειδικά στις αρχιτεκτονικές SMP, όπου η συμφόρηση για εύρος ζώνης μνήμης είναι περισσότερο εμφανής.



Σχήμα 6.10: Ανά πίνακα επίδοση του σχήματος CSX-Sym στα 16 νήματα του συστήματος Gainestown. Η επίδοση του CSX-Sym υπερβαίνει τα 12 Gflop/s στις περιπτώσεις πινάκων με κανονική δομή, ενώ παραμένει κοντά στην επίδοση του βασικού CSR στις περιπτώσεις των λιγότερο κανονικών πινάκων.

πολύ ακανόνιστη δομή και πολύ μεγάλο εύρος ζώνης, το σχήμα CSX-Sym πέτυχε σχεδόν βέλτιστη επίδοση για σχεδόν όλους του πίνακες των ακραίων περιπτώσεων.

6.4.3 Πίνακες μειωμένου εύρους ζώνης

Η εκτέλεση του συμμετρικού πυρήνα SpMV επηρεάζεται ιδιαίτερα από το εύρος ζώνης του αραιού πίνακα. Πέρα από τα προβλήματα επίδοσης του τυπικού

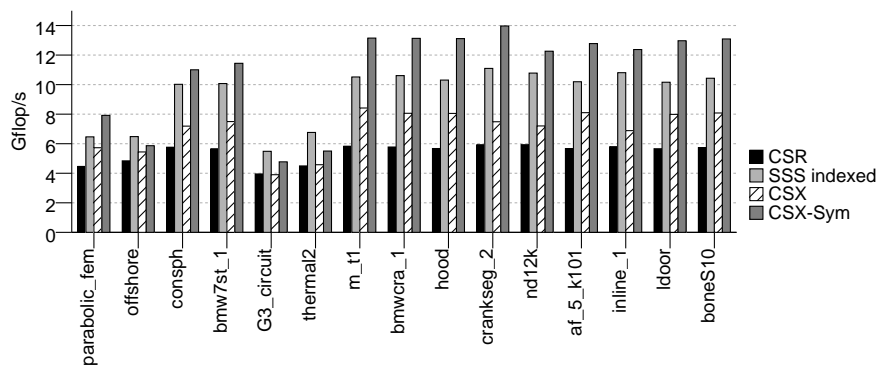
	Dunnington (24 threads)	Gainestown (16 threads)
CSR	24.8%	8.9%
CSX	50.2%	11.1%
SSS	73.6%	34.9%
CSX-Sym	85.3%	38.6%

Πίνακας 6.2: Βελτίωση της επίδοσης του συμμετρικού SpMV λόγω αναδιάταξης του πίνακα (χρήση αλγορίθμου RCM). Η επίδραση της αναδιάταξης είναι πολύ περισσότερο σημαντική στην περίπτωση των συμμετρικών σχημάτων, ενώ παράλληλα το σχήμα CSX τείνει να επωφελείται περισσότερο αυτής της βελτιστοποίησης, καθώς μπορεί να ανιχνεύει και να κωδικοποιεί περισσότερες κανονικότητες.

πυρήνα SpMV, π.χ., ακανόνιστες προσβάσεις στο διάνυσμα εισόδου, αυξημένη επιβάρυνση βρόχων, ανισορροπία φόρτου εργασίας κ.λπ. (βλ. Κεφάλαιο 3, Ενότητα 3.3 για περισσότερες λεπτομέρειες), οι πίνακες μεγάλου εύρους ζώνης τείνουν να αυξήσουν την αλληλεπίδραση των νημάτων του συμμετρικού πυρήνα κατά την φάση εγγραφής στο διάνυσμα εξόδου. Η φάση αναγωγής της απλοϊκής μεθόδου και της μεθόδου ενεργού εύρους δεν επηρεάζεται από αυτή την αλληλεπίδραση, καθώς όλες οι εγγραφές κατευθύνονται προς τα τοπικά διανύσματα. Το προτεινόμενο σχήμα δεικτοδότησης των τοπικών διανυσμάτων, ωστόσο, επηρεάζεται, καθώς η επιβάρυνση της φάσης αναγωγής εξαρτάται συγκεκριμένα από αυτή την αλληλεπίδραση, καθώς ανανεώνουμε στα τοπικά διανύσματα μόνο τα «συγκρουόμενα» στοιχεία. Η μείωση, επομένως, του εύρους ζώνης τέτοιων πινάκων μπορεί να αποδειχθεί ιδιαίτερα ωφέλιμη για τον συμμετρικό πυρήνα SpMV.

Με αρχικό σκοπό την μείωση της επιβάρυνσης της επικοινωνίας σε κατανεμημένες υλοποιήσεις του συμμετρικού πυρήνα SpMV, οι τεχνικές μείωσης του εύρους ζώνης του πίνακα προσπαθούν να φέρουν τα μη μηδενικά στοιχεία του πίνακα όσο το δυνατόν πλησιέστερα στην κύρια διαγώνιό του, αναδιατάσσοντας τις γραμμές και τις στήλες του. Η προφανής επίδραση αυτής της αναδιάταξης των μη μηδενικών στοιχείων στην εκτέλεση του SpMV είναι η ελαχιστοποίηση της αλληλεπίδρασης των διεργασιών/νημάτων. Σε μία κατανεμημένη υλοποίηση του SpMV, αυτό είναι ισοδύναμο με την μείωση του κόστους επικοινωνίας, καθώς λιγότερα δεδομένα των διανυσμάτων εισόδου και εξόδου θα πρέπει να ανταλλαχθούν μεταξύ των συμμετεχόντων επεξεργαστών. Στην περίπτωση μιας πολυνηματικής εκτέλεσης, αυτό είναι ισοδύναμο με την μείωση του κόστους της φάσης αναγωγής. Οι τεχνικές ελαχιστοποίησης του εύρους ζώνης του πίνακα έχουν επωφελή επίδραση και στην περίπτωση του μη συμμετρικού SpMV, καθώς ομογενοποιούν την κατανομή των

6. Εκμετάλλευση συμμετρίας στους αραιούς πίνακες



Σχήμα 6.11: Ανά-πίνακα επίδοση επί των αναδιατεταγμένων αραιών πινάκων (Gainestown, 16 νήματα). Το σχήμα CSX-Sym παρουσιάζει την καλύτερη επίδοση, υπερβαίνοντας το όριο των 12 Gflop/s στην πλειονότητα των πινάκων.

μη μηδενικών στοιχείων και οδηγούν σε καλύτερο σχήμα προσβάσεων στην μνήμη και καλύτερη εξισορρόπηση του φόρτου εργασίας (βλ. Κεφάλαιο 3). Τα σχήματα ανίχνευσης κανονικοτήτων, όπως το CSX, επωφελούνται περαιτέρω, καθώς η ανακατανομή των μη μηδενικών στοιχείων αυξάνει την πιθανότητα ανεύρεσης κανονικοτήτων (βλ. Κεφάλαιο 5). Η εικόνα αυτή επαληθεύεται και στην περίπτωση του συμμετρικού πυρήνα SpMV, όπως απεικονίζεται στον Πίνακα 6.2, ο οποίος παρουσιάζει την μέση βελτίωση στην επίδοση των διαφορετικών υλοποιήσεων του SpMV λόγω της αναδιάταξης του πίνακα. Στο σύστημα Dunnington, η βασική υλοποίηση του CSR κερδίζει 22% επιπλέον, ενώ το σχήμα CSX επωφελείται κατά 63%. Όπως ήταν αναμενόμενο, η επίδραση της αναδιάταξης του πίνακα επί του συμμετρικού πυρήνα είναι σαφώς πιο σημαντική, με την βελτίωση του SSS να υπερβαίνει το 90%, ενώ το σχήμα CSX-Sym κερδίζει μία επιτάχυνση της τάξης του $2\times$. Αντίστοιχη είναι η εικόνα και στο σύστημα Gainestown, αλλά τόσο η βελτίωση της επίδοσης όσο και οι διαφορές ανάμεσα στα διαφορετικά σχήματα είναι μειωμένες. Η συμπεριφορά αυτή –εμφανής σε όλες τις πειραματικές αξιολογήσεις σε αυτή την διατριβή– είναι τυπική των αρχιτεκτονικών NUMA, όπου η συμφόρηση για εύρος ζώνης μνήμης δεν είναι τόσο έντονη όσο στα συστήματα SMP. Τέλος, το Σχήμα 6.11 παρουσιάζει την απόλυτη επίδοση του SpMV για τους αναδιατεταγμένους πίνακες της σούιτας· είναι αξιοσημείωτο ότι η επίδοση του CSX-Sym υπερβαίνει τα 12 Gflop/s σε εννέα πίνακες.

6.4.4 Επίδραση επί της επαναληπτικής μεθόδου CG

Με σκοπό να αξιολογήσουμε την επίδραση της μεθόδου δεικτοδότησης των τοπικών διανυσμάτων και του σχήματος CSX-Sym στο πλαίσιο ενός επαναληπτικού επιλυτή, υλοποιήσαμε μία έκδοση της μεθόδου Conjugate Gradient (CG) χωρίς την χρήση preconditioner. Η μέθοδος CG είναι μία ευρέως χρησιμοποιούμενη επαναληπτική μέθοδος επίλυσης συμμετρικών και θετικά ορισμένων γραμμικών συστημάτων. Αποτελεί, επίσης, μέρος των παραλλήλων μετροπρογραμμάτων NAS, ως ένας τυπικός πυρήνας μη δομημένων υπολογισμών πλέγματος, για την αξιολόγηση της επίδοσης ενός παραλλήλου συστήματος σε συνθήκες ακανόνιστης επικοινωνίας και στο γινόμενο πίνακα-επίδιάνυσμα [Bailey et al., 1991].

```

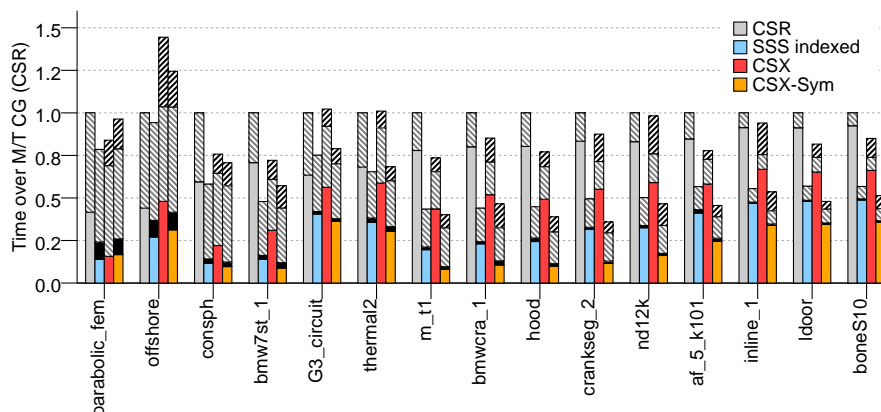
1: procedure CONJUGATEGRADIENT(A, b, x0)
  A: coefficient matrix
  b: target vector
  x0: initial approximate solution
2:   r0 ← b − Ax0
3:   p0 ← r0
4:   i ← 0                                     ▷ Iteration count
5:   loop
6:      $a_i \leftarrow \frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{p}_i^T \mathbf{A} \mathbf{p}_i}$            ▷ SpMV operation
7:     xi+1 ← xi + ai · pi
8:     ri+1 ← ri − ai · pi
9:     if |ri+1| is adequately small then
10:      break
11:      $b_i \leftarrow \frac{\mathbf{r}_{i+1}^T \mathbf{r}_{i+1}}{\mathbf{r}_i^T \mathbf{r}_i}$ 
12:     pi+1 ← ri+1 − bi · pi
13:     i ← i + 1
  return xi+1

```

Αλγόριθμος 6.2: Ο αλγόριθμος Conjugate Gradient χωρίς την χρήση preconditioner.

Το Σχήμα 6.12 παρουσιάζει την διάρθρωση του χρόνου εκτέλεσης της μεθόδου CG χρησιμοποιώντας τα μη συμμετρικά CSR και CSX και τα συμμετρικά SSS και CSX-Sym (με χρήση δεικτοδότησης των τοπικών διανυσμάτων) σχήματα αποθήκευσης. Παρουσιάζονται τα αποτελέσματα στα 24 νήματα του Dunnington για αναδιατεταγμένους κατά RCM πίνακες σε σύνολο 2048 επαναλήψεων της μεθόδου. Η πρώτη γενική παρατήρηση είναι ότι το κόστος των

6. Εκμετάλλευση συμμετρίας στους αραιούς πίνακες



Σχήμα 6.12: Διάρθρωση εκτέλεσης της μεθόδου CG σε αναδιατεταγμένους κατά RCM πίνακες χρησιμοποιώντας 24 νήματα στο σύστημα Dunnington σε σύνολο 2048 επαναλήψεων. Τα συμμετρικά σχήματα αποθήκευσης με χρήση της τεχνικής δεικτοδότησης των τοπικών διανυσμάτων προσφέρουν σημαντική βελτίωση της επίδοσης· το σχήμα CSX-Sym είναι επωφελές σε μεγάλους αραιούς πίνακες, όπου μπορεί να αποσβέσει το κόστος προεπεξεργασίας του. [Υπόμνημα διάρθρωσης χρόνου εκτέλεσης – έγχρωμες: SpMV, μαύρες: αναγωγή του SpMV, αντι-διαγώνιες: διανυσματικές πράξεις, διαγώνιες: προεπεξεργασία CSX.]

διανυσματικών πράξεων μπορεί να γίνει αρκετά σημαντικό σε μικρούς και αρκετά αραιούς πίνακες, όπως είναι οι *parabolic_fem*, *offshore* κ.λπ., υπερβαίνοντας ακόμα και το 50% του συνολικού χρόνου εκτέλεσης του πολυνηματικού πυρήνα της CG. Η CG εκτελεί πολλές διανυσματικές πράξεις, συμπεριλαμβανομένων σημειακών γινομένων, κατά την διάρκεια μιας επανάληψης (βλ. Αλγόριθμο 6.2), αλλά μόνο μία πράξη SpMV. Επομένως, για μικρούς πίνακες που χωρούν στην συνολική κρυφή μνήμη του συστήματος, η επιβάρυνση των διανυσματικών πράξεων μπορεί εύκολα να κυριαρχήσει τον συνολικό χρόνο εκτέλεσης του επιλυτή. Με την εξαίρεση των πολύ αραιών πινάκων *parabolic_fem* and *offshore*, όπου ο υπολογισμός κυριαρχείται από τις διανυσματικές πράξεις, η μέθοδος CG επωφελείται εξαιρετικά από τα συμμετρικά σχήματα αποθήκευσης, καταγράφοντας περισσότερη του 50% βελτίωση της συνολικής της επίδοσης για μεγάλους αραιούς πίνακες. Το σχήμα CSX-Sym παρεμποδίζεται από το κόστος προεπεξεργασίας του στους μικρούς πίνακες και παρέχει αντίστοιχη ή ελαφρώς χαμηλότερη επίδοση σε σχέση με το σχήμα SSS με χρήση της τεχνικής δεικτοδότησης των τοπικών διανυσμάτων. Σε μεγαλύτερους πίνακες, ωστόσο, το σχήμα CSX-Sym αποσβάνει το κόστος προεπεξεργασίας του και επιτυγχάνει περαιτέρω βελτίωση της επίδοσης της μεθόδου CG.

6.5 Περίληψη

Στο κεφάλαιο αυτό, εστίασαμε στην εκμετάλλευση της συμμετρίας των μη μηδενικών στοιχείων των συμμετρικών αραιών πινάκων, με σκοπό την επιτάχυνση της επίδοσης του πυρήνα SpMV. Με την αποθήκευση του μισού αριθμού των μη μηδενικών στοιχείων του αρχικού πίνακα (κάτω τριγωνικός και κύρια διαγώνιος), τα συμμετρικά σχήματα αποθήκευσης μπορούν να ωφελήσουν σημαντικά την επίδοση του SpMV, λόγω της ελάφρυνσης του υποσυστήματος μνήμης. Ωστόσο, ο πολυνηματικός πυρήνας SpMV θα πρέπει να επιλύσει επιτυχώς την εξάρτηση δεδομένων RAW στο διάνυσμα εξόδου, που προκαλείται από τις πράξεις του SpMV στο συμμετρικό (άνω τριγωνικό) τμήμα του πίνακα. Η συνηθέστερη λύση περιλαμβάνει την χρήση τοπικών, ανά νήμα διανυσμάτων, όπου αποθηκεύονται τα ενδιάμεσα αποτελέσματα του SpMV. Σε δεύτερη φάση, τα διανύσματα αυτά ανάγονται παράλληλα στο τελικό διάνυσμα εξόδου. Το κυριο πρόβλημα επίδοσης της φάσης αυτής είναι η γραμμική συσχέτιση της επιβάρυνσής της με το πλήθος των νημάτων, πράγμα αποκλείει την κλιμάκωση του πυρήνα. Βασιζόμενοι στην παρατήρηση ότι τα τοπικά διανύσματα είναι αρκετά αραιά και η «αραιότητά» τους αυξάνεται με την αύξηση των νημάτων, προτείναμε ένα σχήμα δεικτοδότησης των τοπικών διανυσμάτων, το οποίο επιτρέπει την αναγωγή μόνο των συγκρουομένων στοιχείων του διανύσματος εξόδου, μειώνοντας έτσι την συμφόρηση για εύρος ζώνης μνήμης κατά την φάση της αναγωγής. Βελτιστοποιούμε περαιτέρω την εκτέλεση του συμμετρικού SpMV με την επέκταση του σχήματος CSX για υποστήριξη συμμετρικών πινάκων. Η λεγόμενη CSX-Sym συμμετρική παραλλαγή του CSX επιτυγχάνει σχεδόν βέλτιστη συμπίεση του αραιού πίνακα και, σε συνδυασμό, με το σχήμα δεικτοδότησης των τοπικών διανυσμάτων, επιταχύνει κατά $2\times$ τον συμμετρικό πυρήνα SpMV, ενώ η επίδραση των προτεινομένων βελτιστοποιήσεων επιτρέπει την βελτίωση κατά περισσότερο από 50% της επαναληπτικής μεθόδου CG για μεγάλους συμμετρικούς πίνακες.

Θέματα ενεργειακής απόδοσης

Η ενεργειακή απόδοση των υπολογιστικών συστημάτων αποτελεί θέμα ιδιαίτερου ενδιαφέροντος για την κοινότητα των ενσωματωμένων συστημάτων εδώ και αρκετά χρόνια. Καθώς το επίπεδο ολοκλήρωσης των ψηφιακών κυκλωμάτων CMOS οδηγείται προς τα όριά του, τα θέματα ενεργειακής απόδοσης κερδίζουν ολοένα το ενδιαφέρον της κοινότητας HPC από το επίπεδο του υλικού μέχρι το επίπεδο του λογισμικού. Η κατανάλωση ισχύος λόγω διαρροών γίνεται ολοένα και πιο σημαντική καθώς η κλίμακα ολοκλήρωσης συρρικνώνεται στο επίπεδο των λίγων νανομέτρων και τείνει να κυριαρχήσει στην συνολική κατανάλωση ισχύος του τσιπ. Τεχνικές για την δυναμική απενεργοποίηση αδρανών επεξεργαστικών στοιχείων έρχονται να συμπληρώσουν πιο παραδοσιακές τεχνικές, όπως η Δυναμική Ρύθμιση Τάσης και Συχνότητας (Dynamic Voltage and Frequency Scaling – DVFS), σε μία προσπάθεια για την ελαχιστοποίηση της συνολικής κατανάλωσης ισχύος του επεξεργαστή. Ωστόσο, η επίδοση των εφαρμογών HPC είναι κρίσιμος και καθοριστικός παράγοντας και θα πρέπει να παραμείνει στο προσκήνιο. Η εξισορρόπηση αυτής της διεκυστίνας, επομένως, αποτελεί βασικό στοιχείο για την υλοποίηση ενεργειακά αποδοτικών συστημάτων και εφαρμογών υψηλών επιδόσεων.

