



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

## Ανάλυση Επίδοσης Μηχανισμών TLB Prefetching

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΒΑΒΟΥΛΙΩΤΗΣ ΓΕΩΡΓΙΟΣ

Επιβλέπων : Γκούμας Γεώργιος  
Επικουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2018





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

## Ανάλυση Επίδοσης Μηχανισμών TLB Prefetching

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΒΑΒΟΥΛΙΩΤΗΣ ΓΕΩΡΓΙΟΣ

Επιβλέπων : Γκούμας Γεώργιος  
Επίκουρος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19η Μαρτίου 2018.

.....  
Κοζύρης Νεκτάριος  
Καθηγητής Ε.Μ.Π.

.....  
Παγουρτζής Αριστείδης  
Αν. Καθηγητής Ε.Μ.Π.

.....  
Γκούμας Γεώργιος  
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2018

.....  
**Βαβουλιώτης Γεώργιος**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Βαβουλιώτης Γεώργιος, 2018.  
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.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Πολλοί είναι οι μηχανισμοί που έχουν προταθεί για τη διατήρηση υψηλής απόδοσης του συστήματος μνημών. Χαρακτηριστικό παράδειγμα αποτελεί η προσωρινή αποθήκευση δεδομένων (caching). Ωστόσο, οι πολυάριθμες αστοχίες της Κρυφής Μνήμης Αναζήτησης Μετάφρασης (TLB) αποτελούν ακόμα ένα από τα σοβαρότερα προβλήματα στη προσπάθεια διατήρησης υψηλής απόδοσης. Επιτυχείς αναζητήσεις στην TLB, ικανοποιούν την ανάγκη για ταχύτερη μετάφραση διευθύνσεων. Αν οι αναζητήσεις στη TLB οδηγούν σε αποτυχίες, γίνεται διάσχιση του πίνακα σελίδων στην ιεραρχία μνήμης για εύρεση της ζητούμενης μετάφρασης, κάτι που προκαλεί μεγάλες επιβαρύνσεις. Τα υπολογιστικά συστήματα που χρησιμοποιούν τη τεχνική της Εικονικής Μνήμης έχουν ανάγκη από όσο το δυνατόν ταχύτερη μετάφραση εικονικών διευθύνσεων σε φυσικές. Καθώς τα working sets αυξάνονται με ρυθμό τον οποίο δεν μπορεί να ακολουθήσει η αύξηση του μεγέθους της TLB, οι αστοχίες της TLB γίνονται ολοένα και περισσότερες, προκαλώντας μεγάλες επιβαρύνσεις και επηρεάζοντας τελικά τη συνολική απόδοση του υπολογιστικού συστήματος. Στην παρούσα Διπλωματική Εργασία εστιάζουμε στην βελτίωση της επίδοσης της TLB μέσω Προανάκλησης Μεταφράσεων (TLB Prefetching). Σκοπός είναι να εξεταστεί αν και κατά πόσο η ιδέα του TLB Prefetching μπορεί να ανταποκριθεί στη προσπάθεια μείωσης των TLB αστοχιών με ταυτόχρονη μείωση της επιβάρυνσης όταν κάποιες από τις αστοχίες δεν μπορούν να εξαιρεθούν. Συγκεκριμένα, υλοποιούνται διάφορα ήδη γνωστά Σχήματα TLB Prefetching, όπως για παράδειγμα ο Ακολουθιακός Προανακλητής, ο Προανακλητής Απόστασης και ο Προανακλητής Τυχαίου Βήματος με κάποιες διαφοροποιήσεις από την αρχική τους έκδοση. Τα πειραματικά αποτελέσματα δείχνουν πως η Προανάκληση Μεταφράσεων μπορεί να βελτιώσει την επίδοση ορισμένων εφαρμογών. Όμως δεν υπάρχει ένας μοναδικός απλός μηχανισμός προανάκλησης που πετυχαίνει τα καλύτερα αποτελέσματα για όλες τις εφαρμογές. Συνεπώς, στην μελέτη αυτή παρουσιάζουμε τον σχεδιασμό και την υλοποίηση νέων μηχανισμών, και πιο συγκεκριμένα κάποια Υβριδικά Σχήματα TLB Prefetching που συνδυάζουν ταυτόχρονα πολλαπλούς απλούς προανακλητές, καθώς επίσης και έναν Tournament Προβλέπτη ο οποίος επιλέγει την ενεργοποίηση μόνο ενός από πολλαπλούς προανακλητές. Τα πειραματικά αποτελέσματα δείχνουν πως τα σχήματα αυτά καταφέρνουν να συνδυάσουν τα οφέλη πολλαπλών προανακλητών και να πετύχουν καλύτερα αποτελέσματα στο σύνολο των εφαρμογών. Ωστόσο, παρατηρείται πως σε κάποιες φάσεις της εκτέλεσης κάθε εφαρμογής, θα ήταν ευεργετική η απενεργοποίηση του TLB Prefetching λόγω πολλαπλών ανεπιτυχών προβλέψεων. Για το λόγο αυτό, προτείνουμε Μηχανισμούς Ελέγχου Προανάκλησης που είναι υπεύθυνοι για την αποφυγή άχρηστων προβλέψεων που επιφέρουν επιβάρυνση σε λειτουργίες μνήμης, με ταυτόχρονη διατήρηση της απόδοσης του εκάστοτε Σχήματος TLB Prefetching. Υλοποιούμε τέσσερις διαφορετικούς Μηχανισμούς Ελέγχου Προανάκλησης, οι δύο λαμβάνουν υπόψη τις πετυχημένες αναζητήσεις στον Prefetch Buffer ενώ οι άλλοι δύο το κόστος που έχει η προανάκληση κάθε μετάφρασης σε λειτουργίες μνήμης. Η πειραματική αξιολόγηση δείχνει πως ο Tournament Προβλέπτης με το Μηχανισμό MMU Confidence Throttling είναι ο καλύτερος Μηχανισμός TLB Prefetching της εν λόγω Διπλωματικής Εργασίας.

## Λέξεις κλειδιά

Κρυφή Μνήμη Αναζήτησης Μετάφρασης, Προανάκληση Μεταφράσεων, Εικονική Μνήμη, Μετάφραση Διευθύνσεων, Ακολουθιακός Προανακλητής, Προανακλητής Τυχαίου Βήματος, Προανακλητής Απόστασης, Προανακλητής Μηδενικού Κόστους, Υβριδικοί Μηχανισμοί Προανάκλησης,

Tournament Προανακλητής, Κόστος Μετάφρασης, Προανάκληση Μεταφράσεων βασισμένη σε Κόστος, Πίνακας Σελίδων, MMU Κρυφή Μνήμη, Μονάδα Διαχείρισης Μνήμης, Μεγάλες Σελίδες, BadgerTrap

## Abstract

Many are the mechanisms that have been proposed for maintaining a high-performance memory system. A typical example is caching. However, the numerous TLB misses are yet one of the most serious problems in keeping up high performance. Successful searches on the TLB satisfy the need for fast address translation. If searches on the TLB lead to failures, the page table is crossed in the memory hierarchy to find the requested translation, something that causes great overheads. Computer systems using Virtual Memory need the fastest address translation possible. As the working sets grow at a rate that can not be followed by the increase in TLB size, TLB failures are becoming more and more, causing great overheads and ultimately affecting the overall performance of the computing system. In this Diploma Thesis we focus on improving the performance of the TLB through TLB Prefetching. The aim is to examine whether and how the TLB Prefetching concept can respond to the TLB misses reduction effort while reducing the load when some of the misses can not be eliminated. In particular, various already known TLB Prefetching schemes are implemented, such as the Sequential Prefetcher, Distance Prefetcher and the Arbitrary Stride Prefetcher with some differentiations from their original version. The experimental results show that TLB Prefetching may improve the performance of certain applications. But there is no single simple prefetch mechanism that achieves the best results for all the applications. Therefore, in this study we present the design and implementation of new mechanisms, and more specifically some Hybrid TLB Prefetching Schemes that combine multiple simple prefetchers, as well as a Tournament Predictor who chooses to activate only one of multiple predictors. The experimental results show that these schemes manage to combine the benefits of multiple prefetchers and achieve better results across applications. However, it is noticeable that in some phases of execution of each application, it would be beneficial to disable TLB Prefetching due to multiple unsuccessful predictions. For this reason, we suggest Control Prefetching Mechanisms that are responsible for avoiding unnecessary predictions that impose overheads in memory references, while maintaining the performance of each TLB Prefetching Scheme. We implement four different Control Prefetching Mechanisms, the two take into account the successful searches in the Prefetch Buffer while the other two the cost of preloading each translation into memory references. The experimental evaluation shows that the Tournament Predictor with the MMU Confidence Throttling is the best TLB Prefetching mechanism of this Diploma Thesis.

## Key words

TLB, Prefetching, Virtual Memory, Address Translation, Stride Prefetcher, Arbitrary Stride Prefetcher, Distance Prefetcher, Free Prefetcher, Hybrid Prefetching Mechanisms, Tournament Prefetcher, Address Translation Cost, Cost Based TLB Prefetching, Page Table, MMU Cache, MMU, Huge Pages, BadgerTrap





## Ευχαριστίες

Για την πραγματοποίηση της παρούσας Διπλωματικής Εργασίας συνέβαλαν πολλά άτομα, το καθένα με τον δικό του ιδιαίτερο τρόπο. Αρχικά θα ήθελα να ευχαριστήσω τους κυρίους Κοζύρη Νεκτάριο και Γκούμα Γεώργιο για τη δυνατότητα που μου έδωσαν να αναλάβω τη συγκεκριμένη εργασία και για όλα τα τεχνικά μέσα που μου παρείχαν για τη διευκόλυνση του έργου μου. Κατά τη διάρκεια της μελέτης αυτής, συνεργάστηκα άψογα με τον κύριο Καρακώστα Βασίλη και του οφείλω ένα μεγάλο ευχαριστώ για τη πρακτική βοήθεια, τις ιδέες και τις συμβουλές που μου παρείχε. Ευχαριστώ πολύ επίσης τη Χλόη Αλβέρτη και τον Νίκα Κωνσταντίνο για τις ιδέες και τη βοήθεια που μου έδωσαν απλόχερα. Στη συνέχεια θα ήθελα να ευχαριστήσω τους φίλους μου Αναστάση, Δημήτρη και Νίκο για όλες τις όμορφες και άσχημες στιγμές που περάσαμε μαζί σαν παρέα καθώς και τις πολλές εμπειρίες που ζήσαμε όλα αυτά τα χρόνια. Ένα μεγάλο ευχαριστώ οφείλω επίσης στη κοπέλα μου Κωνσταντίνα, η οποία με στήριξε καθ' όλη τη διάρκεια της μελέτης αυτής και με βοήθησε να ξεπεράσω τα διάφορα εμπόδια που συνάντησα. Το μεγαλύτερο ευχαριστώ όμως το αξίζουν οι γονείς μου, Σπύρο και Ελένη, οι οποίοι έδωσαν τη δυνατότητα σε μένα και την αδερφή μου να σπουδάσουμε αυτό που επιθυμούσαμε χωρίς κανένα περιορισμό και πάντα φρόντιζαν να μας στηρίζουν και να μη μας λείπει απολύτως τίποτα.

Βαβουλιώτης Γεώργιος,  
Αθήνα, 19η Μαρτίου 2018



# Περιεχόμενα

Περίληψη . . . . .	5
Abstract . . . . .	7
Ευχαριστίες . . . . .	9
Περιεχόμενα . . . . .	11
Κατάλογος πινάκων . . . . .	13
Κατάλογος σχημάτων . . . . .	15
<b>1. Εισαγωγή . . . . .</b>	<b>19</b>
1.1 Εικονική Μνήμη . . . . .	19
1.1.1 Αρχιτεκτονική Υποστήριξη . . . . .	19
1.2 Κίνητρα . . . . .	20
1.3 Προσέγγιση και Συνεισφορά Μελέτης . . . . .	21
1.4 Οργάνωση Μελέτης . . . . .	21
<b>2. Θεωρητικό Υπόβαθρο . . . . .</b>	<b>23</b>
2.1 Βασικοί Όροι και Έννοιες . . . . .	23
2.2 Αρχιτεκτονική Υποστήριξη . . . . .	25
2.2.1 Πίνακας Σελίδων (Page Table) . . . . .	26
2.2.2 Κρυφή Μνήμη Αναζήτησης Μετάφρασης (TLB) . . . . .	29
2.2.3 MMU Κρυφή Μνήμη (MMU Cache) . . . . .	31
<b>3. Σχεδίαση και Υλοποίηση . . . . .</b>	<b>33</b>
3.1 Εισαγωγή . . . . .	33
3.2 Εισαγωγή της Προανάκλησης Μεταφράσεων στη TLB . . . . .	34
3.3 Τεχνικές Προανάκλησης Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης . . . . .	34
3.3.1 Ακολουθιακός Προανακλητής . . . . .	35
3.3.2 Προανακλητής Τυχαίου Βήματος . . . . .	35
3.3.3 Προανακλητής Απόστασης . . . . .	36
3.3.4 Προανακλητής Μηδενικού Κόστους . . . . .	37
3.4 Υβριδικές Τεχνικές Προανάκλησης Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης . . . . .	38
3.4.1 Υβριδικός Προανακλητής Μηδενικού Κόστους . . . . .	38
3.4.2 Υβριδικός Ακολουθιακός Προανακλητής . . . . .	39
3.5 Tournament Προβλέπτης . . . . .	40
3.6 Η Έννοια του Κόστους στη Προανάκληση Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης . . . . .	41
3.6.1 Μηχανισμοί Ελέγχου Προανάκλησης Επιτυχόντων Προβλέψεων . . . . .	41

3.6.1.1	Μηχανισμός Ελέγχου Προανάκλησης Επιτυχόντων Προβλέψεων βασισμένο σε Εκπαίδευση (Hit Interval Throttling) . . . . .	41
3.6.1.2	Δενδρικός Μηχανισμός Ελέγχου Προανάκλησης Επιτυχόντων Προ- βλέψεων (Hit Confidence Throttling) . . . . .	42
3.6.2	Μηχανισμοί Ελέγχου Προανάκλησης MMU Κρυφής Μνήμης . . . . .	43
3.6.2.1	Μηχανισμός Ελέγχου Προανάκλησης MMU Κρυφής Μνήμης βα- σιμένος σε Εκπαίδευση (MMU Interval Throttling) . . . . .	43
3.6.2.2	Δενδρικός Μηχανισμός Ελέγχου Προανάκλησης MMU Κρυφής Μνήμη (MMU Confidence Throttling) . . . . .	44
<b>4.</b>	<b>Μεθοδολογία</b> . . . . .	<b>47</b>
4.1	Εισαγωγή . . . . .	47
4.2	Πειραματικό Περιβάλλον . . . . .	47
4.3	BadgerTrap . . . . .	48
4.4	Μετροπρογράμματα . . . . .	49
4.5	Μετρικές Αξιολόγησης . . . . .	52
<b>5.</b>	<b>Αξιολόγηση</b> . . . . .	<b>55</b>
5.1	Εισαγωγή . . . . .	55
5.2	Σύγκριση Αποτελεσμάτων . . . . .	55
5.3	Αποτελέσματα Ακολουθιακού Προανακλητή . . . . .	60
5.4	Αποτελέσματα Προανακλητή Τυχαίου Βήματος . . . . .	64
5.5	Αποτελέσματα Προανακλητή Μηδενικού Κόστους . . . . .	68
5.6	Αποτελέσματα Προανακλητή Απόστασης . . . . .	71
5.7	Αποτελέσματα Υβριδικού Προανακλητή Μηδενικού Κόστους . . . . .	75
5.8	Αποτελέσματα Υβριδικού Ακολουθιακού Προανακλητή . . . . .	79
5.9	Αποτελέσματα Tournament Προβλέπτη . . . . .	83
5.10	Μελέτη Παραμέτρων . . . . .	87
5.11	Huge Pages . . . . .	89
<b>6.</b>	<b>Σχετική Βιβλιογραφία</b> . . . . .	<b>91</b>
6.1	Μηχανισμοί Προανάκλησης Μεταφράσεων για Μονοπύρηνια Συστήματα . . . . .	91
6.2	Μηχανισμοί Προανάκλησης Μεταφράσεων για Πολυπύρηνια Συστήματα . . . . .	92
6.3	Μηχανισμοί χωρίς Προανάκληση Μεταφράσεων . . . . .	92
<b>7.</b>	<b>Συμπεράσματα και Μελλοντικές Επεκτάσεις</b> . . . . .	<b>93</b>
7.1	Συμπεράσματα . . . . .	93
7.2	Μελλοντικές Επεκτάσεις . . . . .	94
	<b>Βιβλιογραφία</b> . . . . .	<b>95</b>

## Κατάλογος πινάκων

3.1	Όλες οι πιθανές μεταβολές του 2-bit προβλέπτη C2. . . . .	41
3.2	Όλες οι πιθανές μεταβολές του 2-bit προβλέπτη C1. . . . .	41
3.3	Όλες οι πιθανές μεταβολές του 2-bit προβλέπτη C0, για το Μηχανισμό Hit Confidence Throttling. . . . .	43
4.1	Ιεραρχία Κρυφής Μνήμης Αναζήτησης Μετάφρασης (TLB) για κάθε πυρήνα του υπολογιστικού μηχανήματος. . . . .	47
4.2	Αποτελέσματα του εργαλείου perf και τα αντίστοιχα performance events [5]. . . .	48
5.1	Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης για καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf. . .	59
5.2	Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Ακολουθιακού Προανακλητή. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf. . . . .	63
5.3	Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Προανακλητή Τυχαίου Βήματος. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf. . . . .	67
5.4	Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Προανακλητή Μηδενικού Κόστους. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf. . . . .	70
5.5	Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Προανακλητή Απόστασης. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf. . . . .	74
5.6	Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Υβριδικού Προανακλητή Μηδενικού Κόστους. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf. . . . .	78
5.7	Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Υβριδικού Ακολουθιακού Προανακλητή. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf. . . . .	82
5.8	Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Tournament Προβλέπτη. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf. . . . .	86



## Κατάλογος σχημάτων

2.1	Αφαιρετική αναπαράσταση των αντιστοιχίσεων από Εικονική σε Φυσική Μνήμη για δύο εφαρμογές που καταλαμβάνουν τον ίδιο εικονικό χώρο διευθύνσεων αλλά αντιστοιχούν σε διαφορετικές φυσικές περιοχές μνήμης. . . . .	23
2.2	Μηχανισμοί αρχιτεκτονικής υποστήριξης για τη μετάφραση εικονικών διευθύνσεων σε φυσικές διευθύνσεις. . . . .	25
2.3	Οι βασικές πληροφορίες που περιέχει μία Καταχώρηση Πίνακα Σελίδων, ανεξαρτήτως αρχιτεκτονικής. . . . .	26
2.4	Χωρισμός εικονικής διεύθυνσης σε ομάδες από δυαδικά ψηφία για τη δεικτοδότηση των διαφόρων επιπέδων του Πίνακα Σελίδων. . . . .	27
2.5	Σχεδιασμός του Πίνακα Σελίδων σε αρχιτεκτονική x86-64. Πρόκειται για Ιεραρχικό Τρόπο Οργάνωσης με τέσσερα (4) επίπεδα. . . . .	28
2.6	Τα πεδία της Κρυφής Μνήμης Αναζήτησης Μετάφρασης (TLB). . . . .	29
2.7	Υλοποίηση της Κρυφής Μνήμης Αναζήτησης Μετάφρασης (TLB) με CAM. . . . .	30
2.8	Η διαδικασία που λαμβάνει χώρα όταν μία εφαρμογή αιτεί μία λειτουργία μνήμης. . . . .	31
2.9	Η δομή της Memory Management Unit (MMU) cache. . . . .	32
3.1	Η διαδικασία που ακολουθείται σε κάθε TLB αστοχία όταν χρησιμοποιείται κάποιο Σχήμα Προανάκλησης Μεταφράσεων. Η διαδικασία παραγωγής νέων προβλέψεων γίνεται στο background. . . . .	34
3.2	Ο μηχανισμός του Ακολουθιακού Προανακλητή (Stride Prefetcher). . . . .	35
3.3	Ο μηχανισμός του Προανακλητή Τυχαιού Βήματος (Arbitrary Stride Prefetcher). . . . .	35
3.4	Ο μηχανισμός του Προανακλητή Απόστασης (Distance Prefetcher). . . . .	37
3.5	Ο μηχανισμός του Προανακλητή Μηδενικού Κόστους (Free Prefetcher). . . . .	38
3.6	Ο μηχανισμός του Υβριδικού Προανακλητή Μηδενικού Κόστους (Hybrid Distance and Free Prefetcher). . . . .	39
3.7	Ο μηχανισμός του Υβριδικού Ακολουθιακού Προανακλητή (Hybrid Distance and Stride Prefetcher). . . . .	39
3.8	Ο μηχανισμός του Tournament Προανακλητή (Tournament Prefetcher). . . . .	40
3.9	Αφαιρετική αναπαράσταση του Μηχανισμού Hit Interval Throttling. . . . .	42
3.10	Αφαιρετική αναπαράσταση του Μηχανισμού Hit Confidence Throttling. . . . .	43
3.11	Αφαιρετική αναπαράσταση του Μηχανισμού MMU Interval Throttling. . . . .	44
3.12	Αφαιρετική αναπαράσταση του Μηχανισμού MMU Confidence Throttling. . . . .	45
4.1	Ποσοστό κύκλων που ξοδεύονται για TLB αστοχίες, σύμφωνα με τα αποτελέσματα του εργαλείου perf. . . . .	51
4.2	Ποσοστό λάθους BadgerTrap όσο αφορά το πλήθος των TLB αστοχιών, χωρίς κανένα Σχήμα Προανάκλησης Μεταφράσεων. . . . .	51
4.3	Ποσοστό λάθους BadgerTrap όσο αφορά το πλήθος των TLB αστοχιών, μετά την εισαγωγή κάποιου Σχήματος Προανάκλησης Μεταφράσεων. . . . .	52
5.1	Ποσοστό Επιτυχημένων Προβλέψεων για καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων. . . . .	56

5.2	Ποσοστό Ακρίβειας Προβλέψεων για καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων. . . . .	57
5.3	Ποσοστό Προβλέψεων Μηδενικού Κόστους για καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων. . . . .	57
5.4	Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης για καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα κάθε Σχήματος Προανάκλησης Μεταφράσεων. Η σειρά παρουσίασης τους είναι: Ακολουθιακός Προανακλητής, Προανακλητής Τυχαίου Βήματος, Προανακλητής Μηδενικού Κόστους, Προανακλητής Απόστασης, Υβριδικός Προανακλητής Μηδενικού Κόστους, Υβριδικός Ακολουθιακός Προανακλητής και Tournament Προβλέπτης. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις. . . . .	58
5.5	Ποσοστό Επιτυχημένων Προβλέψεων Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	60
5.6	Ποσοστό Ακρίβειας Προβλέψεων Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	61
5.7	Ποσοστό Προβλέψεων Μηδενικού Κόστους Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	61
5.8	Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Ακολουθιακού Προανακλητή. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις. . . . .	62
5.9	Ποσοστό Επιτυχημένων Προβλέψεων Προανακλητή Τυχαίου Βήματος για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	64
5.10	Ποσοστό Ακρίβειας Προβλέψεων Προανακλητή Τυχαίου Βήματος για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	65
5.11	Ποσοστό Προβλέψεων Μηδενικού Κόστους Προανακλητή Τυχαίου Βήματος για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	65
5.12	Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Προανακλητή Τυχαίου Βήματος. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Προανακλητή Τυχαίου Βήματος για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις. . . . .	66
5.13	Ποσοστό Επιτυχημένων Προβλέψεων Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	68
5.14	Ποσοστό Ακρίβειας Προβλέψεων Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	69



5.15	Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Προανακλητή Μηδενικού Κόστους. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις. . . . .	69
5.16	Ποσοστό Επιτυχημένων Προβλέψεων Προανακλητή Απόστασης για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	71
5.17	Ποσοστό Ακρίβειας Προβλέψεων Προανακλητή Απόστασης για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	72
5.18	Ποσοστό Προβλέψεων Μηδενικού Κόστους Προανακλητή Απόστασης για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	72
5.19	Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Προανακλητή Απόστασης. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Προανακλητή Απόστασης για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις. . . . .	73
5.20	Ποσοστό Επιτυχημένων Προβλέψεων Υβριδικού Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	75
5.21	Ποσοστό Ακρίβειας Προβλέψεων Υβριδικού Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	76
5.22	Ποσοστό Προβλέψεων Μηδενικού Κόστους Υβριδικού Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	76
5.23	Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Υβριδικού Προανακλητή Μηδενικού Κόστους. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Υβριδικού Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις. . . . .	77
5.24	Ποσοστό Επιτυχημένων Προβλέψεων Υβριδικού Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	79
5.25	Ποσοστό Ακρίβειας Προβλέψεων Υβριδικού Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	80
5.26	Ποσοστό Προβλέψεων Μηδενικού Κόστους Υβριδικού Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	80

5.27	Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Υβριδικού Ακολουθιακού Προανακλητή. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Υβριδικού Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις. . . . .	81
5.28	Ποσοστό Επιτυχημένων Προβλέψεων Tournament Προβλέπτη για τις περιπτώσεις No Throttling, Hit Confidence Throttling, MMU Confidence Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	83
5.29	Ποσοστό Ακρίβειας Προβλέψεων Tournament Προβλέπτη για τις περιπτώσεις No Throttling, Hit Confidence Throttling, MMU Confidence Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	84
5.30	Ποσοστό Προβλέψεων Μηδενικού Κόστους Tournament Προβλέπτη για τις περιπτώσεις No Throttling, Hit Confidence Throttling, MMU Confidence Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων. . . . .	84
5.31	Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Tournament Προβλέπτη. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Tournament Προβλέπτη για τις περιπτώσεις No Throttling, Hit Confidence Throttling, MMU Confidence Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις. . . . .	85
5.32	Σύγκριση απόδοσης Tournament Προβλέπτη με την ιδανική περίπτωση απόδοσης.	87
5.33	Σύγκριση απόδοσης Tournament Προβλέπτη με την ιδανική περίπτωση απόδοσης.	88
5.34	Ποσοστό μείωσης TLB αστοχιών λόγω ενεργοποίησης του μηχανισμού των Huge Pages. . . . .	89

## Κεφάλαιο 1

### Εισαγωγή

Στο κεφάλαιο αυτό θα αναλυθούν οι βασικές έννοιες που συνθέτουν την Εικονική Μνήμη (Virtual Memory), η οποία βασίζεται σε σελιδοποίηση (paging) καθώς και η εξέλιξη των μηχανισμών υποστήριξης από πλευράς αρχιτεκτονικής για τη μετάφραση εικονικών διεύθυνσεων σε φυσικές διευθύνσεις (address translation). Στη συνέχεια αναλύονται τα κίνητρα που οδήγησαν στη πραγματοποίηση της μελέτης αυτής καθώς και τα διάφορα προβλήματα που συναντήθηκαν κατά την εκπόνησή της. Τέλος, αναφέρονται συνοπτικά όλα τα θέματα που θα αναλυθούν στη συνέχεια και γίνεται μία μικρής έκτασης παρουσίαση των συνεισφορών της μελέτης αυτής.

#### 1.1 Εικονική Μνήμη

Η Εικονική Μνήμη είναι μία τεχνική διαχείρισης μνήμης η οποία απαλοίζει το πρόβλημα της πεπερασμένης φυσικής μνήμης και παρέχει την ψευδαίσθηση στο προγραμματιστή πως έχει στη διάθεσή του απεριόριστη μνήμη. Η ψευδαίσθηση αυτή οδηγεί σε αύξηση της παραγωγικότητάς του. Επιπρόσθετα, οι διάφορες εφαρμογές εκτελούνται απομονωμένες αφού δεν έχουν να διαχειριστούν μοιραζόμενες περιοχές μνήμης. Αποτέλεσμα αυτού είναι η αύξηση της ασφάλειας και η καλύτερη χρησιμοποίηση της μνήμης.