Στο κεφάλαιο αυτό, πραγματοποιούμε ένα πρώτο βήμα προς την αναγνώριση και διερεύνηση της διεκυστίνας επίδοσης-κατανάλωσης ενέργειας του πυρήνα SpMV στις σύγχρονες πολυπύρηνες αρχιτεκτονικές, μεταβάλλοντας την συχνότητα του επεξεργαστή και την τοποθέτηση των νημάτων στους διαθέσιμους πυρήνες. Χρησιμοποιώντας την έννοια του κατά Pareto βελτίστου, χαρακτηρίζουμε τους διαφορετικούς συμβιβασμούς και προτείνουμε μία μεθοδολογία για την επιλογή των ρυθμίσεων εκτέλεσης (συχνότητα επεξεργαστή, τοποθέτηση νημάτων) του SpMV που οδηγούν σε βέλτιστους συμβιβασμούς επίδοσης-κατανάλωσης ενέργειας για τον εκάστοτε πίνακα εισόδου. Το κεφάλαιο αυτό ανοίγει ένα σύνολο θεμάτων, τα οποία θα πρέπει να απαντηθούν επακριβέστερα στο μέλλον και αποτελεί τροφή για περαιτέρω σκέψεις επί της αξιολόγησης της πραγματικής επίδρασης των ενεργειακών βελτιστοποιήσεων σε επίπεδο λογισμικού σε ένα σύστημα HPC.

7.1 Βασικές αρχές κατανάλωσης ισχύος του επεξεργαστή

7.1.1 Πηγές κατανάλωσης ισχύος

Η κατανάλωση ισχύος μιας συσκευής CMOS στο στοιχειώδες επίπεδο των πυλών του τρανζίστορ υπολογίζεται ως το άθροισμα τριών κυρίων στοιχείων: τις απώλειες μεταγωγής (switching loss), τις απώλειες διαρροών (leakage) και τις απώλειες βραχυκυκλώματος (short-circuit) [Brooks et al., 2000; Kaxiras και Martonosi, 2008]. Πιο συγκεκριμένα, η κατανάλωση ισχύος P μίας συσκευής CMOS προσεγγίζεται από την ακόλουθη σχέση:

$$P = \frac{1}{2} CV^2 \alpha f + P_{\text{leakage}} + P_{\text{short-circuit}} \quad (7.1)$$

Ο πρώτος όρος, γνωστός επίσης και ως *δυναμική κατανάλωση ισχύος*, πρόκειται για την κατανάλωση ισχύος λόγω της μεταγωγής των τρανζίστορ και σχετίζεται άμεσα με την χωρητικότητα του κυκλώματος C , την τάση τροφοδοσίας V , την συχνότητα ρολογιού f της συσκευής και τον *συντελεστή ενεργοποίησης* α . Η χωρητικότητα του κυκλώματος είναι μία παράμετρος σχεδίασης και εξαρτάται σε σημαντικό βαθμό από την καλωδίωση των επιμέρους στοιχείων του τσιπ. Ο συντελεστής ενεργοποίησης είναι ένα κλάσμα με εύρος τιμών $[0, 1]$, που υποδηλώνει πόσο συχνά η συσκευή διατρέχεται από το σήμα του ρολογιού, δηλαδή, πόσο συχνά στην πράξη χρησιμοποιείται· όταν είναι αδρανείς, οι συσκευές αποσυνδέονται από την πηγή ρολογιού, ώστε να εξοικονομείται ενέργεια (*clock gating*). Οι *απώλειες διαρροών* πρόκεινται για την κατανάλωση ισχύος λόγω διαρροών ρεύματος κατά την διάρκεια που η πύλη του τρανζίστορ είναι ανενεργή. Παρόλο που τα τρανζίστορ θεωρούνται ως ιδανικοί διακόπτες on-off, στην πράξη, μία μικρή ποσότητα ρεύματος τα διαρρέει κατά την κατάσταση «off», προκαλώντας απώλεια ισχύος. Οι *απώλειες βραχυκυκλώματος* ή *απώλειες glitching* πρόκεινται για την κατανάλωση ισχύος λόγω του στιγμιαίου βραχυκυκλώματος κατά την μετάβαση από την κατάσταση «on» στην κατάσταση «off», που οφείλεται στους μη ιδανικούς χρόνους μετάβασης των τρανζίστορ.

Η δυναμική κατανάλωση ισχύος αποτελούσε την κύρια πηγή κατανάλωσης ισχύος των συσκευών CMOS για αρκετά χρόνια. Η τετραγωνική συσχέτισή της με την τάση τροφοδοσίας οδήγησε σε σημαντική εξοικονόμηση ισχύος, καθώς η νέα τεχνολογία επέτρεπε την ασφαλή μείωση της τάσης. Η συχνότητα του ρολογιού έχει διπλή επίδραση επί της δυναμικής κατανάλωσης ισχύος: πέρα από την άμεση συσχέτισή της με την συνολική κατανάλωση ισχύος, όπως αυτό καθορίζεται από την σχέση (7.1), η διατήρηση υψηλότερης συχνότητας απαιτεί μία αναλογικά υψηλότερη τάση τροφοδοσίας, οδηγώντας έτσι σε μία κυβική συσχέτιση μεταξύ της συχνότητας και της δυναμικής κατανάλωσης ισχύος [Brooks et al., 2000; Kaxiras και Martonosi, 2008]. Η σημαντική αυ-

τή επίδραση της συχνότητας επί της συνολικής κατανάλωσης ισχύος οδήγησε στην ευρύτερη διάδοση της δυναμικής διαχείρισης της τάσης και της συχνότητας, με σκοπό την μείωση της συνολικής κατανάλωσης με τους ελάχιστους συμβιβασμούς στην επίδοση. Οι τεχνικές *Δυναμικής Κλιμάκωσης της Τάσης και της Συχνότητας* (*Dynamic Voltage and Frequency Scaling – (DVFS)*) επιτρέπουν την επιλεκτική κλιμάκωση αμφοτέρων της τάσης και της συχνότητας κατά τις λιγότερο εστιασμένες επί της CPU φάσεις μιας εφαρμογής, π.χ., φάσεις προσβάσεων στη κύρια μνήμη, ώστε να επιτευχθεί υψηλή επίδοση με ένα χαμηλότερο προϋπολογισμό κατανάλωσης ισχύος [Xie et al., 2003; Choi et al., 2004]. Πιο προχωρημένες τεχνικές επιτρέπουν τον διαχωρισμό του επεξεργαστή σε πολλές πεδία ρύθμισης της τάσης, επιτρέποντας με αυτό τον τρόπο την επιλεκτική κλιμάκωση της τάσης σε διαφορετικά τμήματα του επεξεργαστή, ανάλογα με τις ανάγκες της τρέχουσας εφαρμογής [Semeraro et al., 2002; Herbert και Marculescu, 2007; Mattson et al., 2010].

Καθώς η κλίμακα ολοκλήρωσης συρρικνώνεται σε επίπεδα λίγων νανομέτρων, οι απώλειες διαρροών γίνονται σημαντικός παράγοντας της συνολικής κατανάλωσης ισχύος των συγχρόνων επεξεργαστών. Υπάρχουν δύο τύποι απωλειών διαρροών: οι *διαρροές κατωφλίου* (sub-threshold leakage) και *διαρροές πύλης* (gate leakage). Οι διαρροές κατωφλίου οφείλονται στο ηλεκτρικό ρεύμα που διατρέχει το τρανζίστορ στην κατάσταση «off», δηλαδή, όταν η τάση του βρίσκεται χαμηλότερα ενός κατωφλίου που υποδηλώνει την κατάσταση «off». Οι διαρροές κατωφλίου αυξάνουν εκθετικά με την μείωση της τάσης κατωφλίου [Kaxiras και Martonosi, 2008]. Η μείωση της τάσης κατωφλίου έρχεται συνήθως ως αποτέλεσμα μιας ανάλογης μείωσης της τάσης τροφοδοσίας V της σχέσης (7.1), καθώς απαιτείται μία ελάχιστη διαφορά τάσης μεταξύ των καταστάσεων «on» και «off» του τρανζίστορ. Οι διαρροές πύλης, από την άλλη πλευρά, οφείλονται στην άμεση διάσχιση ηλεκτρονίων διαμέσου του μονωτικού που διαχωρίζει την πύλη από τον δίαυλο του τρανζίστορ. Καθώς το στρώμα του μονωτικού γίνεται ολοένα και πιο λεπτό, λόγω της μείωσης της κλίμακας ολοκλήρωσης, οι διαρροές πύλης αυξάνονται εκθετικά και κατ' επέκταση και οι συνολικές απώλειες διαρροών [Borkar, 1999; Kim et al., 2003; Agarwal et al., 2006].

Είναι φανερό ότι η αρχιτεκτονική υπολογιστών προσκρούσει σε ένα «τοίχο κατανάλωσης ισχύος», που ορθώνεται κυρίως λόγω της τρομακτικής αύξησης των απωλειών διαρροών [Ramirez, 2011; Ahmed και Schuegraf, 2011]. απαιτούνται, επομένως, δραστικές μετατοπίσεις της τεχνολογίας, ώστε να διατηρηθεί ζωντανός ο νόμος του Moore πέραν του ορίου των 22 nm [Hisamoto et al., 2000; Yang et al., 2004; Skotnicki et al., 2005; Hu, 2012]. Οι σύγχρονες μικροαρχιτεκτονικές αντιμετωπίζουν το πρόβλημα των απωλειών διαρροών ενσωματώνοντας προχωρημένες τεχνικές διαχείρισης της κατανάλωσης ισχύος. Εκτός των τεχνικών δυναμικής κλιμάκωσης της τάσης και της συχνότητας, οι

7. Θέματα ενεργειακής απόδοσης

οποίες υφίστανται εδώ και κάποια χρόνια [Semeraro et al., 2002], οι σύγχρονοι επεξεργαστές επιτρέπουν επίσης την δυναμική απενεργοποίηση αδρανών πυρήνων. Για παράδειγμα, ο τελευταίος οκτα-πύρηνος, τετρα-νηματικός επεξεργαστής Power7 της IBM μπορεί να φέρει ένα πυρήνα σε μία από δύο καταστάσεις «ύπνωσης», ώστε να εξοικονομήσει όσο το δυνατόν περισσότερη ενέργεια από τους αδρανείς πυρήνες [Kalla et al., 2010]. Αντίστοιχα, η μικροαρχιτεκτονική Sandy Bridge της Intel μεταβάλλει δυναμικά την κατάσταση ισχύος διαφορετικών τμημάτων του επεξεργαστή, με σκοπό να βελτιστοποιήσει την συνολική κατανάλωση ισχύος, ενώ προσφέρει στον χρήστη και μία προγραμματιστική διεπαφή για την παρακολούθηση της κατανάλωσης ισχύος, ώστε εκμεταλλευόμενό την το λογισμικό να μπορεί να μειώσει την συνολική κατανάλωση ενέργειας [Rotem et al., 2012].

Κλείνοντας την συζήτηση επί των πηγών κατανάλωσης ισχύος στις σύγχρονες μικροαρχιτεκτονικές, θα πρέπει να γίνει μία ειδική μνεία στις κρυφές μνήμες. Η ενσωμάτωση πολλών, γρήγορων πυρήνων εντός του ίδιου επεξεργαστή έθεσε ένα σημαντικό φορτίο επί του υποσυστήματος μνήμης. Τόσο με όρους καθυστέρησης όσο και εύρους ζώνης, η κύρια μνήμη δεν μπορεί να ακολουθήσει τους εξαιρετικά υψηλούς ρυθμούς κατανάλωσης δεδομένων των συγχρόνων επεξεργαστών· και η διαφορά αυτή τείνει να αυξάνεται εκθετικά με κάθε νέα γενιά επεξεργαστών [Wulf και McKee, 1995]. Οι μεγάλες κρυφές μνήμες έρχονται, επομένως, για να καλύψουν και να αποκρύψουν αυτό το κενό επίδοσης και δεν είναι πλέον παράλογο να συναντά κανείς κρυφές μνήμες της τάξης των 24 MiB ανά επεξεργαστή. Οι τόσο μεγάλες κρυφές μνήμες, ωστόσο, καταλαμβάνουν ένα σημαντικό ποσοστό της συνολικής επιφάνειας του τσιπ, συμβάλλοντας κατ' αυτό τον τρόπο σημαντικά στην συνολική κατανάλωση ισχύος λόγω διαρροών¹ [Kaxiras και Martonosi, 2008]. Αρκετές τεχνικές έχουν προταθεί για την μείωση των απωλειών διαρροών των κρυφών μνημών, ξεκινώντας από το επίπεδο των κελιών μνήμης [Powell et al., 2000] και φθάνοντας σε τεχνικές απενεργοποίησης ολοκλήρων τμημάτων της κρυφής μνήμης [Yang et al., 2001] ή συγκεκριμένων αχρησιμοποίητων γραμμών της [Kaxiras et al., 2001; Flautner et al., 2001]. Οι τεχνικές αυτές έχουν σιγά-σιγά αρχίσει να καθιερώνονται, επιτρέποντας την σχεδίαση ενεργειακά αποδοτικών κρυφών μνημών που υπερβαίνουν ακόμα και τα 20 MiB [Chang et al., 2009; Huang et al., 2012].

¹ Η δυναμική κατανάλωση ισχύος των πολύ μεγάλων κρυφών μνημών δεν είναι τόσο σημαντική, καθώς βρίσκονται συνήθως σε διαφορετικό πεδίο τάση και συχνότητας από τον υπόλοιπο επεξεργαστή, «τρέχοντας» σε πολύ χαμηλότερες συχνότητες.

7.1.2 Γινόμενα Energy-Delay

Από απόψεως λογισμικού, η ελαχιστοποίηση της κατανάλωσης ισχύος της υποκείμενης αρχιτεκτονικής είναι η μία όψη του νομίσματος· η επίδοση είναι η άλλη και είναι, επίσης, πολύ σημαντική. Παρά το αντικρουόμενο των στόχων της υψηλής επίδοσης και της χαμηλής κατανάλωσης ισχύος ενός υπολογιστικού συστήματος, η υψηλή επίδοση δεν σημαίνει απαραίτητα αναποτελεσματικότητα στο επίπεδο της κατανάλωσης ενέργειας. Η κατανάλωση ισχύος δεν είναι τίποτα άλλο από τον ρυθμό με τον οποίο ένα υπολογιστικό σύστημα καταναλώνει ενέργεια· επομένως, όσο πιο γρήγορος είναι ο υπολογισμός, τόσο λιγότερο είναι και το συνολικό ποσό της καταναλούμενης ενέργειας. Σε τελική ανάλυση, αυτό που πληρώνει κανείς είναι η κατανάλωση ενέργειας. Για τον λόγο αυτό, η μετρική $\frac{\text{flop/s}}{\text{W}}$ αρχίζει να χρησιμοποιείται ευρέως για τον χαρακτηρισμό της ενεργειακής απόδοσης ενός υπολογιστικού συστήματος. Την στιγμή της συγγραφής αυτής της διατριβής, ο γρηγορότερος υπερυπολογιστής στον κόσμο, επιτυγχάνοντας την εξαιρετικά υψηλή επίδοση των 17.6 Pflop/s, είναι επίσης και ο τρίτος πιο ενεργειακά αποδοτικός², επιτυγχάνοντας επίδοση 2.1 Gflop/s ανά Watt καταναλουμένης ισχύος [Top500, 2012; Green500, 2012].

Μία προσεκτικότερη ματιά στην μετρική της επίδοσης ανά Watt αποκάλυπτει ότι δεν είναι τίποτα περισσότερο από το αντίστροφο της κατανάλωσης ενέργειας. Ένας κοινός τρόπος για την αξιολόγηση του συμβιβασμού επίδοσης-κατανάλωσης ενέργειας της σχεδίασης ενός συστήματος είναι τα *the γινόμενα energy-delay* ή ED^n , $n \geq 0$. Για $n = 0$, το γινόμενο energy-delay αντιστοιχεί στην κατανάλωση ενέργειας του συστήματος, ενώ μεγαλύτερες τιμές του n ευνοούν την υψηλότερη επίδοση. Όπως θα δούμε στην Ενότητα 7.2, τα γινόμενα energy-delay αποτελούν βέλτιστους συμβιβασμούς σχεδίασης υπό την έννοια του κατά Pareto βελτίστου, δηλαδή, αποκλείεται η ύπαρξη σχεδίασης που να μην υποβιβάζει τουλάχιστον ένα από τους στόχους της επίδοσης ή της κατανάλωσης ενέργειας.

7.1.3 Ενεργειακή απόδοση από πλευράς λογισμικού

Το λογισμικό μπορεί να παίξει σημαντικό ρόλο στην μείωση της συνολικής κατανάλωσης ισχύος ενός υπολογιστικού συστήματος. Η δυναμική κατανάλωση ισχύος του επεξεργαστή επηρεάζεται άμεσα από την χρησιμοποίηση των δομικών του μονάδων κατά τις διαφορετικές φάσεις μιας εφαρμογής. Από πλευράς κατανάλωσης ισχύος, η χρησιμοποίηση μιας μονάδας υλικού μπορεί να θεωρηθεί ως ένδειξη του συντελεστή ενεργοποίησής της. Βασιζόμενοι σε αυτή την υπόθεση, οι Isci και Martonosi [2003] δημιούργησαν ένα αρκετά ακριβές μοντέλος πρόβλεψης της κατανάλωσης ισχύος των βασιζόμενων στην μικρο-

² Ο πιο ενεργειακά αποδοτικός επιτυγχάνει επίδοση 2.5 Gflop/s ανά Watt.

αρχιτεκτονική Intel Netburst μικροεπεξεργαστών [Koufaty και Marr, 2003]. Οι συγγραφείς συμπεραίνουν τον συντελεστή χρησιμοποίησης 22 μονάδων υλικού (κρυφές μνήμες, TLBs, μονάδες εκτέλεσης κ.λπ.), υπολογίζοντας την χρησιμοποίησή τους βάσει των μετρητών επίδοσης του υλικού (hardware performance counters), και στη συνέχεια χρησιμοποιούν ένα σταθμισμένο άθροισμα των ανά μονάδα προβλέψεων, ώστε να υπολογίσουν την συνολική κατανάλωση ισχύος του επεξεργαστή. Μία πιο λεπτομερής και γενικευμένη προσέγγιση για την πρόβλεψη της κατανάλωσης ισχύος ενός σύγχρονου επεξεργαστή είναι η υποδομή McPAT [Sheng et al., 2009]. Η υποδομή McPAT προβλέπει επίσης την κατανάλωση ισχύος μέσω της χρήση των μετρητών επίδοσης, ωστόσο, αντίθετα με το μοντέλο των Isci και Martonosi [2003], δεν περιορίζεται σε κάποια συγκεκριμένη αρχιτεκτονική. Αντίθετα, χρησιμοποιεί μοντέλα χαμηλού επιπέδου για την προσομοίωση βασικών μονάδων ενός πολυπύρηνου επεξεργαστή, επιτρέποντας έτσι την εξερεύνηση συμβιβασμών σχεδίασης για νέες αρχιτεκτονικές.

Η συλλογή μιας πληθώρας μετρικών επίδοσης, με σκοπό την ακριβή εκτίμηση της κατανάλωσης ισχύος, μπορεί να απαιτεί πολλαπλές εκτελέσεις της υπό εξέταση εφαρμογής, καθώς το πλήθος των καταχωρητών παρακολούθησης της επίδοσης (performance monitoring registers) είναι αρκετά περιορισμένο στις τρέχουσες μικροαρχιτεκτονικές. Κατά συνέπεια, οι μέθοδοι αυτοί δεν είναι κατάλληλες για online εκτίμηση της κατανάλωσης ισχύος μίας τρέχουσας εφαρμογής. Οι Curtis-Maury et al. [2008] ακολουθούν μία υβριδική offline/online προσέγγιση για την πρόβλεψη των ρυθμίσεων εκτέλεσης (πλήθος νημάτων και συχνότητα πυρήνων), η οποία παρέχει μία βέλτιστη ισορροπία μεταξύ επίδοσης και κατανάλωσης ενέργειας. Το σύστημα εκτέλεσης που προτείνουν συλλέγει την πληροφορία των μετρικών επίδοσης για την τρέχουσα εφαρμογή και τροφοδοτεί ένα μοντέλο παλινδρόμησης (regression), εκπαιδευθέν offline, το οποίο προβλέπει τις βέλτιστες ρυθμίσεις εκτέλεσης για την επόμενη φάση της εφαρμογής. Επί της ουσίας, η τεχνική αυτή προσπαθεί να καταχωρήσει τους ελάχιστους δυνατούς πόρους (συχνότητα, πλήθος νημάτων) σε μία εφαρμογή έτσι, ώστε να εκτελείται με ένα χαμηλό προϋπολογισμό κατανάλωσης ισχύος, χωρίς να υποβαθμίζεται η επίδοσή της. Μία αντίστοιχη υβριδική προσέγγιση υιοθετείται και από τους Singh et al. [2009] σε ένα σχήμα χρονοδρομολόγησης, το οποίο προσπαθεί να διατηρήσει την συνολική κατανάλωση ισχύος του συστήματος εντός ενός καθορισμένου από τον χρήστη ορίου.

Ενώ ο έλεγχος από το λογισμικό επί της δυναμικής κατανάλωσης ισχύος είναι δυνατός, χάρη στην έξυπνη χρήση της συχνότητας και της τοποθέτησης των νημάτων, ο έλεγχος επί των απωλειών διαρροών δεν έχει υιοθετηθεί ακόμα ευρέως. Για να επιτευχθεί αυτό, θα πρέπει το λογισμικό να μπορεί να απενεργοποιεί συγκεκριμένες δομικές μονάδες του επεξεργαστή (π.χ., ALU, FPU

κ.λπ.) ή συγκεκριμένα τμήματα της κρυφής μνήμης, βάσει των αναγκών του. Θεωρώντας μία τέτοια υποστήριξη από το υλικό για ένα λεπτομερή έλεγχο των απωλειών λόγω διαρροών, οι Zhang et al. [2003] και οι You et al. [2006] εξετάζουν τεχνικές συμπίλησης για την αναγνώριση βασικών μονάδων προγράμματος, όπου συγκεκριμένες μονάδες του επεξεργαστή δεν χρησιμοποιούνται, εμπλουτίζοντας τον παραγόμενο κώδικα με ρητές εντολές ενεργοποίησης/απενεργοποίησης δομικών μονάδων. Αντίστοιχα, οι Zhang et al. [2002] θεωρούν την δυνατότητα λεπτομερούς ελέγχου επί των γραμμών της κρυφής μνήμης εντολών και προτείνουν τεχνικές βελτιστοποίησης σε επίπεδο συμπιληστή για να τίθενται επιλεκτικά οι γραμμές της κρυφής μνήμης σε λειτουργία χαμηλών διαρροών.

7.2 Διελκυστίνδα επίδοσης-ενέργειας στον πυρήνα SpMV

Η επίδοση του υπολογιστικού πυρήνα πολλαπλασιασμού αραιού πίνακα με διάνυσμα περιορίζεται από την αυξημένη ένταση των προσβάσεων στην μνήμη. Αυτό κατέστη σαφές από την ανάλυση των προηγούμενων κεφαλαίων, όπου προτείναμε πρωτότυπες τεχνικές για την αντιμετώπιση αυτού του κείριου προβλήματος επίδοσης. Δείξαμε ότι από την στιγμή που το μονοπάτι των δεδομένων προς την κύρια μνήμη κορεστεί πλήρως, δεν υπάρχει κανένας λόγος αύξησης της παραλληλίας του πυρήνα, καθώς το όφελος ελαχιστοποιείται. Η παρατήρηση αυτή είναι σημαντική και από την πλευρά της κατανάλωσης ισχύος, καθώς η προσθήκη περισσότερων πυρήνων στον υπολογισμό αυξάνει την κατανάλωση ισχύος για ένα ελάχιστο όφελος επίδοσης.

Η τοποθέτηση των νημάτων παίζει σημαντικό ρόλο κατά την εκτέλεση του πυρήνα SpMV. Μία λογική πολιτική πλήρωσης των πυρήνων μπορεί να επιτρέψει την κλιμάκωση του SpMV, καθυστερώντας τον κορεσμό του εύρους ζώνης μνήμης μέσω μίας καλύτερης εξισορρόπησης της κίνησης προς την κύρια μνήμη. Για παράδειγμα, η πολιτική «share-nothing», που παρουσιάστηκε στο Κεφάλαιο 3, κατανέμει τα νήματα σε όλους τους διαθέσιμους επεξεργαστές, ώστε να επιτυγχάνεται ο ελάχιστος διαμοιρασμός των πόρων. Το πλεονέκτημα της πολιτικής αυτής είναι η ελαχιστοποίηση της συμφόρησης στους κοινούς πόρους του μονοπατιού δεδομένων (π.χ., κρυφές μνήμες, διεπαφή εμπροσθίου διαύλου, ελεγκτής μνήμης κ.λπ.). Συγκεκριμένα για τις κρυφές μνήμες, μία τέτοια τοποθέτηση αυξάνει το μέγεθος της συνολικά διαθέσιμης κρυφής μνήμης του συστήματος, «ελαφρύνοντας» περαιτέρω το υποσύστημα της κύριας μνήμης. Ως αποτέλεσμα, η επίδοση του SpMV κλιμακώνεται με σταθερό ρυθμό μέχρι την 12-νηματική εκτέλεση στο σύστημα Dunnington, προτού τελικά ο δίαυλος μνήμης κορεστεί (βλ. Κεφάλαιο 3, Ενότητα 3.3.2). Δυστυχώς, αυτή η βελτίωση της επίδοσης δεν έρχεται δωρεάν. Όχι μόνο η αυ-

7. Θέματα ενεργειακής απόδοσης

ξημένη χρησιμοποίηση του επεξεργαστή αυξάνει την κατανάλωση ισχύος του, αλλά, πιο σημαντικά, η χρησιμοποίηση πολλαπλών κρυφών μνημών μπορεί να οδηγήσει σε σημαντική αύξηση των απωλειών διαρροών σε ένα σύστημα που υποστηρίζει δυναμική διαχείριση των απωλειών αυτών και μπορεί να θέτει σε «ύπνωση» ανενεργές κρυφές μνήμες ή ακόμα και ολόκληρους επεξεργαστές. Η μετατόπιση αυτή της κατανάλωσης ισχύος μπορεί ωστόσο να αποζημιωθεί από την γρηγορότερη εκτέλεση και μία ενδεχόμενη μείωση της συχνότητας μέσω του μηχανισμού DVFS.

Τέλος, η επίδοση του πυρήνα SpMV εξαρτάται άμεσα από την δομή του πίνακα εισόδου. Εκτός από τις ενδείξεις επί της επίδοσης του SpMV που ένας αραιός πίνακας μπορεί να παράσχει (συμφόρηση στη μνήμη, ακανόνιστες προσβάσεις, ανισορροπία φόρτου εργασίας κ.λπ.), το ενεργειακό αποτύπωμα του πυρήνα είναι πιθανό να μεταβάλεται από πίνακα σε πίνακα. Για παράδειγμα, η ανάθεση πολλών επεξεργαστών στον υπολογισμό του πυρήνα επί ενός ακανονιστού πίνακα με ανισορροπία στην κατανομή του φόρτου εργασίας μπορεί να αποτελεί σπατάλη ενέργειας, την στιγμή που μία τέτοια τοποθέτηση είναι πιο λογική για ένα πίνακα με κανονική δομή. Είναι επομένως σημαντικό να ποσοτικοποιήσει κανείς και να λάβει υπόψη του αυτή την πληροφορία κατά την πρόβλεψη των βελτίστων συμβιβασμών επίδοσης-κατανάλωσης ενέργειας για τον πυρήνα SpMV.

7.2.1 Πειραματική διαρρύθμιση

Για την αξιολόγηση της διελκυστίνδας επίδοσης-κατανάλωσης ενέργειας του πυρήνα SpMV, χρησιμοποιήσαμε το σύστημα Dunnington για τις μετρήσεις επιδόσεων (βλ. Κεφάλαιο 3, Ενότητα 3.2) και την υποδομή McPAT [Sheng et al., 2009] για ακριβείς εκτιμήσεις της κατανάλωσης ισχύος. Το σύστημα Dunnington αποτελείται από τέσσερις επεξεργαστές Intel X7460 στην τεχνολογία ολοκλήρωσης των 45 nm. Κάθε επεξεργαστής είναι στην ουσία ένα σύνολο τριών διπύρηνων μονάδων με 3 MiB κοινή κρυφή μνήμη, ενώ ολόκληρος ο επεξεργαστής φιλοξενεί μία ενοποιημένη κρυφή μνήμη L3 των 16 MiB. Ο επεξεργαστής υποστηρίζει την τεχνική DVFS και παρέχει δύο επίπεδα συχνότητας, στα 2.14 GHz και στα 2.67 GHz (πλήρης ταχύτητα). Κάθε επίπεδο συχνότητας σχετίζεται με συγκεκριμένη τάση τροφοδοσίας, η οποία ρυθμίζεται κατά την μετάβαση στην αντίστοιχη συχνότητα [Int, 2008]. Για την μεταβολή της συχνότητας του επεξεργαστή, χρησιμοποιούμε τα εργαλεία χώρου χρήστη `cpufrequtils` του Linux.

Θεωρούμε τρεις εναλλακτικές πολιτικές τοποθέτησης των νημάτων στους πυρήνες ανάλογα με το επίπεδο διαμοιρασμού των κοινών πόρων:

- (α') Πολιτική *full sharing*, κατά την οποία τα νήματα τοποθετούνται όσο το δυνατόν πιο κοντά μεταξύ τους, μοιραζόμενα όλα τα επίπεδα της ιεραρχί-

ας της κρυφής μνήμης (ίδια με την πολιτική «share-all» που περιγράφηκε στο Κεφάλαιο 3).

- (β') Πολιτική *semi sharing*, κατά την οποία τα νήματα τοποθετούνται έτσι, ώστε να μοιράζονται μόνο την κρυφή μνήμη L3, αλλά όχι την L2.
- (γ') Πολιτική *no sharing*, κατά την οποία τα νήματα τοποθετούνται όσο το δυνατόν «αραιότερα», ώστε να ελαχιστοποιηθεί ο διαμοιρασμός της ιεραρχίας της κρυφής μνήμης (ίδια με την πολιτική «share-nothing» που περιγράφηκε στο Κεφάλαιο 3).

Χρησιμοποιούμε τον συμβολισμό $XrYsZc$ για την περιγραφή μιας τοποθέτησης που χρησιμοποιεί συνολικά Z πυρήνες, Y διπύρηνες μονάδες (κοινή L2) και X επεξεργαστές (κοινή L3). Για παράδειγμα, οι δυνατές τοποθετήσεις τεσσάρων νημάτων είναι οι 1r2s4c, 2r4s4c και 4r4s4c. Συνολικά, θεωρούμε 26 διαφορετικές τοποθετήσεις για τα ακόλουθα πλήθη νημάτων: 1, 2, 3, 4, 6, 8, 9, 12, 15, 18, 24.

Η υποδομή McPAT προβλέπει την κατανάλωση ισχύος βασιζόμενη σε συγκεκριμένες καταμετρήσεις γεγονότων του επεξεργαστή (π.χ., επεξεργαστικούς κύκλους, προσβάσεις/αστοχίες στην κρυφή μνήμη κ.λπ.). Επειδή προορίζεται για χρήση από προσομοιωτές επεξεργαστών, υποστηρίζει ένα πολύ λεπτομερή κατάλογο γεγονότων, τα οποία δεν όλα διαθέσιμα στους πραγματικούς επεξεργαστές. Για τον λόγο αυτό, συμπεριλάβαμε γεγονότα σχετικά με τα απαιτούμενα από το McPAT που συμβάλλουν στην συνολική κατανάλωση ισχύος του επεξεργαστή [Isci και Martonosi, 2003; Singh et al., 2009]. Συνολικά, χρησιμοποιούμε καταμετρήσεις 14 γεγονότων, οι οποίες παρατίθενται στον Πίνακα 7.1. Η αρχιτεκτονική Intel Core του συστήματος Dunnington υποστηρίζει μόνο δύο προγραμματιζόμενους καταχωρητές καταμέτρησης γεγονότων επίδοσης, επομένως, απαιτήθηκαν πολλαπλές εκτελέσεις του πυρήνα SpMV για την συλλογή του συνόλου των γεγονότων. Τέλος, δεν συμπεριλαμβάνουμε στις μετρήσεις μας συνολικής κατανάλωσης ισχύος την συνεισφορά των ανενεργών επεξεργαστών, σαν να είχαμε την δυνατότητα να τους θέσουμε σε κατάσταση ύπνωσης με χαμηλές απώλειες διαρροών.

7.2.2 Χαρακτηρισμός των συμβιβασμών

Το Σχήμα 7.1 παρουσιάζει τους συμβιβασμούς επίδοσης-κατανάλωσης ενέργειας των 52 διαφορετικών ρυθμίσεων εκτέλεσης (τοποθέτηση νημάτων, συχνότητα πυρήνων) του πυρήνα SpMV στο σύστημα Dunnington για τέσσερις αραιούς πίνακες με διαφορετικά υπολογιστικά χαρακτηριστικά. Πιο συγκεκριμένα, οι πίνακες `xenon2` και `parabolic_fem` είναι αμφότεροι μικροί, χωρώντας στην συνολική κρυφή μνήμη του συστήματος, ενώ ο δεύτερος είναι σχετικά ακανόνιστος. Από την άλλη, οι `boneS10` και `thermal2` είναι αμφότεροι αρκετά

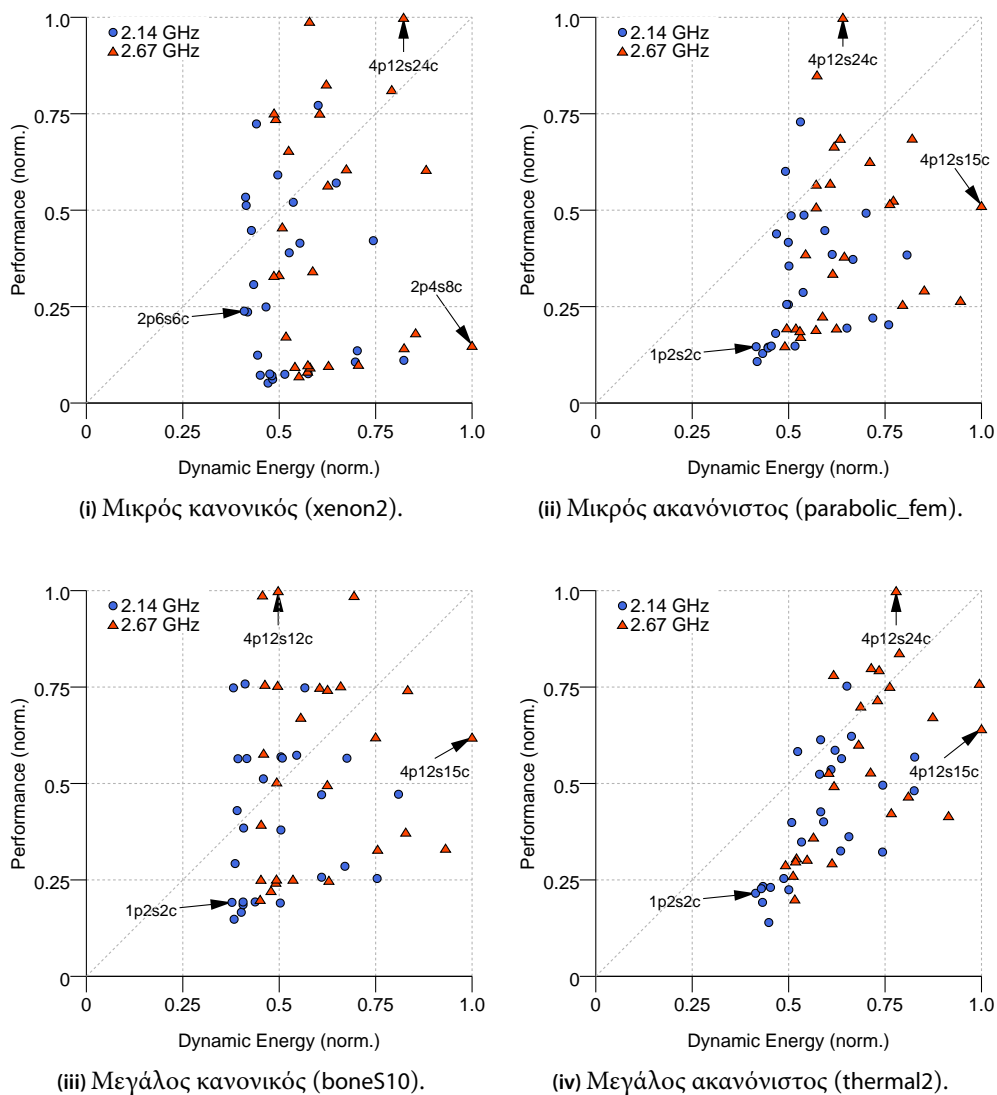
7. Θέματα ενεργειακής απόδοσης

Event Mnemonic	Description
CORE EVENTS	
Unhalted Core Cycles	Unhalted core cycles
UOPS_RETIRED.ANY	Micro-ops retired
SIMD_UOPS_EXEC	SIMD micro-ops executed
BRANCH UNIT EVENTS	
Branch Instruction Retired	Branch instructions retired
Branch Misses Retired	Mispredicted branch instructions retired
CACHES EVENTS	
L1I_READS	Instruction fetches
L1I_MISSES	Instruction fetch unit misses
L1D_REPL	Cache lines allocated in the L1 data cache
L1D_ALL_REF.ANY	All references to the L1 data cache
L2_RQSTS	L2 cache requests
L2_LINES_IN	L2 cache misses
LLC Reference	Last level cache references
LLC Misses	Last level cache misses
BUS EVENTS	
BUS_TRANS_ANY	All bus transactions

Πίνακας 7.1: Γεγονότα παρακολούθησης της επίδοσης που χρησιμοποιήθηκαν για την πρόβλεψη της κατανάλωσης ισχύος από την υποδομή McPAT. Για περισσότερες πληροφορίες επί των γεγονότων των μικροαρχιτεκτονικών της Intel, ανατρέξτε στο [Int, 2010] χρησιμοποιώντας το κλειδί (mnemonic) του γεγονότος.

μεγαλύτεροι της συνολικής κρυφής μνήμης του συστήματος, με τον δεύτερο να έχει ακανόνιστη διάταξη. Όλοι οι πίνακες είναι αποθηκευμένοι με το σχήμα CSX. Η επίδοση και η δυναμική κατανάλωση ενέργειας (δυναμική κατανάλωση ισχύος \times χρόνος εκτέλεσης) έχουν κανονικοποιηθεί επί της απόλυτης καλύτερης επίδοσης και του απολύτου μεγίστου της κατανάλωσης ενέργειας, αντίστοιχα. Η πρώτη βασική παρατήρηση είναι ότι το «τοπίο» της επίδοσης-ενέργειας είναι αρκετά ευρύ και δύο διαφορετικές ρυθμίσεις εκτέλεσης μπορεί να είναι αρκετά μακριά στο επίπεδο της ενεργειακής απόδοσης. Επίσης, στο ίδιο επίπεδο ενέργειας υπάρχουν πολλές ρυθμίσεις εκτέλεσης με διαφορετική επίδοση και αντιστρόφως. Αυτό απαιτεί μία μέθοδο για τον χαρακτηρισμό κάθε συμβιβασμού, ένα θέμα που συζητάμε αργότερα στην ενότητα αυτή. Άλλη μία βασική παρατήρηση είναι ότι το τοπίο επίδοσης-ενέργειας διαφέρει μεταξύ μικρών και μεγάλων πινάκων, ενώ οι διαφορές μεταξύ των ακανονίστων πινάκων δεν είναι τόσο σημαντικές.

7.2. Διελκυστίδα επίδοσης-ενέργειας στον πυρήνα SpMV



Σχήμα 7.1: Συμβιβασμοί επίδοσης-κατανάλωσης ενέργειας του πυρήνα SpMV για ένα σύνολο πινάκων με διαφορετικά χαρακτηριστικά επίδοσης. Οι διαφορετικές τοποθετήσεις των νημάτων σημειώνονται αναγράφοντας στην μορφή $XpYsZc$, υποδηλώνοντας την χρήση X επεξεργαστών, Y διπύρηνων μονάδων (κοινή L2) και Z πυρήνων συνολικά.