Ωστόσο, η χρήση της Εικονικής Μνήμης δεν έχει μόνο θετικές αλλά και αρνητικές επιπτώσεις για το υπολογιστικό σύστημα. Για να είναι ορθή και συνεπής η λειτουργία της Εικονικής Μνήμης, είναι αναγκαία η εισαγωγή μίας διεπαφής μεταξύ του εικονικού και του φυσικού χώρου διευθύνσεων. Η διεπαφή αυτή ονομάζεται Πίνακας Σελίδων (Page Table). Στη πράξη κάθε λειτουργία μνήμης που πυροδοτείται από κάποια ενεργή εφαρμογή του συστήματος αφορά μία εικονική διεύθυνση. Επομένως, αρχικά θα πρέπει να βρεθεί η αντιστοίχιση της εικονικής διεύθυνσης σε φυσική διεύθυνση και μετά να εκτελεστεί η ζητούμενη λειτουργία μνήμης. Η διαδικασία αυτή πολλές φορές προκαλεί μεγάλη επιβάρυνση τόσο στην απόδοση του υπολογιστικού συστήματος [16, 17, 26, 32, 33] όσο και στη κατανάλωση ενέργειας [21, 27, 28, 29].

##### 1.1.1 Αρχιτεκτονική Υποστήριξη

Η υλοποίηση της Εικονικής Μνήμης για να είναι ορθή και συνεπής απαιτεί την συνύπαρξη μηχανισμών τόσο λογισμικού όσο και υλικού. Ο Πίνακας Σελίδων (Page Table) είναι η δομή λογισμικού που χρησιμοποιείται από την Εικονική Μνήμη, στην οποία είναι αποθηκευμένες οι αντιστοιχίσεις εικονικών σε φυσικές σελίδες. Η Μονάδα Διαχείρισης Μνήμης (MMU) είναι ο μηχανισμός υλικού ο οποίος είναι υπεύθυνος για την επιτάχυνση της μετάφρασης των εικονικών διευθύνσεων σε φυσικές. Στην αρχική της μορφή η Μονάδα Διαχείρισης Μνήμης (MMU) αποτελούταν μόνο από τη Κρυφή Μνήμη Αναζήτησης Μετάφρασης (TLB).

**Κρυφή Μνήμη Αναζήτησης Μετάφρασης (Translation Lookaside Buffer).** Η Κρυφή Μνήμη Αναζήτησης Μετάφρασης (TLB) είναι μία μικρού μεγέθους κρυφή μνήμη (cache), η οποία αποθηκεύει τις πιο πρόσφατες μεταφράσεις εικονικής μνήμης σε φυσική μνήμη. Η απόδοση της TLB είναι εκείνη που καθορίζει το ποσοστό επιτάχυνσης της μετάφρασης εικονικών σε φυσικές διευθύνσεις.

Σε κάθε λειτουργία μνήμης η επεξεργαστική μονάδα (CPU) αναζητά στη TLB την ζητούμενη μετάφραση. Σε περίπτωση επιτυχημένης αναζήτησης στη TLB, η μετάφραση επιστρέφεται γρήγορα στη CPU και συνεχίζεται η εκτέλεση της ζητούμενης λειτουργίας μνήμης. Σε περίπτωση αστοχίας, εκτελείται μία διάσχιση σελίδας (page walk), με σκοπό να βρεθεί η ζητούμενη αντιστοίχιση στο Πίνακα Σελίδων. Βέβαια κάθε διάσχιση σελίδας επιβαρύνει στο σύστημα με νέες λειτουργίες μνήμης ώστε να βρεθεί η ζητούμενη μετάφραση και να μεταφερθεί στη TLB. Συγκεκριμένα στην x86-64 αρχιτεκτονική κάθε διάσχιση σελίδας προσθέτει τέσσερις νέες λειτουργίες μνήμης. Αξίζει να σημειωθεί ότι νεότερες αρχιτεκτονικές υποστηρίζουν Page Tables με πέντε επίπεδα, όποτε κάθε page walk θα κοστίζει πέντε λειτουργίες μνήμης.

Με το πέρασμα των χρόνων, καθώς η Εικονική Μνήμη άρχισε να χρησιμοποιείται καθολικά, μελέτες έδειξαν τη σημαντικότητα της μετάφρασης σελίδων (address translation) για την απόδοση του υπολογιστικού συστήματος. Για το λόγο αυτό, πλέον η Μονάδα Διαχείρισης Μνήμης (MMU) δεν αποτελείται μόνο από τη TLB, αλλά περιλαμβάνει και νέες δομές. Πλέον χρησιμοποιούνται Πολυεπίπεδες Κρυφές Μνήμες Αναζήτησης Μετάφρασης για καθένα από τους πυρήνες του συστήματος (per-core multi-level TLBs). Η πιο κοινή περίπτωση περιλαμβάνει δύο TLB επίπεδα για κάθε πυρήνα, η L1 TLB και η L2 TLB. Η L1 TLB έχει μικρότερο μέγεθος και ως εκ τούτου είναι ταχύτερο στην αναζήτηση λόγω μικρότερου χρόνου πρόσβασης (access time). Επίσης υπάρχει υποστήριξη για Μεγάλες Σελίδες (Huge Pages), γεγονός το οποίο βοηθά στη βελτίωση της απόδοσης της TLB αλλά και την αύξηση του *TLB Reach* (2.2.3). Για παράδειγμα, στην αρχιτεκτονική x86-64, εκτός από 4KB σελίδες υπάρχει υποστήριξη από πλευράς TLB και για σελίδες μεγέθους 2MB και 1GB. Επίσης η Μονάδα Διαχείρισης Μνήμης περιλαμβάνει μία νέα δομή, την MMU Κρυφή Μνήμη (MMU Cache), η οποία αποτελείται από μικρού μεγέθους κρυφές μνήμες, αρμοδιότητα των οποίων είναι να αποθηκεύουν καταχωρήσεις από τα ενδιάμεσα επίπεδα του Πίνακα Σελίδων. Με το τρόπο αυτό μειώνουν το κόστος ενός Page Walk, αφού μία επιτυχία στη δομή της MMU Cache είναι ικανή να γλιτώσει μία, δύο ή και τρεις λειτουργίες μνήμης, δηλαδή να αποφύγει το διάβασμα κάποιων επιπέδων του Πίνακα Σελίδων από τη μνήμη. Τέλος, για μεγαλύτερη απόδοση πολλές φορές χρησιμοποιούνται και ξεχωριστά TLBs για εντολές και δεδομένα.

## 1.2 Κίνητρα

Συνήθης στόχος των υπολογιστικών συστημάτων που χρησιμοποιούν την Εικονική Μνήμη σαν τεχνική διαχείρισης μνήμης είναι η υψηλή απόδοση. Τροχοπέδη στην επιθυμία τους αυτή, αποτελεί η επιβάρυνση που εισάγεται λόγω της μετάφρασης διευθύνσεων από τον εικονικό στο φυσικό χώρο διευθύνσεων. Καθώς τα χρόνια περνούν, οι απαιτήσεις των εφαρμογών για μνήμη (memory footprint) αυξάνονται με πολύ γρήγορο ρυθμό σε αντίθεση με το μέγεθος της TLB το οποίο αδυνατεί να ακολουθήσει το ρυθμό αυτό. Ο λόγος είναι πως η TLB είναι στο κρίσιμο τμήμα και μία ενδεχόμενη απόπειρα αύξησης του μεγέθους της με ρυθμό παρόμοιο με αυτό των απαιτήσεων των εφαρμογών για μνήμη, δεν θα οδηγούσε στο επιθυμητό αποτέλεσμα, διότι παρά τις περισσότερες επιτυχημένες αναζητήσεις θα αυξανόταν πολύ το latency της TLB αλλά και το ποσό της καταναλισκόμενης ενέργειας, καθώς κάθε ζητούμενη λειτουργία μνήμης απαιτεί μία αναζήτηση στη TLB. Εν κατακλείδι, εφαρμογές με υψηλές απαιτήσεις μνήμης και κακή τοπικότητα μνήμης (memory locality) αναγκάζουν τη TLB να παρουσιάζει πολύ χαμηλή απόδοση, γεγονός καταστροφικό για τη συνολική απόδοση του συστήματος.

### 1.3 Προσέγγιση και Συνεισφορά Μελέτης

Όπως επισημάνθηκε στο υποκεφάλαιο 1.1, η επιβάρυνση που εισάγει η μετάφραση διευθύνσεων έχει σοβαρή επιρροή στην απόδοση ενός υπολογιστικού συστήματος. Στη μελέτη αυτή, αρχικά εντοπίζουμε το μέγεθος της επιβάρυνσης χρησιμοποιώντας μία ομάδα από μετροπρογράμματα (benchmarks), τα οποία αναγκάζουν τη TLB να έχει κακή απόδοση. Στη συνέχεια αφού επικυρώσουμε τη κακή αυτή απόδοση με χρήση του εργαλείου perf, η χρησιμοποίηση του οποίου αναλύεται στο Κεφάλαιο 4, προτείνουμε κάποιους Μηχανισμούς Προανάκλησης Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης (TLB Prefetching Mechanisms) ώστε να μειωθεί η επιβάρυνση αυτή. Δηλαδή η προσπάθεια βελτίωσης της απόδοσης της TLB στη μελέτη αυτή επικεντρώνεται στην μεταφορά μεταφράσεων στη TLB πριν αυτές ζητηθούν από την εκάστοτε εφαρμογή. Κάποιες από τις Τεχνικές Προανάκλησης Μεταφράσεων είναι ήδη γνωστές από παλαιότερες μελέτες, όπως ο Ακολουθιακός Προανακλητής, ο Προανακλητής Απόστασης και ο Προανακλητής Τυχαίου Βήματος με κάποιες διαφοροποιήσεις από την αρχική τους έκδοση, ενώ άλλες υλοποιήθηκαν για τους σκοπούς της μελέτης αυτής, όπως Προανακλητής Μηδενικού Κόστους, Υβριδικό Μηχανισμοί Προανάκλησης και Tournament Προβλέπτης. Αφού παρουσιαστούν τα αποτελέσματα των διαφόρων Σχημάτων TLB Prefetching θα γίνει εισαγωγή της έννοιας του κόστους σε αυτά. Στους Μηχανισμούς Ελέγχου Προανάκλησης λαμβάνεται υπόψη ο αριθμός των λειτουργιών μνήμης που απαιτούνται για να γίνουν οι προβλέψεις και αποφασίζεται αν η πραγματοποίηση τους συμβάλει στην βελτίωση της απόδοσης του συστήματος ή όχι. Προτείνονται τέσσερις διαφορετικοί Μηχανισμοί Ελέγχου Προανάκλησης, ενσωματώνονται στα Σχήματα TLB Prefetching και παρουσιάζεται η επίδραση τους στα τελικά αποτελέσματα. Για την αξιολόγηση των αποτελεσμάτων και την εξαγωγή ορθών συμπερασμάτων χρησιμοποιούνται οι μετρικές που αναλύονται στο Κεφάλαιο 4. Στο Κεφάλαιο 5, τα πειραματικά αποτελέσματα δείχνουν ότι ο Tournament Προβλέπτης δίνει τα καλύτερα αποτελέσματα απόδοσης, σε κάποια benchmarks φτάνει και το 60%, εισάγοντας μικρές επιβαρύνσεις όσον αφορά τις λειτουργίες μνήμης στο σύστημα. Η ενσωμάτωση του Μηχανισμού MMU Confidence Throttling στον Tournament Προβλέπτη διατηρεί την απόδοση στα ίδια επίπεδα με ταυτόχρονη μείωση των απαιτούμενων λειτουργιών μνήμης. Σε κάποιες περιπτώσεις οι απαιτούμενες λειτουργίες μνήμης είναι λιγότερες και από εκείνες που θα εισάγονταν στο σύστημα αν δεν υπήρχε κανένα Σχήμα Προανάκλησης Μεταφράσεων. Τέλος, κατά μέσο όρο καταφέρνει να μειώσει το συνολικό ποσοστό κύκλων που ξοδεύει καθένα από τα benchmarks σε TLB αστοχίες κατά 23%.

### 1.4 Οργάνωση Μελέτης

Το Κεφάλαιο 2 προσφέρει επιπρόσθετες πληροφορίες όσο αφορά την εικονική μνήμη, την αρχιτεκτονική υποστήριξη που παρέχεται για την ορθή λειτουργία της και τη μετάφραση εικονικών διευθύνσεων σε φυσικές διευθύνσεις.

Το Κεφάλαιο 3 παρουσιάζει όλα τα Σχήματα Προανάκλησης Μεταφράσεων που υλοποιήθηκαν στα πλαίσια της μελέτης αυτής, δίνοντας λεπτομέριες για την υλοποίηση καθενός από αυτά. Επίσης παρουσιάζει τους Μηχανισμούς Ελέγχου Προανάκλησης που αναπτύχθηκαν καθώς και την ιδέα πάνω στην οποία στηρίζεται η υλοποίηση του καθενός.

Το Κεφάλαιο 4 αφορά το πειραματικό περιβάλλον στο οποίο έλαβε χώρα η πειραματική αξιολόγηση. Περιέχει παρουσίαση των χαρακτηριστικών του υπολογιστικού μηχανήματος που χρησιμοποιήθηκε για τη μελέτη αυτή, των μετροπρογραμμάτων που χρησιμοποιήθηκαν, των μετρικών που επιλέχθηκαν για την αξιολόγηση των αποτελεσμάτων καθώς και του BadgerTrap το οποίο είναι το εργαλείο το οποίο χρησιμοποιήθηκε για τον εντοπισμό των αστοχιών της TLB.

Το Κεφάλαιο 5 παρουσιάζει τα πειραματικά αποτελέσματα με τη βοήθεια των μετρικών του υποκεφαλαίου 4.5. Η παρουσίαση συνοδεύεται από εκτενή σχολιασμό των αποτελεσμάτων για την εξαγωγή χρήσιμων συμπερασμάτων.

Το Κεφάλαιο 6 περιέχει βιβλιογραφία σχετική με κάποιους άλλους Μηχανισμούς Προανάκλησης

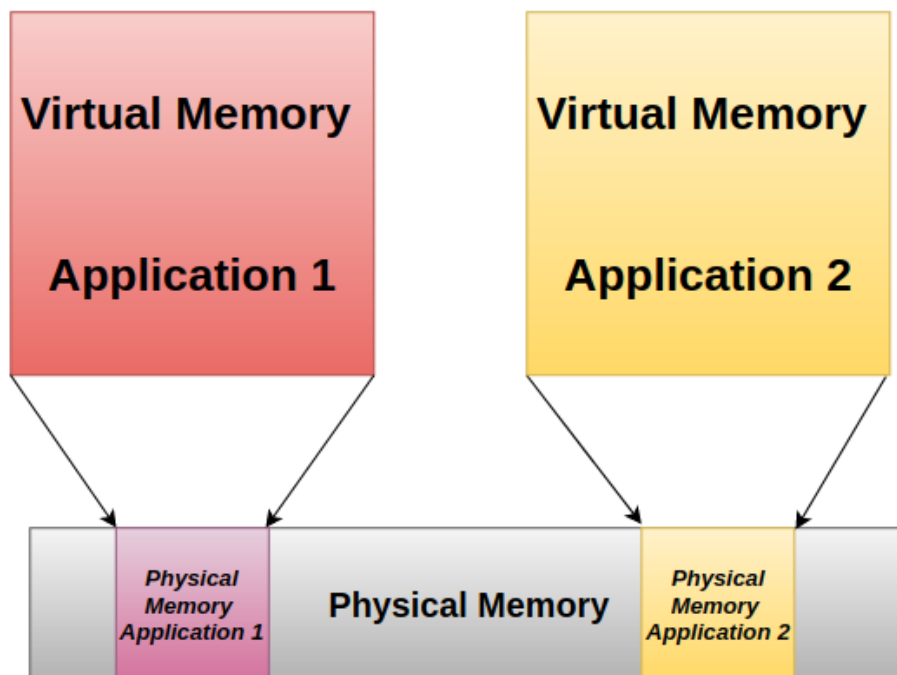
Μεταφράσεων στη TLB που δεν υλοποιήθηκαν στη μελέτη αυτή καθώς και Μηχανισμούς βελτίωσης της απόδοσης της TLB οι οποίοι δεν χρησιμοποιούν τη τεχνική της προανάκλησης. Το Κεφάλαιο 7 παραθέτει κάποια τελικά συμπεράσματα όσο αφορά τη μελέτη αυτή και προτείνει κάποιες μελλοντικές κατευθύνσεις της έρευνας όσο αφορά τη βελτίωση της απόδοσης της TLB.

## Κεφάλαιο 2

### Θεωρητικό Υπόβαθρο

Στο κεφάλαιο αυτό θα ασχοληθούμε με την Εικονική Μνήμη (Virtual Memory) η οποία βασίζεται σε σελιδοποίηση, αλλά και την υποστήριξη που παρέχεται από πλευράς αρχιτεκτονικής για να υλοποιηθεί η μετάφραση των εικονικών διευθύνσεων σε φυσικές διευθύνσεις. Θα πρέπει να τονιστεί ότι τα μηχανήματα τα οποία χρησιμοποιήθηκαν για την εκπόνηση της εργασίας αυτής βασίζονται στην x86-64 αρχιτεκτονική. Για το λόγο αυτό, η παρακάτω εκτενής ανάλυση αφορά τη συγκεκριμένη αρχιτεκτονική και το τρόπο με τον οποίο η μετάφραση διευθύνσεων εκτελείται σ' εκείνη. Για περισσότερες λεπτομέρειες πάνω σε Εικονική Μνήμη και διάφορες αρχιτεκτονικές, ο κάθε ενδιαφερόμενος αναγνώστης μπορεί να απευθυνθεί στα [24, 25].

#### 2.1 Βασικοί Όροι και Έννοιες



*Σχήμα 2.1: Αφαιρετική αναπαράσταση των αντιστοιχίσεων από Εικονική σε Φυσική Μνήμη για δύο εφαρμογές που καταλαμβάνουν τον ίδιο εικονικό χώρο διευθύνσεων αλλά αντιστοιχούν σε διαφορετικές φυσικές περιοχές μνήμης.*



Στην ενότητα αυτή θα δοθούν ορισμοί βασικών εννοιών που θα χρησιμοποιηθούν στη συνέχεια και κρίνεται αναγκαία η κατανόηση τους σε βάθος για να γίνει αντιληπτό το αντικείμενο της μελέτης αυτής.

**Εικονική Μνήμη (Virtual Memory).** Η Εικονική Μνήμη είναι μία τεχνική διαχείρισης μνήμης, η οποία παρέχει την ψευδαίσθηση στο προγραμματιστή πως έχει στη διάθεση του απεριόριστη μνήμη, υπερκερνώντας έτσι το πρόβλημα που υπήρχε για πολλά χρόνια λόγω της πεπερασμένης φυσικής μνήμης. Στη πραγματικότητα το λειτουργικό σύστημα (operating system) απεικονίζει τις διευθύνσεις μνήμης κάθε ενεργής εφαρμογής σε φυσικές διευθύνσεις και αποθηκεύει την απεικόνιση αυτή στο Πίνακα Σελίδων (Page Table), του οποίου η λειτουργία θα αναλυθεί στο υποκεφάλαιο 2.2.1. Μια αφαιρετική απεικόνιση της εικονικής μνήμης, της φυσικής μνήμης καθώς και μιας τυχαίας απεικόνισης αυτών για δύο (2) οποιεσδήποτε εφαρμογές φαίνεται στο Σχήμα 2.1. Τα πλεονεκτήματα που προσφέρει η Εικονική Μνήμη είναι αρχικά η ελευθερία η οποία δίνεται στο προγραμματιστή, αφού πλέον δεν ανησυχεί για ενδεχόμενη έλλειψη μνήμης, αυξάνοντας με το τρόπο αυτό την παραγωγικότητά του. Επίσης οι διάφορες εφαρμογές που εκτελούνται σε ένα υπολογιστικό σύστημα δεν έχουν να διαχειριστούν μοιραζόμενες περιοχές μνήμης, γεγονός το οποίο οδηγεί σε απομόνωση μεταξύ των εφαρμογών, αύξηση της ασφάλειας και καλύτερη χρησιμοποίηση της μνήμης. Η απειλή για υπεράριθμες Διασχίσεις Σελίδας (Page Walks) είναι ένα από τα κύρια μειονεκτήματα της Εικονικής Μνήμης αφού επηρεάζουν αρνητικά τη συνολική απόδοση του υπολογιστικού συστήματος.

**Εικονικός Χώρος Διευθύνσεων (Virtual Address Space).** Ο εικονικός χώρος διευθύνσεων είναι ένα σύνολο από εικονικές διευθύνσεις μνήμης οι οποίες είναι ορατές σε μία μόνο εφαρμογή. Το Λειτουργικό Σύστημα είναι υπεύθυνο για τη δέσμευση και την διάθεση τους στη συγκεκριμένη εφαρμογή.

**Φυσικός Χώρος Διευθύνσεων (Physical Address Space).** Ο φυσικός χώρος διευθύνσεων είναι ένα εύρος από διακριτές θέσεις φυσικής μνήμης, είναι δηλαδή η Κύρια Μνήμη (Main Memory) που διαθέτει ένα υπολογιστικό σύστημα. Το Λειτουργικό Σύστημα είναι υπεύθυνο να την διαχειριστεί κατάλληλα και να κάνει τη καλύτερη δυνατή αντιστοίχιση ανάμεσα στη διαθέσιμη μνήμη και τις απαιτήσεις σε εικονική μνήμη των διαφόρων ενεργών εφαρμογών.

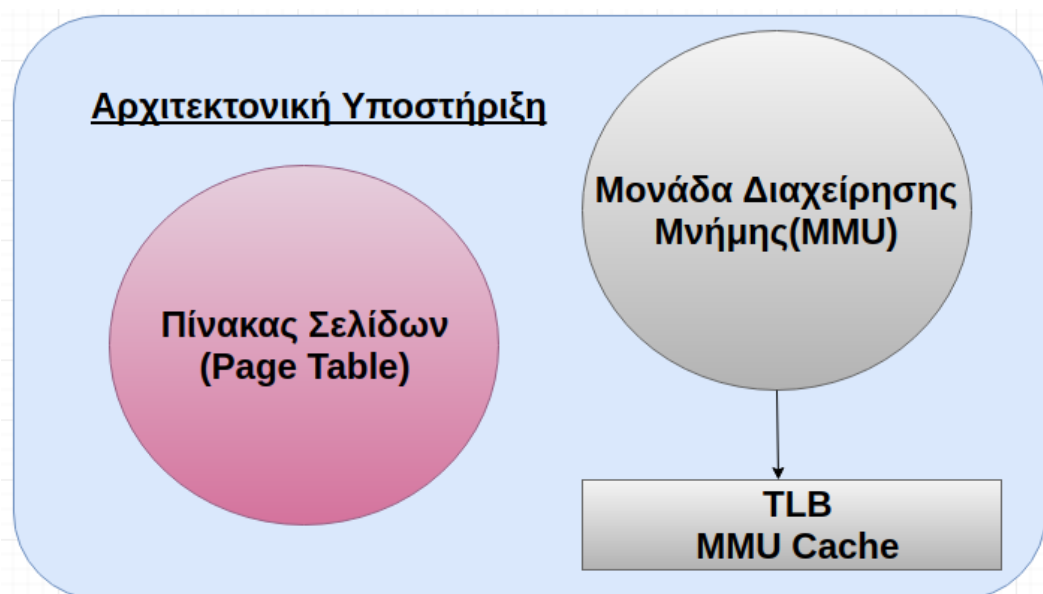
**Σελιδοποίηση (Paging).** Η Σελιδοποίηση είναι ένα πολύ σημαντικό κομμάτι πάνω στο οποίο στηρίζεται η ιδέα της εικονικής μνήμης στα σύγχρονα λειτουργικά συστήματα. Το δομικό στοιχείο πάνω στο οποίο στηρίζεται η ιδέα της Σελιδοποίησης είναι η σελίδα (page). Τόσο η εικονική, όσο και η φυσική μνήμη χωρίζονται σε σελίδες, δηλαδή συνεχή, καθορισμένου μεγέθους κομμάτια μνήμης. Η διαφορά είναι ότι η εικονική μνήμη χωρίζεται σε εικονικές σελίδες (virtual pages) ενώ η φυσική μνήμη σε φυσικές σελίδες (physical pages ή page frames). Το μέγεθος των σελίδων είναι 4KB στη πλειοψηφία των αρχιτεκτονικών που υπάρχουν σήμερα. Μια εικονική σελίδα αναγνωρίζεται ανάμεσα στις υπόλοιπες χρησιμοποιώντας το δικό της μοναδικό αριθμό εικονικής σελίδας (virtual page number). Επομένως, δοσμένης κάποιας εικονικής διεύθυνσης μνήμης μπορούμε να ανακαλύψουμε σε ποια εικονική σελίδα ανήκει απλά αποκόπτοντας τα 12 ελαχίστως σημαντικά δυαδικά της ψηφία (page offset). Το ακριβώς ανάλογο μπορεί να γίνει και με τις φυσικές διευθύνσεις, δηλαδή μία φυσική σελίδα αναγνωρίζεται ανάμεσα στις υπόλοιπες χρησιμοποιώντας τον δικό της μοναδικό αριθμό φυσικής σελίδας (physical frame number).



**Μετάφραση Εικονικής σε Φυσική Διεύθυνση (Address Translation).** Μια εφαρμογή η οποία εκτελείται σε μία υπολογιστική μηχανή στέλνει συνεχόμενα αιτήματα πρόσβασης σε περιοχές μνήμης τις οποίες το Λειτουργικό Σύστημα της έχει επιτρέψει τη πρόσβαση. Δεδομένου ότι η υπολογιστική μηχανή διαθέτει Εικονική Μνήμη, τα διάφορα αιτήματα πρόσβασης στη μνήμη που κάνει η εφαρμογή, αφορούν εικονικές περιοχές μνήμης, αφού μόνο αυτές μπορεί να δει. Επομένως, κάθε τέτοιο αίτημα, απαιτεί μία μετάφραση από τον εικονικό χώρο διεύθυνσεων στο φυσικό χώρο διεύθυνσεων. Η μέθοδος που έχει οριστεί να κάνει τη μετάφραση διεύθυνσεων ουσιαστικά δέχεται την προς μετάφραση εικονική διεύθυνση και παράγει στην έξοδο της μία φυσική διεύθυνση, η οποία είναι η αντιστοίχιση που είχε ορίσει το Λειτουργικό Σύστημα γι' αυτή την εικονική διεύθυνση. Αφού ολοκληρωθεί η μετάφραση, η εφαρμογή πλέον μπορεί να εκτελέσει τη ζητούμενη ενέργεια (διάβασμα, γράψιμο) στη συγκεκριμένη φυσική διεύθυνση. Η διαδικασία αυτή επαναλαμβάνεται όπως αναφέρθηκε παραπάνω για καθένα από τα αιτήματα πρόσβασης στη μνήμη κάθε εκτελούμενης εφαρμογής στο σύστημα.

## 2.2 Αρχιτεκτονική Υποστήριξη

Η υλοποίηση της Εικονικής Μνήμης για να είναι ορθή και συνεπής απαιτεί την συνύπαρξη μηχανισμών τόσο λογισμικού όσο και υλικού. Όπως αναλύθηκε παραπάνω το λειτουργικό σύστημα είναι υπεύθυνο για να κάνει τις αντιστοιχίσεις ανάμεσα σε εικονικές και φυσικές διεύθυνσεις. Όλες αυτές οι αντιστοιχίσεις είναι αποθηκευμένες σε μία δομή λογισμικού που ονομάζεται Πίνακας Σελίδων (Page Table). Απο πλευράς υλικού υπάρχει η Μονάδα Διαχείρισης Μνήμης (MMU), από την οποία περνούν όλες οι αναφορές στη μνήμη. Η χρήση της έχει ως στόχο την επιτάχυνση της μετάφρασης των εικονικών διεύθυνσεων σε φυσικές. Θα πρέπει να τονιστεί ότι η Μονάδα Διαχείρισης Μνήμης είτε υλοποιείται σαν μέρος του επεξεργαστή (cpu), είτε είναι ένα αυτόνομο κύκλωμα. Αποτελείται από δυο ειδικού σκοπού κρυφές μνήμες, τη Κρυφή Μνήμη Αναζήτησης Μετάφρασης (TLB), η οποία κρατάει αποθηκευμένες τις πιο πρόσφατες μεταφράσεις εικονικής σε φυσική μνήμη και την MMU Κρυφή Μνήμη (MMU Cache) η οποία κρατάει κάποια ενδιάμεσα επίπεδα του Πίνακα Σελίδων για να επιταχύνει τη μετάφραση. Στο Σχήμα 2.2 φαίνονται αφαιρετικά οι μηχανισμοί που αναφέρθηκαν παραπάνω, οι οποίοι θα αναλυθούν περισσότερο δίνοντας αρκετές λεπτομέρειες για τον καθένα στη συνέχεια.