7. Θέματα ενεργειακής απόδοσης

Μία πιο προσεκτική μελέτη του τοπίου του πίνακα xenon2 (31 MiB στο σχήμα CSX) αποκαλύπτει ότι η ενεργειακά ωφέλιμη εκτέλεση είναι εκείνη που χρησιμοποιεί έξι νήματα κατανεμημένα αραιά τω τρόπω (μη διαμοιρασμός της κρυφής μνήμης L2) σε δύο επεξεργαστές με συχνότητα στα 2.14 GHz. Από την άλλη πλευρά, η χρήση δύο επεξεργαστών και οκτώ νημάτων με διαμοιρασμό της κρυφής μνήμης L2 στα 2.67 GHz είναι η λιγότερο ενεργειακά αποδοτική εκτέλεση, καταναλώνοντας περίπου 2.5 φορές περισσότερη ενέργεια για ένα κλάσμα μόνος της επίδοσης. Αυτό αποτελεί ένα τυπικό παράδειγμα της ωφέλειας μιας σωστής τοποθέτησης σε χαμηλότερη συχνότητα. Παρόλο που η εξα-νηματική εκτέλεση χρησιμοποιεί περισσότερους πόρους υλικού, η χαμηλότερη συχνότητα επιτρέπει την κατά 35% μείωση της κατανάλωσης ισχύος σε σύγκριση με την οκτω-νηματική εκτέλεση στην υψηλότερη συχνότητα (4.4 W έναντι 6.8 W). Από την άποψη της επίδοσης, η εξα-νηματική εκτέλεση είναι επίσης ωφέλιμη, καθώς εξαλείφει την συμφόρηση στη κοινή κρυφή μνήμη L2, η οποία πλέον λειτουργεί ως αποκλειστική μνήμη, παρέχοντας έτσι σημαντικά υψηλότερη επίδοση (+37%), παρά την χαμηλότερη συχνότητα. Μία βασική παρατήρηση, συχνή σε πίνακες με κανονική δομή, είναι η απότομη αύξηση στην ενεργειακή απόδοση. Για παράδειγμα, η εκτέλεση 3p9s9c στα 2.14 GHz του xenon2 επιτυγχάνει διπλάσια επίδοση από την πιο ενεργειακά συμφέρουσα 2p6s6c, με σχεδόν την ίδια κατανάλωση ισχύος³. Αυτό οφείλεται σε μία σχεδόν ταυτόσημη αύξηση στην επίδοση και στην κατανάλωση ισχύος, η οποία είναι τυπική της γραμμικής κλιμάκωσης του πυρήνα SpMV στους κανονικούς πίνακες όταν αυξάνονται οι επεξεργαστές του συστήματος. Παρά την αντίστοιχα απότομη αύξηση στην ενεργειακή τους απόδοση, οι μεγάλοι πίνακες που υπερβαίνουν την συνολική κρυφή μνήμη του συστήματος, παρουσιάζουν ένα ελαφρώς διαφορετικό τοπία επίδοσης-ενέργειας. Για παράδειγμα, η πιο ενεργειακά συμφέρουσα εκτέλεση για τον πίνακα bone510 είναι η δι-νηματική εκτέλεση 1p2s2c στα 2.14 GHz, ενώ η λιγότερο αποδοτική είναι η 15-νηματική στα 2.67 GHz. Αξίζει να σημειώσουμε εδώ την συμφόρηση στο εύρος ζώνης μνήμης και την επίδρασή της στην κατανάλωση ενέργειας. Η πιο αποδοτικές από άποψη επίδοσης εκτελέσεις για τον πίνακα bone510 είναι η 12-νηματική και η οκτω-νηματική εκτέλεση χρησιμοποιώντας και τους τέσσερις επεξεργαστές στα 2.67 GHz. Ο διάυλος μνήμης έχει ήδη κορεστεί από την οκτω-νηματική εκτέλεση και η προσθήκη περισσότερων νημάτων προσφέρει ελάχιστη βελτίωση στην επίδοση· η προσπάθεια προσθήκης ακόμα περισσότερων νημάτων πρόκειται απλά για σπατάλη ενέργειας, η οποία τελικά διπλασιάζεται στην 15-νηματική εκτέλεση⁴.

Το τοπίο των συμβιβασμών επίδοσης-κατανάλωσης ενέργειας στους ακα-

³ Ο διπλασιασμός της επίδοσης κατά την μετάβαση από τα κτώ στα εννέα νήματα οφείλεται άνετη φιλοξενία του συνόλου εργασίας στην συνολική κρυφή μνήμη των τριών επεξεργαστών.

⁴ Η 15-νηματική εκτέλεση υποφέρει, επίσης, και από ανισορροπία του φόρτου εργασίας, καθώς

νόμιστους πίνακες είναι παρόμοιο με αυτό των μεγάλων πινάκων, αλλά μία λιγότερο απότομη αύξηση της ενεργειακής απόδοσης. Το μέγεθος του πίνακα φαίνεται να μην παίζει σημαντικό ρόλο, καθώς τόσο ο μικρός `parabolic_fem` όσο και ο μεγαλύτερος `thermal2` παρουσιάζουν μέγιστη και ελάχιστη ενεργειακή απόδοση στις ίδιες εκτελέσεις, συγκεκριμένα στην `1p2s2c` στα 2.14 GHz και στην 15-νηματική στα 2.67 GHz, αντίστοιχα. Αυτό είναι, γενικά, αναμενόμενο, καθώς το βασικό πρόβλημα επίδοσης των πινάκων αυτών δεν είναι η συμφόρηση στο εύρος ζώνης μνήμης (βλ. Κεφάλαιο 3), επομένως, το μέγεθος του πίνακα δεν είναι τόσο κρίσιμο. Για τον ίδιο λόγο, δεν παρατηρείται και απότομη αύξηση στην ενεργειακή τους απόδοση· εφόσον η κλιμάκωση του SpMV εμποδίζεται, το κέρδος στην επίδοση από την χρήση περισσότερων πυρήνων είναι λιγότερο σημαντικό από την αντίστοιχη αύξηση στην κατανάλωση ισχύος.

Βέλτιστοι συμβιβασμοί

Έχοντας εξετάσει σε λεπτομέρεια το τοπία επίδοσης-ενέργειας του πυρήνα SpMV, το ερώτημα που ανακύπτει είναι τι καθιστά καλό ένα συμβιβασμό. Προφανώς, τα σημεία στην άνω αριστερή γωνία του τοπία επίδοσης-ενέργειας είναι καλοί συμβιβασμοί, καθώς μεγιστοποιούν την επίδοση, διατηρώντας τις ενεργειακές απαιτήσεις σε χαμηλά επίπεδα. Ωστόσο, το ερώτημα που ανακύπτει τώρα είναι πώς συγκρίνουμε δύο διαφορετικούς συμβιβασμούς και εάν υπάρχει ένας μοναδικός βέλτιστος συνδυασμός. Ας υποθέσουμε δύο σημεία, έστω (e, p) και (e', p') , στο τοπία επίδοσης-ενέργειας που αντιστοιχούν στις εκτελέσεις E και E' . Εάν $e < e'$ και $p \geq p'$, τότε η εκτέλεση E είναι σίγουρα καλύτερος συμβιβασμός από την E' , εφόσον με χαμηλότερη κατανάλωση ενέργειας επιτυγχάνει τουλάχιστον την ίδια επίδοση· το ίδιο ισχύει και για τις εκτελέσεις όπου $e \leq e'$ και $p > p'$. Στη περίπτωση αυτή, η εκτέλεση E *κυριαρχεί αυστηρά* επί της εκτέλεσης E' . Από την άλλη πλευρά, εάν $e \leq e'$ και $p \leq p'$, τότε δεν μπορούμε να αποφανθούμε εάν η εκτέλεση E είναι καλύτερη από την E' . Το σύνολο των σημείων (και των αντιστοιχών εκτελέσεων) που δεν κυριαρχούνται από κανένα άλλο σημείο στο τοπία επίδοσης-ενέργειας σχηματίζουν το *μη κυριαρχούμενο μέτωπο Pareto* [Luke, 2011]. Οι εκτελέσεις επί του μετώπου Pareto αποτελούν το σύνολο των βελτίστων συμβιβασμών, καθώς είναι αυστηρά καλύτερες από οποιαδήποτε άλλη εκτέλεση σε τουλάχιστον ένα στόχο (επίδοση ή κατανάλωση ενέργειας). Μία σημαντική ιδιότητα του μετώπου Pareto είναι ότι η μετάβαση από σημείο σε σημείο του υπόκειται στην ελάχιστη απώλεια ενέργειας και στο μέγιστο κέρδος στην επίδοση, και αντιστρόφως. Υπό αυτή την έννοια, όλα τα σημεία του μετώπου Pareto είναι θεωρητικά ισο-

έξι νήματα μοιράζονται την κρυφή μνήμη L2, ενώ τα υπόλοιπα την χρησιμοποιούν αποκλειστικά.

7. Θέματα ενεργειακής απόδοσης

δύναμοι συμβιβασμοί. Επομένως, δεν υπάρχει ένας μοναδικός βέλτιστος συμβιβασμός επίδοσης-κατανάλωσης ενέργειας (εκτός και αν το μέτωπο Pareto εκφυλίζεται σε ένα σημείο), αλλά ένα σύνολο βελτίστων συμβιβασμών.

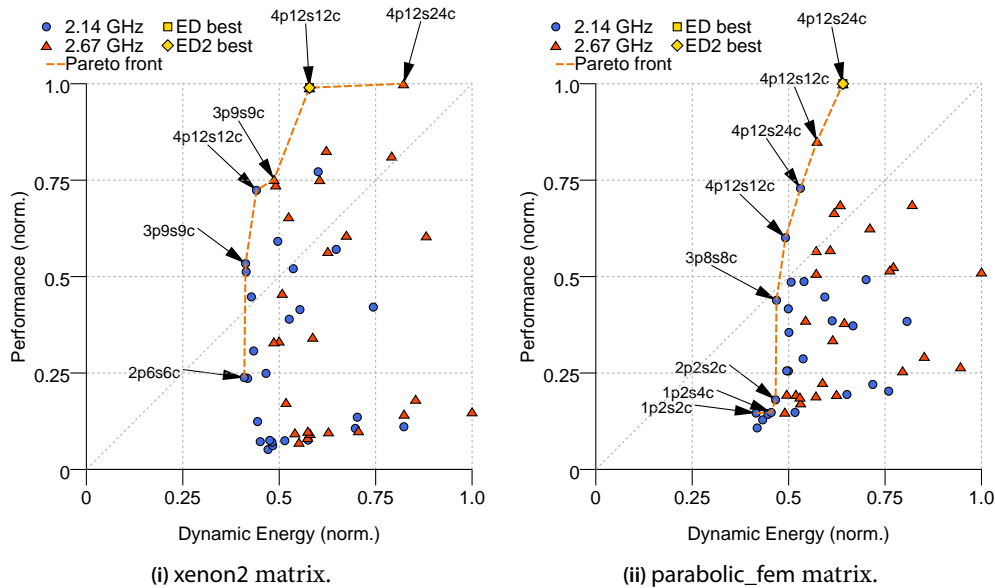
Τα γινόμενα energy-delay είναι βέλτιστα Κάθε συμβιβασμός που αντιστοιχεί σε ένα ελάχιστο γινόμενο energy-delay για κάποιο $n \geq 0$ είναι βέλτιστος κατά την έννοια του Pareto. Αυτό μπορεί εύκολα να αποδειχθεί: έστω ότι η εκτέλεση M ελαχιστοποιεί το γινόμενο ED^n για κάποιο $n \geq 0$, δηλαδή, $e_M d_M^n < e_{M'} d_{M'}^n$ για κάθε $M' \neq M$. Εάν υποθέσουμε ότι το σημείο (e_M, d_M) δεν βρίσκεται επί του μετώπου Pareto, τότε θα πρέπει να υπάρχει μία εκτέλεση C , τέτοια ώστε είτε $e_C < e_M$ και $d_C \leq d_M$ είτε $e_C \leq e_M$ και $d_C < d_M$. Το γινόμενο energy-delay αυτή της εκτέλεσης θα είναι, επομένως, $e_C d_C^n \leq e_M d_M^n$ για $n = 0$ ή $e_C d_C^n < e_M d_M^n$ για $n > 0$, πράγμα το οποίο έρχεται σε αντίθεση με την αρχική μας υπόθεση ότι η εκτέλεση M ελαχιστοποιεί το γινόμενο energy-delay. Επομένως, η M πρόκειται για μία μη κυριαρχούμενη λύση, δηλαδή, αντιστοιχεί σε σημείο επί του μετώπου Pareto. Κατά συνέπεια, κάθε σημείο του μετώπου Pareto αντιστοιχεί σε ένα ελάχιστο γινόμενο energy-delay για κάποιο $n \geq 0$.

Τα πιο συνήθη γινόμενα energy-delay που χρησιμοποιούνται στην πράξη είναι η ενέργεια (E), το energy-delay (ED) και το energy-delay στο τετράγωνο (ED^2), διατεταγμένα κατά αύξουσα έννοια ως προς την επίδοση. Το Σχήμα 7.2 παρουσιάζει το μέτωπο Pareto των πινάκων `xenon2` και `parabolic_fem` καθώς και τα σημεία που αντιστοιχούν στα γινόμενα ED και ED^2 .

7.3 Πρόβλεψη των βελτίστων ρυθμίσεων εκτέλεσης

Καθώς η επίδοση του πυρήνα `SprMV` εξαρτάται σε σημαντικό βαθμό από την δομή του πίνακα εισόδου, είναι επιθυμητό, δεδομένου ενός αγνώστου πίνακα, να μπορεί κανείς να προβλέψει εκείνες τις ρυθμίσεις εκτελέσεως που θα οδηγήσουν σε ένα ιδανικό συμβιβασμό επίδοσης-κατανάλωσης ενέργειας, με σκοπό την εξοικονόμηση πόρων και την επίτευξη πιθανών κερδών στην κατανάλωση ενέργειας. Σύμφωνα με την συζήτηση και την πειραματική αξιολόγηση του Κεφαλαίου 3, δύο από τις πιο καθοριστικές παραμέτρους της επίδοσης του πυρήνα `SprMV` είναι το μέγεθος του πίνακα και ο λόγος `float:byte`, ο οποίος σχετίζεται άμεσα με το μέσο πλήθος μη μηδενικών στοιχείων ανά γραμμή. Εάν ο πίνακας είναι αρκετά μικρός, ώστε να χωρά στην συνολική κρυφή μνήμη της υποκειμένης αρχιτεκτονικής, η βελτίωση της επίδοσης του `SprMV` θα είναι σημαντική, εφόσον εξαλείφεται η συμφόρηση στο εύρος ζώνης μνήμης. Ωστόσο, υπάρχουν πίνακες, για τους οποίους αυτή η συμφόρηση δεν αποτελεί το κύριο πρόβλημα επίδοσης, καθώς υποφέρουν περισσότερο από ακανόνιστες προσβάσεις στην κύρια μνήμη, επιβαρύνσεις βρόχων, ανισορροπίες φόρτου

7.3. Πρόβλεψη των βελτίστων ρυθμίσεων εκτέλεσης



Σχήμα 7.2: Οι βέλτιστοι συμβιβασμοί επίδοσης-κατανάλωσης ενέργειας δύο αραιών πινάκων. Τα σημεία του μετώπου Pareto σημειώνονται με τα χαρακτηριστικά των αντιστοιχών εκτελέσεων. Παρουσιάζονται, επίσης, οι εκτελέσεις που ελαχιστοποιούν τα γινόμενα energy-delay.

εργασίας κ.λπ. Δείξαμε στο Κεφάλαιο 3 ότι υπάρχει μία σημαντική συσχέτιση μεταξύ του λόγου flops:byte ενός πίνακα και της επίδοσής του, ενώ στην προηγούμενη ενότητα αναγνωρίσαμε επίσης διαφορές στο τοπία επίδοσης-ενέργειας του SpMV, εξαρτώμενες από το μέγεθος και την «αραιότητα» του πίνακα. Η πληροφορία αυτή, επομένως, μπορεί να διαφανεί χρήσιμη για την πρόβλεψη των βελτίστων ρυθμίσεων εκτέλεσης του πυρήνα SpMV.

Ο στόχος μας είναι να προβλέψουμε ένα ευρύ σύνολο βελτίστων ρυθμίσεων εκτέλεσης· για τον λόγο αυτό, επαφιομάστε σε μία προσέγγιση μάθησης μηχανών, βασισμένη στην τεχνική της ομαδοποίησης, με σκοπό να προσεγγίσουμε το το μέτωπο Pareto των βελτίστων εκτελέσεων. Η βασική ιδέα πίσω από αυτή την προσέγγιση είναι ότι αναμένουμε πίνακες με παρόμοια δομικά χαρακτηριστικά, π.χ., μεγάλοι κανονικοί πίνακες, να παρουσιάζουν αντίστοιχους συμβιβασμούς επίδοσης-ενέργειας. Η διαδικασία μάθησης βασίζεται σε μία αρχική ομαδοποίηση των πινάκων σύμφωνα με το μέγεθός τους και το μέσο πλήθος μη μηδενικών τους στοιχείων ανά γραμμή. Για κάθε σχηματιζόμενη ομάδα, κατασκευάζουμε ένα αντιπροσωπευτικό μέτωπο Pareto από τις ρυθμίσεις εκτελέσεις των επιμέρους μετώπων Pareto της ομάδας. Στην συνέχεια,

κάθε ομάδα αντιπροσωπεύεται από το γεωμετρικό της κέντρο (στο πεδίο των χαρακτηριστικών του πίνακα) και το δικό της μέτωπο Pareto.

7.3.1 Ομαδοποίηση των πινάκων

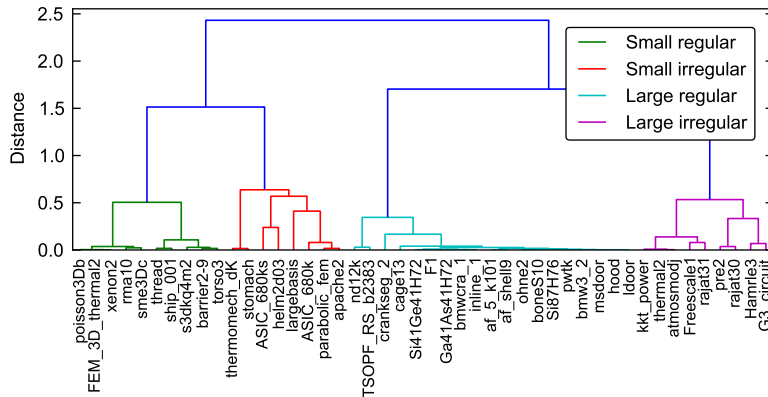
Με σκοπό την δημιουργία ενός ακόμα πιο αντιπροσωπευτικού συνόλου πινάκων και της συλλογής περισσότερων δειγμάτων για την εκπαίδευση του μοντέλου μας, επεκτείναμε την αρχική σουίτα πινάκων (βλ. Κεφάλαιο 3, Πίνακας 3.1) στους 50 αραιούς πίνακες, όλοι επιλεγμένοι από την συλλογή αραιών πινάκων του Πανεπιστημίου της Φλόριδα [Davis και Hu, 2011]. Σε κάθε πίνακα αναθέτουμε ένα διάνυσμα δύο χαρακτηριστικών, που αντικατοπτρίζουν την «αραιότητα» του και το μέγεθός του. Εφόσον η ομαδοποίηση εξαρτάται από τις αποστάσεις των πινάκων στο πεδίο των χαρακτηριστικών τους, είναι κρίσιμο οι τιμές των χαρακτηριστικών αυτών να αντιπροσωπεύουν ένα έγκυρο μέτρο ομοιότητας. Για παράδειγμα, θεωρώντας για το χαρακτηριστικό του μεγέθους το σχετικό ως προς την συνολική κρυφή μνήμη του συστήματος μέγεθος του πίνακα μπορεί να οδηγηθούμε σε παραπλανητικά συμπεράσματα, καθώς πολύ μεγάλοι πίνακες, όπως ο `boneS10`, είναι πολύ πιθανό να καταστούν εσφαλμένα *outliers*, δηλαδή, μη δυνάμενοι να κατηγοριοποιηθούν σε καμιά ομάδα. Στην πράξη, δεν μας αφορά πόσο μεγάλος είναι κάποιος πίνακας, εάν το μέγεθός του υπερβαίνει κατά πολύ το μέγεθος της συνολικής κρυφής μνήμης του συστήματος· επομένως, η μετρική αυτή δεν είναι ένα έγκυρο μέτρο ομοιότητας. Ακριβώς το ίδιο ισχύει και για οποιαδήποτε μετρική είτε του απόλυτου είτε του σχετικού πλήθους των μη μηδενικών στοιχείων ανά γραμμή του πίνακα. Για τον λόγο αυτό, αναθέτουμε τιμές στα χαρακτηριστικά κλιμακωτά, προσπαθώντας να αναπαραστήσουμε καλύτερα την ομοιότητα των πινάκων. Πιο συγκεκριμένα, θεωρώντας ως C την συνολική κρυφή μνήμη του συστήματος και S το μέγεθος του πίνακα, υπολογίζουμε το χαρακτηριστικό του μεγέθους ως εξής, διακρίνοντας διαφορετικές περιπτώσεις για πίνακες που χωρούν στην συνολική κρυφή μνήμη δύο, τριών ή τεσσάρων επεξεργαστών (πλήρες σύστημα)⁵:

$$a_m = \begin{cases} 1, & S/C \in (0, 1/2) \\ 1.5, & S/C \in [1/2, 3/4) \\ 3, & S/C \in [3/4, 1) \\ 3.5, & S/C \in [1, \infty) \end{cases} \quad (7.2)$$

Αντίστοιχα, θέτουμε ως όριο των πολύ αραιών πινάκων τα 15 μη μηδενικά στοιχεία ανά γραμμή και υπολογίζουμε το χαρακτηριστικό της «αραιότητας»

⁵ Δεν συμπεριλαμβάνουμε την περίπτωση των πινάκων που χωρούν στην κρυφή μνήμη L3 ενός μόνο επεξεργαστή, καθώς η σουίτα μας δεν περιλαμβάνει τόσο μικρούς πίνακες.