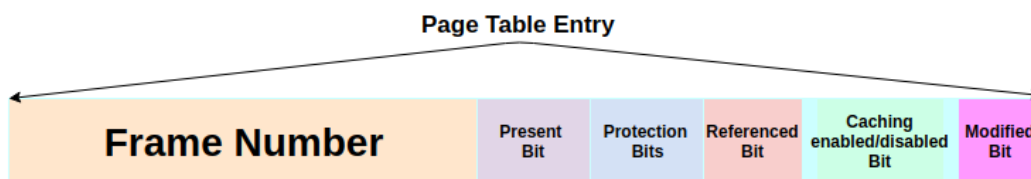


**Σχήμα 2.2:** Μηχανισμοί αρχιτεκτονικής υποστήριξης για τη μετάφραση εικονικών διεύθυνσεων σε φυσικές διεύθυνσεις.

## 2.2.1 Πίνακας Σελίδων (Page Table)

Ο Πίνακας Σελίδων είναι μία δομή δεδομένων λογισμικού, την οποία διαχειρίζεται το λειτουργικό σύστημα και αποτελεί τη διεπαφή ανάμεσα σε εικονικό και φυσικό χώρο διεύθυνσεων. Η αρμοδιότητά του είναι να διατηρεί στη μνήμη όλες τις αντιστοιχίσεις από εικονική σε φυσική μνήμη για κάθε ενεργή εφαρμογή του υπολογιστικού συστήματος.

**Καταχώρηση Πίνακα Σελίδων (Page Table Entry).** Ο Πίνακας Σελίδων αποτελείται από πολλά Page Table Entries, καθένα από τα οποία κρατάει όλες τις απαιτούμενες πληροφορίες για την αντιστοίχιση μίας εικονικής σε μία φυσική διεύθυνση. Ένα Page Table Entry θα πρέπει να περιέχει σίγουρα τον αριθμό φυσικής σελίδας στον οποίο γίνεται η αντιστοίχιση αλλά και επιπλέον πληροφορίες, οι οποίες εξαρτώνται από την αρχιτεκτονική του συστήματος, όπως φαίνεται και στο Σχήμα 2.3. Στη συνέχεια παρουσιάζονται μερικές από τις πληροφορίες που διατηρεί ένα PTE, ανεξάρτητα από την αρχιτεκτονική του συστήματος:

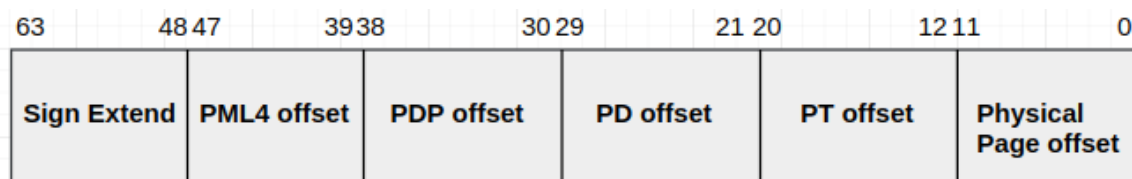


**Σχήμα 2.3:** Οι βασικές πληροφορίες που περιέχει μία Καταχώρηση Πίνακα Σελίδων, ανεξαρτήτως αρχιτεκτονικής.

1. **Αριθμός Φυσικής Σελίδας (Frame Number).** Η πληροφορία αυτή δίνει την αντιστοίχιση από την εικονική στη φυσική μνήμη. Ο αριθμός των δυαδικών ψηφίων που απαιτούνται για τη πληροφορία αυτή εξαρτάται από το μέγεθος ενός πλαισίου φυσικής μνήμης και από το μέγεθος της φυσικής μνήμης.
2. **Δυαδικό Ψηφίο Παρουσίας (Present bit).** Η τιμή αυτού του δυαδικού ψηφίου καθορίζει αν είναι έγκυρο ή όχι το περιεχόμενο του Page Table Entry. Για το λόγο αυτό πολλές φορές ονομάζεται και δυαδικό ψηφίο εγκυρότητας (valid bit). Αν δεν είναι έγκυρο και ζητηθεί τότε πυροδοτείται Σφάλμα Σελίδας (Page Fault), του οποίου η λειτουργία θα αναλυθεί στη συνέχεια.
3. **Δυαδικά Ψηφία Προστασίας (Protection bits).** Δηλώνει ποιές από τις λειτουργίες μνήμης μπορούν να εκτελεστούν στη σελίδα αυτή, όπως αποκλειστικό διάβασμα (read only), διάβασμα-/εγγραφή (read/write).
4. **Δυαδικό Ψηφίο Αναφοράς (Referenced bit).** Ενημερώνει σχετικά με το αν η σελίδα αυτή είχε αναφερθεί στο τελευταίο κύκλο ρολογιού.
5. **Δυαδικό Ψηφίο Ενημέρωσης Ιεραρχίας Μνήμης (Caching enabled/disabled bit).** Το δυαδικό αυτό ψηφίο ενεργοποιεί/απενεργοποιεί την ενημέρωση όλης της ιεραρχίας μνήμης για τη συγκεκριμένη σελίδα. Συνήθως το δυαδικό αυτό ψηφίο είναι μηδέν, δηλαδή απενεργοποιείται η διαδικασία της ενημέρωσης της ιεραρχίας της μνήμης για λόγους συνεκτικότητας των πληροφοριών.
6. **Δυαδικό Ψηφίο Τροποποίησης (Modified bit).** Δηλώνει αν μία σελίδα έχει τροποποιηθεί ή όχι, δηλαδή αν έχουν γραφτεί νέα δεδομένα σε αυτή. Αν κάτι τέτοιο έχει γίνει, τότε θα πρέπει να γραφτούν τα δεδομένα πίσω στη κύρια μνήμη πριν αυτή ανακτηθεί από την εφαρμογή.

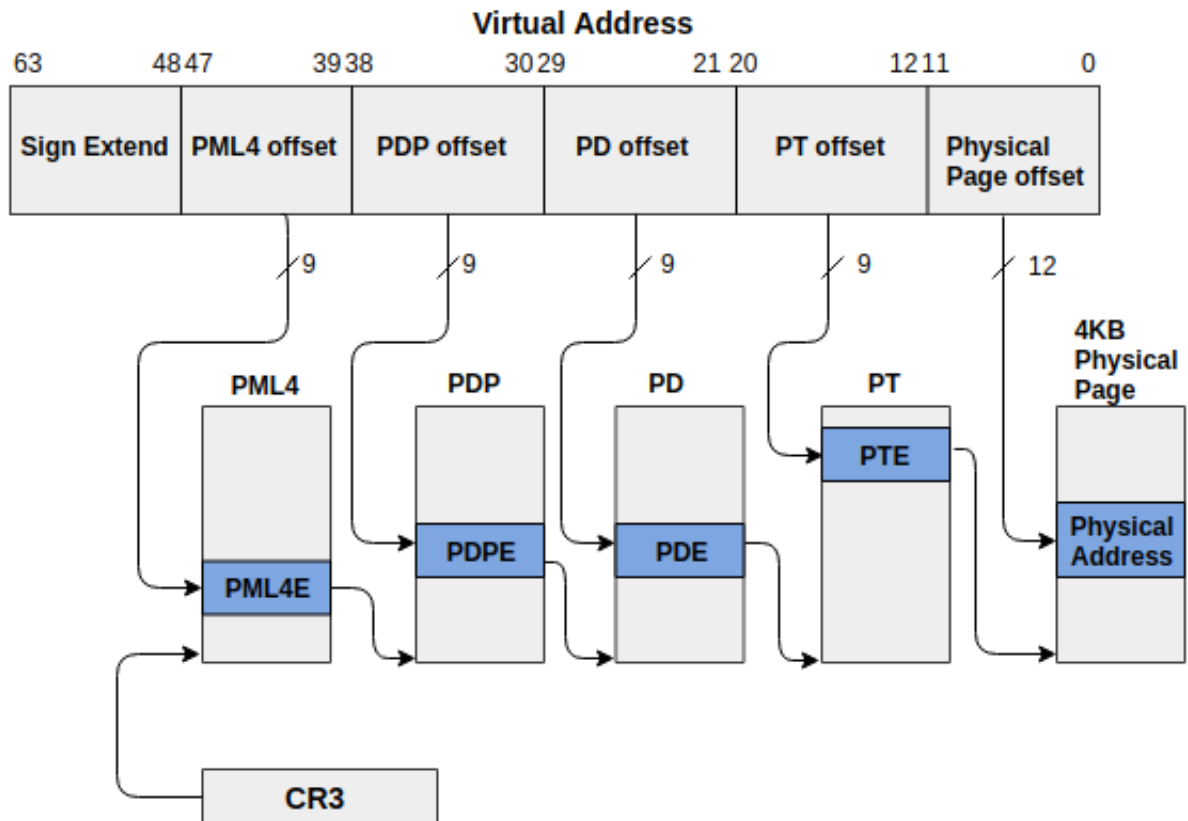
**Οργάνωση του Πίνακα Σελίδων.** Κάποιες προτάσεις για τον τρόπο οργάνωσης του Πίνακα Σελίδων που έχουν γίνει κατά καιρούς φαίνονται στη συνέχεια:

1. **Επίπεδος Πίνακας.** Η πλέον εύκολη οργάνωση του Πίνακα Σελίδων είναι σαν ένας πίνακας πολύ μεγάλου μεγέθους, ο οποίος έχει αποθηκευμένες όλες τις αντιστοιχίσεις από εικονικές σε φυσικές διευθύνσεις για κάθε ενεργή εφαρμογή του υπολογιστικού συστήματος. Η προσέγγιση αυτή παρότι έχει ιδιαίτερα απλή υλοποίηση, είναι μη αποτελεσματική καθώς ο πολύ μεγάλου μεγέθους πίνακας θα πρέπει να είναι αποθηκευμένος στη κύρια μνήμη και ως εκ τούτου θα καταλαμβάνει ένα πολύ μεγάλο μέρος της.
2. **Ανεστραμμένος Πίνακας Σελίδων (Inverted Page Table).** Η οργάνωση αυτή συνδυάζει ένα Πίνακα Σελίδων και ένα Πίνακα Πλαισίων σε μία κοινή δομή δεδομένων. Το κύριο πρόβλημα που αντιμετωπίζει η οργάνωση αυτή είναι η κακή τοπικότητα όσο αφορά της κρυφές μνήμες (cache locality). Επειδή τέτοιο τρόπο οργάνωσης χρησιμοποιούν αρχιτεκτονικές όπως PowerPC, the UltraSPARC και IA-64 αλλά όχι η x86-64 αρχιτεκτονική η οποία χρησιμοποιείται στη μελέτη αυτή, κάθε ενδιαφερόμενος αναγνώστης μπορεί να απευθυνθεί στα εξής για περαιτέρω εμβάθυνση [24, 25].
3. **Ιεραρχικός Τρόπος Οργάνωσης (Hierarchical Page Table Organization).** Ο Ιεραρχικός Τρόπος Οργάνωσης στηρίζεται στην ιδέα του διαχωρισμού του Πίνακα Σελίδων σε διάφορα επίπεδα (levels). Πλεον ο Πίνακας Σελίδων οργανώνεται σαν δενδρική δομή και κάθε επίπεδο είναι υπεύθυνο για άλλη περιοχή μνήμης. Όσο πλησιάζουμε τη ρίζα του δένδρου αυξάνεται το μέγεθος της περιοχής μνήμης που καλύπτει το κάθε επίπεδο.



**Σχήμα 2.4:** Χωρισμός εικονικής διεύθυνσης σε ομάδες από δυαδικά ψηφία για τη δεικτοδότηση των διαφόρων επιπέδων του Πίνακα Σελίδων.

Εφόσον η μελέτη αυτή στηρίζεται στην αρχιτεκτονική x86-64, θα επικεντρωθούμε στο τρόπο με τον οποίο είναι υλοποιημένος ο Πίνακας Σελίδων σ' αυτή. Συγκεκριμένα ο τρόπος οργάνωσης στην x86-64 είναι Ιεραρχικός με τέσσερα (4) επίπεδα των οποίων τα συμβολικά ονόματα είναι PML4, PDP, PD και PT. Στη πράξη, η εικονική διεύθυνση χωρίζεται σε ομάδες από δυαδικά ψηφία όπως δείχνει αναλυτικά και το Σχήμα 2.4. Αυτά τα δυαδικά ψηφία λειτουργούν σαν δείκτες στα επίπεδα του Πίνακα Σελίδων και κάθε καταχώρηση ενός επιπέδου είναι δείκτης για το ακριβώς επόμενο επίπεδο, όπως φαίνεται πλήρως στο Σχήμα 2.5, το οποίο δείχνει αναλυτικά τον Ιεραρχικό Τρόπο Οργάνωσης του Πίνακα Σελίδων στην x86-64 αρχιτεκτονική.

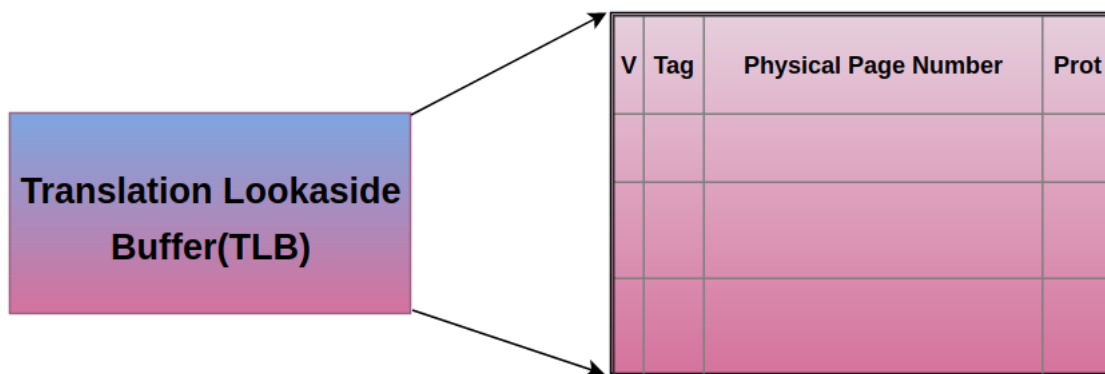


Σχήμα 2.5: Σχεδιασμός του Πίνακα Σελίδων σε αρχιτεκτονική x86-64. Πρόκειται για Ιεραρχικό Τρόπο Οργάνωσης με τέσσερα (4) επίπεδα.

**Διάσχιση Σελίδας (Page Walk).** Το Page Walk είναι η διαδικασία η οποία εκτελείται όταν θέλουμε να βρούμε την αντιστοίχιση μίας εικονικής διεύθυνσης στο φυσικό χώρο διευθύνσεων. Υπεύθυνος για τη διαδικασία αυτή είναι ο Page Table Walker, μία μηχανή πεπερασμένης κατάστασης (finite state machine) υλοποιημένης στο υλικό. Αρχικά γίνεται η πρόσβαση στο Πίνακα Σελίδων και στη συνέχεια η αναζήτηση της καταχώρησης του Πίνακα Σελίδων που αντιστοιχεί στη συγκεκριμένη εικονική διεύθυνση. Αν τελικά βρεθεί η ζητούμενη καταχώρηση, τότε θα περιέχει τη ζητούμενη μετάφραση. Σε αντίθετη περίπτωση, προκαλείται μία διακοπή λογισμικού (software interrupt) με όνομα Σφάλμα Σελίδας (Page Fault), η οποία θα αναλυθεί στη συνέχεια. Στην αρχιτεκτονική x86-64 η διαδικασία του Page Walk πραγματοποιείται με τη βοήθεια ενός ειδικού καταχωρητή με όνομα CR3, ο οποίος είναι ορατός από τη αρχιτεκτονική του συστήματος και στην ουσία κρατάει ένα δείκτη στη ρίζα του επιπέδου του Πίνακα Σελίδων με όνομα PML4. Αρχικά το περιεχόμενο του CR3 χρησιμοποιείται μαζί με τα δυαδικά ψηφία 47-39 της εικονικής διεύθυνσης για να βρεθεί η αντίστοιχη καταχώρηση του PML4, η οποία με τη σειρά της είναι δείκτης προς τη ρίζα του επόμενου επιπέδου του Πίνακα Σελίδων, του PDP. Στη συνέχεια τα δυαδικά ψηφία 38-30 της εικονικής διεύθυνσης μαζί με το δείκτη προς τη ρίζα του PDP χρησιμοποιούνται για να βρεθεί η αντίστοιχη καταχώρηση του PDP, η οποία με τη σειρά της είναι δείκτης προς τη ρίζα του επόμενου επιπέδου του Πίνακα Σελίδων, του PD. Έπειτα, τα δυαδικά ψηφία 29-21 της εικονικής διεύθυνσης μαζί με το δείκτη προς τη ρίζα του PD χρησιμοποιούνται για να βρεθεί η αντίστοιχη καταχώρηση του PD, η οποία με τη σειρά της είναι δείκτης προς τη ρίζα του Πίνακα Σελίδων, του PT. Τα δυαδικά ψηφία 20-12 της εικονικής διεύθυνσης μαζί με το δείκτη προς τη ρίζα του PT χρησιμοποιούνται για να βρεθεί η αντίστοιχη καταχώρηση του PT, η οποία μαζί με τα δυαδικά ψηφία 11-0 οδηγούν τελικά στη ζητούμενη αντιστοίχιση στο φυσικό χώρο διευθύνσεων.

**Σφάλμα Σελίδας (Page Fault).** Το Σφάλμα Σελίδας είναι όπως αναφέρθηκε και παραπάνω μία διακοπή λογισμικού (software interrupt) και συγκεκριμένα μία εξαίρεση (exception) η οποία εγείρεται είτε όταν μία εφαρμογή προσπαθήσει να αποκτήσει πρόσβαση σε κάποια σελίδα για την οποία δεν υπάρχει κάποια αντιστοίχιση στο Πίνακα Σελίδων, είτε όταν δεν υπάρχει έγκυρη αντιστοίχιση στο Πίνακα Σελίδων (NULL pointer), είτε όταν παραβιάσει τα δικαιώματα πρόσβασης που της έχουν δοθεί. Όταν γίνει η διακοπή, το λειτουργικό σύστημα εκτελεί τη ρουτίνα εξυπηρέτησης (handler) της εν λόγω εξαίρεσης για να αποφανθεί τελικά για το τρόπο δράσης του. Στις πρώτες δύο περιπτώσεις δίνεται λύση από το λειτουργικό σύστημα ενώ στη τρίτη περίπτωση η εφαρμογή τερματίζεται με κατάλληλο μήνυμα σφάλματος.

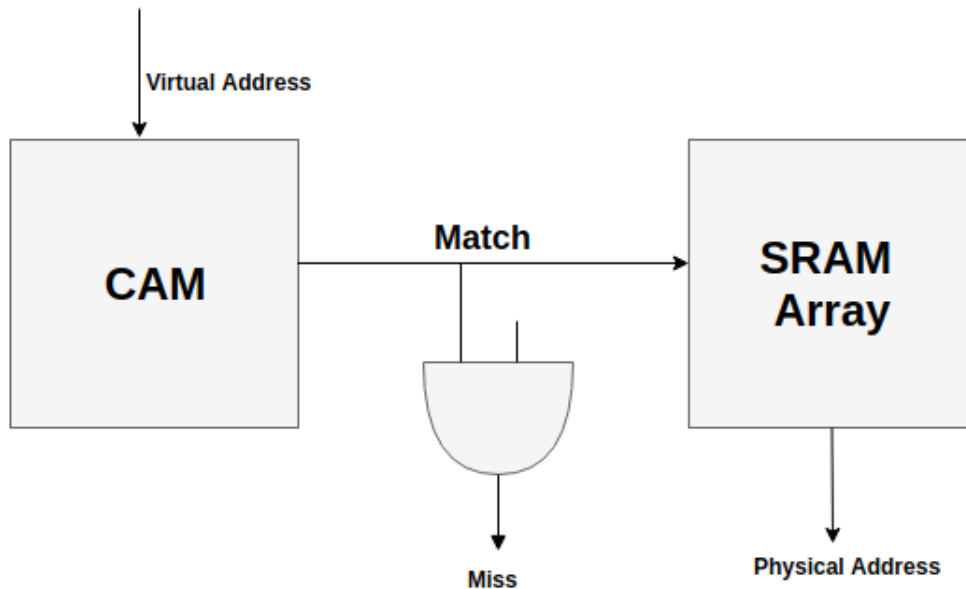
### 2.2.2 Κρυφή Μνήμη Αναζήτησης Μετάφρασης (TLB)



Σχήμα 2.6: Τα πεδία της Κρυφής Μνήμης Αναζήτησης Μετάφρασης (TLB).

Ενα σημαντικό πρόβλημα της Εικονικής Μνήμης είναι ότι για κάθε λειτουργία μνήμης (memory reference) που αιτεί μία εφαρμογή, θα πρέπει να γίνει αρχικά μετάφραση της εικονικής διεύθυνσης σε φυσική διεύθυνση, ενέργεια που απαιτεί 4 επιπρόσθετες λειτουργίες μνήμης για να βρεθεί η αντιστοίχιση από το Πίνακα Σελίδων και μετά να πραγματοποιηθεί η πραγματική λειτουργία μνήμης. Κάτι τέτοιο επιβαρύνει το σύστημα με πολλές λειτουργίες μνήμης, γεγονός το οποίο είναι εξαιρετικά ζημιογόνο για την απόδοση του. Για να μειωθεί αυτή η αρνητική επίδραση της Εικονικής Μνήμης, επινοήθηκε η Κρυφή Μνήμη Αναζήτησης Μετάφρασης (Translation Lookaside Buffer ή TLB), η οποία είναι μία μικρού μεγέθους κρυφή μνήμη (memory cache), η οποία αποθηκεύει τις πιο πρόσφατες μεταφράσεις εικονικής μνήμης σε φυσική μνήμη, όπως φαίνεται και στο Σχήμα 2.6. Για το λόγο αυτό ονομάζεται και address-translation cache στη διεθνή βιβλιογραφία και είναι μέρος της Μονάδας Διαχείρισης Μνήμης (MMU). Σε επίπεδο υλοποίησης, η Κρυφή Μνήμη Αναζήτησης Μετάφρασης μπορεί να υλοποιηθεί με Content-Addressable Memory (CAM) με κλειδί αναζήτησης του CAM την εικονική διεύθυνση και το αποτέλεσμα είναι η φυσική διεύθυνση. Το Σχήμα 2.7 δείχνει την υλοποίηση της Κρυφής Μνήμης Αναζήτησης Μετάφρασης με CAM. Η Κρυφή Μνήμη Αναζήτησης Μετάφρασης μπορεί να κατοικεί είτε ανάμεσα στον επεξεραστή και το σύστημα κρυφών μνημών του επεξεραστή, είτε ανάμεσα στο σύστημα κρυφών μνημών του επεξεραστή και τη φυσική μνήμη, είτε τέλος ανάμεσα στα διάφορα επίπεδα των κρυφών μνημών. Στη πράξη, σε κάθε λειτουργία μνήμης η επεργαστική μονάδα (cpu) αρχικά αναζητά την εικονική διεύθυνση στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης. Αν τη βρεί, επιστρέφει αμέσως στην επεξεργαστική μονάδα την ζητούμενη μετάφραση χωρίς να προκαλέσει καμία επιπλέον λειτουργία μνήμης, δηλαδή δεν ψάχνει καθόλου στο Πίνακα Σελίδων. Αν δεν τη βρεί, πυροδοτείται ένα Page Walk, όπως αναλύθηκε παραπάνω και στο τέλος μεταφέρεται ένα αντίγραφο της καταχώρησης του Πίνακα Σελίδων στην οποία αντιστοιχούσε η εικονική διεύθυνση στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης, για ενδεχόμενη μελλοντική χρήση. Η παραπάνω διαδικασία φαίνεται αναλυτικά στο Σχήμα 2.8. Θα πρέπει να επισημανθεί ότι το Page Walk μπορεί να εκτελεστεί από το υλικό (hardware-managed TLB) είτε

από το λογισμικό (software-managed TLB). Η αρχιτεκτονική x86-64 χρησιμοποιεί ελεγχόμενη από το υλικό Κρυφή Μνήμη Αναζήτησης Μετάφρασης (hardware-managed TLB), όποτε κάθε φορά που θα αναφερόμαστε σε Page Walk θα εννοείται ότι η διαδικασία που αναλύθηκε παραπάνω εκτελείται από το υλικό. Τέλος, κατά τη διάρκεια του Page Walk εντολές που δεν έχουν κάποια εξάρτηση με την εντολή αυτή εισέρχονται στα διάφορα στάδια της σωλήνωσης (pipeline stages).



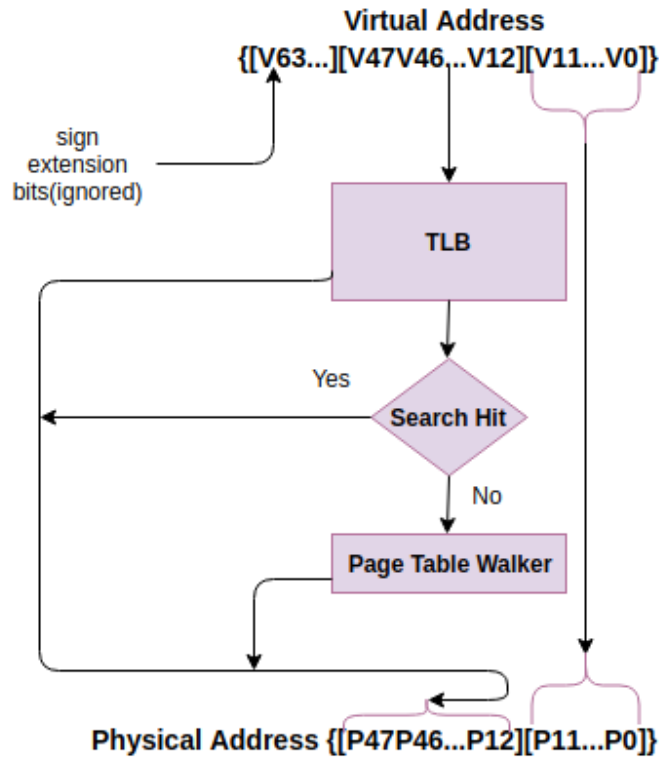
*Σχήμα 2.7: Υλοποίηση της Κρυφής Μνήμης Αναζήτησης Μετάφρασης (TLB) με CAM.*

Πολύ σημαντικό ζήτημα είναι το μέγεθος της μνήμης το οποίο μπορεί να εξυπηρετεί η TLB με μεταφράσεις εικονικών σε φυσικές διευθύνσεις, χωρίς να αποτυγχάνει ποτέ. Το ζήτημα αυτό είναι διεθνώς γνωστό ως TLB Reach [31, 36, 37]. Ωστόσο οι εφαρμογές με την πάροδο του χρόνου έχουν όλο και μεγαλύτερες ανάγκες για μνήμη, το μέγεθος της οποίας είναι σχεδόν πάντα μεγαλύτερο από το μέγεθος της μνήμης το οποίο μπορεί να εξυπηρετεί η TLB χωρίς καμία αποτυχία. Το γεγονός αυτό οδηγεί σε ανάγκη για αντικατάσταση μιας καταχώρησης της TLB με κάποια άλλη πολλαπλές φορές, το οποίο οδηγεί τελικά σε μείωση της απόδοσης της TLB. Μια προφανής λύση είναι η αύξηση του μεγέθους της TLB. Κάτι τέτοιο δεν είναι δυνατό να γίνει, διότι η TLB πρέπει να είναι μικρή σε μέγεθος για να έχει μικρό χρόνο πρόσβασης (access time). Δηλαδή υπάρχει ένα άνω φράγμα όσο αφορά το πόσο μεγάλο μπορεί να γίνει το μέγεθος της TLB.