7.3. Πρόβλεψη των βελτίστων ρυθμίσεων εκτέλεσης



Σχήμα 7.3: Οι τέσσερις βασικές κατηγορίες πινάκων της επεκτεταμένης σουίτας μας.

ως εξής:

$$a_s = \begin{cases} \frac{1}{15} \cdot \frac{NNZ}{N}, & \frac{NNZ}{N} \leq 15 \\ 2, & \frac{NNZ}{N} > 15 \end{cases} \quad (7.3)$$

Το Σχήμα 7.3 παρουσιάζει τις σχηματιζόμενες ομάδες χρησιμοποιώντας μία τεχνική ιεραρχικής ομαδοποίησης. Η σωστή ανάθεση τιμών στα χαρακτηριστικά επέτρεψε στον αλγόριθμο της ομαδοποίησης να αναγνωρίσει με ακρίβεια τις τέσσερις βασικές κατηγορίες πινάκων, συγκεκριμένα τους «μικρούς και αραιούς», τους «μικρούς και κανονικούς», τους «μεγάλους και αραιούς» και τους «μεγάλους και κανονικούς», χωρίς να αφήνει εκτός μεμονωμένους πίνακες.

7.3.2 Κατασκευή του μετώπου Pareto της ομάδας

Θεωρώντας ότι η ομαδοποίηση των πινάκων αντικατοπτρίζει σωστά τα κοινά τους χαρακτηριστικά επίδοσης, το μέτωπο Pareto της ομάδας θα αποτελείται από τις πιο κοινές ρυθμίσεις εκτέλεσης στα μέτωπα Pareto των πινάκων της ομάδας. Για την κατασκευή του μετώπου Pareto της ομάδας, διασχίζουμε τα σημεία των επιμέρους μετώπων Pareto της ομάδας από το σημείο με την υψηλότερη επίδοση προς το σημείο με την λιγότερη κατανάλωση ενέργειας, δηλαδή, από το άνω δεξιά στο κάτω αριστερά σημείο του μετώπου. Κατά την διάσχιση, καταγράφουμε τις ρυθμίσεις εκτελέσεις που αντιστοιχούν στο τρέχον σημείο εκτέλεσης για κάθε πίνακα της ομάδας και επιλέγουμε την πιο συχνά εμφανιζόμενη. Εάν δύο ή περισσότερες εκτελέσεις έχουν την ίδια συχνότητα εμφάνισης, επιλέγουμε και τις δύο εάν η συχνότητά τους είναι μεγαλύτερη του 25% του μεγέθους της ομάδας, αλλιώς επιλέγουμε την εκτέλεση με τον μικρότερο αριθμό νημάτων, ώστε να καλύψουμε καλύτερα τις εκτελέσεις χαμηλής

7. Θέματα ενεργειακής απόδοσης

<i>Matrix #1</i>	<i>Matrix #2</i>	<i>Matrix #3</i>	<i>Cluster Pareto</i>
4p12s24c@2.67	4p12s24c@2.67	4p12s24c@2.67	4p12s24c@2.67
4p12s12c@2.67	4p12s12c@2.67	4p12s12c@2.67	4p12s12c@2.67
4p8s8c@2.67	4p8s8c@2.67	4p8s8c@2.67	4p8s8c@2.67
4p12s24c@2.14	4p12s24c@2.14	4p12s24c@2.14	4p12s24c@2.14
4p12s12c@2.14	4p12s12c@2.14	4p12s12c@2.14	4p12s12c@2.14
4p8s8c@2.14	4p8s8c@2.14	4p8s8c@2.14	4p8s8c@2.14
2p6s6c@2.14	1p2s2c@2.14	2p6s6c@2.14	2p6s6c@2.14
1p3s3c@2.14	n/a	1p3s6c@2.14	1p3s3c@2.14
1p2s2c@2.14	n/a	1p2s4c@2.14	1p2s2c@2.14
n/a	n/a	1p2s2c@2.14	n/a

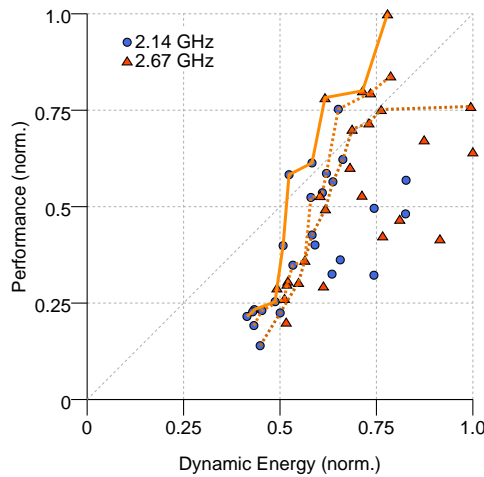
Σχήμα 7.4: Κατασκευή του μετώπου Pareto ομάδας τριών πινάκων. Οι εκτελέσεις που επιλέγονται ως σημεία του κατασκευαζόμενου μετώπου σημειώνονται με έντονη γραμματοσειρά.

ενεργείας. Χρησιμοποιούμε, επίσης, αυτή την τεχνική και για την πρόβλεψη των ρυθμίσεων εκτέλεσης που ελαχιστοποιούν τα γινόμενα ED και ED^2 .

Αποφυγή υπερεκπαίδευσης Τα μέτωπα Pareto των διαφορετικών πινάκων μιας ομάδας δεν έχουν απαραίτητα τον ίδιο αριθμό σημείων. Η προτεινόμενη μέθοδος για την κατασκευή του μετώπου Pareto της ομάδας ενέχει τον κίνδυνο της δημιουργίας ενός μετώπου με τόσα πολλά σημεία, ώστε να μην είναι πιο αντιπροσωπευτικό της ομάδας, καλύπτοντας ακόμα και ακραίες περιπτώσεις. Για την αποφυγή αυτής της *υπερεκπαίδευσης*, σταματάμε την κατασκευή του μετώπου Pareto μόλις εξαντληθούν τα σημεία για περισσότερους από τους μισούς πίνακες της ομάδας. Το Σχήμα 7.4 παρουσιάζει ένα παράδειγμα κατασκευής του μετώπου Pareto της ομάδας.

7.3.3 Κατηγοριοποίηση και αξιολόγηση

Μετά το πέρας της διαδικασίας εκπαίδευσης, αποθηκεύουμε τα κέντρα των ομάδων και τα αντίστοιχα μέτωπα Pareto για μελλοντική χρήση. Το κέντρο μιας ομάδας είναι ο γεωμετρικός μέσος των χαρακτηριστικών των πινάκων που την απαρτίζουν, ενώ το μέτωπο Pareto της ομάδας κατασκευάζεται όπως περιγράφηκε προηγουμένως. Τα δύο αυτά χαρακτηριστικά καθορίζουν κάθε κατηγορία πίνακα. Μόλις ένας νέος πίνακας εμφανιστεί, υπολογίζουμε τα χαρακτηριστικά του μεγέθους και της «αραιότητάς» του, σύμφωνα με τις σχέσεις (7.2) και (7.3) και τον κατηγοριοποιούμε στην πλησιέστερη ομάδα στο πεδίο των χαρακτηριστικών. Στην συνέχεια, ανακτούμε τις ρυθμίσεις εκτέλεσης του μετώπου Pareto της ομάδας που επιλέχθηκε και τις αναθέτουμε στον καινούργιο πίνακα.



Σχήμα 7.5: Τα πρώτα τρία μέτωπα Pareto του πίνακα thermal2. Οι ρυθμίσεις εκτέλεσης που κείνται επί του πρώτου μετώπου είναι βέλτιστες, ενώ αυτές που κείνται επί του i -οστού αποτελούν τις i -οστές καλύτερες λύσης.

Για να αξιολογήσουμε την ποιότητα των προβλέψεών μας, αναθέτουμε ένα βαθμό σε κάθε ρύθμιση εκτέλεσης του προβλεφθέντος μετώπου Pareto αντικατοπτρίζοντας την ποιότητά του. Για να το επιτύχουμε αυτό, αρχικά υπολογίζουμε επαναληπτικά ένα σύνολο μετώπων Pareto για όλους τους συμβιβασμούς επίδοσης-ενέργειας του πίνακα-στόχου. Το πρώτο μέτωπο Pareto είναι το σύνολο των βελτίστων ρυθμίσεων εκτέλεσης, όπως περιγράφηκε στην Ενότητα 7.2.2 (βαθμός βελτίστου 0). Σε κάθε επανάληψη i , αγνοούμε τα σημεία των προηγούμενως υπολογισθέντων μετώπων και υπολογίζουμε ένα νέο μέτωπο Pareto· τα σημεία αυτού του μετώπου είναι οι i -οστές καλύτερες ρυθμίσεις εκτέλεσης (βαθμός βελτίστου i). Το Σχήμα 7.5 παρουσιάζει ένα παράδειγμα των τριών πρώτων μετώπων Pareto του πίνακα του πίνακα thermal2. Σε κάθε σημείο του προβλεφθέντος μετώπου, επομένως, ανατίθεται ο βαθμός βελτίστου του πραγματικού μετώπου Pareto, επί του οποίου κείται. Ο μέσος όρος των βαθμών όλων των προβλεφθέντων ρυθμίσεων εκτέλεσης αποτελεί το συνολικό μέτρο ποιότητας της πρόβλεψής μας· ονομάζουμε το μέτρο αυτό *βαθμό του προβλεφθέντος μετώπου Pareto*.

Δοκιμάζουμε την μέθοδο πρόβλεψης που προτείνουμε με την τεχνική *five-cross-validation* (5-CV). Η τεχνική cross-validation είναι μία συνήθης στατιστική τεχνική για την αξιολόγηση της ακρίβειας ενός μοντέλου πρόβλεψης και αποτελείται από τα ακόλουθα βήματα:

- (α') Ομοιόμορφη αναδιάταξη των αρχικών δεδομένων και διαχωρισμός τους σε ένα σταθερό αριθμό ίσων διαμερίσεων (γνωστών και ως *πτυχών* (*folds*)).

7. Θέματα ενεργειακής απόδοσης

Στην περίπτωση μας, το σύνολο δεδομένων είναι η σουίτα των 50 πινάκων, η οποία χωρίζεται σε πέντε πτυχές.

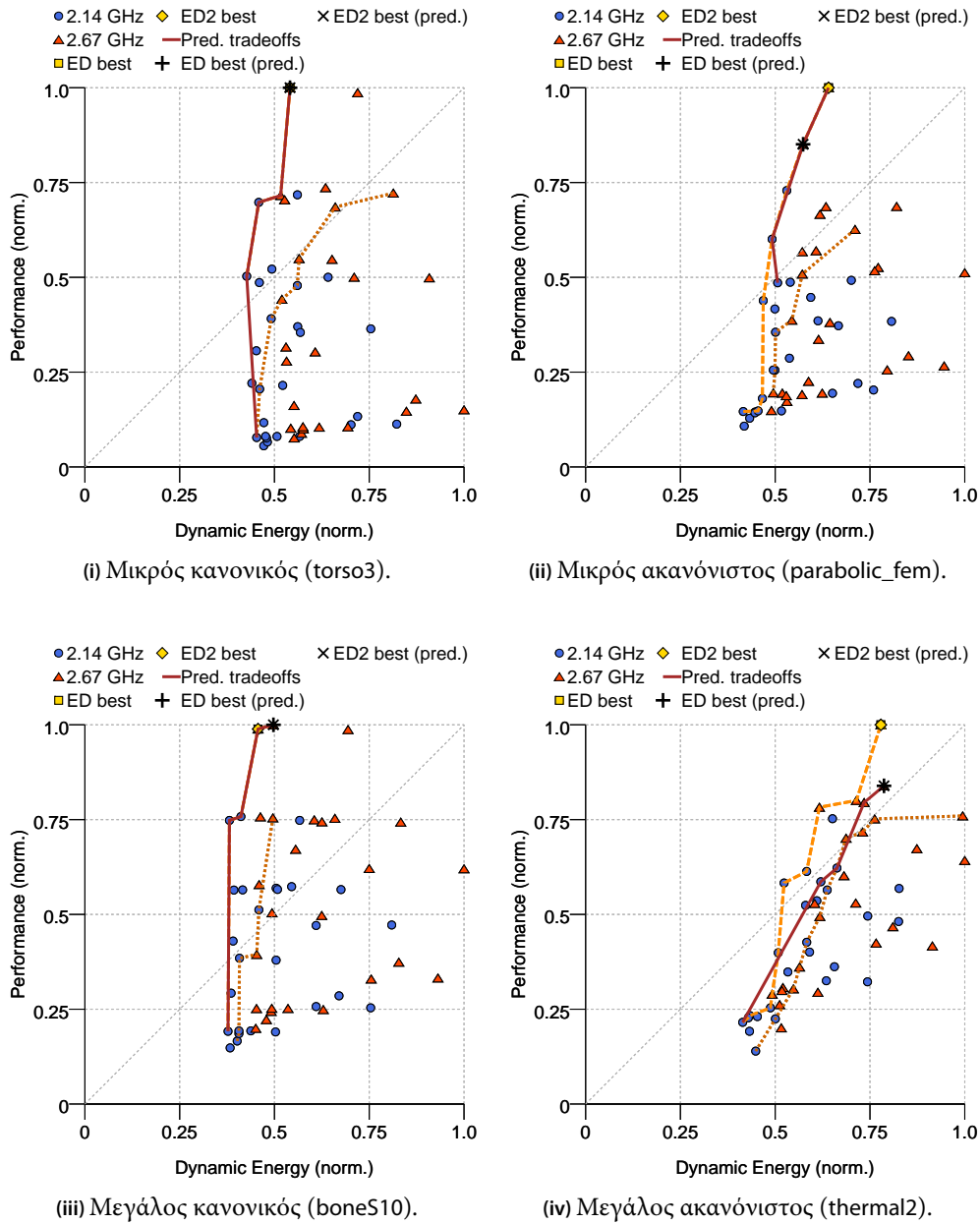
(β') Διατήρηση μίας πτυχής για δοκιμή και χρήση των υπολοίπων για εκπαίδευση.

(γ') Επανάληψη του προηγούμενου βήματος μέχρις ότου δοκιμαστούν όλες οι πτυχές.

Ο πίνακας 7.2 παρουσιάζει μία σύνοψη των αποτελεσμάτων της δοκιμής 5-CV. Για κάθε πτυχή, παρουσιάζεται ο βαθμός του προβλεφθέντος μετώπου Pareto και το ποσοστό των προβλέψεων με βαθμό χαμηλότερο ενός ορισμένου κατωφλίου. Η μεθοδολογία μας επιτυγχάνει αρκετά ακριβείς προβλέψεις με ένα μέσο βαθμό βελτίστου κοντά στο μηδέν. Πράγματι, 90% των προβλεφθέντων Pareto έχουν βαθμό μικρότερο του ένα κατά μέσο όρο, ενώ όλα έχουν βαθμό μικρότερο του δύο. Το Σχήμα 7.6 παρουσιάζει τυπικές προβλέψεις για πίνακες κάθε κατηγορίας. Οι προβλέψεις για τους κανονικούς πίνακες είναι αρκετά ακριβείς και τείνουν προς το τέλειο στους μεγαλύτερους από αυτούς. Από την άλλη, οι προβλέψεις για τους πολύ αραιούς πίνακες, π.χ., thermal2, τείνουν να είναι λιγότερο ακριβείς. Αυτό είναι κάτι το αναμενόμενο, καθώς το χαρακτηριστικό «αραιότητας» που χρησιμοποιούμε (μέσος αριθμός μη μηδενικών στοιχείων ανά γραμμή) αντικατοπτρίζει με ακρίβεια μόνο την επιβάρυνση βρόχου των πολύ αραιών πινάκων. Παρόλο που οι αραιοί πίνακες είναι πολύ πιθανό να υποφέρουν και από ακανόνιστες προσβάσεις στη μνήμη και από ανισορροπίες του φόρτου εργασίας, ο κανόνας αυτός έχει εξαιρέσεις· για παράδειγμα, μία αναδιατεταγμένη έκδοση ενός ακανονίστου πίνακα δεν έχει αυτά τα προβλήματα (βλ. Κεφάλαιο 3, Ενότητα 3.3.2). Ωστόσο, επιλέξαμε να μην κατακερματίσουμε περαιτέρω την κατηγορία των πολύ αραιών πινάκων, καθώς το αρχικό μας σύνολο δεδομένων είναι αρκετά μικρό, ώστε να επιτρέπει την ακριβή εκπαίδευση τέτοιων υποκατηγοριών.

Τέλος, ο Πίνακας 7.3 παρουσιάζει μία περίληψη της ακρίβειας των προβλέψεων των γινομένων energy-delay. Για κάθε ένα από τα τρία επιλεγμένα γινόμενα, συγκεκριμένα το της ενέργειας, το energy-delay, και energy-delay στο τετράγωνο, παρουσιάζεται η απόσταση των προβλέψεων από τα πραγματικά ελάχιστα γινόμενα. Για παράδειγμα, οι προβλέψεις για την πιο ενεργειακά αποδοτική εκτέλεση οδηγούν σε κατά 7.6% κάτω μέσο όρο μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την πραγματικά βέλτιστη. Αντίστοιχης ακρίβειας είναι και οι προβλέψεις των υπολοίπων γινομένων, με τα ED να προσεγγίζονται μία μία διαφορά της τάξης του 5.1%, τα δε ED² με διαφορά 9.3% κατά μέσο όρο.

7.3. Πρόβλεψη των βελτίστων ρυθμίσεων εκτέλεσης



Σχήμα 7.6: Προβλέψεις βελτίστων ρυθμίσεων εκτέλεσης και προβλέψεις για τα ελάχιστα γινόμενα ED και ED^2 . Παρουσιάζονται, επίσης, τα μέτωπα Pareto με βαθμούς ένα και δύο.

7. Θέματα ενεργειακής απόδοσης

Fold No.	Prediction Rank	Rank < 1	Rank < 2
#1	0.45	100.0%	100.0%
#2	0.27	77.8%	100.0%
#3	0.37	88.9%	100.0%
#4	0.19	100.0%	100.0%
#5	0.27	88.9%	100.0%
Average	0.31	91.1%	100.0%

Πίνακας 7.2: Αποτελέσματα cross-validation για την προτεινόμενη μεθοδολογία πρόβλεψης. Παρουσιάζονται ο βαθμός των προβλεφθέντων μετώπων Pareto και το ποσοστό των προβλέψεων με βαθμό μικρότερο ενός ορισμένου ορίου.

Fold No.	Energy	ED	ED ²
#1	6.9%	1.6%	6.6%
#2	12.5%	10.5%	17.7%
#3	13.2%	4.1%	8.0%
#4	2.1%	0.9%	0.0%
#5	3.5%	8.5%	14.2%
Average	7.6%	5.1%	9.3%

Πίνακας 7.3: Ακρίβεια πρόβλεψης γινόμενων energy-delay. Παρουσιάζονται ανά πτυχή οι μέσες αποστάσεις από τα πραγματικά ελάχιστα γινόμενα.