Μια ακόμα ιδέα που έχει προταθεί και στα περισσότερα σύγχρονα υπολογιστικά συστήματα έχει εφαρμοστεί είναι η αντικατάσταση της απλής TLB από μία Ιεραρχική Οργάνωση δύο ή περισσότερων επιπέδων από TLBs. Αν υποθέσουμε ότι η οργάνωση αυτή αποτελείται από  $N$  το πλήθος επίπεδα, τότε το επίπεδο  $i$  θα έχει TLB μικρότερου μεγέθους από το επίπεδο  $i+1$  για να επιτευχθεί όσο το δυνατόν γρηγορότερη αναζήτηση των μεταφράσεων. Επίσης όταν οι απαιτήσεις για απόδοση είναι μεγάλες, υλοποιούνται 2 ξεχωριστές TLB, η μία για εντολές και η άλλη για δεδομένα, όπως ακριβώς στο μοντέλο αρχιτεκτονικής Harvard.

Η TLB επινοήθηκε για να επιταχύνει τη μετάφραση εικονικών διευθύνσεων σε φυσικές διευθύνσεις, δηλαδή να αυξήσει την απόδοση της Εικονικής Μνήμης. Βρίσκεται στο κρίσιμο τμήμα κάθε λειτουργίας μνήμης και έχει αποδειχθεί ότι η απόδοση της επηρεάζει σε πολύ μεγάλο βαθμό τη συνολική απόδοση του υπολογιστικού συστήματος [16, 17, 26, 32, 33] καθώς και τη κατανάλωση ενέργειας [21, 27, 28, 29]. Για το λόγο αυτό έχουν γίνει πολλές έρευνες και μελέτες για να μπορέσει να αυξηθεί η απόδοση της, το οποίο είναι και το αντικείμενο μελέτης της εργασίας αυτής.



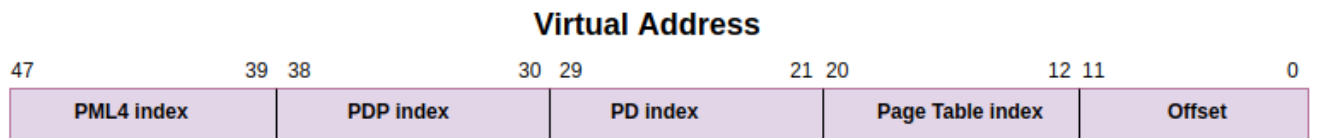


Σχήμα 2.8: Η διαδικασία που λαμβάνει χώρα όταν μία εφαρμογή αιτεί μία λειτουργία μνήμης.

### 2.2.3 MMU Κρυφή Μνήμη (MMU Cache)

Η παραπάνω ανάλυση έδειξε ότι για εφαρμογές με μεγάλες απαιτήσεις για μνήμη, δηλαδή για εφαρμογές που φανερώνουν το πρόβλημα του TLB Reach, θα συμβούν αρκετές αστοχίες της TLB, οι οποίες θα οδηγήσουν σε page walks, τα οποία θα προκαλέσουν 4 επιπλέον λειτουργίες μνήμης (μία για κάθε επίπεδο του Πίνακα Σελίδων). Για να μειωθούν οι απαιτούμενες λειτουργίες μνήμης για ένα page walk, υλοποιήθηκαν οι MMU Caches [14, 17]. Πρόκειται για μικρού μεγέθους κρυφές μνήμες, οι οποίες αποθηκεύουν καταχωρήσεις από τα ενδιαμέσα επίπεδα του Πίνακα Σελίδων.

Στην αρχιτεκτονική x86-64, όπως φαίνεται και στο Σχήμα 2.9, υπάρχουν 3 MMU Caches με διαφορετικά χαρακτηριστικά και πολιτικές αντικατάστασης, μία για καθένα από τα επίπεδα PML4, PDP και PD του Πίνακα Σελίδων. Η MMU Cache του επιπέδου PD διευθυνσιοδοτείται με τα δυαδικά ψηφία 47-21 της εικονικής διεύθυνσης που προκάλεσε Page Walk, η MMU Cache του επιπέδου PDP διευθυνσιοδοτείται με τα δυαδικά ψηφία 47-30 της εικονικής διεύθυνσης και η MMU Cache του επιπέδου PML4 διευθυνσιοδοτείται με τα δυαδικά ψηφία 47-39 της εικονικής διεύθυνσης. Ο Page Table Walker αρχικά ψάχνει παράλληλα στις MMU Caches. Όταν ολοκληρωθεί η αναζήτηση, μπορεί να έχει πετύχει το πολύ 3 επιτυχείς αναζητήσεις (hits), μία σε καθεμία από τις MMU Caches. Αν υπάρχει έστω και μία επιτυχής αναζήτηση σε κάποια από τις MMU Caches, τότε θα μειωθούν οι απαιτούμενες λειτουργίες μνήμης (κατά μία έως και τρεις) για το συγκεκριμένο page walk. Μια επιτυχής αναζήτηση στην MMU Cache του επιπέδου PD γλιτώνει 3 λειτουργίες μνήμης, μία επιτυχής αναζήτηση στην MMU Cache του επιπέδου PDP γλιτώνει 2 λειτουργίες μνήμης και μία επιτυχής αναζήτηση στην MMU Cache του επιπέδου PML4 γλιτώνει 1 λειτουργία μνήμης. Επομένως, στο τέλος της αναζήτησης ο Page Walker επιλέγει τη πετυχημένη αναζήτηση (hit) της MMU Cache που χρησιμοποιεί τα περισσότερα δυαδικά ψηφία δεικτοδότησης, ώστε να μειωθούν περισσότερο οι απαιτούμενες λειτουργίες μνήμης.



**PML4 cache**

PML4 index(bits 47-39)	Physical Address	Flags

**PDP cache**

PML4 index  PDP index  (bits 47-30)	Physical Address	Flags

**PD cache**

PML4 index  PDP index  PD index  (bits 47-21)	Physical Address	Flags

*Σχήμα 2.9: Η δομή της Memory Management Unit (MMU) cache.*



## Κεφάλαιο 3

# Σχεδίαση και Υλοποίηση

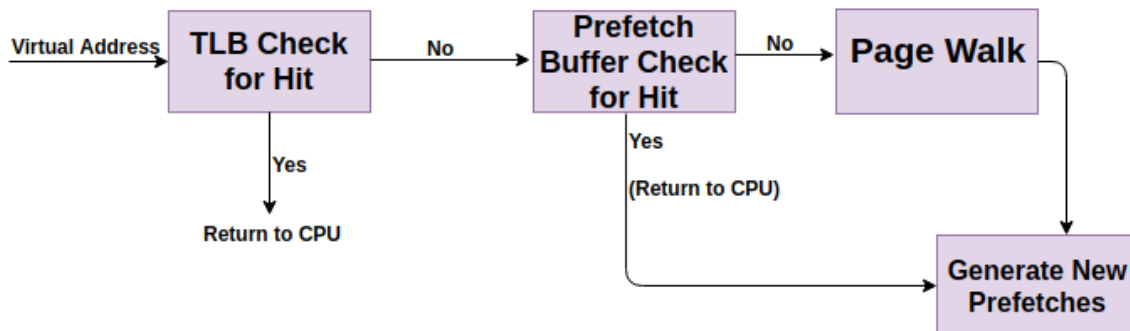
### 3.1 Εισαγωγή

Διάφορες μελέτες έχουν δείξει ότι οι σύγχρονες και μοντέρνες εφαρμογές παρουσιάζουν μεγάλες αποκλίσεις όσο αφορά τον αναμενόμενο χρόνο εκτέλεσης. Σε πολλές περιπτώσεις το ποσοστό της αύξησης φτάνει και το 50%, λόγω των διασχίσεων σελίδων (page walks) που απαιτούν επιπρόσθετες λειτουργίες μνήμης (memory references), όπως αναλύθηκε στο κεφάλαιο 2. Η μετάφραση εικονικών σελίδων σε φυσικές με χρήση της Κρυφής Μνήμης Αναζήτησης Μετάφρασης (TLB) βρίσκεται στο κρίσιμο τμήμα του επεξεργαστή και απαιτεί μικρό χρόνο πρόσβασης. Αυτό σημαίνει ότι το μέγεθος της TLB θα πρέπει να παραμένει μικρό, σε αντίθεση με το μέγεθος της μνήμης που απαιτούν οι διάφορες εφαρμογές, το οποίο αυξάνεται συνεχώς και με υψηλό ρυθμό. Το γεγονός αυτό οδηγεί σε πολλές αστοχίες της TLB. Είναι λοιπόν ξεκάθαρο ότι η κακή απόδοση που παρουσιάζει η TLB επηρεάζει αρνητικά σε μεγάλο βαθμό τη συνολική απόδοση του υπολογιστικού συστήματος.

Η ανάγκη για βελτίωση της απόδοσης της TLB έχει επισημανθεί από πολλές μελέτες [13, 19, 23, 26, 34, 38]. Πολλές είναι οι μελέτες που προσπάθησαν να κάνουν πράξη αυτή την απαίτηση. Συγκεκριμένα, προτάθηκαν και υλοποιήθηκαν ιδέες όπως Ιεραρχική Οργάνωση της TLB, το Superpaging [40], το Hugeraging [4, 12] και τα Multipage Mappings τα οποία χρησιμοποιούν μία καταχώρηση του TLB για να κρατούν πολλαπλές σελίδες (8-16 σελίδες για κάθε καταχώρηση) [37]. Στο πρόσφατο παρελθόν, η επιστημονική κοινότητα άρχισε να εξετάζει σε πιο πολύ βάθος τη τεχνική της Προανάκλησης Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης (TLB Prefetching). Στην ενότητα αυτή, θα αναλύσουμε, θα εξετάσουμε και θα υλοποιήσουμε κάποια ήδη γνωστά σχήματα TLB Prefetching με κάποιες βελτιστοποιήσεις και θα προτείνουμε κάποια νέα ώστε να καταφέρουμε καλύτερη απόδοση, δηλαδή η TLB να αποκτήσει καλύτερη απόδοση κάνοντας προβλέψεις. Σε δεύτερο στάδιο, αφού έχουμε ήδη υλοποιήσει τα διάφορα Σχήματα Προανάκλησης Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης, θα προτείνουμε δύο διαφορετικές τεχνικές οι οποίες λαμβάνουν υπόψη τον κάποιο κόστος, διαφορετικό η καθεμία, για να αποφασίσουν αν θα κάνουν κάποια πρόβλεψη ή όχι. Οι δύο αυτές τεχνικές εφαρμόζονται για καθένα από τα υλοποιημένα TLB Prefetching Σχήματα.

### 3.2 Εισαγωγή της Προανάκλησης Μεταφράσεων στη TLB

Στην ενότητα αυτή θα δοθούν οι βασικές ιδέες πάνω στις οποίες στηρίζεται η τεχνική της Προανάκλησης Μεταφράσεων στη TLB. Υποθέτοντας πως κάποια εφαρμογή του συστήματος ζητάει τη μετάφραση κάποιας εικονικής διεύθυνσης, γίνεται αναζήτηση της στη TLB. Η επιτυχής αναζήτηση μεταφέρει τη μετάφραση στην επεξεργαστική μονάδα και η εκτέλεση της εφαρμογής συνεχίζεται κανονικά. Σε περίπτωση αστοχίας της TLB, δεν πυροδοτείται κατευθείαν ένα page walk, αλλά πρώτα γίνεται αναζήτηση της προς μετάφραση διεύθυνσης σε μία νέα δομή δεδομένων, το Πίνακα Προβλέψεων (Prefetch Buffer). Ο Πίνακας Προβλέψεων είναι μία δομή δεδομένων, η οποία γεμίζει με τις προβλέψεις του εκάστοτε Σχήματος Προανάκλησης Μεταφράσεων. Ο λόγος χρησιμοποίησης της δομής αυτής, είναι ότι δεν πρέπει να γεμίζει η TLB με προβλέψεις οι οποίες δεν είναι γνωστό εκ των προτέρων αν θα φανούν χρήσιμες. Το μέγεθός του διατηρείται αρκετά μικρό, π.χ. 16 καταχωρήσεων, ώστε να μη συγκρίνεται με το μέγεθος της TLB (περίπου 1500 καταχωρήσεις του L2 TLB) και να έχει μικρό χρόνο πρόσβασης (access time). Περισσότερες πληροφορίες για το μέγεθος του θα δοθούν στη συνέχεια. Επομένως αν η αναζήτηση στον Prefetch Buffer είναι επιτυχής μεταφέρεται η μετάφραση στην επεξεργαστική μονάδα και η εκτέλεση της εφαρμογής συνεχίζεται κανονικά. Σε αντίθετη περίπτωση ενεργοποιείται ο Page Table Walker και ακολουθείται η διαδικασία που αναλύθηκε στο κεφάλαιο 2. Τέλος, γίνονται οι νέες προβλέψεις και μεταφέρονται στη κατάλληλη θέση του Prefetch Buffer, αν δεν υπάρχουν ήδη σ'αυτόν. Η διαδικασία αυτή φαίνεται καλύτερα στο Σχήμα 3.1. Για τις Τεχνικές Προανάκλησης Μεταφράσεων στη TLB που θα αναλυθούν στη συνέχεια, η διαδικασία αυτή συμβαίνει σε κάθε TLB αστοχία.

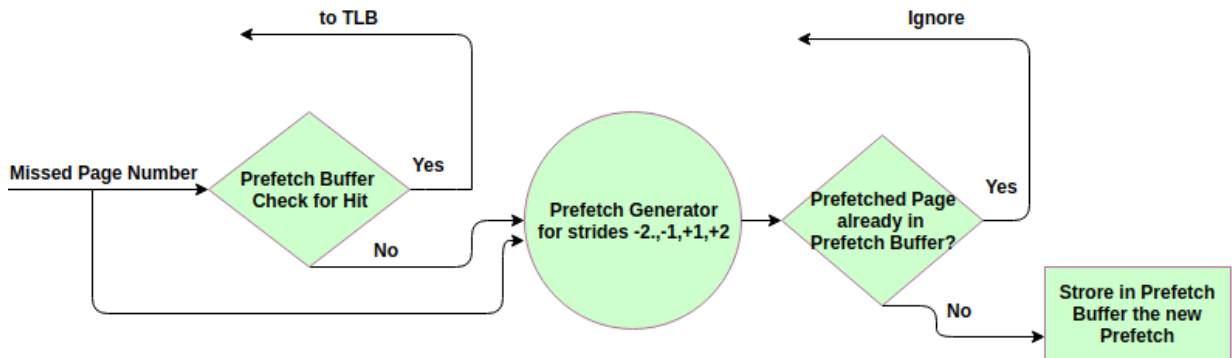


*Σχήμα 3.1: Η διαδικασία που ακολουθείται σε κάθε TLB αστοχία όταν χρησιμοποιείται κάποιο Σχήμα Προανάκλησης Μεταφράσεων. Η διαδικασία παραγωγής νέων προβλέψεων γίνεται στο background.*

### 3.3 Τεχνικές Προανάκλησης Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης

Στην ενότητα αυτή θα παρουσιαστούν κάποια γνωστά TLB Prefetching Σχήματα με ορισμένες αλλαγές ώστε να αποδίδουν καλύτερα και ένα νέο TLB Prefetching Σχήμα, ο Προανακλητής Μηδενικού Κόστους, του οποίου η ιδέα υλοποίησης στηρίζεται σε κάποια χαρακτηριστικά του συστήματος κρυφών μνημών. Θα γίνει εκτενής ανάλυση του τρόπου λειτουργίας τους, χωρίς ωστόσο να δοθούν συγκεκριμένες τιμές για τα μεγέθη των δομών που χρησιμοποιεί το καθένα από αυτά, αφού οι τιμές αυτές θα παρουσιαστούν στο επόμενο κεφάλαιο.

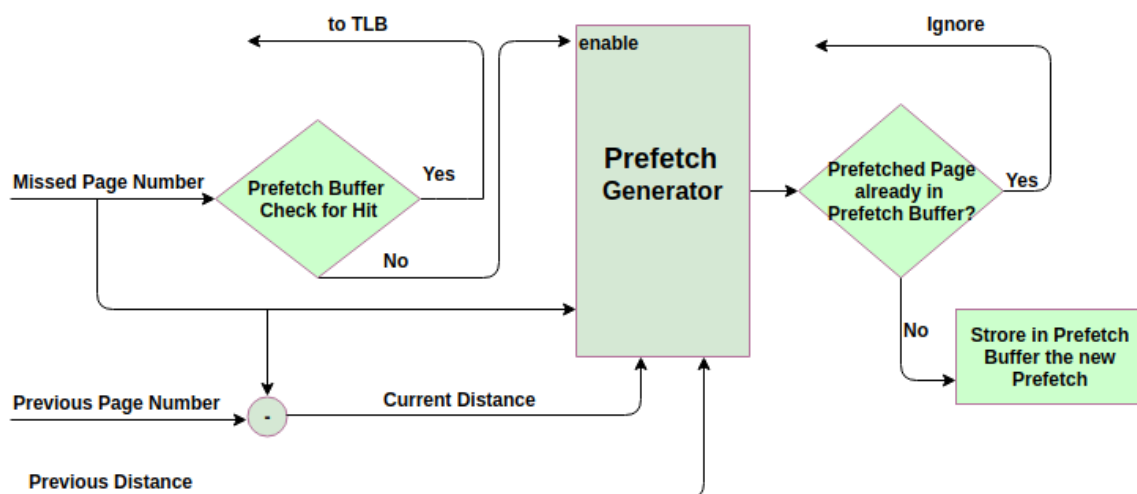
### 3.3.1 Ακολουθιακός Προανακλητής



Σχήμα 3.2: Ο μηχανισμός του Ακολουθιακού Προανακλητή (Stride Prefetcher).

Ο μηχανισμός πρόβλεψης του Ακολουθιακού Προανακλητή (Stride Prefetcher) προβλέπει πως οι αναφορές στη μνήμη που κάνει μία εφαρμογή είναι ακολουθιακές (σειριακές) με βάθος κατ' απόλυτη τιμή μέχρι και 2 και γι' αυτό σε κάθε TLB αστοχία φέρνει στο Πίνακα Προβλέψεων τις προβλέψεις αυτές. Αν για παράδειγμα μία αναφορά στη θέση μνήμης A, ακολουθείται από μία αναφορά σε μία από τις θέσεις μνήμης A+1, A-1, A+2, A-2, τότε θα ανιχνευθεί ορθά από το μηχανισμό αυτό, αλλιώς όχι. Θα πρέπει να τονιστεί ότι η παραπάνω ανάλυση γίνεται σε επίπεδο σελίδων. Το Σχήμα 3.2 δείχνει το μηχανισμό του Ακολουθιακού Προανακλητή και τη διαδικασία που ακολουθείται σε κάθε TLB αστοχία. Το πρώτο στάδιο είναι ο έλεγχος αν υπάρχει ευστοχία στον Prefetch Buffer ή όχι, όπως αναλύθηκε στο υποκεφάλαιο 3.2. Το δεύτερο στάδιο, αφορά τις νέες προβλέψεις και είναι ανεξάρτητο από το πρώτο. Στο στάδιο αυτό, για καθένα από τα βήματα (strides) 1, 2, -1, -2 δημιουργείται η αντίστοιχη πρόβλεψη με βάση τη τρέχουσα TLB αστοχία. Αν δεν υπάρχει ήδη στο Prefetch Buffer λόγω παλαιότερων προβλέψεων, μπαίνει στη κατάλληλη θέση του. Η διαδικασία αυτή επαναλαμβάνεται για κάθε TLB αστοχία. Ο συγκεκριμένος Ακολουθιακός Προανακλητής είναι μία πιο επιθετική εκδοχή του κλασσικού Ακολουθιακού Προανακλητή (Sequential Prefetcher) [30, 42], ο οποίος προέβλεπε μόνο την αμέσως επόμενη σελίδα από εκείνη που προκάλεσε TLB αστοχία.

### 3.3.2 Προανακλητής Τυχαίου Βήματος



Σχήμα 3.3: Ο μηχανισμός του Προανακλητή Τυχαίου Βήματος (Arbitrary Stride Prefetcher).

Ο μηχανισμός πρόβλεψης του Προανακλητή Τυχαίου Βήματος (Arbitrary Stride Prefetcher) αποτελεί μία διαφορετική προσέγγιση από τις ήδη γνωστές [30, 42]. Αποτελεί μία πιο απλή εκδοχή τους, αφού δεν χρησιμοποιεί κάποιο Πίνακα Παραγωγής Προβλέψεων ούτε κάποια μηχανή πεπερασμένης κατάστασης. Το Σχήμα 3.3 δείχνει το μηχανισμό του Προανακλητή Τυχαίου Βήματος και τη διαδικασία που ακολουθείται σε κάθε TLB αστοχία. Το πρώτο στάδιο είναι ο έλεγχος αν υπάρχει ευστοχία στον Prefetch Buffer ή όχι, όπως αναλύθηκε στο υποκεφάλαιο 3.2. Για να γίνει κατανοητή η διαδικασία που ακολουθείται στο δεύτερο στάδιο, το στάδιο των προβλέψεων, θα πρέπει να οριστεί σαν  $d(x,y)$  η απόσταση της σελίδας  $y$  από τη σελίδα  $x$ . Ο συγκεκριμένος προβλέπτης προσπαθεί να συσχετίσει τις αποστάσεις των τριών (3) τελευταίων σελίδων που προκάλεσαν TLB αστοχία, για να δημιουργήσει δύο νέες προβλέψεις. Αν υποθέσουμε τυχαία ότι οι τρεις τελευταίες σελίδες που προκάλεσαν TLB αστοχία είναι οι A,B,C τότε οι προβλέψεις που κάνει είναι οι εξής:

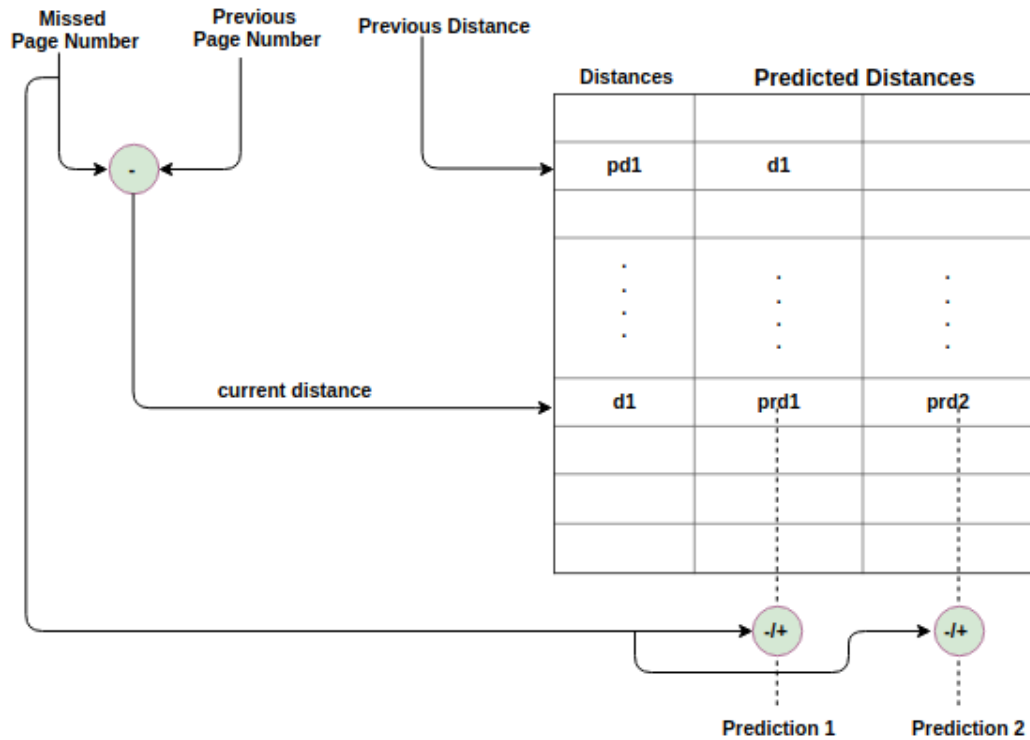
$$PR1 = C + d(C, B) \quad (3.1)$$

$$PR2 = C + d(B, A) \quad (3.2)$$

Θα πρέπει να τονιστεί ότι αν κάποια πρόβλεψη υπάρχει ήδη στον Prefetch Buffer τότε δεν πραγματοποιείται εκχώρηση της. Αν δεν υπάρχει τότε μπαίνει στη κατάλληλη θέση του. Η διαδικασία αυτή επαναλαμβάνεται για κάθε TLB αστοχία.

### 3.3.3 Προανακλητής Απόστασης

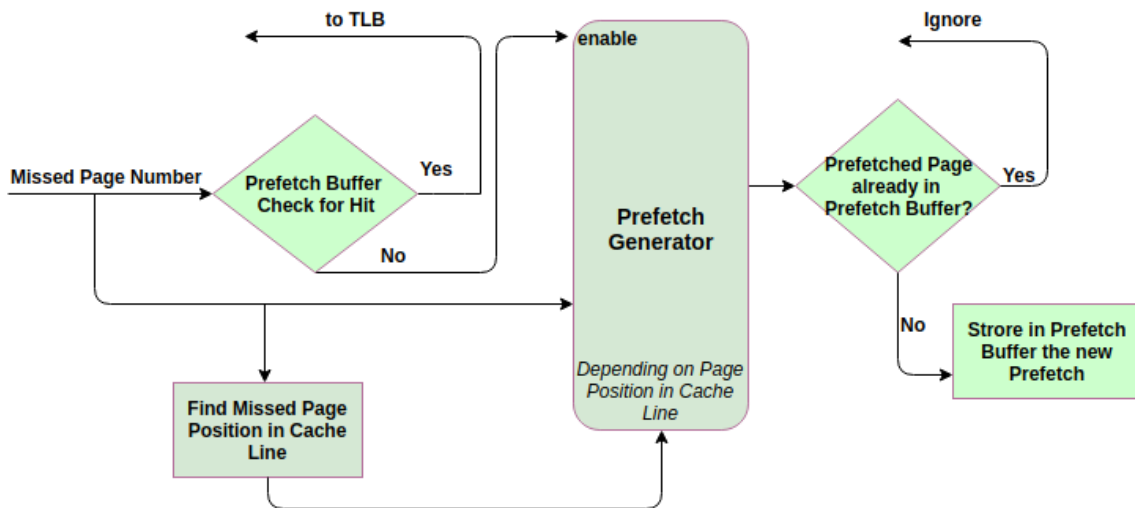
Ο μηχανισμός πρόβλεψης του Προανακλητή Απόστασης (Distance Prefetcher) υλοποιήθηκε με τη βοήθεια του [30], που παρείχε τις βασικές γνώσεις για να κατανοηθεί η λειτουργία του. Διατηρεί μία επιπλέον δομή δεδομένων, τον Πίνακα Αποστάσεων (Distance Table). Το μέγεθος του Πίνακα Αποστάσεων δεν μπορεί να είναι συγκρίσιμο με αυτό της TLB, ωστόσο μία ενδεικτική τιμή του μεγέθους του είναι 16 καταχωρήσεις, καθεμία από τις οποίες έχει δύο δικές της καταχωρήσεις. Το Σχήμα 3.4 δείχνει το μηχανισμό του Προανακλητή Απόστασης και τη διαδικασία που ακολουθείται σε κάθε TLB αστοχία. Το πρώτο στάδιο είναι ο έλεγχος αν υπάρχει ευστοχία στον Prefetch Buffer ή όχι, όπως αναλύθηκε στο υποκεφάλαιο 3.2. Το δεύτερο στάδιο ελέγχει αν η τρέχουσα απόσταση υπάρχει στον Πίνακα Αποστάσεων. Αν υπάρχει, ο προανακλητής χρησιμοποιεί τη τρέχουσα σελίδα μαζί με τις δύο προβλέψεις απόστασης της τρέχουσας απόστασης για να κάνει δύο νέες προβλέψεις. Αν δεν υπάρχουν ήδη στο Prefetch Buffer, εκχωρούνται σ' αυτόν. Στη συνέχεια, γίνεται ενημέρωση του Πίνακα Αποστάσεων, αφού η προηγούμενη απόσταση αποθηκεύει σαν πρόβλεψη της, την τρέχουσα απόσταση. Αν δεν υπάρχει η τρέχουσα απόσταση στο Πίνακα Αποστάσεων τότε αποθηκεύεται στη κατάλληλη θέση, φροντίζοντας οι προβλέψεις της να είναι άκυρες αφού μόλις έγινε η καταχώρησή της. Όμοια με πριν, η προηγούμενη απόσταση αποθηκεύει σαν πρόβλεψη της τη τρέχουσα απόσταση. Η διαδικασία αυτή επαναλαμβάνεται για κάθε TLB αστοχία.



Σχήμα 3.4: Ο μηχανισμός του Προανακλητή Απόστασης (Distance Prefetcher).

### 3.3.4 Προανακλητής Μηδενικού Κόστους

Ο Προανακλητής Μηδενικού Κόστους (Free Prefetcher) είναι ένας νέος μηχανισμός πρόβλεψης, ο οποίος για πρώτη φορά θα προταθεί στη μελέτη αυτή. Πρόκειται για ένα μηχανισμό, ο οποίος εκμεταλλεύεται τη χωρητικότητα του συστήματος κρυφών μνημών. Το Σχήμα 3.5 δείχνει το μηχανισμό του Προανακλητή Μηδενικού Κόστους και τη διαδικασία που ακολουθείται σε κάθε TLB αστοχία. Το πρώτο στάδιο είναι ο έλεγχος αν υπάρχει ευστοχία στον Prefetch Buffer ή όχι, όπως αναλύθηκε στο υποκεφάλαιο 3.2. Για να γίνει κατανοητή η διαδικασία του δεύτερου σταδίου θα πρέπει να αναφερθεί πως όταν μία σελίδα προκαλεί TLB αστοχία και δεν υπάρχει σαν καταχώρηση στον Prefetch Buffer, στο τέλος του page walk εκτός από τη TLB μεταφέρεται και στο σύστημα των κρυφών μνημών. Στην αρχιτεκτονική x86-64 το μέγεθος μιας καταχώρησης του συστήματος κρυφών μνημών (cache line) επαρκεί για να χωρέσουν οκτώ (8) page table entries. Το δεύτερο στάδιο, εκμεταλλεύεται αυτό ακριβώς το γεγονός, ότι δηλαδή κατοικούν εφτά (7) πιθανές προβλέψεις στο ίδιο cache line, οι οποίες είναι μηδενικού κόστους αφού δεν χρειάζεται καμία επιπλέον λειτουργία μνήμης για να προανακληθούν στο Prefetch Buffer. Αν υποθέσουμε ότι η σελίδα A προκάλεσε TLB αστοχία, ο Προανακλητής Μηδενικού Κόστους αρχικά βρίσκει τη θέση που βρίσκεται η σελίδα αυτή στη κατάλληλη γραμμή του συστήματος κρυφών μνημών. Στη συνέχεια εκμεταλλεύεται και τις εφτά (7) προβλέψεις που είναι μηδενικού κόστους, δηλαδή τις τοποθετεί στο Prefetch Buffer, αν δεν υπάρχουν ήδη μέσα του. Η διαδικασία αυτή επαναλαμβάνεται για κάθε TLB αστοχία. Αν υποθέσουμε ότι συμβαίνει μία TLB αστοχία στη σελίδα C, τότε η σελίδα αυτή στο τέλος του Page Walk θα βρίσκεται εκτός από τη TLB και σε κάποιο cache line. Στο ίδιο cache line θα υπάρχουν, όπως εξηγήθηκε παραπάνω, οι σελίδες A,B,D,E,F,G και H. Οι σελίδες αυτές είναι οι μηδενικού κόστους προβλέψεις που κάνει ο συγκεκριμένος προανακλητής.



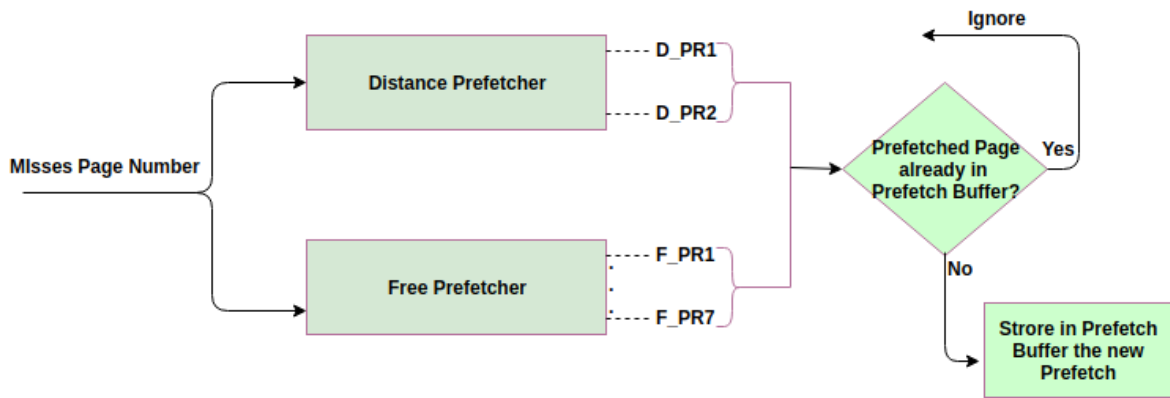
Σχήμα 3.5: Ο μηχανισμός του Προανακλητή Μηδενικού Κόστους (Free Prefetcher).

### 3.4 Υβριδικές Τεχνικές Προανάκλησης Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης

Οι Υβριδικές Τεχνικές Προανάκλησης Μεταφράσεων υλοποιήθηκαν διότι παρατηρήθηκε πως στα benchmarks στα οποία πετύχαινε καλή απόδοση ο Προανακλητής Απόστασης, ο Ακολουθιακός Προανακλητής και ο Προανακλητής Μηδενικού Κόστους παρουσιάζαν κακή απόδοση και αντίστροφα. Το γεγονός ότι οι προανακλητές αυτοί είναι ορθογώνιοι μεταξύ τους, οδήγησε στην υλοποίηση των Υβριδικών Τεχνικών Προανάκλησης Μεταφράσεων. Σε κάθε TLB αστοχία και οι δύο Μηχανισμοί Προανάκλησης Μεταφράσεων που περιέχονται στον εκάστοτε υβριδικό μηχανισμό, παράγουν προβλέψεις και τις τοποθετούν στο κοινό Πίνακα Προβλέψεων. Στη συνέχεια παρουσιάζονται οι δύο Υβριδικοί Προανακλητές που υλοποιήθηκαν στη μελέτη αυτή.

#### 3.4.1 Υβριδικός Προανακλητής Μηδενικού Κόστους

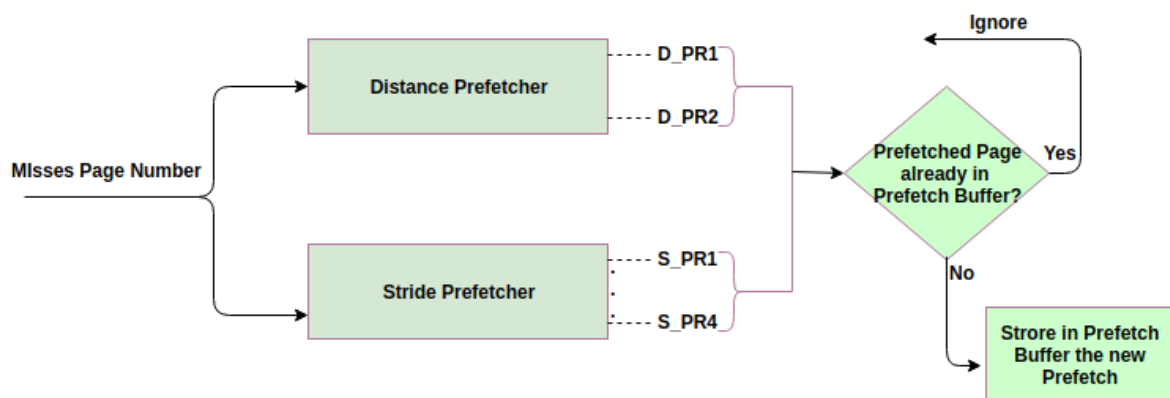
Ο Υβριδικός Προανακλητής Μηδενικού Κόστους (Hybrid Distance and Free Prefetcher) είναι ένας νέος μηχανισμός πρόβλεψης, ο οποίος συνδυάζει τις ιδέες των μηχανισμών που αναλύθηκαν στα υποκεφάλαια 3.3.4 και 3.3.3. Διατηρεί μία επιπλέον δομή δεδομένων, το Πίνακα Αποστάσεων (Distance Table), το μέγεθος του οποίου δεν μπορεί να είναι συγκρίσιμο με αυτό της TLB, ωστόσο μία ενδεικτική τιμή του μεγέθους του είναι 16 καταχωρήσεις, καθεμία από τις οποίες έχει δύο δικές της καταχωρήσεις. Ο μηχανισμός αυτός σε κάθε TLB αστοχία επιτρέπει και στα δύο Σχήματα Προανάκλησης Μεταφράσεων να κάνουν τις δικές τους προβλέψεις και να τις τοποθετήσουν στον κοινό Πίνακα Προβλέψεων, αν δεν υπάρχουν ήδη σε αυτόν. Ο τρόπος παραγωγής των προβλέψεων για καθένα από τα δύο αυτά σχήματα αναλύεται στα υποκεφάλαια 3.3.4 και 3.3.3. Το Σχήμα 3.6 δείχνει το μηχανισμό του Υβριδικού Προανακλητή Μηδενικού Κόστους.



Σχήμα 3.6: Ο μηχανισμός του Υβριδικού Προανακλητή Μηδενικού Κόστους (Hybrid Distance and Free Prefetcher).

### 3.4.2 Υβριδικός Ακολουθιακός Προανακλητής

Ο Υβριδικός Ακολουθιακός Προανακλητής (Hybrid Distance and Stride Prefetcher) είναι ένας νέος μηχανισμός πρόβλεψης, ο οποίος συνδυάζει τις ιδέες των μηχανισμών που αναλύθηκαν στα υποκεφάλαια 3.3.1 και 3.3.3. Διατηρεί μία επιπλέον δομή δεδομένων, το Πίνακα Αποστάσεων (Distance Table), το μέγεθος του οποίου δεν μπορεί να είναι συγκρίσιμο με αυτό της TLB, ωστόσο μία ενδεικτική τιμή του μεγέθους του είναι 16 καταχωρήσεις, καθεμία από τις οποίες έχει δύο δικές της καταχωρήσεις. Ο μηχανισμός αυτός σε κάθε TLB αστοχία επιτρέπει και στα δύο Σχήματα Προανάκλησης Μεταφράσεων να κάνουν τις δικές τους προβλέψεις και να τις τοποθετήσουν στον κοινό Πίνακα Προβλέψεων, αν δεν υπάρχουν ήδη σε αυτόν. Ο τρόπος παραγωγής των προβλέψεων για καθένα από τα δύο αυτά σχήματα αναλύεται στα υποκεφάλαια 3.3.1 και 3.3.3. Το Σχήμα 3.7 δείχνει το μηχανισμό του Υβριδικού Ακολουθιακού Προανακλητή.

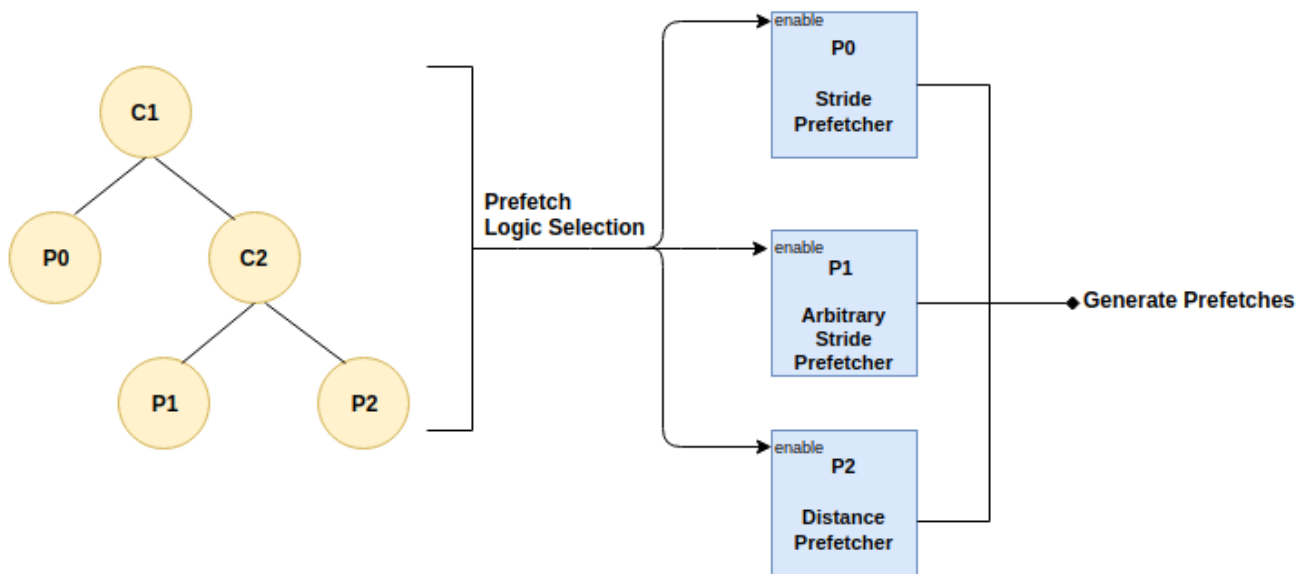


Σχήμα 3.7: Ο μηχανισμός του Υβριδικού Ακολουθιακού Προανακλητή (Hybrid Distance and Stride Prefetcher).



### 3.5 Tournament Προβλέπτης

Οι Tournament Προβλέπτες που χρησιμοποιούνται για τη Πρόβλεψη Διακλαδώσεων (Branch Prediction) ήταν η ιδέα πάνω στην οποία στηρίχτηκε η υλοποίηση του Tournament Προβλέπτη (Tournament Predictor) [3]. Πρόκειται για ένα νέο μηχανισμό πρόβλεψης, ο οποίος προσπαθεί να συνδυάσει τις ιδέες των μηχανισμών που αναλύθηκαν στα υποκεφάλαια 3.3.1, 3.3.2 και 3.3.3. Εκτός από το πραγματικό Πίνακα Προβλέψεων (Prefetch Buffer) διαθέτει έναν Πίνακα Αποστάσεων (Distance Table) και άλλους τρεις Ψεύτικους Πίνακες Προβλέψεων (Fake Prefetch Buffers). Το μέγεθος όλων των δομών δεδομένων που αναφέρθηκαν παραπάνω δεν μπορεί να είναι συγκρίσιμα με αυτό της TLB. Οι τρεις Fake Prefetch Buffers είναι υπεύθυνοι για να παράγουν εικονικές προβλέψεις, οι οποίες θα βοηθήσουν τελικά τον Tournament Προβλέπτη να ανακαλύπτει δυναμικά ποιό από τα τρία Σχήματα Προανάκλησης Μεταφράσεων είναι το πλέον κατάλληλο για να παράγει πραγματικές προβλέψεις σε κάθε TLB αστοχία. Οι εικονικές προβλέψεις δεν κοστίζουν σε λειτουργίες μνήμης αφού στη πράξη δεν κάνουν κάποια πραγματική προανάκληση στον Prefetch Buffer. Ο Tournament Προβλέπτης είναι πρακτικά μία δενδρική δομή, η οποία έχει στη ρίζα της ένα 2-bit προβλέπτη, στο επόμενο επίπεδο δύο παιδιά, εκ των οποίων το ένα είναι κάποιο από τα τρία Σχήματα Πρόβλεψης και το άλλο είναι ένας άλλος 2-bit προβλέπτης, ο οποίος με τη σειρά του έχει σαν παιδιά του τα άλλα δύο Σχήματα Πρόβλεψης. Οι επιτυχίες και οι αποτυχίες σε καθένα από τους Fake Prefetch Buffers αυξάνει, μειώνει ή διατηρεί κατάλληλα τους μετρητές των 2-bit προβλεπτών. Θα πρέπει να τονιστεί ότι αν κάποιος από τους 2-bit προβλέπτες είναι 0 ή 1 επιλέγεται το αριστερό παιδί ενώ αν είναι 2 ή 3 το δεξί παιδί. Οι Πίνακες 3.1, 3.2 δείχνουν τις αλλαγές που προκαλούνται στους 2-bit προβλέπτες για κάθε πιθανό συνδυασμό επιτυχιών/αποτυχιών των τριών Σχημάτων Πρόβλεψης. Το Σχήμα 3.8 δείχνει το μηχανισμό του Tournament Προανακλητή. Στη πράξη αρχικά ενημερώνονται οι μετρητές C1 και C0 ανάλογα με το ποιό Προανακλητές Μεταφράσεων πρόβλεψαν σωστά ή όχι τη τωρινή TLB αστοχία. Στη συνέχεια ελέγχεται η τιμή του C1 και μόνο αν η τιμή του είναι μεγαλύτερη του δύο (2) ελέγχεται η τιμή του C2 ώστε να επιλεγεί ο κατάλληλος προανακλητής. Για παράδειγμα αν υποθέσουμε ότι σε κάποιο στιγμιότυπο ισχύει ότι  $(C1, C2) = (1, 2)$  και  $(Stride, Arbitrary Stride, Distance) = (Hit, Miss, Miss)$ . Ο C2 επιλέγει τον Distance επειδή έχει τιμή 2. Ο C1 αλλάζει τιμή σε 0 διότι  $(Stride, Distance) = (Hit, Miss)$ . Ο C2 δεν αλλάζει τιμή αφού  $(Arbitrary Stride, Distance) = (Miss, Miss)$ . Αφού ο C1 έχει τιμή μηδέν επιλέγεται ο Stride Prefetcher να παράγει προβλέψεις στη συγκεκριμένη φάση. Εν κατακλείδι, με το τρόπο αυτό γίνεται η επιλογή του καταλληλότερου Σχήματος Προανάκλησης Μεταφράσεων και στη συνέχεια του δίνεται η ευκαιρία να παράγει πρόβλεψεις.



Σχήμα 3.8: Ο μηχανισμός του Tournament Προανακλητή (Tournament Prefetcher).



Arbitrary Stride	Distance	C2
Miss	Miss	no change
Miss	Hit	+1
Hit	Miss	-1
Hit	Hit	no change

Πίνακας 3.1: Όλες οι πιθανές μεταβολές του 2-bit προβλέπτη C2.

Stride	Selected from C2	C1
Miss	Miss	no change
Miss	Hit	+1
Hit	Miss	-1
Hit	Hit	no change

Πίνακας 3.2: Όλες οι πιθανές μεταβολές του 2-bit προβλέπτη C1.

### 3.6 Η Έννοια του Κόστους στη Προανάκληση Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης

Σκοπός της Προανάκλησης Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης είναι η βελτίωση της απόδοσης της TLB ώστε να αποφευχθεί όσο το δυνατόν περισσότερο η επιβάρυνση που εισάγουν οι λειτουργίες μνήμης που απαιτούνται για τη μετάφραση εικονικών σε φυσικές διευθύνσεις. Ωστόσο, η Προανάκληση Μεταφράσεων εισάγει κι αυτή με τη σειρά της επιβάρυνση στο σύστημα, αφού δημιουργεί αιτήματα πρόσβασης στη μνήμη ώστε να εξασφαλίσει την προανάκληση των ζητούμενων μεταφράσεων. Επομένως, η Προανάκληση Μεταφράσεων είναι ένα φιλόδοξο εγχείρημα, το οποίο μπορεί να βελτιώσει την απόδοση του συστήματος αλλά μπορεί και να την κάνει ακόμη χειρότερη, αν η προς εκτέλεση εφαρμογή κάνει προσβάσεις στη μνήμη με τρόπο ο οποίος δεν μπορεί αν ανιχνευθεί ορθά από το Σχήμα Προανάκλησης που χρησιμοποιείται. Στο κεφάλαιο αυτό θα αναλυθούν δύο Μηχανισμοί Ελέγχου Προανάκλησης που ενσωματώνονται στα Σχήματα Προανάκλησης Μεταφράσεων που αναλύθηκαν στα υποκεφάλαια 3.3, 3.4 και 3.5 με σκοπό τη μείωση των άχρηστων προβλέψεων και την απενεργοποίηση του Prefetching όταν οι προβλέψεις δεν προσφέρουν βελτίωση της απόδοσης.

#### 3.6.1 Μηχανισμοί Ελέγχου Προανάκλησης Επιτυχόντων Προβλέψεων

Στην ενότητα αυτή θα παρουσιαστούν δύο Μηχανισμοί Ελέγχου Προανάκλησης Επιτυχόντων Προβλέψεων (Hit Throttling). Η πρώτη εκδοχή, θα εφαρμοστεί στους Μηχανισμούς Προανάκλησης Μεταφράσεων των Κεφαλαίων 3.3 και 3.4, ενώ η δεύτερη σ' αυτόν του Κεφαλαίου 3.5.

##### 3.6.1.1 Μηχανισμός Ελέγχου Προανάκλησης Επιτυχόντων Προβλέψεων βασισμένο σε Εκπαίδευση (Hit Interval Throttling)

Ο μηχανισμός Ελέγχου Προανάκλησης Επιτυχόντων Προβλέψεων που βασίζεται σε Εκπαίδευση ενσωματώνεται σε καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων των Κεφαλαίων 3.3 και 3.4. Ο συγκεκριμένος μηχανισμός, διακρίνει δύο περιόδους, οι οποίες επαναλαμβάνονται κυκλικά. Η πρώτη ονομάζεται περίοδος εκμάθησης και η δεύτερη περίοδος εφαρμογής. Κατά τη περίοδο της εκμάθησης ο μηχανισμός προσπαθεί να αποκτήσει μία όσο το δυνατόν πιο ισχυρή άποψη σχετικά με το αν πρέπει ή όχι να ενεργοποιηθεί το Σχήμα Προανάκλησης Μεταφράσεων στην επόμενη περίοδο εφαρμογής. Η άποψή του διαμορφώνεται σύμφωνα με το πλήθος το επιτυχημένων προβλέψεων κατά τη διάρκεια κάθε περιόδου εκμάθησης, αφού υπάρχει ένα κατώφλι πάνω από το οποίο ενεργοποιείται το Σχήμα Προανάκλησης Μεταφράσεων στην επόμενη περίοδο εφαρμογής,

αλλιώς απενεργοποιείται. Η διάρκεια της περιόδου εκμάθησης είναι προφανώς μικρότερη από τη διάρκεια της περιόδου εφαρμογής αλλά η διάρκεια καθεμίας μπορεί να πάρει διάφορες τιμές, όπως θα αναλυθεί και σε επόμενα κεφάλαια αναλυτικά. Εν κατακλείδι, ο Μηχανισμός του Hit Interval Throttling ενεργοποιεί το Σχήμα Προανάκλησης Μεταφράσεων σε κάθε περίοδο εκμάθησης και αν οι επιτυχείς προβλέψεις υπερβούν κάποιο κάτω φράγμα τότε η προανάκληση συνεχίζεται και στη περίοδο εφαρμογής. Το Σχήμα 3.9 δείχνει την ιδέα πάνω στην οποία στηρίζεται ο Μηχανισμός του Hit Interval Throttling.

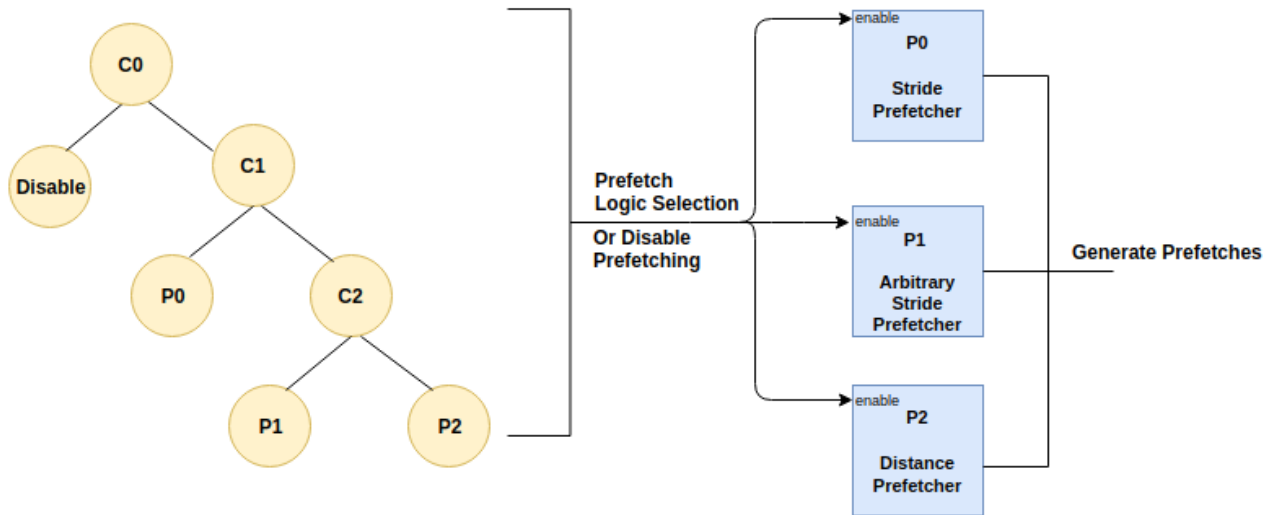


Σχήμα 3.9: Αφαιρετική αναπαράσταση του Μηχανισμού Hit Interval Throttling.

### 3.6.1.2 Δενδρικός Μηχανισμός Ελέγχου Προανάκλησης Επιτυχόντων Προβλέψεων (Hit Confidence Throttling)

Ο Δενδρικός Μηχανισμός Ελέγχου Προανάκλησης Επιτυχόντων Προβλέψεων ενσωματώνεται στο Σχήμα Προανάκλησης Μεταφράσεων του Κεφαλαίου 3.5. Ο λόγος ανάπτυξης διαφορετικού μηχανισμού ελέγχου προανάκλησης για το συγκεκριμένο Σχήμα Προανάκλησης είναι η ευκαιρία εκμετάλλευσης της δενδρικής δομής που διατηρεί. Ο Μηχανισμός του Hit Confidence Throttling προσθέτει ένα ακόμα επίπεδο στο δένδρο του Σχήματος 3.8. Όπως φαίνεται στο Σχήμα 3.10, στη ρίζα του δένδρου τοποθετείται ένας νέος 2-bit προβλέπτης, ο οποίος αποφασίζει αν θα ενεργοποιήσει ή όχι την προανάκληση μεταφράσεων. Οι ταυτόχρονες αποτυχίες σε καθένα από τους ψεύτικους Πίνακες Προβλέψεων, οι οποίοι αποτελούν μέρος της υλοποίησης που περιγράφεται αναλυτικά στην ενότητα 3.5, τροποποιεί κατάλληλα τον 2-bit προβλέπτη της ρίζας του δένδρου ώστε να αποκτήσει μεγαλύτερη αυτοπεποίθηση όσο αφορά την απενεργοποίηση της προανάκλησης μεταφράσεων. Σε αντίθετη περίπτωση, δηλαδή όταν υπάρχει έστω και μία επιτυχία, η τροποποίηση του 2-bit προβλέπτη έχει αντίθετες επιπτώσεις. Θα πρέπει να τονιστεί ότι αν κάποιος από τους 2-bit προβλέπτες είναι 0 ή 1 επιλέγεται το αριστερό παιδί ενώ αν είναι 2 ή 3 το δεξί παιδί.