7.3.4 Περιορισμοί

Η προτεινόμενη μεθοδολογία για την πρόβλεψη των ρυθμίσεων εκτέλεσης του SrMV που οδηγούν σε βέλτιστους συμβιβασμούς επίδοσης-κατανάλωσης ενέργειας μπορεί να παράσχει αρκετά ακριβείς προβλέψεις για ένα ευρύ φάσμα αραιών πινάκων. Ωστόσο, κάποιοι περιορισμοί παραμένουν ακόμα. Καταρχήν, παρά τις γενικά ακριβείς προβλέψεις, η μέθοδός μας δεν μπορεί επί του παρόντος να προβλέψει πόσο μακριά κείνται οι ρυθμίσεις εκτελέσεις στο πεδίο επίδοσης-ενέργειας. Για παράδειγμα, οι εκτελέσεις 1p2s2c και 4p8s8c στα 2.14 GHz για τον πίνακα boneS10 (Σχήμα 7.1), παρόλο που είναι ισοδύναμες με την έννοια του κατά-Pareto βελτίστου, καταναλώνουν περίπου την ίδια δυναμική ενέργεια έχοντας περισσότερη από $3 \times$ στην επίδοση· είναι φανερό ότι η εκτέλεση 1p2s2c θα μπορούσε να παραλειφθεί τελείως ως ένας μη ιδανικός συμβιβασμός στην πράξη. Ο περιορισμός αυτός μπορεί να αρθεί με μία πιο «χαλαρή» κατασκευή του μετώπου Pareto, το οποίο θα συμπεριλαμβάνει μόνο τα

σημεία των οποίων οι στόχοι μεταβάλλονται περισσότερο από ένα ορισμένο κατώφλι.

Ένας δεύτερος περιορισμός αφορά στην ομαδοποίηση των πινάκων. Ενώ το χαρακτηριστικό του μεγέθους είναι σωστά δομημένο, το χαρακτηριστικό της «αραιότητας» δεν λαμβάνει υπόψη του συγκεκριμένες ιδιαιτερότητες των πολύ αραιών πινάκων, που επηρεάζουν την επίδοση του SpMV, π.χ., την ακανόνιστη κατανομή των μη μηδενικών στοιχείων και την ανισορροπία του φόρτου εργασίας. Συνεπώς, παρατηρείται μία πτώση της ακρίβειας σε αυτές τις περιπτώσεις. Η συμπερίληψη αυτής της πληροφορίας κατά την ομαδοποίηση των πινάκων, απαιτεί την πλήρη ανίχνευση/διάσχιση του πίνακα και μία λογική ποσοτικοποίηση αυτών των χαρακτηριστικών, η οποία θα αντικατοπτρίζει σωστά την ομοιότητα ή ανομοιότητα των πινάκων υπό όρους επίδοσης.

7.4 Ανοιχτά θέματα

Στο Κεφάλαιο αυτό, πραγματοποιήσαμε μία πρώτη προσέγγιση στην αξιολόγηση της διελκυστίνδας επίδοσης-κατανάλωσης ενέργειας του πυρήνα SpMV και στην πρόβλεψη των ρυθμίσεων εκτέλεσης που οδηγούν σε βέλτιστους συμβιβασμούς επίδοσης-ενέργειας, ανάλογα με τον πίνακα εισόδου. Ένα σύνολο θεμάτων, ωστόσο, παραμένει ακόμα ανοιχτό και θα πρέπει να διερευνηθεί στο μέλλον. Η προσέγγισή μας εστιάζει αποκλειστικά στην δυναμική κατανάλωση ισχύος του επεξεργαστή, αγνοώντας τις απώλειες διαρροών. Σύμφωνα με τις προβλέψεις του McPAT, οι απώλειες διαρροών του συστήματος Dunnington αγγίζουν τα 88 W, ενώ η μέγιστη δυναμική κατανάλωση ισχύος που καταγράφει είναι κατά τι ανώτερη των 50 W. Κατά μέσο όρο, οι πίνακες που χωρούν στην συνολική κρυφή μνήμη του συστήματος αυξάνουν κατά 37% την συνδυασμένη κατανάλωση ισχύος όλων των επεξεργαστών του συστήματος, ενώ η συνολική ισχύς για μεγάλους πίνακες αυξάνεται μόνο κατά 12.5%. Η διαφορά αυτή είναι αναμενόμενη, καθώς ο πυρήνας SpMV υποχρησιμοποιεί των επεξεργαστή για πολύ αραιούς πίνακες ($IPC < 1$), λόγω της συμφόρησης στην κύρια μνήμη. Ωστόσο, ακόμα και αυτοί οι αριθμοί εκθέτουν την μισή αλήθεια, καθώς η κατανάλωση ισχύος της κύριας μνήμης δεν συνυπολογίζεται. Οι προσβάσεις στην κύρια μνήμη καταναλώνουν περίπου 50 φορές περισσότερη ενέργεια σε σχέση με τις αριθμητικές πράξεις [Dongara, 2012]. Πράγματι, οι Kamil et al. [2008] δείχνουν ότι εφαρμογές με έντονο ρυθμό προσβάσεων στην μνήμη, όπως είναι τα μετροπρογράμματα STREAM και CG, οδηγούν σε μεγαλύτερη κατανάλωση στην κλίμακα του συνολικού συστήματος σε σχέση με αυτές που εντείνουν την χρήση της CPU. Και ο πυρήνας SpMV δεν αποτελεί εξαίρεση. Είναι επομένως σημαντικό να θεωρήσει κανείς την διελκυστίνδα επίδοσης-κατανάλωσης ενέργειας και από την πλευρά της πλευρά της χρη-

7. Θέματα ενεργειακής απόδοσης

σιμοποίησης του συνολικού συστήματος. Παρά τα διάφορα ερωτήματα που πρέπει ακόμα να απαντηθούν προς αυτή την κατεύθυνση, πιστεύουμε ότι τα πρωτογενή αποτελέσματα που παρουσιάστηκαν σε αυτό το κεφάλαιο για την αναγνώριση και πρόβλεψη των βελτίστων ρυθμίσεων εκτέλεσης θα παραμείνουν στο προσκήνιο και στην περίπτωση της ανάλυσης επίδοσης-ενέργειας του συνολικού συστήματος, όπου η κατανάλωση ισχύος της κύριας μνήμης θα παίζει καθοριστικό ρόλο.

Συμπεράσματα

Η διατριβή αυτή εστίασε στην βελτιστοποίηση του υπολογιστικού πυρήνα Πολλαπλασιασμού Αραιού Πίνακα με Διάνυσμα στις σύγχρονες πολυπύρηνες αρχιτεκτονικές υπολογιστών. Πραγματοποιήσαμε μία εις βάθος ανάλυση του πυρήνα και αναδείξαμε τα κυριότερα προβλήματα επίδοσης. Αυτό μας επέτρεψε να προτείνουμε ένα προχωρημένο σχήμα αποθήκευσης αραιών πινάκων, το σχήμα Compressed Sparse eXtended (CSX), το οποίο εστιάζει στοχευμένα στην ελαχιστοποίηση του μεγέθους αναπαράστασης του αραιού πίνακα. Το σχήμα αυτό μπορεί να παράσχει σημαντική βελτίωση της επίδοσης του SpMV σε μία πληθώρα διαφορετικών πινάκων και πολυπύρηνων αρχιτεκτονικών, διατηρώντας μία σημαντική σταθερότητα στην επίδοση. Τέλος, μελετήσαμε την επίδοση του SpMV υπό το πρίσμα της ενεργειακής του απόδοσης, με σκοπό την αναγνώριση των ρυθμίσεων εκτέλεσης που οδηγούν σε βέλτιστους συμβιβασμούς επίδοσης-κατανάλωσης ενέργειας. Το κεφάλαιο αυτό συνοψίζει τα βασικά συμπεράσματα και επιτεύγματα της διατριβής αυτής, παρέχοντας, επίσης, την θεώρηση του γράφοντος επί των μελλοντικών σχετικών προοπτικών και κατευθύνσεων έρευνας.

8.1 Ο πυρήνας SpMV: Θύμα της αλγοριθμικής του φύσης

Ο πυρήνας SpMV βρίσκεται στην καρδιά των επαναληπτικών μεθόδων επίλυσης αραιών γραμμικών συστημάτων, τα οποία προκύπτουν σε μία πληθώρα επιστημονικών πεδίων. Το βασικό πρόβλημα επίδοσης του SpMV προέρχεται πρωτίστως από την ίδια του την αλγοριθμική φύση, η οποία του επιβάλλει ένα πολύ χαμηλό λόγο flops:byte, ασκώντας τελικά ασφυκτική πίεση στο υποσύστημα μνήμης. Πραγματοποιώντας μία εις βάθος ανάλυση της επίδοσης του πυρήνα SpMV σε αυτή την διατριβή, αναδείξαμε αυτό το χαρακτηριστικό –κάπως αγνοημένο κατά το παρελθόν– ως το κύριο πρόβλημα επίδοσης του πυρήνα στις σύγχρονες πολυπύρηνες αρχιτεκτονικές. Στις αρχιτεκτονικές SMP, το πρόβλημα αυτό καθορίζει πλήρως την πολυνηματική επίδοση του SpMV σε μεγάλους πίνακες, καθώς ο κοινός εμπρόσθιος δίαυλος των συστημάτων

8. Συμπεράσματα

αυτών κορέννεται πολύ γρήγορα. Στις αρχιτεκτονικές NUMA, από την άλλη πλευρά, οι ενσωματωμένοι ελεγκτές μνήμης αφήνουν μεγαλύτερα περιθώρια και το μονοπάτι μνήμης δεν κορέννεται μέχρι την πλήρη χρησιμοποίηση του συστήματος· το γεγονός αυτό επιτρέπει την χρήση και υπολογιστικών βελτιστοποιήσεων, οι οποίες δεν έχουν κανένα αντίκτυπο στην επίδοση στις αρχιτεκτονικές SMP. Ωστόσο, η επίδοση του SpMV στις αρχιτεκτονικές NUMA είναι ιδιαίτερα ευάλωτη στην τοποθέτηση των δεδομένων των νημάτων στους κόμβους μνήμης του συστήματος, καθώς το δίκτυο διασύνδεσης των επεξεργαστών μπορεί εύκολα να κορεστεί. Η διατήρηση της σωστής τοποθέτησης των κοινών δομών δεδομένων μπορεί να απαιτεί σημαντική προγραμματιστική προσπάθεια και να οδηγήσει σε σημαντική αναδιάρθρωση ένα ήδη υπάρχοντα κώδικα λογικής SMP. Για να διευκολύνουμε την διαδικασία αυτή, προτείναμε την χρήση ενός κατά-NUMA παρεμβλητικού εκχωρητή μνήμης, ο οποίος τοποθετεί διαφανώς ως προς τον χρήστη τα διαφορετικά τμήματα μιας κοινής δομής δεδομένων στους σωστούς κόμβους, χωρίς να επηρεάζει την κατά-SMP λογική του κώδικα· η απλή αντικατάσταση των κλήσεων εκχώρησης μνήμης αρκεί για την κατά-NUMA μετατροπή του κώδικα.

Στην διατριβή αυτή, εξετάσαμε επίσης σε βάθος τα σχήματα αποθήκευσης αραιών πινάκων σε μπλοκ. Η τεχνική αποθήκευσης σε μπλοκ είχε χρησιμοποιηθεί επιτυχώς στο παρελθόν ως μία εναλλακτική μέθοδος αποθήκευσης αραιών πινάκων, καθώς όχι μόνο επιτρέπει την μείωση του μεγέθους αναπαράστασης του πίνακα, αλλά συγχρόνως προσφέρει και καλά υπολογιστικά χαρακτηριστικά. Πραγματοποιήσαμε μία λεπτομερή ανάλυση της επίδοσης των πιο τυπικών σχημάτων αποθήκευσης σε μπλοκ, αναγνωρίζοντας τα πλεονεκτήματα και τα μειονεκτήματα του καθενός. Το βασικό συμπέρασμα αυτής της ανάλυσης ήταν η διελκυστίνδα μεταξύ συμπίεσης και πολυνηματικής επίδοσης μεταξύ των μεθόδων μεταβλητού και σταθερού μεγέθους μπλοκ. Παρά το σημαντικό κέρδος στο μέγεθος της αναπαράστασης του πίνακα, ειδικά σε πίνακες με ακανόνιστη δομή, οι μέθοδοι μπλοκ μεταβλητού μεγέθους θα πρέπει να πληρώσουν το κόστος των επιπλέον δομών που απαιτούνται για την διατήρηση των μεγεθών των μπλοκ. Η αυξημένη συμπίεση αρχίζει να αποδίδει καρπούς καθώς αυξάνεται το πλήθος των νημάτων και η πίεση στο υποσύστημα μνήμης γίνεται κρίσιμη για την επίδοση του πυρήνα SpMV· στις πολυνηματικές εκτελέσεις, επομένως, οι μέθοδοι μπλοκ μεταβλητού μεγέθους μπορεί να υπερβούν κατά πολύ την επίδοση των μεθόδων σταθερού μεγέθους μπλοκ. Από την άλλη πλευρά, οι μέθοδοι σταθερού μεγέθους μπλοκ, οδηγούν σε απλότερο πυρήνα, επιτρέποντας την παραγωγή καλύτερου κώδικα από τον συμπιληστή, ενώ επιτρέπουν και υπολογιστικές βελτιστοποιήσεις, όπως είναι η τεχνική «vectorization». Εστιάζοντας στο σχήμα BCSR, το πιο αντιπροσωπευτικό σχήμα αποθήκευσης σε μπλοκ σταθερού μεγέθους, εξετάσαμε την επίδραση των διαστάσεων του μπλοκ στην επίδοση του πυρήνα SpMV. Η επίδραση αυτή

μπορεί να καταστεί ιδιαίτερα σημαντική σε περιπτώσεις όπου το εύρος ζώνης μνήμης δεν έχει κορεστεί πλήρως και, ως εκ τούτου, υπολογιστικές βελτιστοποιήσεις μπορούν να προσφέρουν σημαντικό όφελος επίδοσης. Προς αυτή την κατεύθυνση, αναπτύξαμε ένα μοντέλο επίδοσης που λαμβάνει υπόψη του τόσο το τμήμα προσβάσεων στην μνήμη όσο και το τμήμα υπολογισμών του πυρήνα SpMV και μπορεί να προβλέψει με ακρίβεια το βέλτιστο μπλοκ του BCSR.

8.2 CSX: Βιώσιμη λύση προς την υψηλή επίδοση του SpMV

Δύο είναι τα βασικά μειονεκτήματα των προηγούμενων προσεγγίσεων για την βελτιστοποίηση της επίδοσης του πυρήνα SpMV. Πρώτον, οι προσεγγίσεις αυτές δεν είναι αρκετά εστιασμένες στην ελαχιστοποίηση του μεγέθους αναπαράστασης του αραιού πίνακα, καθώς προέρχονται από διαφορετικά υπόβαθρα, π.χ., βελτιστοποίηση χρήσης καταχωρητών, και, δεύτερον, ακολουθούν μία τακτική «όλα-ή-τίποτα». Για παράδειγμα, το σχήμα BCSR θα παράσχει σημαντική βελτίωση της επίδοσης σε πίνακες που κυριαρχούνται από πυκνές διαστάσεις περιοχές, αλλά η επίδοσή του θα καταρρακωθεί στην αντίθετη περίπτωση. Με κίνητρο αυτές τις ελλείψεις των εναλλακτικών σχημάτων για τον πυρήνα SpMV, προτείναμε σε αυτή την διατριβή το σχήμα Compressed Sparse eXtended format (CSX), ένα ενοποιημένο σχήμα αποθήκευσης άμεσης συμπίεσης δεδομένων. Το σχήμα CSX μπορεί να ανιχνεύει και να κωδικοποιεί στην ίδια αναπαράσταση μία πληθώρα κανονικοτήτων μη μηδενικών στοιχείων, που εμφανίζονται στους αραιούς πίνακες, και συγκεκριμένα οριζόντιες, κατακόρυφες, διαγώνιες, αντιδιαγώνιες και διαστάτες κανονικότητες (μπλοκ). Σε συνδυασμό με την πολύ συμπιεσμένη αναπαράσταση, βασιζόμενη στην κωδικοποίηση run-length των στηλών του πίνακα, το σχήμα CSX μπορεί να μειώσει το μέγεθος αναπαράστασης του πίνακα κοντά στο θεωρητικό όριο. Ως αποτέλεσμα, το σχήμα CSX παρέχει σημαντικά οφέλη επίδοσης (υπερβαίνοντας το 50% κατά μέσο όρο) σε συστήματα SMP, όπου η συμφόρηση στη μνήμη είναι περισσότερο έντονη. Στις αρχιτεκτονικές NUMA, λαμβάνουμε ιδιαίτερη μνεία στην εξισορρόπηση της επιβάρυνσης της αποσυμπίεσης. Πιο συγκεκριμένα, «χαλαρώνουμε» το σχήμα συμπίεσης και χρησιμοποιούμε μία προχωρημένη ευρετική συνάρτηση επιλογής κανονικοτήτων, η οποία συνυπολίζει την υπολογιστική επιβάρυνση της αποσυμπίεσης. Οι βελτιστοποιήσεις αυτές επιτρέπουν στο CSX να παράσχει μία βελτίωση μεγαλύτερη του 20% κατά μέσο όρο σε συστήματα NUMA, υπερβαίνοντας την επίδοση των άλλων εναλλακτικών σχημάτων.

Ένα σημαντικό χαρακτηριστικό του CSX είναι η σταθερότητα της επίδοσής του. Παρά την σημαντική βελτίωση της επίδοσης σε πίνακες με πολλές κανονικότητες, η επίδοσή του είναι τουλάχιστον συγκρίσιμη με του αυτή του

βασικού σχήματος CSR στους πολύ αραιούς και ακανόνιστους πίνακες, αποδεικνύοντας έτσι μία υψηλή δυνατότητα προσαρμογής στην δομή του πίνακα. Επιπλέον, καταβάλαμε σημαντική προσπάθεια για την μείωση του κόστους προεπεξεργασίας (ανίχνευση και κωδικοποίηση κανονικοτήτων), με σκοπό να καταστεί το CSX ένα βιώσιμο εναλλακτικό σχήμα υπό το πρίσμα πραγματικών επιλυτών. Πράγματι, το CSX μπόρεσε να επιταχύνει την επίδοση του επιλυτή Elmer κατά περίπου 15%, παρά το κόστος προεπεξεργασίας του και το αυξημένο κόστος preconditioning των προβλημάτων αξιολόγησης, που χρησιμοποιήθηκαν.

Μία σημαντική επέκταση του σχήματος CSX, που προτάθηκε σε αυτή την διατριβή, είναι η υποστήριξη συμμετρικών αραιών πινάκων. Οι συμμετρικοί αραιοί πίνακες εμφανίζονται συνήθως κατά την διακριτοποίηση των ΜΔΕ και είναι αρκετά ελκυστικό –στο πλαίσιο του πυρήνα SpMV– να αποθηκεύσει κανείς μόνο το κάτω τριγωνικό τους κομμάτι και την κύρια διαγώνιο. Παρά το εμφανές όφελος αυτής της διάταξης στην μείωση του μεγέθους αναπαράστασης του πίνακα, η πολυνηματική εκτέλεση του συμμετρικού πυρήνα SpMV εισάγει μία εξάρτηση RAW στο διάνυσμα εξόδου, καθιστώντας την αποδοτική υλοποίηση του πυρήνα προβληματική. Οι περισσότερες προσεγγίσεις απαλείφουν αυτή την εξάρτηση με χρήση είτε τοπικών, ανα-νήμα, διανυσμάτων είτε επιλεκτικών κλειδωμάτων. Η χρήση ανά-νήμα τοπικών διανυσμάτων απαιτεί ένα επιπλέον βήμα αναγωγής, του οποίου η επιβάρυνση αυξάνει γραμμικά με το πλήθος των νημάτων, περιορίζοντας έτσι σημαντικά την κλιμάκωση της επίδοσης, ενώ η χρήση υπερβολικού αριθμού κλειδωμάτων, από την άλλη πλευρά, είναι απαγορευτική. Η δική μας προσέγγιση στην βελτιστοποίηση του συμμετρικού πυρήνα SpMV εκμεταλλεύεται το γεγονός ότι τα τοπικά διανύσματα γίνονται ολοένα και πιο αραιά καθώς αυξάνεται το πλήθος των νημάτων. Για τον λόγο αυτό, χρησιμοποιούμε μία δομή δεικτοδότησης των τοπικών διανυσμάτων, ώστε να αναγάγουμε μόνο τα συγκρουόμενα στοιχεία· όλες οι άλλες εγγραφές κατευθύνονται απευθείας στο τελικό διάνυσμα εξόδου. Η τεχνική αυτή αποσυνδέει στην πράξη το κόστος της αναγωγής από το πλήθος των νημάτων και επιτρέπει την κλιμάκωση της επίδοσης του συμμετρικού πυρήνα SpMV. Σε συνδυασμό με την παραλλαγή του CSX για συμμετρικούς πίνακες, που οδηγεί σε βαθμούς συμπίεσης κοντά στο 67%, μπορέσαμε να επιταχύνουμε περισσότερο από $2\times$ την επίδοση του πυρήνα SpMV, μία βελτίωση ευδιάκριτη και στο πλαίσιο της επαναληπτικής μεθόδου CG.

8.3 Προς ένα ενεργειακά αποδοτικό πυρήνα SpMV

Στην διατριβή αυτή πραγματοποιήσαμε ένα πρώτο βήμα προς την αναγνώριση των συμβιβασμών επίδοσης-κατανάλωσης ενέργειας του πυρήνα SpMV. Κι-

νητοποιημένοι από την ένταση των προσβάσεων στην κύρια μνήμη, εξετάσαμε διαφορετικές συχνότητες ρολογιού των πυρήνων και τοποθετήσεις των νημάτων, με σκοπό να επιτύχουμε υψηλή επίδοση με ένα χαμηλό προϋπολογισμό ενέργειας. Οι σύγχρονοι επεξεργαστές υποστηρίζουν την δυναμική ρύθμιση της τάσης τροφοδοσίας τους και της συχνότητάς του, επιτρέποντας την σημαντική εξοικονόμηση ενέργειας, χωρίς να θυσιάζεται η επίδοση σε περιπτώσεις εφαρμογών με μικρή υπολογιστική ένταση. Ως τέτοια εφαρμογή, η επίδοση του πυρήνα SpMV εξαρτάται σε μεγάλο βαθμό από την τοποθέτηση των νημάτων στους διαθέσιμους πυρήνες, καθώς μία λογική τοποθέτηση μπορεί να ελαφρύνει την πίεση στο υποσύστημα μνήμης. Κατά συνέπεια, ο συνδυασμός χαμηλής συχνότητας του επεξεργαστή και μιας αποδοτικής τοποθέτησης των νημάτων μπορεί να οδηγήσει σε σημαντικά οφέλη επίδοσης, θυσιάζοντας μόνο ένα μικρό μέρος της συνολικής επίδοσης. Βασιζόμενοι στην υπόθεση ότι πίνακες με όμοια χαρακτηριστικά επίδοσης (κυρίως λόγω της δομής τους) θα παρουσιάζουν και αντίστοιχη συμπεριφορά στην κατανάλωση ενέργειας, προτείνουμε μία τεχνική, βασισμένη στη μάθηση μηχανών, για την πρόβλεψη των ρυθμίσεων εκτέλεσης που οδηγούν σε βέλτιστους συμβιβασμούς επίδοσης-κατανάλωσης ενέργειας. Τα πρώτα αποτελέσματα είναι αρκετά ελπιδοφόρα και προετοιμάζουν το έδαφος για μία περαιτέρω εξέταση του της ενεργειακής απόδοσης του πυρήνα SpMV.

8.4 Μελλοντικές ερευνητικές κατευθύνσεις

Η μεγάλη ένταση αναφορών στην κύρια μνήμη είναι εγγενές πρόβλημα του πυρήνα SpMV. Στην πράξη, πρόκειται για ένα από τους πυρήνες με τον πιο μικρό λόγο flop:byte και η επίδοσή του στο άμεσο μέλλον αναμένεται να περιορίζεται από το εύρος ζώνης μνήμης, εκτός και εάν πραγματοποιηθεί κάποια σημαντικό επίτευγμα στην τεχνολογία των κυρίων μνημών. Κατά συνέπεια, αναμένουμε η ιδέα της συμπίεσης δεδομένων, που χρησιμοποιείται και από το σχήμα CSX σε αυτή την διατριβή, να κεντρίσει περαιτέρω το ενδιαφέρον της κοινότητας τα επόμενα χρόνια. Ωστόσο, το σχήμα CSX έχει φθάσει στο όριο της συμπίεσης των μεταδεδομένων του πίνακα και, επομένως, είναι αναγκαία η επέκταση της συμπίεσης και στα καθαυτά δεδομένα του πίνακα, δηλαδή, στις τιμές των μη μηδενικών του στοιχείων. Το κέρδος επίδοσης των συμμετρικών σχημάτων αποθήκευσης προσφέρει μία καθαρή ένδειξη του οφέλους μιας αποδοτικής συμπίεσης των μη μηδενικών στοιχείων. Η κυριώτερη πρόκληση σε αυτή την προσέγγιση, ωστόσο, είναι ότι η συμπίεση θα πρέπει να διατηρεί την διπλή ακρίβεια των τιμών, ενώ απαιτείται και μία καλή εξισορρόπηση του κόστους αποσυμπίεσης, ώστε αυτό να μην εμποδίζει τον συνολικό υπολογισμό.

8. Συμπεράσματα

Οι τεχνικές συμπίεσης δεδομένων κεντρίζουν πρόσφατα το ενδιαφέρον όχι μόνο ως μέσο για την αντιμετώπιση της διαφοράς ταχύτητας κύριας μνήμης-επεξεργαστή στις εφαρμογές με μεγάλη ένταση προσβάσεων στην μνήμη, αλλά και ως μέσο για την μείωση της κατανάλωσης ενέργειας του υποσυστήματος μνήμης. Αυτό εισάγει μία επιπλέον διάσταση στην βελτιστοποίηση των εφαρμογών με μεγάλη ένταση προσβάσεων στην κύρια μνήμη και ανοίγει ενδιαφέρουσες ερευνητικές κατευθύνσεις. Στο πλαίσιο μιας επιστημονικής εφαρμογής, ένας συνδυασμός συμπίεσης δεδομένων και προχωρημένων τεχνικών διαχείρισης της κατανάλωσης ισχύος του επεξεργαστή θα μπορούσε να αποτελέσει την βάση για την ελαχιστοποίηση του ενεργειακού της αποτυπώματος και της μεγιστοποίησης της επίδοσής της.

Τέλος, από τεχνικής απόψεως, είναι σημαντικό να εμπλουτιστεί το σχήμα CSX με μία λειτουργική διεπαφή βιβλιοθήκης έτσι, ώστε να μπορεί να αξιοποιηθεί εύκολα από την κοινότητα HPC. Προς αυτή την κατεύθυνση, μία περαιτέρω βελτίωση του κόστους προεπεξεργασίας, μέσω της χρήση προχωρημένων τεχνικών απόκρυψής του, θα καταστήσει το σχήμα CSX περισσότερο ελκυστικό και θα βοηθήσει στην περαιτέρω διάχυσή του στην κοινότητα.

Κατάλογος δημοσιεύσεων

Κατά την διάρκεια της διατριβής αυτής παρήχθη ένα σύνολο δημοσιεύσεων σε έγκριτα διεθνή περιοδικά και συνέδρια υψηλής ποιότητας της κοινότητας HPC.

Διεθνή περιοδικά

- V. Karakasis, T. Gkountouvas, K. Kourtis, G. Goumas και N. Koziris. An extended compression format for the optimization of sparse matrix-vector multiplication. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*. To Appear.
- G. Goumas, K. Kourtis, N. Anastopoulos, V. Karakasis και N. Koziris. Performance evaluation of the sparse matrix-vector multiplication on modern architectures. *The Journal of Supercomputing*, 50(1):36–77, 2009.

Διεθνή συνέδρια

- V. Karakasis, G. Goumas, K. Nikas, N. Koziris, J. Ruokolainen και P. Råback. Using state-of-the-art sparse matrix optimizations for accelerating the performance of multiphysics simulations. Στο *PARA 2012: Workshop on State-of-the-Art in Scientific and Parallel Computing*, Helsinki, Finland, 2012. Springer.
- V. Karakasis, G. Goumas και N. Koziris. Exploring the performance-energy tradeoffs in sparse matrix-vector multiplication. Στο *Workshop on Emerging Supercomputing Technologies (WEST), ICS'11*, Tucson, AZ, USA, 2011.
- K. Kourtis, V. Karakasis, G. Goumas και N. Koziris. CSX: An extended compression format for SpMV on shared memory systems. Στο *16th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming (PPoPP'11)*, San Antonio, TX, USA, 2011. ACM.

- V. Karakasis, G. Goumas και N. Koziris. Performance models for blocked sparse matrix-vector multiplication kernels. Στο *38th International Conference on Parallel Processing (ICPP'09)*, pages 356–364, Vienna, Austria, 2009. IEEE Computer Society.
- V. Karakasis, G. Goumas και N. Koziris. A comparative study of blocking storage methods for sparse matrices on multicore architectures. Στο *12th IEEE International Conference on Computational Science and Engineering (CSE-09)*, Vancouver, Canada, 2009. IEEE Computer Society.
- V. Karakasis, G. Goumas και N. Koziris. Exploring the effect of block shapes on the performance of sparse kernels. Στο *IEEE International Symposium on Parallel & Distributed Processing (IPDPS'09)*, pages 1–8, Rome, Italy, 2009. IEEE Computer Society.
- G. Goumas, K. Kourtis, N. Anastopoulos, V. Karakasis και N. Koziris. Understanding the performance of sparse matrix-vector multiplication. Στο *Proceedings of the 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP'08)*, Toulouse, France, 2008. IEEE Computer Society.
- G. Goumas, N. Drosinos, V. Karakasis και N. Koziris. Coarse-grain parallel execution for 2-dimensional PDE problems. Στο *21st IEEE International Parallel and Distributed Processing Symposium (IPDPS'07)*, pages 1–8, Long Beach, CA, USA, 2007. IEEE.

Βιβλιογραφία

- A. Agarwal, S. Mukhopadhyay, A. Raychowdhury, K. Roy και C. H. Kim. Leakage power analysis and reduction for nanoscale circuits. *IEEE Micro*, 26(2): 68–80, 2006.
- R. C. Agarwal, F. G. Gustavson και M. Zubair. A high performance algorithm using pre-processing for the sparse matrix-vector multiplication. Στο *Proceedings of Supercomputing'92*, pages 32–41, Minneapolis, MN, USA, 1992. IEEE Computer Society.
- K. Ahmed και K. Schuegraf. Transistor wars. *IEEE Spectrum*, 48(11):50–66, 2011.
- K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams και K. A. Yelick. The landscape of parallel computing research: A view from Berkeley. Technical Report UCB/EECS-2006-183, University of California, Berkeley, 2006.
- D. H. Bailey, E. Barszcz, J. T. Barton., D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan και S. K. Weeratunga. The NAS parallel benchmarks summary and preliminary results. Στο *Proceedings of the 1991 ACM/IEEE Conference on Supercomputing*, pages 158–165, Albuquerque, NM, USA, 1991. ACM.
- R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine και H. V. der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, 1987. ISBN 0-89871-328-5.
- V. H. F. Batista, G. O. Ainsworth Jr. και F. L. B. Ribeiro. Parallel structurally-symmetric sparse matrix-vector products on multi-core processors. *Computing Research Repository (CoRR)*, abs/1003.0952, 2010.

- M. Belgin, G. Back και C. J. Ribbens. Pattern-based sparse matrix representation for memory-efficient SMVM kernels. Στο *Proceedings of the 23rd international conference on Supercomputing (ICS'09)*, pages 100–109, Yorktown Heights, NY, USA, 2009. ACM.
- N. Bell και M. Garland. Implementing sparse matri-vector multiplication on throughput-oriented processors. Στο *SC'09 Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, Portland, OR, USA, 2009. ACM.
- S. Borkar. Design challenges of technology scaling. *IEEE Micro*, 19(4):23–29, 1999.
- D. M. Brooks, P. Bose, S. E. Schuster, H. Jacobson, P. N. Kudva, A. Buyuktosunoglu, J.-D. Wellman, V. Zyuban, M. Gupta και P. W. Cook. Power-Aware Microarchitecture: Design and Modeling Challenges for Next-Generation Microprocessors. *IEEE Micro*, 20(6):26–44, 2000.
- A. Buluç, J. T. Fineman, M. Frigo, J. R. Gilbert και C. E. Leiserson. Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks. Στο *Proceedings of the twenty-first annual Symposium on Parallelism in Algorithms and Architectures*, pages 233–244, Calgary, Canada, 2009. ACM.
- A. Buluç, S. Williams, L. Oliker και J. Demmel. Reduced-bandwidth multithreaded algorithms for sparse matrix-vector multiplication. Στο *IEEE International Parallel & Distributed Processing Symposium*, pages 721–733, Anchorage, AK, USA, 2011. IEEE Computer Society.
- A. Buttari, V. Eijkhout, J. Langou και S. Filippone. Performance optimization and modeling of blocked sparse kernels. *International Journal of High Performance Computing Applications*, 21(4):467–484, 2007.
- A. Buttari, J. Dongarra, J. Kurzak, P. Luszczek και S. Tomov. Using mixed precision for sparse matrix computations to enhance the performance while achieving 64-bit accuracy. *ACM Transactions on Mathematical Software*, 34(4):17–39, 2008.
- Ü. V. Çatalyürek και C. Aykanat. Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transactions on Parallel and Distributed Systems*, 10(7):673–693, 1999.

- J. Chang, S.-L. Chen, W. Chen, S. Chiu, R. Faber, R. Ganesan, M. Grgek, V. Lukka, W.-W. Mar, J. Vash, S. Rusu και K. Zhang. A 45nm 24MB on-die L3 cache for the 8-core multi-threaded Xeon®processor. Στο *Symposium on VLSI Circuits*, pages 152–153, Kyoto, Japan, 2009. IEEE.
- J. W. Choi, A. Singh και R. W. Vuduc. Model-driven autotuning of sparse matrix-vector multiply on gpus. Στο *Proceedings of the 15th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, Bangalore, India, 2010. ACM.
- K. Choi, R. Soma και M. Pedram. Dynamic voltage and frequency scaling based on workload decomposition. Στο *Proceedings of the 2004 International Symposium on Low Power Electronics and Design*, pages 174–179, Newport, CA, USA, 2004. ACM.
- P. Colella. Defining software requirements for scientific computing, 2004. DARPA's High Productivity Computer Systems (pres.).
- M. Curtis-Maury, A. Shah, F. Blagojevic, D. S. Nikolopoulos, B. R. de Supinski και M. Schulz. Prediction models for multi-dimensional power-performance optimization on many cores. Στο *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, pages 250–259, Toronto, Ontario, Canada, 2008. ACM.
- E. Cuthill και J. McKee. Reducing the bandwidth of sparse symmetric matrices. Στο *Proceedings of the 1969 24th National Conference*. ACM, 1969.
- T. Davis και Y. Hu. The university of Florida sparse matrix collection. *ACM Transactions on Mathematical Software*, 38:1–25, 2011. URL <http://www.cise.ufl.edu/research/sparse/matrices>.
- T. A. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, 2006. ISBN 0-89871-613-6.
- J. Dongara. Algorithmic and software challenges when moving towards exascale. Στο *PARA 2012: Workshop on the State-of-the-Art in Scientific and Parallel Computing*, Helsinki, Finland, 2012. Springer. (keynote pres.).
- I. S. Duff και J. K. Reid. Some design features of a sparse matrix code. *ACM Transactions on Mathematical Software*, 5(1):18–35, 1979.
- I. S. Duff, A. M. Erisman και J. K. Reid. *Direct Methods for Sparse Matrices*. Oxford University Press, 1989. ISBN 0-19853-421-3.

- S. C. Eisenstat, M. C. Gursky, M. H. Schultz και A. H. Sherman. Yale sparse matrix package I: The symmetric codes. *International Journal for Numerical Methods in Engineering*, 18(8):1145–1151, 1982.
- L. F. Escudero. Solving systems of sparse linear equations. *Advances in Engineering Software*, 6(3):141–147, 1984.
- J. H. Ferziger και M. Peric. *Computational Methods for Fluid Dynamics*. Springer, 3rd edition, 2001. ISBN 3-54042-074-6.
- K. Flautner, K. Nam Sung, S. Martin, D. Blaauw και T. Mudge. Drowsy caches: Simple techniques for reducing leakage power. Στο *29th Annual International Symposium on Computer Architecture*, pages 148–157, Anchorage, AL, USA, 2001. IEEE.
- A. Fog. *Instruction tables: Lists of instruction latencies, throughputs and micro-operation breakdowns for Intel, AMD and VIA CPUs*. Copenhagen University College of Engineering, Φεβρουάριος 2012.
- J. A. George. Computer implementation of the finite element method. Technical Report STAN-CS-71-208, Stanford University, Computer Science Department, 1971.
- R. Geus και S. Röllin. Towards a fast parallel sparse matrix-vector multiplication. *Parallel Computing*, 27:883–896, 2001.
- N. E. Gibbs, W. G. Poole και P. K. Stockmeyer. An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM Journal on Numerical Analysis*, 13(2):236–250, 1976.
- G. Goumas, K. Kourtis, N. Anastopoulos, V. Karakasis και N. Koziris. Understanding the performance of sparse matrix-vector multiplication. Στο *Proceedings of the 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing*, Toulouse, France, 2008. IEEE Computer Society.
- G. Goumas, K. Kourtis, N. Anastopoulos, V. Karakasis και N. Koziris. Performance evaluation of the sparse matrix-vector multiplication on modern architectures. *The Journal of Supercomputing*, 50(1):36–77, 2009.
- Green500. The Green500 List, Νοέμβριος 2012. URL <http://www.green500.org>.
- W. D. Gropp, D. K. Kaushik, D. E. Keyes και B. F. Smith. Toward realistic performance bounds for implicit CFD codes. Στο *Proceedings of Parallel CFD*, pages 233–240, Williamsburg, VA, USA, 1999. Elsevier.

- D. Guo και W. Gropp. Optimizing sparse data structures for matrix-vector multiply. *High Performance Computing Applications*, 25(1):115–131, 2011.
- M. Harris. Mapping computational concepts to GPUs. Στο *ACM SIGGRAPH 2005 Courses, Chapter 31*. ACM, 2005.
- S. Herbert και D. Marculescu. Analysis of dynamic voltage/frequency scaling in chip-multiprocessors. Στο *ACM/IEEE International Symposium on Low Power Electronics and Design*, pages 38–43, Portland, OR, USA, 2007. IEEE.
- M. R. Hestenes και E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6): 409–436, 1952.
- D. Hisamoto, W.-C. Lee, J. Kedzierski, H. Takeuchi, K. Asano, C. Kuo, E. Anderson, T.-J. King, J. Bokor και C. Hu. FinFET – A self-aligned double-gate MOSFET scalable to 20 nm. *IEEE Transactions on Electron Devices*, 47(12): 2320–2325, 2000.
- M. Hoemmen. *Communication-Avoiding Krylov Subspace Methods*. PhD thesis, University of California, Berkeley, 2010.
- C. Hu. Thin-body FinFET as scalable low voltage transistor. Στο *International Symposium on VLSI Technology, Systems, and Applications (VLSI-TSA)*, pages 1–4, Hsinchu, Taiwan, 2012. IEEE.
- M. Huang, M. Mehalel, R. Arvapalli και He. S. An energy efficient 32nm 20MB L3 cache for Intel®Xeon®processor E5 family. Στο *IEEE Custom Integrated Circuits Conference*, pages 1–4, San Jose, CA, USA, 2012. IEEE.
- E.-J. Im. *Optimizing the performance of sparse matrix-vector multiplication*. PhD thesis, University of California, Berkeley, 2000.
- E.-J. Im και K. A. Yelick. Optimizing sparse matrix computations for register reuse in SPARSITY. Στο *Proceedings of the International Conference on Computational Sciences – Part I*, pages 127–136. Springer-Verlag, 2001.
- E.-J. Im, K. Yelick και R. Vuduc. Sparsity: Optimization framework for sparse matrix kernels. *International Journal of High Performance Computing Applications*, 18:135–158, 2004.
- Intel®Xeon®Processor 7400 Series – Datasheet*. Intel Corp., Οκτώβριος 2008.
- Intel®64 and IA-32 Architectures Optimization Reference Manual*. Intel Corp., Μάρτιος 2009.