Οι Πίνακες 3.1, 3.2, οι οποίοι παρουσιάστηκαν στο υποκεφάλαιο 3.5, δείχνουν τις αλλαγές που προκαλούνται στους 2-bit προβλέπτες C1 και C2 για κάθε πιθανό συνδυασμό επιτυχιών/αποτυχιών των τριών Σχημάτων Πρόβλεψης. Επομένως, η λειτουργία των C1 και C2 έχει παραμείνει η ίδια. Ο Πίνακας 3.3 εξηγεί την λειτουργία του 2-bit προβλέπτη C0. Για παράδειγμα αν υποθέσουμε ότι σε κάποιο στιγμιότυπο ισχύει ότι  $(C0, C1, C2) = (0, 2, 1)$  και  $(Stride, Arbitrary Stride, Distance) = (Miss, Hit, Miss)$ , τότε ο C2 επιλέγει τον Arbitrary Stride επειδή έχει τιμή 1 και η νέα του τιμή θα είναι 0 αφού  $(Arbitrary Stride, Distance) = (Hit, Miss)$ , ενώ ο C1 επιλέγει την επιλογή που έκανε ο C2, δηλαδή τον Distance επειδή έχει τιμή 2 και δεν θα αλλάζει τιμή διότι  $(Stride, Distance) = (Miss, Miss)$ . Εφόσον ο C0 έχει τιμή 0 τότε για τη συγκεκριμένη TLB αστοχία απενεργοποιείται το Prefetching. Επίσης, επειδή έστω και ένας από τους Μηχανισμούς Πρόβλεψης έκανε σωστή πρόβλεψη ο C0 θα πάρει τη τιμή 1. Εν κατακλείδι, με το τρόπο αυτό λαμβάνεται η απόφαση σχετικά με το αν πρέπει να ενεργοποιηθεί κάποιο Σχήμα Προανάκλησης Μεταφράσεων ή όχι.



Σχήμα 3.10: Αφαιρετική αναπαράσταση του Μηχανισμού Hit Confidence Throttling.

Stride	Arbitrary Stride	Distance	C0
Miss	Miss	Miss	-1
Miss	Miss	Hit	+1
Miss	Hit	Miss	+1
Miss	Hit	Hit	+1
Hit	Miss	Miss	+1
Hit	Miss	Hit	+1
Hit	Hit	Miss	+1
Hit	Hit	Hit	+1

Πίνακας 3.3: Όλες οι πιθανές μεταβολές του 2-bit προβλέπτη C0, για το Μηχανισμό Hit Confidence Throttling.

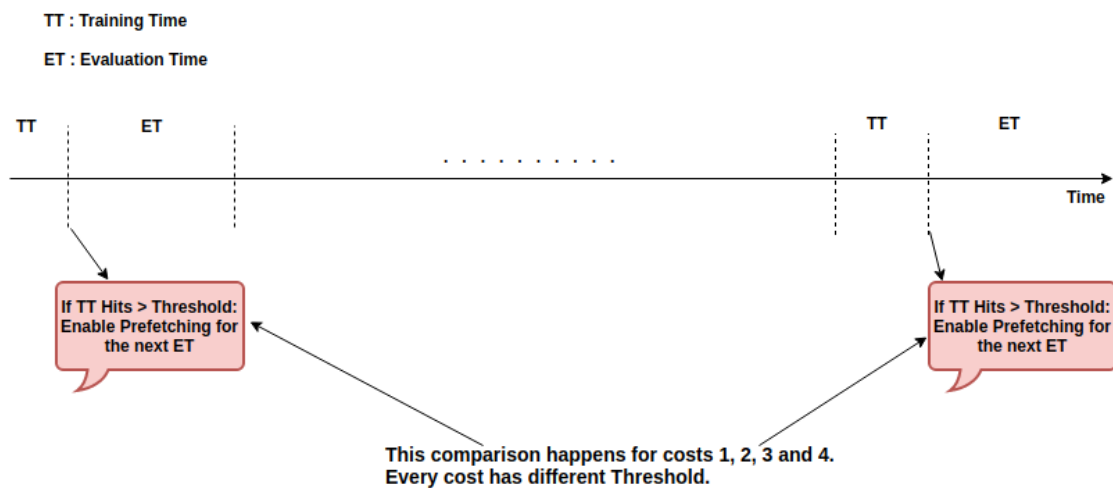
### 3.6.2 Μηχανισμοί Ελέγχου Προανάκλησης MMU Κρυφής Μνήμης

Στην ενότητα αυτή θα παρουσιαστούν δύο Μηχανισμοί Ελέγχου Προανάκλησης MMU Κρυφής Μνήμης. Η ιδέα πάνω στην οποία στηρίζονται οι μηχανισμοί αυτοί είναι ο διαχωρισμός των προβλέψεων ανάλογα με το κόστος που έχει η προανάκληση τους οσον αφορά τις λειτουργίες μνήμης που απαιτούνται για την κάθε προανάκληση. Επομένως, οι διάφορες προβλέψεις ομαδοποιούνται ανάλογα με το κόστος που έχει η ενδεχόμενη προανάκλησή τους. Η πρώτη εκδοχή, θα εφαρμοστεί στους Μηχανισμούς Προανάκλησης Μεταφράσεων των Κεφαλαίων 3.3 και 3.4, ενώ η δεύτερη στο μηχανισμό του Κεφαλαίου 3.5.

#### 3.6.2.1 Μηχανισμός Ελέγχου Προανάκλησης MMU Κρυφής Μνήμης βασισμένος σε Εκπαίδευση (MMU Interval Throttling)

Ο Μηχανισμός του MMU Interval Throttling ενσωματώνεται σε καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων των Κεφαλαίων 3.3 και 3.4. Ο συγκεκριμένος μηχανισμός, διακρίνει δύο περιόδους, οι οποίες επαναλαμβάνονται κυκλικά. Η πρώτη ονομάζεται περίοδος εκμάθησης και η δεύτερη περίοδος εφαρμογής. Στην αρχιτεκτονική x86-64, όπως φαίνεται και στο Σχήμα 2.9, υπάρχουν 3 MMU Κρυφές Μνήμες με διαφορετικά χαρακτηριστικά και πολιτικές αντικατάστασης, μία για καθένα από τα επίπεδα PML4, PDP και PD του Πίνακα Σελίδων. Περαιτέρω ανάλυση αυτών παρέχεται στο υποκεφάλαιο 2.2.3. Ο συγκεκριμένος μηχανισμός ελέγχου προανάκλησης, για την

ορθή λειτουργία του, απαιτεί σε κάθε περίοδο εκμάθησης να διατηρεί σε κάποιους μετρητές το πλήθος των προβλέψεων που κοστίζουν μία, δύο, τρεις ή και τέσσερις επιπλέον λειτουργίες μνήμης για να πραγματοποιηθούν. Στο υποκεφάλαιο 2.2.3 αναλύθηκε ότι μία επιτυχής αναζήτηση στο επίπεδο PD, PDP και PML4 της MMU Κρυφής Μνήμης μειώνει τις απαιτούμενες λειτουργίες μνήμης κατά τρεις, δύο και μία αντίστοιχα. Προφανώς αν υπάρχει MMU αστοχία τότε δεν συμβαίνει καμία μείωση. Επίσης, διατηρεί σε ένα μονοδιάστατο πίνακα το πλήθος των προβλέψεων κάθε επιπέδου της MMU Κρυφής Μνήμης που οδήγησαν σε επιτυχία. Με το τρόπο αυτό στο τέλος κάθε περιόδου εκμάθησης, υπάρχουν τέσσερα το πλήθος ποσοστά, τα οποία φανερώνουν την απόδοση των τριών επιπέδων της MMU Κρυφής Μνήμης καθώς και την απόδοση των αστοχιών της. Επομένως, χρησιμοποιώντας τα ποσοστά αυτά και κατάλληλα κατώφλια για καθένα από αυτά, ο μηχανισμός ελέγχου προανάκλησης μπορεί να αποφανθεί σχετικά με το αν προβλέψεις με κόστος ένα, δύο, τρία ή τέσσερα είναι ευεργετικές για την απόδοση του συστήματος. Η περίοδος εφαρμογής ελέγχει τα ποσοστά επιτυχίας και όποια υπερβαίνουν το αντίστοιχο κατώφλι τους αποφασίζεται να συνεχίσουν να παράγουν προβλέψεις κατά τη διάρκεια αυτής της περιόδου. Όσα ποσοστά επιτυχίας δεν είναι ικανοποιητικά, απενεργοποιούν τις προβλέψεις τους για τη συγκεκριμένη περίοδο εφαρμογής. Ενδεικτικά ποσοστά που χρησιμοποιήθηκαν σαν κατώφλια στη πειραματική αξιολόγηση είναι το 12,5% και το 25%. Τέλος, η διάρκεια της περιόδου εκμάθησης είναι προφανώς μικρότερη από τη διάρκεια της περιόδου εφαρμογής αλλά η διάρκεια καθεμίας μπορεί να πάρει διάφορες τιμές, όπως θα αναλυθεί και σε επόμενα κεφάλαια αναλυτικά. Το Σχήμα 3.11 δείχνει το μηχανισμό του MMU Interval Throttling.

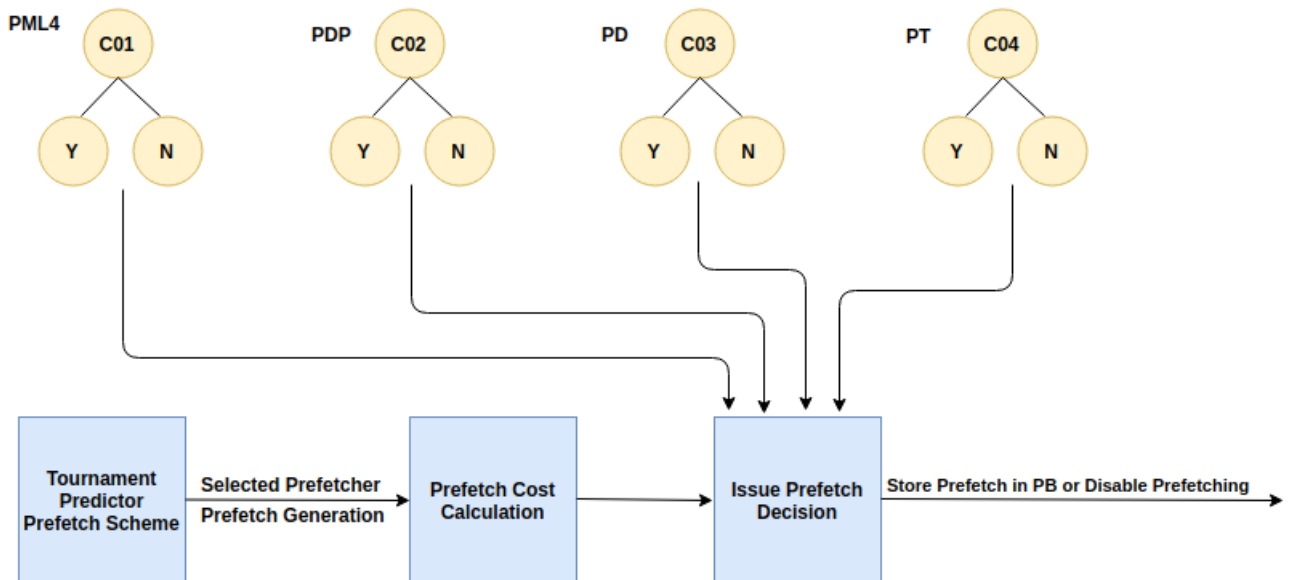


Σχήμα 3.11: Αφαιρετική αναπαράσταση του Μηχανισμού MMU Interval Throttling.

### 3.6.2.2 Δενδρικός Μηχανισμός Ελέγχου Προανάκλησης MMU Κρυφής Μνήμη (MMU Confidence Throttling)

Ο Μηχανισμός του MMU Confidence Throttling ενσωματώνεται στο Σχήμα Προανάκλησης Μεταφράσεων του Κεφαλαίου 3.5. Ο μηχανισμός αυτός δημιουργεί μία νέα δενδρική δομή, η οποία αποτελείται από ένα κόμβο με δύο παιδιά, για καθένα από τα επίπεδα της δομής της MMU Κρυφής Μνήμης, για τα οποία περισσότερες πληροφορίες παρέχονται στο υποκεφάλαιο 2.2.3. Στη ρίζα κάθε δενδρικής δομής υπάρχει ένας 4-bit προβλέπτης, ο οποίος αποφασίζει δυναμικά αν θα επιτρέψει ή όχι την Προανάκληση Μεταφράσεων με το αντίστοιχο κόστος. Δηλαδή καθεμία από τις δενδρικές δομές είναι υπεύθυνη με ένα μόνο κόστος, αφού κάθε επίπεδο της MMU Κρυφής Μνήμης εισάγει διαφορετικό κόστος προανάκλησης (2.2.3). Εφόσον οι καταστάσεις είναι δύο για κάθε δενδρική δομή, θα αρκούσε και ένας 1-bit προβλέπτης στη ρίζα. Στη πράξη δεν ισχύει αυτό, αφού για την απενεργοποίηση της Προανάκλησης Μεταφράσεων για κάποιο κόστος θα πρέπει να υπάρχει μεγάλη βεβαιότητα ότι η ενέργεια αυτή θα επιφέρει θετικά αποτελέσματα. Ο 4-bit προβλέπτης έχει

μεγαλύτερη ανοχή σε κακές προβλέψεις και για το λόγο αυτό επιλέχθηκε. Για τη σωστή λειτουργία του μηχανισμού κάθε προανάκληση, πραγματική ή ψεύτικη, σε κάποιον από τους Πίνακες Προβλέψεων, πραγματικό ή ψεύτικο, μεταφέρει και το κόστος που είχε για να πραγματοποιηθεί. Στο σημείο αυτό πρέπει να τονιστεί οι εικονές προβλέψεις που παράγονται δεν κοστίζουν σε λειτουργίες μνήμης αφού στη πράξη δεν κάνουν κάποια πραγματική προανάκληση, παρά μόνο βοηθούν στη δυναμική επιλογή του καταλληλότερου Σχήματος Προανάκλησης. Επομένως, μία επιτυχής αναζήτηση σε κάποιον από τους Πίνακες Προβλέψεων οδηγεί σε αύξηση της εμπιστοσύνης για το συγκεκριμένο κόστος, ανανεώνοντας κατάλληλα τον αντίστοιχο 4-bit προβλέπτη. Η μείωση της εμπιστοσύνης για κάποιο κόστος συμβαίνει όταν κάποια πρόβλεψη που βρίσκεται σε κάποιο Πίνακα Προβλέψεων αντικατασταθεί πριν προλάβει να προκαλέσει μία εύστοχη αναζήτηση, δηλαδή η ανίχνευση άχρηστων προβλέψεων μειώνει την εμπιστοσύνη για το αντίστοιχο κόστος. Η απενεργοποίηση της Προανάκλησης Μεταφράσεων για κάποιο κόστος αποφασίζεται όταν συμβούν 16 συνεχόμενες άχρηστες προβλέψεις αυτού του κόστους. Είναι σημαντικό να σημειωθεί πως όταν κάποιος από τους 4-bit προβλέπτες φτάσει στη τιμή που απενεργοποιεί την Προανάκληση Μεταφράσεων για το συγκεκριμένο κόστος, τότε μόνο επιτυχείς αναζητήσεις στους ψεύτικους Πίνακες Προβλέψεων για το συγκεκριμένο κόστος μπορούν να ενεργοποιήσουν και πάλι την Προανάκληση Μεταφράσεων για τον συγκεκριμένο 4-bit προβλέπτη. Το Σχήμα 3.12 δείχνει τον Δενδρικό Μηχανισμό Ελέγχου Προανάκλησης MMU Κρυφής Μνήμη.



Σχήμα 3.12: Αφαιρετική αναπαράσταση του Μηχανισμού MMU Confidence Throttling.



## Κεφάλαιο 4

### Μεθοδολογία

#### 4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιαστούν εκτενώς τα χαρακτηριστικά του υπολογιστικού συστήματος στο οποίο έλαβαν χώρα τα διάφορα πειράματα, δίνοντας έμφαση σε χαρακτηριστικά που αφορούν τη Κρυφή Μνήμη Αναζήτησης Μετάφρασης (TLB). Στη συνέχεια θα παρουσιαστούν τα διάφορα εργαλεία που χρησιμοποιήθηκαν για τη συλλογή των αποτελεσμάτων που χρειαζόμασταν. Τέλος θα γίνει έκθεση των διαφόρων μετροπρογραμμάτων (benchmarks) που χρησιμοποιήθηκαν στη μελέτη αυτή, για την αξιολόγηση της απόδοσης των διαφόρων Σχημάτων Προανάκλησης Μεταφράσεων στη TLB.

#### 4.2 Πειραματικό Περιβάλλον

Η μελέτη αυτή βασίζεται σε ένα τετραπύρνο (4-core) Intel(R) Core(TM) i5-7600 (Kaby Lake), που τρέχει στα 3.5 GHz και είναι εξοπλισμένο με 16GB μνήμη. Καθένας από τους πυρήνες έχει μία ιδιωτική οργάνωση Κρυφής Μνήμης Αναζήτησης Μετάφρασης (TLB). Θα πρέπει να τονιστεί ότι υπάρχουν διαφορετικές TLBs για εντολές και δεδομένα, με ονόματα ITLB και DTLB αντίστοιχα. Συγκεκριμένα διαθέτει ένα L1 DTLB για δεδομένα, ένα L1 ITLB για εντολές και ένα L2 TLB το οποίο μοιράζεται μεταξύ των L1 DTLB και L1 ITLB, όπως φαίνεται στο Πίνακα 4.1. Στη μελέτη αυτή, η προσοχή εστιάζεται όσο αφορά τις DTLB αστοχίες και όχι τις ITLB αστοχίες, διότι οι πρώτες είναι αυτές που ευθύνονται κατά κύριο λόγο για την επιβάρυνση ενός συστήματος. Επομένως κάθε αναφορά σε TLB αστοχία, θα εννοεί DTLB αστοχία στη μελέτη αυτή.

L1 ITLB	4KB 2MB/4MB	128 entries 8 entries/thread	8-way assoc fully assoc
L1 DTLB	4KB 2MB/4MB 1G	64 entries 32 entries 4 entries	4-way assoc 4-way assoc fully assoc
L2 TLB	4KB + 2MB 1G	1536 entries 16 entries	12-way assoc 4-way assoc

Πίνακας 4.1: Ιεραρχία Κρυφής Μνήμης Αναζήτησης Μετάφρασης (TLB) για κάθε πυρήνα του υπολογιστικού μηχανήματος.

Η διανομή που χρησιμοποιεί το συγκεκριμένο υπολογιστικό μηχανήμα είναι Debian, με έκδοση 8.10 (jessie) και η έκδοση του πυρήνα του Linux η οποία χρησιμοποιήθηκε και τροποποιήθηκε είναι η 3.12.13.

Τα πειραματικά αποτελέσματα τα οποία θα παρουσιαστούν στο Κεφάλαιο 5, λόγω των τροποποιήσεων και των προσθηκών που έγιναν στα αρχεία του πυρήνα του Linux, όπως θα εξηγηθεί στην ενότητα 4.3, θα έχουν κάποια απόκλιση από τη πραγματικότητα. Για το λόγο αυτό, τη μέτρηση δηλαδή του ποσοστού αστοχίας του μηχανισμού, θα χρησιμοποιήσουμε τους μετρητές από-



δοσης (performance counters) με τη βοήθεια του εργαλείου *perf* [9]. Τα αποτελέσματα από τους performance counters θα χρησιμοποιηθούν επίσης για τις ανάγκες μίας μετρικής που θα αναλυθεί στο υποκεφάλαιο 4.5. Ο πίνακας 4.2 παρουσιάζει το είδος των δεδομένων που εξήχθησαν με χρήση του εργαλείου *perf* καθώς και τα ορίσματα τα οποία δόθηκαν για τη συλλογή τους.

perf Results	Performance Events
# DTLB Load Misses	DTLB_LOAD_MISSES.WALK_COMPLETED
# %Cycles spent in DTLB Load Misses	DTLB_LOAD_MISSES.WALK_ACTIVE
# DTLB Store Misses	DTLB_STORE_MISSES.WALK_COMPLETED
# %Cycles spent in DTLB Store Misses	DTLB_STORE_MISSES.WALK_ACTIVE

Πίνακας 4.2: Αποτελέσματα του εργαλείου *perf* και τα αντίστοιχα performance events [5].

Συγκεκριμένα χρησιμοποιήθηκε η εντολή *perf stat -e* του εργαλείου *perf* για την συλλογή των αποτελεσμάτων. Η εντολή η οποία έδωσε τα τελικά αποτελέσματα φαίνεται στη συνέχεια.

```
$ perf stat -e cpu-cycles , major-faults , re08 -e cpu/event=0x08 , umask=0x10 , cmask=0x01/ -e re49 -e cpu/event=0x49 , umask=0x10 , cmask=0x01/ ./ benchmark_executable benchmark_in/out_args
```

### 4.3 BadgerTrap

Η Μονάδα Διαχείρισης Μνήμης (MMU), όπως αναλύθηκε και σε προηγούμενα κεφάλαια, είναι πολύ σημαντική για τα σύγχρονα υπολογιστικά συστήματα. Περιέχει τη Κρυφή Μνήμη Αναζήτησης Μετάφρασης (TLB) και την MMU Κρυφή Μνήμη. Λόγω της υψηλής επιβάρυνσης που επιφέρουν οι αστοχίες στη TLB, δημιουργήθηκε το BadgerTrap, το οποίο σε αντίθεση με τους υπάρχοντες προσομοιωτές παρέχει μία απευθείας επαφή με τις αστοχίες της TLB.

Το BadgerTrap, το οποίο είναι διαθέσιμο στο [1], είναι υλοποιημένο για αρχιτεκτονική x86-64. Στη πράξη αυτό που κάνει είναι να μετατρέπει κάθε TLB αστοχία σε Σφάλμα Σελίδας (Page Fault), της οποίας η συνάρτηση χειρισμού είναι μία νέα μέθοδος η οποία εισάγεται στο πυρήνα του μηχανήματος. Επομένως, σε κάθε TLB αστοχία καλείται η συνάρτηση αυτή, στην οποία παρέχονται οι απαιτούμενες πληροφορίες ώστε να μπορεί να εντοπιστεί η ακολουθία των σελίδων που προκαλούν αστοχία. Τονίζεται επίσης ότι υπάρχει κατάλληλη συνάρτηση χειρισμού για TLB αστοχίες οι οποίες αφορούν τόσο σελίδες μεγέθους 4KB, όσο και 2MB (Huge Pages). Μια πλήρης παρουσίαση και επεξήγηση των ενεργειών και των τροποποιήσεων που έγιναν για να μπορέσει το BadgerTrap να λειτουργεί ορθά και να είναι αξιόπιστο σε κάθε υπολογιστικό σύστημα με αρχιτεκτονική x86-64 βρίσκεται στο [2]. Θα πρέπει να τονιστεί ότι η εν λόγω διανομή του BadgerTrap, λειτουργεί χωρίς πρόβλημα στο πυρήνα 3.12.13. Στη μελέτη αυτή χρησιμοποιήθηκε ο συγκεκριμένος πυρήνας, ωστόσο χρειάστηκαν μικρές τροποποιήσεις για να επιτελέσει τις λειτουργίες που απαιτούσε η μελέτη αυτή.

Στη διαθέσιμη υλοποίηση του BadgerTrap, οι συναρτήσεις χειρισμού των TLB αστοχιών το μόνο που έκαναν ήταν να μετρούν και τελικά να εμφανίζουν το πλήθος των TLB αστοχιών για καθεμία από τις εφαρμογές που τρέχουν στο υπολογιστικό σύστημα. Παρέχει λοιπόν τη δυνατότητα ενεργοποίησης/απενεργοποίησης, αναλόγα με το αν μία εφαρμογή έχει ενδιαφέρουσα συμπεριφορά και θέλουμε να την μελετήσουμε. Η επέκταση των συναρτήσεων χειρισμού των TLB αστοχιών αφήνεται στον εκάστοτε ενδιαφερόμενο ερευνητή, ώστε να μην υπάρχει κάποιος περιορισμός στις διάφορες υλοποιήσεις.

Στη μελέτη αυτή, οι υλοποιήσεις των διαφόρων Μηχανισμών Προανάκλησης Μεταφράσεων στη TLB τοποθετήθηκαν στις συναρτήσεις χειρισμού που αναφέρθηκαν παραπάνω. Με το τρόπο αυτό, εξετάστηκε η απόδοση των διαφόρων Μηχανισμών Προανάκλησης που περιγράφονται στο Κεφάλαιο 3, για μία ομάδα μετροπρογραμμάτων, τα οποία ανάγκαζαν τη TLB να παρουσιάζει πολλές



αστοχίες και να επιβαρύνει πολύ την απόδοση του συστήματος. Θα πρέπει να τονιστεί πως κάθε Σχήμα TLB Prefetching που υλοποιείται περιέχει την προσομοίωση της MMU Κρυφής Μνήμης με ιεραρχικό τρόπο οργάνωσης τεσσάρων επιπέδων, διότι σε κάθε TLB αστοχία γίνεται αναζήτηση σε αυτές για την εύρεση της εκάστοτε μετάφρασης. Κατά τη διάρκεια της μελέτης ανακαλύφθηκε ότι το BadgerTrap είχε ένα σφάλμα στην υλοποίηση των συναρτήσεων χειρισμού. Συγκεκριμένα, η υλοποίηση περιελάμβανε έλεγχο για τυχόν ύπαρξη διαδοχικών TLB αστοχιών στην ίδια διεύθυνση. Αν κάτι τέτοιο ανιχνευόταν, δεν λαμβανόταν υπόψη στην ανάλυση η συγκεκριμένη αστοχία, αφού στη πράξη δεν είναι δυνατό να συμβεί κάτι τέτοιο, διότι μετά τη πρώτη αστοχία θα μεταφερόταν στη TLB η σελίδα που περιέχει τη διεύθυνση αυτή και την επόμενη φορά θα οδηγούσε σε επιτυχημένη αναζήτηση. Το σφάλμα ήταν λοιπόν ότι ο έλεγχος αυτός δεν θα έπρεπε να γίνεται σε επίπεδο διευθύνσεων αλλά σελίδων αφού η TLB περιέχει σελίδες. Τελικά, το σφάλμα αυτό διορθώθηκε και πλέον ο μηχανισμός παράγει περισσότερο συνεπή αποτελέσματα.

#### 4.4 Μετροπρογράμματα

Για να γίνει σωστή αξιολόγηση της απόδοσης των Σχημάτων Προανάκλησης Μεταφράσεων στη TLB επιλέχθηκαν εφαρμογές από διαφορετικές σουίτες, οι οποίες προκαλούν πολλές TLB αστοχίες για σελίδες μεγέθους 4KB, επιβαρύνοντας έτσι την απόδοση του συστήματος συνολικά. Οι εφαρμογές αυτές έχουν χρησιμοποιηθεί έχουν χρησιμοποιηθεί και απο αντίστοιχες μελέτες στο παρελθόν. Οι σουίτες από τις οποίες επιλέχθηκαν τα εν λόγω μετροπρογράμματα ήταν η σουίτα Parsec [8] και η σουίτα Spec2006 [11]. Τα μετροπρογράμματα canneal και streamcluster ανήκουν στη σουίτα Parsec ενώ τα υπόλοιπα στη Spec2006. Στη συνέχεια, παρατίθενται μία καταγραφή όλων των μετροπρογραμμάτων που χρησιμοποιήθηκαν στη μελέτη αυτή καθώς και μία σύντομη περιγραφή του καθενός.

**473.astar.** Εντάσσεται στη κατηγορία των μετροπρογραμμάτων που έχουν σχέση με παιχνίδια σε υπολογιστές (computer games), Τεχνητή Νοημοσύνη (Artificial Intelligence) και Εύρεση Μονοπατιού (Path Finding). Χρησιμοποιεί μία 2D βιβλιοθήκη η οποία έχει υλοποιημένους τρεις διαφορετικούς αλγόριθμους Εύρεσης Μονοπατιού. Η πρώτη υλοποίηση είναι ο γνωστός αλγόριθμος A\* για χάρτες με διαβατό και μη διαβατό τύπο εδάφους. Η δεύτερη υλοποίηση περιέχει μία τροποποίηση του τρόπου Εύρεσης Μονοπατιού του αλγόριθμου A\* για χάρτες με διαφορετικά είδη εδάφους και ταχύτητα κίνησης. Η τρίτη υλοποίηση αφορά τον A\* αλγόριθμο για γράφους. Τέλος η βιβλιοθήκη αυτή περιέχει κάποιες μεθόδους για το προσδιορισμό των περιοχών των χαρτών. Το μετροπρόγραμμα αυτό θα χρησιμοποιηθεί με δύο διαφορετικές εισόδους, οι οποίες παρουσιάζουν διαφορετική συμπεριφορά γι' αυτό και θα μεταχειριστούν σαν δύο διαφορετικά μετροπρογράμματα. Η πρώτη είσοδος ονομάζεται *rivers* και η δεύτερη *BigLakes2048*.

**436.cactusADM.** Εντάσσεται στη κατηγορία των μετροπρογραμμάτων που έχουν σχέση με τη Φυσική και τη Θεωρία της Σχετικότητας. Είναι ένας συνδυασμός από ένα ελεύθερο περιβάλλον επίλυσης προβλημάτων με όνομα Cactus και ένα υπολογιστικό πυρήνα που αντιπροσωπεύει πολλές εφαρμογές στην Αριθμητική Σχετικότητα. Το μετροπρόγραμμα αυτό λύνει τις εξισώσεις εξέλιξης του Einstein.

**canneal.** Πρόκειται για ένα μετροπρόγραμμα της σουίτας Parsec και έχει αναπτυχθεί από το Πανεπιστήμιο Princeton. Το canneal χρησιμοποιεί fine-grained παραλληλία για ένα lock-free αλγόριθμο, δηλαδή για έναν αλγόριθμο στον οποίο η παύση ενός νήματος δεν προκαλεί αποτυχία σε κάποιο άλλο νήμα. Ταυτόχρονα χρησιμοποιεί ένα πολύ επιθετικό σχήμα συγχρονισμού το οποίο δεν βασίζεται στη τεχνική της αποφυγής.

**459.GemsFDTD.** Εντάσσεται στη κατηγορία των μετροπρογραμμάτων που αφορούν τον Υπολογιστικό Ηλεκτρομαγνητισμό. Λύνει τις εξισώσεις του Maxwell στον τρισδιάστατο χώρο στο πεδίο του χρόνου χρησιμοποιώντας την FDTD μέθοδο. Το 459.Gems είναι ένα υποσύνολο του κώδικα GemsTD.

**429.mcf.** Εντάσσεται στη κατηγορία των μετροπρογραμμάτων που έχουν σχέση με Συνδυαστική Βελτιστοποίηση. Στηρίζεται σε κώδικα που αφορά το προγραμματισμό οχήματος με μία αποθήκη σε μαζικές μεταφορές και περιλαμβάνει σχεδόν κατ' αποκλειστικότητα αριθμητική ακεραίων.

**471.onmetpp.** Εντάσσεται στη κατηγορία των μετροπρογραμμάτων που αφορούν τη Προσομοίωση Διακριτών Συμβάντων. Το 471.omnetpp εκτελεί μία Προσομοίωση Διακριτών Συμβάντων από ένα πολύ μεγάλης έκτασης δίκτυο Ethernet με τη βοήθεια του OMNeT++ προσομοιωτή [6]. Συγκεκριμένα, το δίκτυο Ethernet που προσομοιώνεται αφορά τη ραχοκοκαλιά μιας ολόκληρης Πανεπιστημιούπολης με πολλά μικρότερα δίκτυα LAN. Το μοντέλο περιέχει περίπου 8000 υπολογιστές, 900 διακόπτες και κόμβους και χρησιμοποιεί όλων των ειδών τις τεχνολογίες Ethernet, όπως Gigabit Ethernet, 100Mb full duplex, 100Mb half duplex, 10Mb UTP και 10Mb bus ("thin Ethernet").

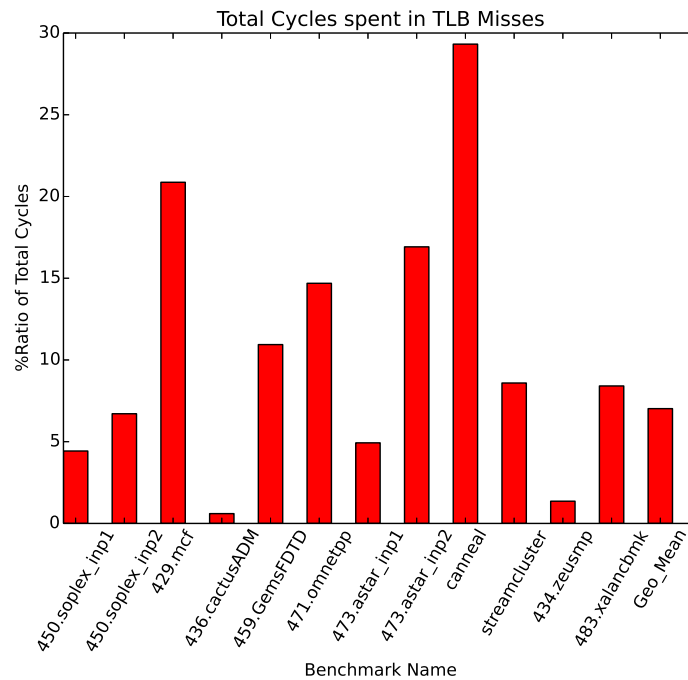
**450.soplex.** Εντάσσεται στη κατηγορία των μετροπρογραμμάτων που αφορούν Γραμμικό Προγραμματισμό (LP). Βασίζεται στο SoPlex έκδοσης 1.2.1 το οποίο χρησιμοποιεί τον αλγόριθμο Simplex για την επίλυση προβλημάτων LP. Το μετροπρόγραμμα αυτό θα χρησιμοποιηθεί με δύο διαφορετικές εισόδους, οι οποίες παρουσιάζουν διαφορετική συμπεριφορά γι' αυτό και θα μεταχειριστούν σαν δύο διαφορετικά μετροπρογράμματα. Η πρώτη είσοδος ονομάζεται *m3500* και η δεύτερη *m45000*.

**streamcluster.** Πρόκειται για ένα μετροπρόγραμμα της σουίτας Parsec που έχει αναπτυχθεί από το Πανεπιστήμιο Princeton. Αφορά το πρόβλημα της ομαδοποίησης (clustering) και είναι ένα πολύ σημαντικό μετροπρόγραμμα λόγω της σημαντικότητας των αλγορίθμων εξόρυξης δεδομένων (data mining) και της επικράτησης προβλημάτων με χαρακτηριστικά ροής.

**483.xalancbmk.** Πρόκειται για ένα μετροπρόγραμμα το οποίο χρησιμοποιεί τον προσομοιωτή Xalan-C++ με έκδοση 1.8 για τη μετατροπή XML αρχείων σε αρχεία HTML, αρχεία κειμένου ή άλλα XML είδη αρχείων.

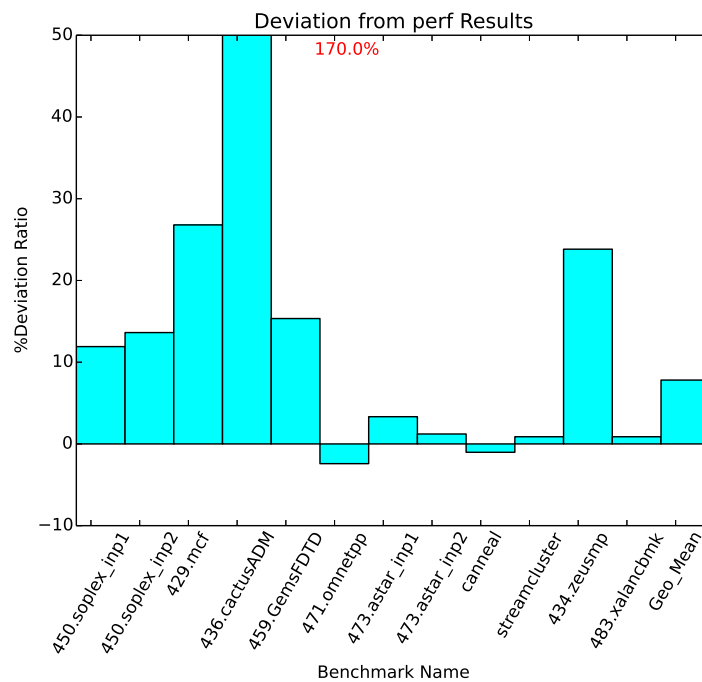
**434.zeusmp.** Εντάσσεται στη κατηγορία των μετροπρογραμμάτων που αφορούν τη Φυσική και τη Μαγνητοϋδροδυναμική. Βασίζεται στο κώδικα ZEUS-MP, ο οποίος αναπτύχθηκε για τη προσομοίωση αστροφυσικών φαινομένων. Στη πράξη λύνει εξισώσεις ιδανικής (μη αντιστάσιμης), μη σχετικιστικής, υδροδυναμικής και μαγνητοϋδροδυναμικής, συμπεριλαμβανομένων των εξωτερικά εφαρμοζόμενων βαρυτικών πεδίων. Το πρόβλημα Φυσικής το οποίο λύνει είναι ένα προσομοιωμένο τρισδιάστατο blastwave με την παρουσία ενός ομοιόμορφου μαγνητικού πεδίου κατά μήκος της κατεύθυνσης του άξονα x, χρησιμοποιώντας ένα καρτεσιανό πλέγμα.

Περισσότερες λεπτομέρειες για τα παραπάνω μετροπρογράμματα παρέχονται στα [7, 10]. Στο Σχήμα 4.1, παρουσιάζονται τα αποτελέσματα τα οποία έδωσε το εργαλείο perf και έχουν να κάνουν με το ποσοστό των κύκλων που ξοδεύουν τα διάφορα μετροπρογράμματα στη συνάρτηση χειρισμού των TLB αστοχιών.

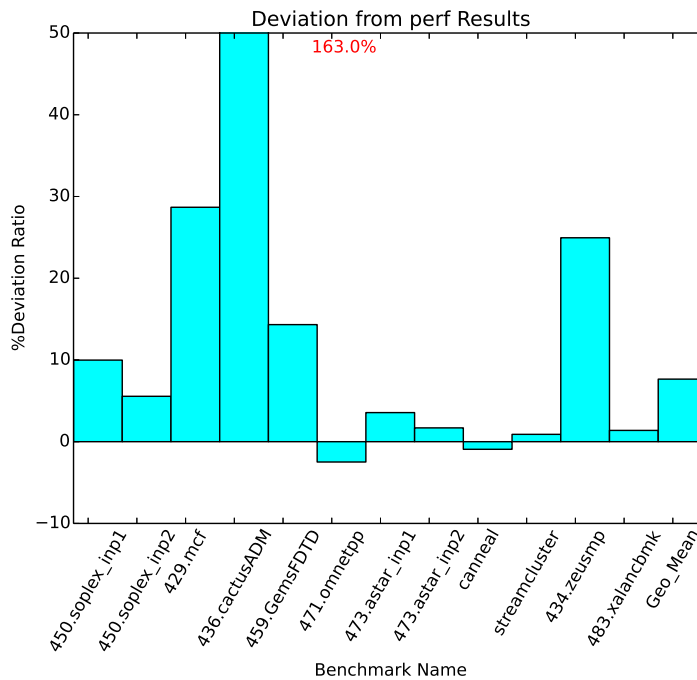


Σχήμα 4.1: Ποσοστό κύκλων που ξοδεύονται για TLB αστοχίες, σύμφωνα με τα αποτελέσματα του εργαλείου perf.

Το BadgerTrap, το οποίο χρησιμοποιούμε για τη προσομοίωση των TLB αστοχιών έχει κάποιες αποκλίσεις από τα πραγματικά νούμερα. Το Σχήμα 4.2 δείχνει το ποσοστό λάθους του BadgerTrap όσο αφορά το πλήθος των TLB αστοχιών που εντοπίζονται, χωρίς την εισαγωγή κάποιου από τα Σχήματα Προανάκλησης Μεταφράσεων. Το Σχήμα 4.3 δείχνει το ποσοστό λάθους μετά την εισαγωγή ενός Σχήματος Προανάκλησης Μεταφράσεων, από αυτά που υλοποιήθηκαν. Η σύγκριση γίνεται με βάση τα αποτελέσματα που έδωσαν οι performance counters.



Σχήμα 4.2: Ποσοστό λάθους BadgerTrap όσο αφορά το πλήθος των TLB αστοχιών, χωρίς κανένα Σχήμα Προανάκλησης Μεταφράσεων.



**Σχήμα 4.3:** Ποσοστό λάθους BadgerTrap όσο αφορά το πλήθος των TLB αστοχιών, μετά την εισαγωγή κάποιου Σχήματος Προανάκλησης Μεταφράσεων.

Τα Σχήματα 4.1, 4.2 και 4.3 δείχνουν αρχικά πως το benchmark 436.cactusADM δεν ξοδεύει μεγάλο ποσοστό των κύκλων του σε TLB αστοχίες και το ποσοστό λάθους του BadgerTrap είναι πολύ υψηλό. Ο λόγος για τον οποίο επιλέχθηκε να μην εξαιρεθεί από τη μελέτη αυτή είναι ότι πολλές παλαιότερες μελέτες έχουν δείξει ότι το συγκεκριμένο benchmark σε μηχανήματα παλαιότερης γενιάς προκαλούσε αρκετά προβλήματα στη TLB. Λόγω των μεγάλων αποκλίσεων που παρουσιάζει τα συμπεράσματα που εξάγονται από αυτό θα πρέπει να αντιμετωπιστούν επιφυλακτικά. Αν εξαιρεθεί το συγκεκριμένο benchmark, τα υπόλοιπα δείχνουν να έχουν μικρές αποκλίσεις από τη πραγματικότητα, όσο αφορά τις TLB αστοχίες και να ξοδεύουν ένα ικανοποιητικό ποσοστό των κύκλων τους στις TLB αστοχίες που παρουσιάζουν. Τέλος, θα πρέπει να τονιστεί ότι η εισαγωγή του κώδικα προσομοίωσης για την αξιολόγηση των TLB Prefetching Σχημάτων δεν επηρεάζει τα αποτελέσματα, όπως φαίνεται και στα Σχήματα 4.2 και 4.3.

## 4.5 Μετρικές Αξιολόγησης

Για την αξιολόγηση της χρησιμότητας των Σχημάτων Προανάκλησης Μεταφράσεων στη TLB, τα οποία αναλύθηκαν στο κεφάλαιο 3, θα χρησιμοποιηθούν κάποιες μετρικές. Στη συνέχεια γίνεται μία παρουσίαση των μετρικών που χρησιμοποιήθηκαν.

1. **Μετρική Απόδοσης.** Η μετρική της απόδοσης είναι πολύ σημαντική, αφού αναπαριστά το ποσοστό των αστοχιών της TLB που προβλέφθηκαν σωστά από το εκάστοτε Σχήμα Προανάκλησης Μεταφράσεων.
2. **Μετρική Ακρίβειας.** Η μετρική της ακρίβειας είναι μία μετρική που αναπαριστά το ποσοστό των προβλέψεων που έγιναν από το εκάστοτε Σχήμα Προανάκλησης Μεταφράσεων και οδήγησαν σε επιτυχημένες αναζητήσεις στον Πίνακα Προβλέψεων. Δείχνει δηλαδή κατά πόσο το Σχήμα Προανάκλησης Μεταφράσεων που χρησιμοποιείται κάνει ακριβείς προβλέψεις.
3. **Μετρική Προβλέψεων Μηδενικού Κόστους.** Η μετρική αυτή είναι μία νέα μετρική, η οποία αναπτύχθηκε για τα δεδομένα της εν λόγω μελέτης. Αναπαριστά το ποσοστό των προβλέψεων

του εκάστοτε Σχήματος Προανάκλησης Μεταφράσεων οι οποίες είναι μηδενικού κόστους από άποψη επιπρόσθετων λειτουργιών μνήμης για τη μεταφορά τους στο Πίνακα Προβλέψεων. Αυτό που προσπαθεί να ανακαλύψει η μετρική αυτή είναι το κατά πόσο καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων περιέχει μέσα του την έννοια που εισάγει ο Προανακλητής Μηδενικού Κόστους 3.3.4. Η μετρική αυτή μπορεί να αναιρέσει ή να ενισχύσει τα αποτελέσματα της Μετρικής Ακρίβειας, αφού είναι πιθανό το ενδεχόμενο κάποιο Σχήμα Προανάκλησης Μεταφράσεων να έχει χαμηλό ποσοστό ακρίβειας αλλά υψηλό ποσοστό προβλέψεων μηδενικού κόστους, γεγονός το οποίο σημαίνει ότι παρά τις πολλές προβλέψεις που κάνει δεν επιβαρύνει πολύ το σύστημα, παρά το χαμηλό ποσοστό ακρίβειας που παρουσιάζει.

4. *Μετρική Λειτουργιών Μνήμης.* Πρόκειται για μία νέα μετρική, η οποία αναπτύχθηκε στα πλαίσια της εν λόγω μελέτης. Αρχικά θεωρεί σαν βάση της τον αριθμό των απαιτούμενων λειτουργιών μνήμης που θα προκαλούσε καθένα από τα μετροπρόγραμματα που χρησιμοποιήθηκαν, αν δεν υπήρχε κανένα Σχήμα Προανάκλησης Μεταφράσεων για την εύρεση των μεταφράσεων που δεν υπάρχουν στη TLB. Χωρίζει τον αριθμό αυτό σε τέσσερις κατηγορίες, ανάλογα με το αν η εύρεση της προς αναζήτηση μετάφρασης ικανοποιήθηκε από κάποιο επίπεδο της MMU Κρυφής Μνήμης ή αν χρειάστηκαν και οι τέσσερις λειτουργίες μνήμης για την εύρεση της. Στη συνέχεια παρουσιάζει την αλλαγή που προκλήθηκε από την εισαγωγή κάποιου από τα Σχήματα Προανάκλησης Μεταφράσεων. Δείχνει το κατά πόσο επηρεάστηκαν οι απαιτούμενες λειτουργίες μνήμης από τις προβλέψεις που έγιναν, καθώς και την επιβάρυνση που δέχτηκε το σύστημα από τις λειτουργίες μνήμης που χρειάστηκαν για να πραγματοποιηθούν οι προβλέψεις. Τόσο οι λειτουργίες μνήμης λόγω αστοχιών της TLB όσο και οι λειτουργίες μνήμης λόγω προβλέψεων χωρίζονται σε τέσσερις κατηγορίες όπως αναλύθηκε παραπάνω.
5. *Γραμμικό Μοντέλο Επιτάχυνσης.* Στη μετρική αυτή θα εξετάσουμε τη μεταβολή που επιφέρει η απόδοση ενός Σχήματος Προανάκλησης Μεταφράσεων στη TLB, στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark σε TLB αστοχίες. Για τη μετρική αυτή θα χρησιμοποιηθούν τα αποτελέσματα της Μετρικής Απόδοσης που αναλύθηκε παραπάνω και τα αποτελέσματα που παρουσιάζονται στο Σχήμα 4.1 και αφορούν το εργαλείο perf. Θα υπολογιστεί λοιπόν ένα νέο ποσοστό κύκλων που ξοδεύονται σε TLB αστοχίες, για κάθε benchmark, το οποίο θα είναι προφανώς μικρότερο από το αρχικό λόγω των διαφόρων Μηχανισμών Προανάκλησης Μεταφράσεων. Η εξίσωση που δίνει το νέο ποσοστό κύκλων φαίνεται στη συνέχεια.

$$\%CYCLES_{new} = \%CYCLES_{old} * \left(1 - \frac{PrefetchBufferHits}{TotalDTLBMisses}\right) \quad (4.1)$$



## Κεφάλαιο 5

# Αξιολόγηση

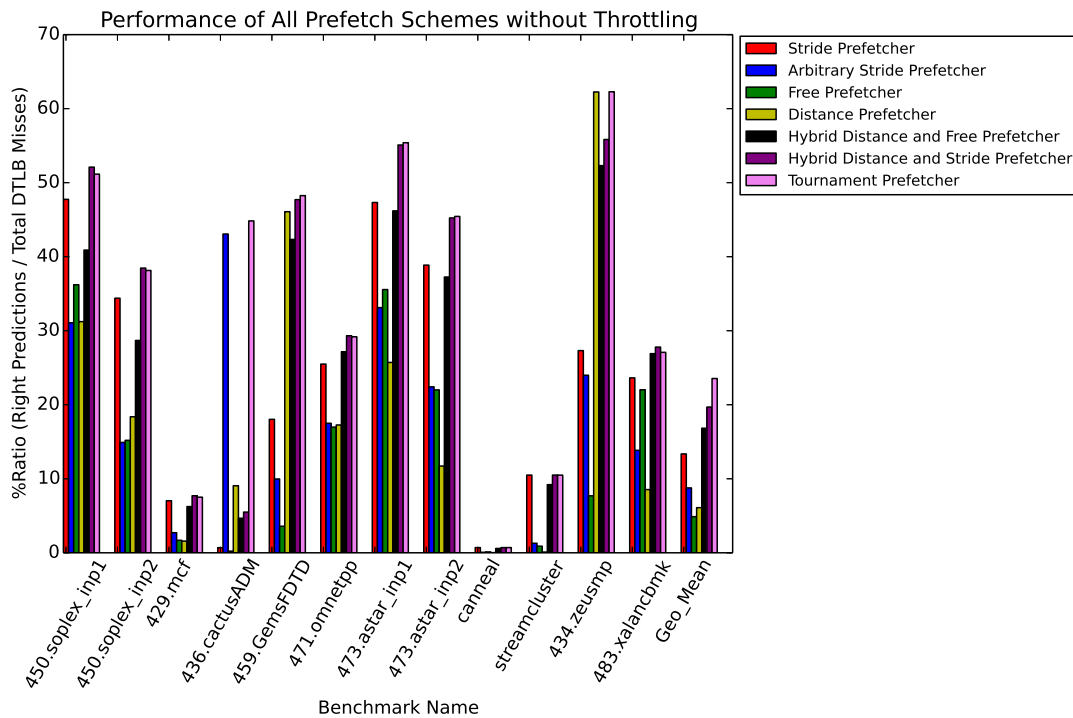
### 5.1 Εισαγωγή

Αντικείμενο του κεφαλαίου αυτού είναι η παρουσίαση των αποτελεσμάτων για τις διάφορες Τεχνικές Προανάκλησης Μεταφράσεων στη TLB, οι οποίες αναλύθηκαν εκτενώς στο Κεφάλαιο 3. Τα αποτελέσματα που θα παρουσιαστούν και θα σχολιαστούν αφορούν τα μετροπρογράμματα, τα οποία παρουσιάστηκαν στο υποκεφάλαιο 4.4. Για να είναι συνεπή τα αποτελέσματα που θα παρουσιαστούν στη συνέχεια και να εξαχθούν χρήσιμα συμπεράσματα, κρίνεται αναγκαίο το μέγεθος του Πίνακα Προβλέψεων να είναι ίδιο για όλα τα Σχήματα Προανάκλησης Μεταφράσεων. Επομένως για τα παρακάτω αποτελέσματα θεωρείται ότι ο Πίνακας Προβλέψεων είναι ένας 16 καταχωρήσεων μονοδιάστατος πίνακας. Κάποια από τα Σχήματα Προανάκλησης Μεταφράσεων χρησιμοποιούν ένα Πίνακα Αποστάσεων (Distance Table). Για τον ίδιο λόγο, όσον αφορά τα παρακάτω αποτελέσματα, θεωρούμε ένα Πίνακα Αποστάσεων άμεσης χαρτογράφησης (direct-mapped) με 16 καταχωρήσεις, καθεμία από τις οποίες διαθέτει δύο καταχωρήσεις για προβλέψεις. Η πολιτική αντικατάστασης του Πίνακα Προβλέψεων και του Πίνακα Αποστάσεων είναι η λιγότερο πρόσφατα χρησιμοποιούμενη (LRU). Η Ιεραρχική Οργάνωση των MMU Κρυφών Μνημών που προσομοιώνονται διαθέτει τρία επίπεδα, το καθένα από τα οποία διαθέτει μια MMU Cache. Η MMU Cache του επιπέδου PML4 έχει δύο (2) καταχωρήσεις, του επιπέδου PDP τέσσερις (4) και του επιπέδου PD δεκαέξι (16) καταχωρήσεις καθεμία από τις οποίες διαθέτει δύο θέσεις για καταχωρήσεις. Όσον αφορά τους Μηχανισμούς Ελέγχου Προανάκλησης στις υλοποιήσεις των Hit Interval Throttling και MMU Interval Throttling θεωρούμε ότι η διάρκεια της περιόδου εκμάθησης είναι 10K TLB αστοχίες και η διάρκεια της περιόδου εφαρμογής 100K TLB αστοχίες. Ο Μηχανισμός Hit Interval Throttling για να επιτρέψει την πραγματοποίηση προβλέψεων σε περίοδο εφαρμογής θα πρέπει η προηγούμενη περίοδος εκμάθησης να έκανε προβλέψεις με ποσοστό ευστοχίας πάνω από 25%. Ο Μηχανισμός MMU Interval Throttling για να επιτρέψει την πραγματοποίηση προβλέψεων σε περίοδο εφαρμογής έθεσε σαν κατώφλια τα ποσοστά 25%, 12.5%, 6.125%, 6.125% και 6.125% για προβλέψεις με κόστος μία, δύο τρεις, τέσσερις ή καμία (μηδενικού κόστους) λειτουργίες μνήμης αντίστοιχα. Για την αξιολόγηση των αποτελεσμάτων χρησιμοποιήθηκαν οι μετρικές που αναλύθηκαν στο υποκεφάλαιο 4.5. Τέλος, πρέπει να τονιστεί ότι καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων προσομοιώνει την ύπαρξη MMU Κρυφής Μνήμης, με Ιεραρχικό Τρόπο Οργάνωσης 3 επιπέδων όπως ακριβώς αναλύθηκε στο υποκεφάλαιο 2.2.3.

### 5.2 Σύγκριση Αποτελεσμάτων

Στην ενότητα αυτή θα γίνει σύγκριση όλων των Σχημάτων Προανάκλησης Μεταφράσεων (Κεφάλαιο 3) για τις μετρικές που αναλύονται στο 4.4. Θα πρέπει να τονιστεί ότι η σύγκριση αφορά τα αποτελέσματα που έδωσαν τα διάφορα Σχήματα Προανάκλησης Μεταφράσεων **χωρίς την ύπαρξη κάποιου Μηχανισμού Ελέγχου Προανάκλησης (No Throttling)**. Για το σκοπό αυτό δημιουργήθηκαν τα παρακάτω γραφήματα.