- Intel®64 and IA-32 Architectures Software Developer's Manual – Volume 3B: System Programming Guide, Part 2*. Intel Corp., Σεπτέμβριος 2010.
- C. Isci και M. Martonosi. Runtime power monitoring in high-end processors: Methodology and empirical data. Στο *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, San Diego, CA, USA, 2003. IEEE Computer Society.
- ITRS. International Technology Roadmap for Semiconductors, 2001. URL <http://www.itrs.net>.
- K. R. James και W. Riha. Convergence criteria for successive overrelaxation. *SIAM Journal on Numerical Analysis*, 12(2):137–143, 1975.
- G. Juckenland, S. Börner, M. Kluge, S. Kölling, W. E. Nagel, S. Pflüger, H. Röding, S. Seidl, T. William και R. Wloch. BenchIT – performance measurement and comparison for scientific applications. Στο *Parallel Computing Software Technology, Algorithms, Architectures and Applications*, volume 13 of *Advances in Parallel Computing*, pages 501–508. Elsevier, 2004.
- R. Kalla, B. Sinharoy, W. J. Starke και M. Floyd. Power7: IBM's next-generation server processor. *IEEE Micro*, 30(2):7–15, 2010.
- S. Kamil, J. Shalf και E. Strohmaier. Power efficiency in high performance computing. Στο *International Symposium on Parallel and Distributed Processing*, pages 1–8, Miami, FL, USA, 2008. IEEE.
- V. Karakasis, G. Goumas και N. Koziris. A comparative study of blocking storage methods for sparse matrices on multicore architectures. Στο *12th IEEE International Conference on Computational Science and Engineering*, Vancouver, Canada, 2009a. IEEE Computer Society.
- V. Karakasis, G. Goumas και N. Koziris. Exploring the effect of block shapes on the performance of sparse kernels. Στο *IEEE International Symposium on Parallel & Distributed Processing*, pages 1–8, Rome, Italy, 2009b. IEEE Computer Society.
- V. Karakasis, G. Goumas και N. Koziris. Performance models for blocked sparse matrix-vector multiplication kernels. Στο *International Conference on Parallel Processing*, Vienna, Austria, 2009c. IEEE.
- V. Karakasis, G. Goumas και N. Koziris. Exploring the performance-energy tradeoffs in sparse matrix-vector multiplication. Στο *Workshop on Emerging Supercomputing Technologies (WEST), ICS'11*, Tucson, AZ, USA, 2011.

- G. Karypis και V. Kumar. METIS – Unstructured graph partitioning and sparse matrix ordering system, version 2.0. Technical report, University of Minnesota, Department of Computer Science, 1995.
- S. Kaxiras και M. Martonosi. *Computer Architecture Techniques for Power-Efficiency*, volume 3 of *Synthesis Lectures on Computer Architecture*. Morgan & Claypool, 2008.
- S. Kaxiras, H. Zhigang και M. Martonosi. Cache decay: Exploiting generational behavior to reduce cache leakage power. Στο *28th Annual International Symposium on Computer Architecture*, pages 240–251, Göteborg, Sweden, 2001. IEEE.
- N. S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir και V. Narayanan. Leakage current: Moore’s law meets static power. *Computer*, 36(12):68–75, 2003.
- D. Koufaty και D. T. Marr. Hyperthreading technology in the Netburst microarchitecture. *IEEE Micro*, 23(2):56–65, 2003.
- K. Kourtis. *Data Compression Techniques for Performance Improvement of Memory-Intensive Applications on Shared Memory Architectures*. PhD thesis, National Technical University of Athens, 2010.
- K. Kourtis, G. Goumas και N. Koziris. Improving the performance of multithreaded sparse matrix-vector multiplication using index and value compression. Στο *Proceedings of the 37th International Conference on Parallel Processing*, Portland, Oregon, USA, 2008a. IEEE Computer Society.
- K. Kourtis, G. Goumas και N. Koziris. Optimizing sparse matrix-vector multiplication using index and value compression. Στο *Proceedings of the 5th conference on Computing Frontiers*, Ischia, Italy, 2008b. ACM.
- K. Kourtis, G. Goumas και N. Koziris. Exploiting compression opportunities to improve SpMxV performance on shared memory systems. *ACM Transactions on Architecture and Code Optimization*, 7(3), 2010.
- N. Kurd, J. Douglas, P. Mosalikanti και R. Kumar. Next generation Intel® microarchitecture (Nehalem) clocking architecture. Στο *IEEE Symposium on VLSI Circuits*, Honolulu, HI, USA, 2008. IEEE.
- C. Lattner. LLVM and Clang: Advancing compiler technology. Στο *Free and Open Source Developers’ European Meeting*, Brussels, Belgium, Φεβρουάριος 2011. URL <http://clang.llvm.org/>.

- C. Lattner και V. Adve. LLVM: A compilation framework for lifelong program analysis & transformation. Στο *International Symposium on Code Generation and Optimization (CGO'04)*, San Jose, CA, USA, 2004. IEEE Computer Society. URL <http://www.llvm.org/>.
- C. L. Lawson, R. J. Hanson, D. R. Kincaid και F. T. Korgh. Basic linear algebra subprograms for Fortran usage. *ACM Transactions on Mathematical Software*, 5(3):308–323, 1979.
- E. Lindholm, J. Nickolls, S. Oberman και J. Montrym. NVIDIA Tesla: A unified graphics and computing architecture. *IEEE Micro*, 28(2):39–55, 2008.
- S. Luke. *Essentials of Metaheuristics*. Lulu, first edition, 2011.
- M. Lyly, J. Ruokolainen και E. Järvinen. ELMER – a finite element solver for multiphysics. Στο *CSC Report on Scientific Computing*, 1999–2000. URL <http://www.csc.fi/english/pages/elmer>.
- T. G. Mattson, M. Riepen, T. Lehnig, P. Brett, W. Hass, P. Kennedy, J. Howard, S. Vangal, N. Borkar, G. Ruhl και S. Dighe. The 48-core SCC processor: the programmer's view. Στο *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC'10)*, pages 1–11, New Orleans, LA, USA, 2010. ACM.
- J. D. McCalpin. STREAM: Sustainable memory bandwidth in high performance computing, 1995. URL <http://www.cs.virginia.edu/stream/>.
- J. Mellor-Crummey και J. Garvin. Optimizing sparse matrix-vector product computations using unroll and jam. *International Journal of High Performance Computing Applications*, 18(2):225–236, 2004.
- T. C. Oppe και D. R. Kincaid. The performance of ITPACK on vector computers for solving large sparse linear systems arising in sample oil reservoir simulation problems. *Communications in Applied Numerical Methods*, 3(1): 23–29, 1987.
- Ch. H. Papadimitriou. The NP-Completeness of the bandwidth minimization problem. *Computing*, 16(3):263–270, 1976.
- J. C. Pichel, D. B. Heras, J. C. Cabaleiro και F. F. Rivera. Improving the locality of the sparse matrix-vector product on shared memory multiprocessors. Στο *Proceedings of the 12th Euromicro Conference on Parallel, Distributed and Network Based Processing*, pages 66–71. IEEE Computer Society, 2004.

- A. Pinar και M. T. Heath. Improving performance of sparse matrix-vector multiplication. Στο *Proceedings of the 1999 ACM/IEEE conference on Supercomputing*, Portland, OR, USA, 1999. ACM.
- U. W. Pooch και A. Nieder. A survey of indexing techniques for sparse matrices. *ACM Computing Surveys*, 5(2):109–133, 1973.
- M. Powell, S.-H. Yang, B. Falsafi, K. Roy και T. N. Vijaykumar. Gated-Vdd: a circuit technique to reduce leakage in deep-submicron cache memories. Στο *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 90–95, Rapallo, Italy, 2000. ACM.
- S. K. Raman, V. Pentkovski και J. Keshava. Implementing streaming SIMD extensions on the Pentium III processor. *IEEE Micro*, 20(4):47–57, 2000.
- A. Ramirez. Supercomputing: Past, present, and a possible future. Στο *International Conference on Embedded Computer Systems (SAMOS)*, page ii, Samos, Greece, 2011. IEEE.
- E. Rotem, A. Naveh, D. Rajwan, A. Ananthakrishnan και E Weissmann. Power-management architecture of the Intel microarchitecture code-named Sandy Bridge. *IEEE Micro*, 32(2):20–27, 2012.
- R. M. Russell. The CRAY-1 computer system. *Communications of the ACM*, 21(1):63–72, 1978.
- Y. Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of Computation*, 37(155):105–126, 1981.
- Y. Saad. *Numerical methods for large eigenvalue problems*. Manchester University Press ND, 1992.
- Y. Saad. SPARSKIT: A basic tool kit for sparse matrix computations, 1994.
- Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd edition, 2003. ISBN 0-89871-534-2.
- Y. Saad και M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.
- G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas και M. L. Scott. Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling. Στο *Proceedings of the Eighth International Symposium on High-Performance Computer Architecture*, pages 29–40, Cambridge, MA, USA, 2002. IEEE.

- L. Sheng, H. A. Jung, R. D. Strong, J. B. Brockman, D. M. Tullsen και N. P. Jouppi. McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. Στο *42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 469–480, New York, NY, USA, 2009. IEEE.
- K. Singh, M. Bhadauria και S. A. McKee. Real time power estimation and thread scheduling via performance counters. *ACM SIGARCH Computer Architecture News*, 37(2):46–55, 2009.
- T. Skotnicki, J. A. Hutchby, T.-J. King, H.-S. P. Wonk και F. Boeuf. The end of CMOS scaling: toward the introduction of new materials and structural changes to improve MOSFET performance. *IEEE Circuits and Devices Magazine*, 21(1):16–26, 2005.
- V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969.
- O. Temam και W. Jalby. Characterizing the behavior of sparse algorithms on caches. Στο *Proceedings of Supercomputing'92*, pages 578–587, Minneapolis, MN, USA, 1992. IEEE Computer Society.
- R. P. Tewarson. *Sparse Matrices*. Academic Press, 1973. ISBN 0-124-11098-3.
- W. F. Tinney και J. W. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. *IEEE Proceedings*, 55(11):1801–1809, 1967.
- S. Toledo. Improving memory-system performance of sparse matrix-vector multiplication. *IBM Journal of Research and Development*, 41:711–725, 1997.
- Top500. TOP500 Supercomputer sites, Νοέμβριος 2012. URL <http://www.top500.org>.
- D. M. Tullsen, S. J. Eggers και H. M. Levy. Simultaneous multithreading: Maximizing on-chip parallelism. Στο *Proceedings of the 22nd Annual International Symposium on Computer Architecture*, S. Margherita Ligure, Italy, 1995. ACM.
- H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.
- R. Vuduc. *Automatic Performance Tuning of Sparse Matrix Kernels*. PhD thesis, University of California, Berkeley, 2003.

- R. Vuduc, J. W. Demmel, K. A. Yelick, S. Kamil, R. Nishtala και B. Lee. Performance optimizations and bounds for sparse matrix-vector multiply. Στο *Proceedings of the ACM/IEEE conference on Supercomputing*, pages 1–35, Baltimore, MD, USA, 2002. IEEE Computer Society.
- R. Vuduc, J. W. Demmel και K. A. Yelick. OSKI: A library of automatically tuned sparse matrix kernels. *Journal of Physics: Conference Series*, 16(521), 2005.
- R. W. Vuduc και H.-J. Moon. Fast sparse matrix-vector multiplication by exploiting variable block structure. Στο *High Performance Computing and Communications*, volume 3726 of *Lecture Notes in Computer Science*, pages 807–816. Springer Berlin/Heidelberg, 2005.
- J. White και P. Sadayappan. On improving the performance of sparse matrix-vector multiplication. Στο *Proceedings of the Fourth International Conference on High-Performance Computing*, pages 66–71. IEEE Computer Society, 1997.
- J. Willcock και A. Lumsdaine. Accelerating sparse matrix computations via data compression. Στο *Proceedings of the 20th annual International conference on Supercomputing*, pages 307–316, Cairns, QLD, Australia, 2006. ACM.
- S. Williams, L. Oiler, R. Vuduc, J. Shalf, K. Yelick και J. Demmel. Optimization of sparse matrix-vector multiplication on emerging multicore platforms. Στο *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, Reno, NV, USA, 2007. ACM.
- S. Williams, A. Waterman και D. Patterson. Roofline: An insightful visual performance model for multicore architectures. *Communications of the ACM – A Direct Path to Dependable Software*, 52(4):65–76, 2009.
- A. Wm. Wulf και S. A. McKee. Hitting the memory wall: implications of the obvious. *ACM SIGARCH Computer Architectures News*, 23(1), 1995.
- F. Xie, M. Martonosi και S. Malik. Compile-time dynamic voltage scaling settings: Opportunities and limits. Στο *Proceedings of the ACM SIGPLAN 2003 conference on Programming Language Design and Implementation*, pages 49–62, San Diego, CA, USA, 2003. ACM.
- F.-L. Yang, D.-H. Lee, H.-Y. Chen, C.-Y. Chang, S.-D. Liu, C.-C. Huang, T.-X. Chung, H.-W. Chen, C.-C. Huang, Y.-H. Liu, C.-C. Wu, C.-C. Chen, S.-C. Chen, Y.-Ts. Chen, Y.-H. Chen, C.-J. Chen, B.-W. Chan, P.-F. Hsu, J.-H. Shieh,

- H.-J. Tao, Y.-C. Yeo, Y. Li, J.-W. Lee, P. Chen, M.-S. Lian και C. Hu. 5nm-gate nanowire FinFET. Στο *Symposium on VLSI Technology*, pages 196–197, Honolulu, HI, USA, 2004. IEEE.
- S. Yang, M. D. Powell, B. Falsafi, K. Roy και T. N. Vijaykumar. An integrated circuit/architecture approach to reducing leakage in deep-submicron high-performance I-caches. Στο *Seventh International Symposium on High-Performance Computer Architecture*, pages 147–157, Monterey, Mexico, 2001. IEEE.
- Y.-P. You, C. Lee και J.-K. Lee. Compilers for leakage power reduction. *ACM Transactions on Design Automation of Electronic Systems*, 11(1):147–164, 2006.
- W. Zhang, J. S. Hu, V. Degalahal, M. Kandemir, N. Vijaykrishnan και M. J. Irwin. Compiler-directed instruction cache leakage optimization. Στο *35th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 208–218, Istanbul, Turkey, 2002. IEEE.
- W. Zhang, M. Kandemir, N. Vijaykrishnan, M. J. Irwin και V. De. Compiler support for reducing leakage energy consumption. Στο *Conference and Exhibition of Design, Automation and Test in Europe*, pages 1146–1147, Munich, Germany, 2003. IEEE.

Ευρετήριο

- 5-CV, *βλέπε* five-cross-validation
- BCSD, *βλέπε* Row Segmented Diagonal
- BCSR, *βλέπε* Blocked Compressed Sparse Row
- COO, *βλέπε* Coordinate
- CSB, *βλέπε* Compressed Sparse Block
- CSC, *βλέπε* Compressed Sparse Column
- CSR, *βλέπε* Compressed Sparse Row
- CSR-DU, *βλέπε* CSR Delta Units
- CSR-VI, *βλέπε* CSR Value Indexed
- CSRC, *βλέπε* Compressed Sparse Row-Column
- CSX, *βλέπε* Compressed Sparse eXtended
- DCSR, *βλέπε* Delta-Coded Sparse Row
- DVFS, *βλέπε* Dynamic Voltage and Frequency Scaling
- ELL, *βλέπε* Ellpack-Itpack
- ELLPACK, *βλέπε* Ellpack-Itpack
- ITPACK, *βλέπε* Ellpack-Itpack
- NOF, *βλέπε* Non-Overlapping Factor
- PBR, *βλέπε* Pattern Block Row
- RSDIAG, *βλέπε* Row Segmented Diagonal
- S-BCSR, *βλέπε* Streaming BCSR
- S-CSR, *βλέπε* Streaming CSR
- SIMD, *βλέπε* Single Instruction Multiple Data
- SS, *βλέπε* Super-Sparse
- SSE, *βλέπε* Streaming SIMD Extensions
- SSS, *βλέπε* Symmetric Sparse Skyline
- VBL, *βλέπε* Variable Block Length
- VBR, *βλέπε* Variable Block Row
- vectorization, 76
- horizontal add, 77
 - shuffle unit, 76
 - Single Instruction Multiple Data, 76
 - Streaming SIMD Extensions, 76
 - ευθυγραμμισμένες φορτώσεις, 77
 - μη ευθυγραμμισμένες φορτώσεις, 77
- απώλειες glitching, *βλέπε* απώλειες βραχυκυκλώματος
- αραιοί πίνακες
- εύρος ζώνης, 37
 - κανονικότητα, 22
 - πυκνότητα, 19
 - συμπύση
 - κωδικοποίηση run-length, 33
 - κωδικοποίηση δέλτα, 33
 - σχήματα μπλοκ

- μεταβλητού μεγέθους, 26
 - σταθερού μεγέθους, 23
 - τεχνική οργάνωσης σε μπλοκ, 22
- βελτιστοποίηση
 - βαθμός Pareto, 173
 - μη κυριαρχούμενο μέτωπο Pareto, 167
- γραμμικά συστήματα
 - διάνυσμα αγνώστων, 1
 - εκ δεξιών διάνυσμα, 1
 - πίνακας συντελεστών, 1
- διαφορική δεικτοδότηση, *βλέπε* κωδικοποίηση δέλτα
- ενεργειακή απόδοση
 - γινόμενα energy-delay, 159
- κατανάλωση ισχύος
 - απώλειες διαρροών, 156
- κατανάλωση ισχύος
 - clock gating, 156
 - Dynamic Voltage and Frequency Scaling, 157
 - απώλειες βραχυκυκλώματος, 156
 - διαρροές κατωφλίου (sub-threshold leakage), 157
 - διαρροές πύλης (gate leakage), 157
 - δυναμική κατανάλωση ισχύος, 156
 - συντελεστής ενεργοποίησης, 156
- κωδικοποίηση run-length, 99
 - διαδρομές (runs), 99
- λόγος flop:byte, *βλέπε* αριθμητική ένταση
- μάθηση μηχανών
 - υπερεκπαίδευση, 172
- μέθοδοι επίλυσης
 - άμεσες, 2
- μοντέλα επίδοσης
 - MEMCOMP, 84
 - MEM, 82
- OVERLAP, 85
- SPARSITY, 83
- Non-Overlapping Factor, 85
- Ακρίβεια επιλογής, 86
- Ακρίβεια πρόβλεψης χρόνου εκτέλεσης, 86
- προφόρτωση δεδομένων, 85
- στατιστική ανάλυση
 - five-cross-validation, 173
 - outliers, 170
 - πτυχή (fold), 173
- σχήματα αποθήκευσης
 - Blocked Compressed Sparse Diagonal, *βλέπε* Row Segmented Diagonal
 - Blocked Compressed Sparse Row, 23
 - Compressed Sparse Block, 31
 - Compressed Sparse Column, 21
 - Compressed Sparse eXtended, 93, 95
 - γενικευμένα στοιχεία, 99
 - κεφαλή, 97
 - μονάδα, 97
 - μονάδα δέλτα, 97
 - μονάδα κανονικότητας, 97
 - πραγμάτωση κανονικότητας, 104
 - σώμα, 97
 - Compressed Sparse Row, 20
 - Compressed Sparse Row-Column, 37
 - Coordinate, 20
 - CSR Delta Units, 34
 - CSR Value Indexed, 31
 - Delta-Coded Sparse Row, 34
 - Ellpack-Itpack, 32
 - Pattern Block Row, 31
 - Row Segmented Diagonal, 24
 - Streaming BCSR, 33
 - Streaming CSR, 33

Super-Sparse, 31
Symmetric Sparse Skyline, 35,
134
Variable Block Length, 27
Variable Block Row, 28
διασπώμενα, 30
υπολογιστική ένταση, 40

Σύντομο Βιογραφικό Σημείωμα

Ο Βασίλειος Καρακάσης γεννήθηκε στις 5 Απριλίου 1982 στην Αθήνα. Αποφοίτησε από το Ενιαίο Λύκειο το 2000, οπότε και εισήχθη στην Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσοβίου Πολυτεχνείου. Παρακολούθησε με επιτυχία τον πενταετή κύκλο σπουδών και αποφοίτησε το 2005 με βαθμό 8,58/10. Το ίδιο έτος ξεκινά και η ερευνητική του συνεργασία με το Εργαστήριο Υπολογιστικών Συστημάτων της Σχολής Η.Μ.Μ.Υ. του Ε.Μ.Π. και τον Δεκέμβριο του 2006 γίνεται δεκτός ως υποψήφιος διδάκτορας της Σχολής Η.Μ.Μ.Υ. υπό την επίβλεψη του Αναπληρωτή Καθηγητή Νεκταρίου Κοζύρη. Τον Δεκέμβριο του 2012 εξετάζεται επιτυχώς στην διδακτορική του διατριβή από την επταμελή εξεταστική επιτροπή και τον Μάρτιο του 2013 ορκίζεται Διδάκτωρ της Σχολής Η.Μ.Μ.Υ. του Ε.Μ.Π.