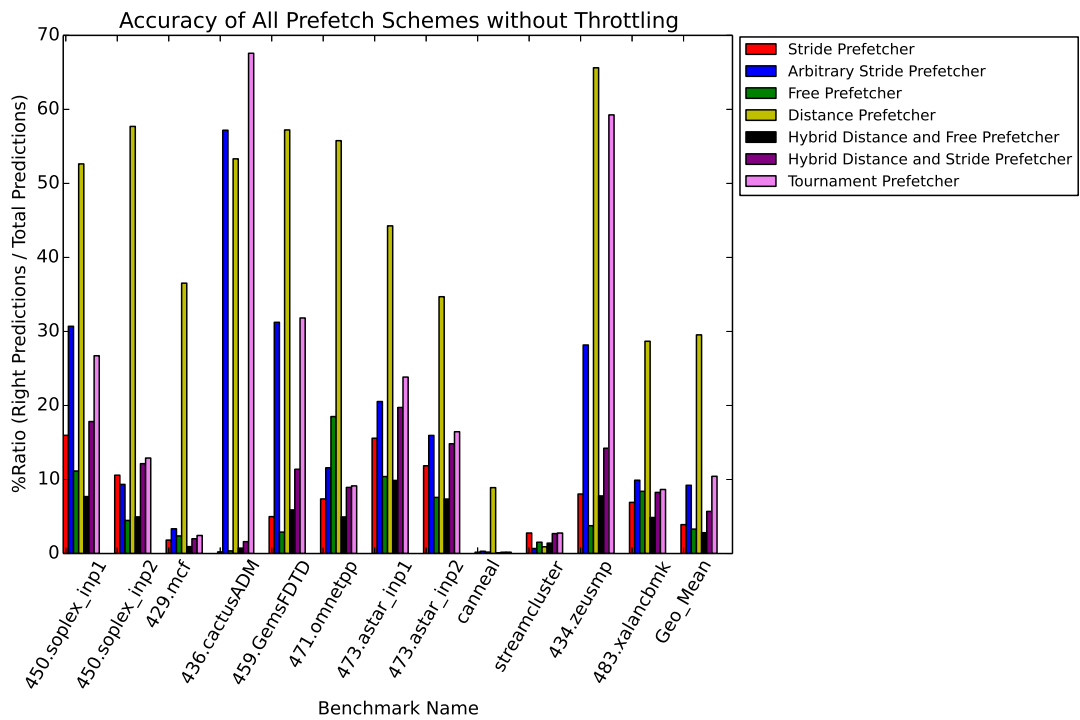


Σχήμα 5.1: Ποσοστό Επιτυχημένων Προβλέψεων για καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων.

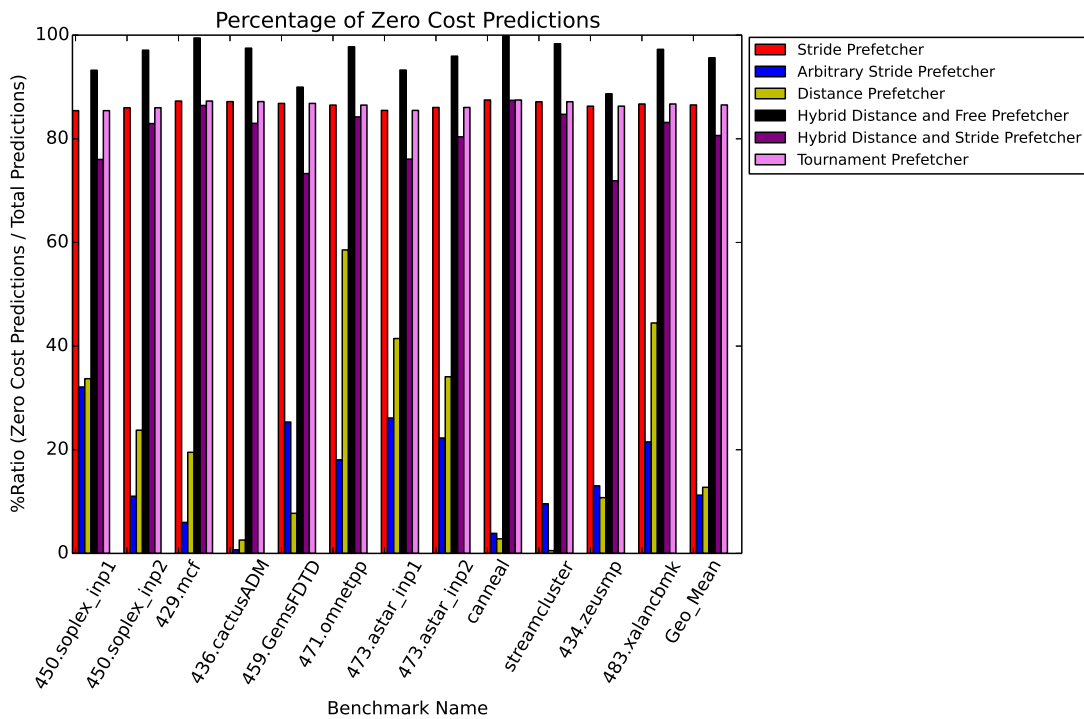
Το Σχήμα 5.1 δείχνει ότι κάποια μετροπρογράμματα παρουσιάζουν καλή απόδοση με Ακολουθιακά Σχήματα Προανάκλησης, κάποια με σχήματα που λαμβάνουν υπόψη την απόσταση διαδοχικών αστοχιών της TLB αλλά σχεδόν κανένα δεν παρουσιάζει εξίσου καλή απόδοση και με τα δύο ταυτόχρονα. Αυτό φανερώνει ότι από άποψη απόδοσης οι δύο αυτές ιδέες Προανάκλησης είναι ορθογώνιες και θα μπορούσαν να συνδυαστούν σε ένα κοινό Σχήμα Προανάκλησης Μεταφράσεων. Αυτό οδήγησε στην ανάπτυξη των Υβριδικών Μηχανισμών Προανάκλησης. Απο άποψη απόδοσης, ο Υβριδικός Προανακλητής Μηδενικού Κόστους φαίνεται να συνδυάζει καλά τους δύο μηχανισμούς που τον αποτελούν μόνο όταν και οι δύο παρουσιάζουν μία σχετικά καλή απόδοση ατομικά. Υπάρχουν μετροπρογράμματα στα οποία ο Προανακλητής Απόστασης δίνει πολύ καλά αποτελέσματα απόδοσης αλλά ο Προανακλητής Μηδενικού Κόστους λόγω της επιθετικότητας που τον διακρίνει όσο αφορά τις προβλέψεις, συχνά διώχνει προβλέψεις από το Πίνακα Προβλέψεων οι οποίες θα έφερναν επιτυχίες και τις αντικαθιστεί με δικές του που δεν θα φέρουν θετικά αποτελέσματα όσο αφορά την απόδοση. Χαρακτηριστικά παραδείγματα είναι τα μετροπρογράμματα 436.cactusADM, 459.GemsFDTD και 434.zeusmp. Κάτι τέτοιο δεν συμβαίνει με τον Υβριδικό Ακολουθιακό Προανακλητή αφού φαίνεται να συνδυάζει καλά τους μηχανισμούς που τον αποτελούν ακόμα και όταν κάποιος από τους δύο δεν παρουσιάζει καλή απόδοση. Μοναδική εξαίρεση αποτελεί το μετροπρόγραμμα 434.zeusmp, το οποίο δείχνει ότι θα ήταν καλύτερο να είχε απενεργοποιηθεί ο Ακολουθιακός Προανακλητής ώστε τα αποτελέσματα να ήταν τα καλύτερα δυνατά. Ο μοναδικός Μηχανισμός Προανάκλησης ο οποίος εκμεταλλεύεται σε κάθε μετροπρόγραμμα το κατάλληλο Σχήμα Προανάκλησης φαίνεται να είναι ο Tournament Προβλέπτης. Καταφέρνει να επιλέγει κάθε φορά το καταλληλότερο και να εκμεταλλεύεται όσο το δυνατόν καλύτερα τους άλλους δύο μηχανισμούς για να πάρει το καλύτερο δυνατό αποτέλεσμα. Δεν παρουσιάζει ποτέ χαμηλότερη απόδοση από το καλύτερο από τα τρία Σχήματα Προανάκλησης που τον αποτελούν και τις περισσότερες φορές τα ξεπερνά, εκμεταλλευόμενος και τα άλλα δύο Σχήματα Προανάκλησης. Στη γενική περίπτωση καταφέρνει να αποδώσει κατά 25% καλύτερα. Βέβαια υπάρχουν και μετροπρογράμματα, όπως το canneal, των οποίων οι αναφορές στη μνήμη δεν μπορούν να προβλεφθούν σωστά από κανένα Σχήμα Προανάκλησης. Στη περίπτωση αυτή σκοπός είναι να μη χειροτερέψει η απόδοση του συστήματος λόγω της Προανάκλησης Μεταφράσεων, γεγονός το οποίο θα



αναλυθεί στο σχολιασμό του Σχήματος 5.4.

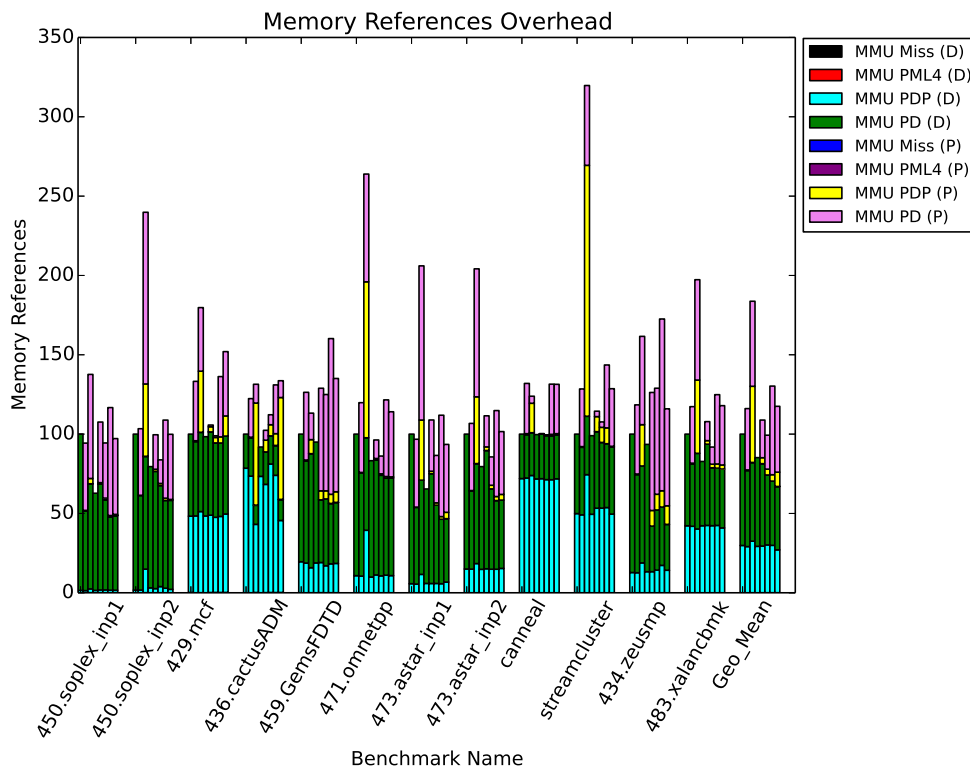


Σχήμα 5.2: Ποσοστό Ακρίβειας Προβλέψεων για καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων.



Σχήμα 5.3: Ποσοστό Προβλέψεων Μηδενικού Κόστους για καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων.

Για να βγούν ασφαλή και σωστά συμπεράσματα όσο αφορά την ευστοχία κάθε Μηχανισμού Προανάκλησης, τα αποτελέσματα των Σχημάτων 5.2, 5.3 θα πρέπει να εξεταστούν παράλληλα. Ο λόγος είναι ότι μπορεί το Σχήμα 5.2 να δείχνει ότι ένας Μηχανισμός Προανάκλησης έχει χαμηλό ποσοστό ακρίβειας προβλέψεων, αλλά αν ληφθεί υπόψη ότι το μεγαλύτερο ποσοστό των προβλέψεων ήταν Μηδενικού Κόστους από το Σχήμα 5.3 τότε το ποσοστό του Σχήματος 5.2 χάνει τη σημαντικότητα του. Παραδείγματα τέτοιων μηχανισμών είναι ο Ακολουθιακός Προανακλητής, ο Υβριδικός Προανακλητής Μηδενικού Κόστους, ο Υβριδικός Ακολουθιακός Προανακλητής και ο Tournament Προβλέπτης. Ωστόσο υπάρχουν Σχήματα Προανάκλησης όπως ο Προανακλητής Τυχαίου Βήματος και ο Προανακλητής Απόστασης οι οποίοι λόγω της ιδέας που υλοποιούν είναι καταδικασμένοι να κάνουν προβλέψεις, οι οποίες με μικρή πιθανότητα θα είναι Μηδενικού Κόστους. Στη περίπτωση αυτή, το Σχήμα 5.2 δίνει ασφαλή συμπεράσματα ακρίβειας. Παρατηρείται ότι ο Προανακλητής Απόστασης έχει σχετικά υψηλό ποσοστό ακρίβειας ενώ ο Προανακλητής Τυχαίου Βήματος όχι. Το γεγονός αυτό θα αποτυπωθεί και στην επιβάρυνση λειτουργιών μνήμης που δείχνει το Σχήμα 5.4. Τέλος, τα υψηλά ποσοστά προβλέψεων Μηδενικού Κόστους που παρουσιάζουν ο Υβριδικός Προανακλητής Μηδενικού Κόστους, ο Υβριδικός Ακολουθιακός Προανακλητής και ο Tournament Προανακλητής οφείλονται στην ύπαρξη του Προανακλητή Μηδενικού Κόστους και του Ακολουθιακού Προανακλητή. Ο Προανακλητής Μηδενικού Κόστους παράγει πάντα προβλέψεις μηδενικού κόστους και οι προβλέψεις του Ακολουθιακού Προανακλητή είναι με μεγάλη πιθανότητα Μηδενικού Κόστους. Το γεγονός αυτό σε συνδυασμό με το ότι παράγουν περισσότερες προβλέψεις από τους υπόλοιπους μηχανισμούς οδηγεί σε αυτά τα υψηλά ποσοστά.



**Σχήμα 5.4:** Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης για καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα κάθε Σχήματος Προανάκλησης Μεταφράσεων. Η σειρά παρουσίασης τους είναι: Ακολουθιακός Προανακλητής, Προανακλητής Τυχαίου Βήματος, Προανακλητής Μηδενικού Κόστους, Προανακλητής Απόστασης, Υβριδικός Προανακλητής Μηδενικού Κόστους, Υβριδικός Ακολουθιακός Προανακλητής και Tournament Προβλέπτης. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις.

Το Σχήμα 5.4 αρχικά επιβεβαιώνει το κακό ποσοστό ευστοχίας του Προανακλητή Τυχαίου Βήματος σε συνδυασμό με το χαμηλό ποσοστό προβλέψεων Μηδενικού Κόστους. Ο προβλέπτης αυτός εισάγει μεγάλη επιβάρυνση στο σύστημα και όπως φαίνεται και από το Σχήμα 5.1 δεν προσφέρει κάποιο ιδιαίτερο ποσοστό απόδοσης. Εξαιρέση αποτελεί το μετροπρόγραμμα 436.cactusADM, στο οποίο είναι ο μοναδικός Προανακλητής που καταφέρνει να έχει καλή απόδοση με σχετικά μικρή επιβάρυνση σε λειτουργίες μνήμης. Αυτός είναι και ο λόγος εισαγωγής του στο Tournament Προβλέπτη. Αξίζει να παρατηρηθεί ότι η επιβάρυνση που εισάγεται στα μετροπρογράμματα που παρουσιάζουν καλή απόδοση είτε με Ακολουθιακά Σχήματα Πρόβλεψης είτε με χρήση του Προανακλητή Απόστασης δεν είναι μεγάλη και σε πολλές περιπτώσεις φαίνεται να αξίζει η εισαγωγή του αφού παρατηρούνται υψηλά ποσοτά απόδοσης. Χαρακτηριστικά παραδείγματα είναι το 450.soplex και με τις δύο διαφορετικές εισόδους, το 473.astar και με τις δύο διαφορετικές εισόδους, 434.zeusmp και 459.GemsFDTD, κυρίως με χρήση του Tournament Προβλέπτη. Το Σχήμα 5.4 φανερώνει και ένα από τα αρνητικά στοιχεία που μπορεί να παρουσιάσει η τεχνική της Προανάκλησης Μεταφράσεων. Κανένα από τα Σχήματα Προανάκλησης δεν μπορεί να πετύχει καλά ποσοστά απόδοσης κυρίως για το μετροπρόγραμμα canneal αλλά και για τα streamcluster και 429.mcf. Από το Σχήμα 5.4 φαίνεται ότι κανένα από τα Σχήματα Προανάκλησης δεν αντιλαμβάνεται την κακή απόδοση του και συνεχίζει να κάνει άχρηστες προβλέψεις επιβαρύνοντας με επιπλέον λειτουργίες μνήμης το σύστημα. Αυτό το αρνητικό χαρακτηριστικό καθώς και την προσπάθεια αποφυγής άχρηστων προβλέψεων κατά τη διάρκεια της εκτέλεσης κάποιας εφαρμογής, οδήγησε στην ανάγκη ανάπτυξης των μηχανισμών που αναλύθηκαν στο υποκεφάλαιο 3.6, τα αποτελέσματα των οποίων θα αναλυθούν στη συνέχεια.

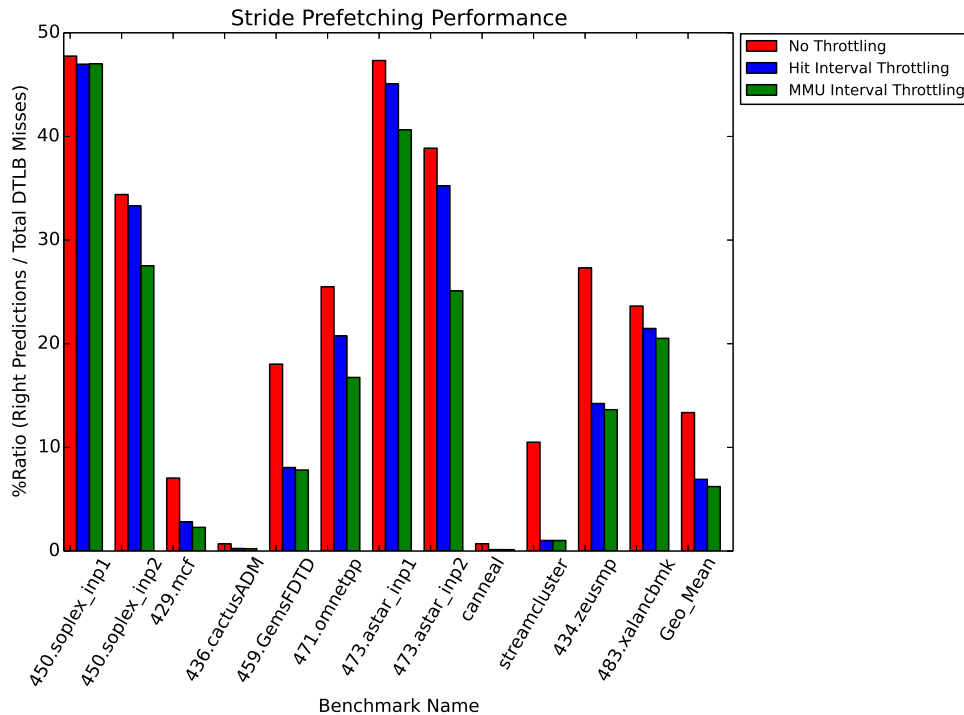
	perf	Stride	Arbitrary Stride	Free	Distance	Hybrid Free	Hybrid Stride	Tournament
450.soplex_inp1	4.43%	2.31%	3.05%	2.83%	3.05%	2.62%	2.12%	2.16%
450.soplex_inp2	6.71%	4.40%	5.71%	5.69%	5.48%	4.78%	4.13%	4.15%
429.mcf	20.87%	19.40%	20.30%	20.52%	20.54%	19.57%	19.26%	19.30%
436.cactusADM	0.60%	0.60%	0.34%	0.60%	0.55%	0.57%	0.57%	0.33%
459.GemsFDTD	10.94%	8.97%	9.85%	10.55%	5.90%	6.31%	5.72%	5.66%
471.omnetpp	14.69%	10.94%	12.12%	12.20%	12.15%	10.70%	10.38%	10.40%
473.astar_inp1	4.93%	2.60%	3.30%	3.18%	3.66%	2.65%	2.21%	2.20%
473.astar_inp2	16.92%	10.34%	13.13%	13.20%	14.94%	10.62%	9.27%	9.23%
canneal	29.32%	29.11%	29.30%	29.28%	29.31%	29.15%	29.11%	29.11%
streamcluster	8.59%	7.69%	8.48%	8.51%	8.58%	7.80%	7.69%	7.69%
434.zeusmp	1.36%	0.99%	1.03%	1.26%	0.51%	0.65%	0.60%	0.51%
483.xalancbmk	8.41%	6.42%	7.25%	6.56%	7.69%	6.15%	6.07%	6.13%
Geo_Mean	7.02%	6.08%	6.41%	6.68%	6.59%	5.84%	5.64%	5.37%

**Πίνακας 5.1:** Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης για καθένα από τα Σχήματα Προανάκλησης Μεταφράσεων. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf.

Ο Πίνακας 5.1 αποτυπώνει όλα όσα σχολιάστηκαν στο Σχήμα 5.1, δηλαδή πως ο προανακλητής με την καλύτερη απόδοση καταφέρνει να μειώσει περισσότερο το ποσοστό των κύκλων που απαιτούνται για τις TLB αστοχίες. Ο Tournament Προβλέπτης είναι εκείνος που καταφέρνει την μεγαλύτερη μείωση, η οποία στη γενική περίπτωση φτάνει το 23%, όπως μαρτυρά ο γεωμετρικός μέσος όρος, αφού από 7.02% το ποσοστό των κύκλων πέφτει σε 5.37%. Κοντά στα αποτελέσματα αυτά είναι και ο Υβριδικός Προανακλητής Απόστασης, ο οποίος σε κάποιες περιπτώσεις καταφέρνει και μεγαλύτερη μείωση από τον Tournament Προβλέπτη. Μικρή μείωση καταφέρνει ο Προανακλητής Μηδενικού Κόστους, αλλά είναι ικανοποιητική λόγω της απλότητας του. Το ίδιο συμβαίνει και με τον Προανακλητή Απόστασης, ωστόσο εκείνος μειώνει κατά σχεδόν 50% το ποσοστό του benchmark 459.GemsFDTD. Ο Προανακλητής Τυχαίου Βήματος παρότι πετυχαίνει καλή απόδοση στο benchmark 436.cactusADM, δεν πετυχαίνει υψηλή μείωση στη γενική περίπτωση αφού το benchmark αυτό έχει πολύ μικρό ποσοστό.

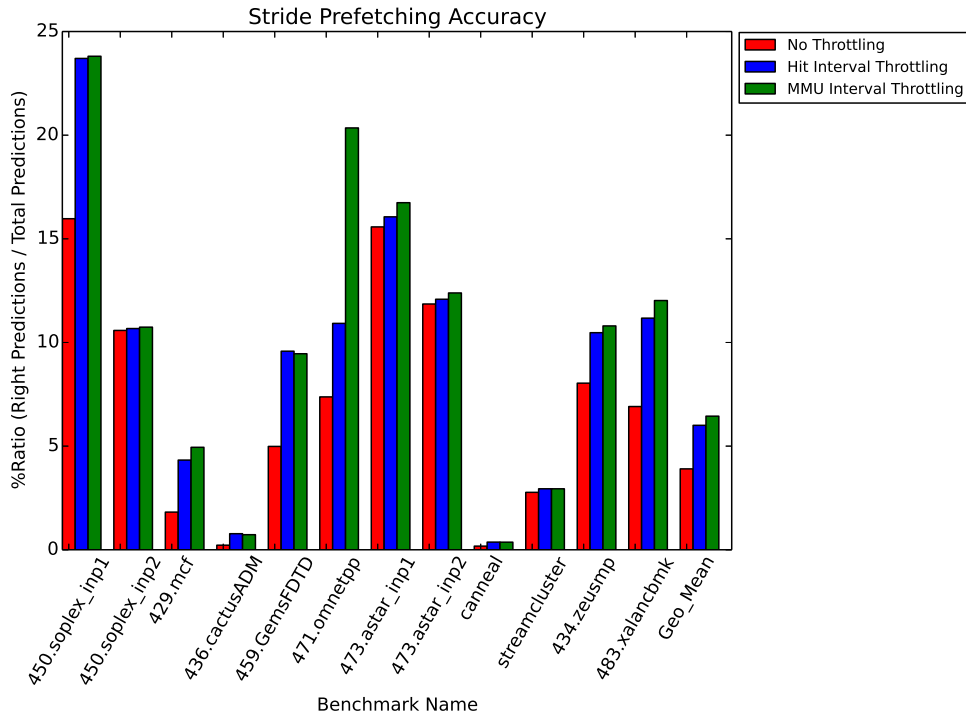
### 5.3 Αποτελέσματα Ακολουθιακού Προανακλητή

Στην ενότητα αυτή θα παρουσιαστούν τα αποτελέσματα του Ακολουθιακού Προανακλητή για τις μετρικές που αναλύθηκαν στο υποκεφάλαιο 4.5. Θα παρουσιαστούν επίσης τα αποτελέσματα της εισαγωγής των Μηχανισμών Hit Interval Throttling και MMU Interval Throttling.

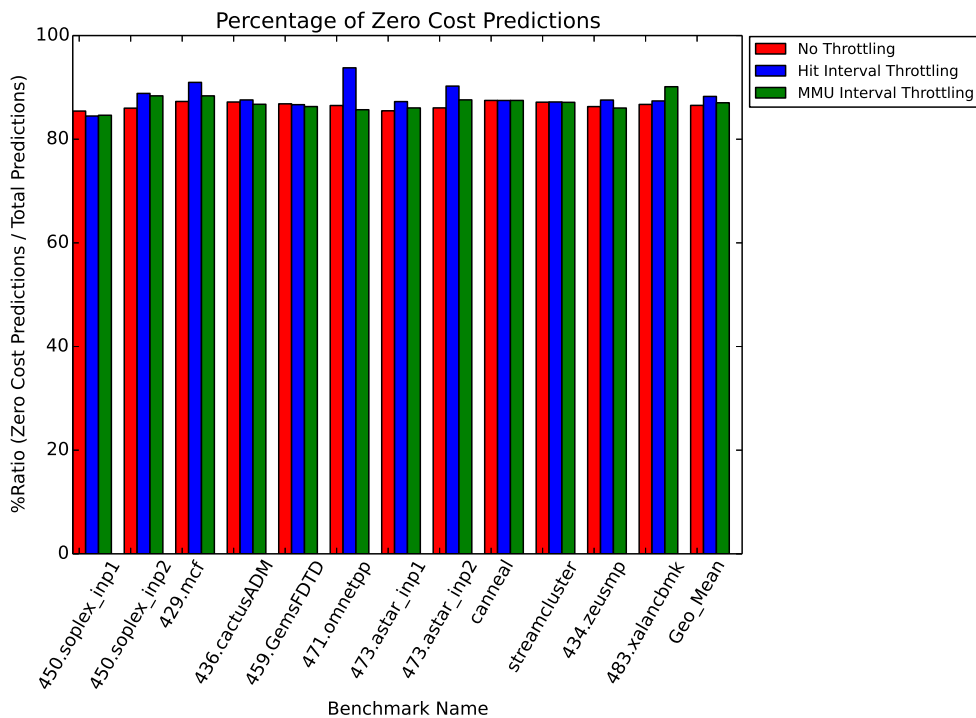


Σχήμα 5.5: Ποσοστό Επιτυχημένων Προβλέψεων Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.

Το Σχήμα 5.5 δείχνει ότι ο No Throttling μηχανισμός του Ακολουθιακού Προανακλητή δίνει τα καλύτερα αποτελέσματα απόδοσης. Μπορεί ο μηχανισμός Hit Interval Throttling να λειτουργεί κάπως καλύτερα από τον MMU Interval Throttling αλλά δεν καταφέρνει κανένας από αυτούς να δώσει αποτελέσματα ανάλογα με αυτά του No Throttling. Απορρίπτουν δηλαδή πολλές προβλέψεις οι οποίες θα έπρεπε να γίνουν και η αιτία αυτού είναι τα κατώφλια κάτω από τα οποία απενεργοποιούν το Σχήμα Προανάκλησης Μεταφράσεων. Η επιλογή καταλληλότερων κατωφλίων θα έδινε αποτελέσματα πιο κοντά στο Μηχανισμό No Throttling.

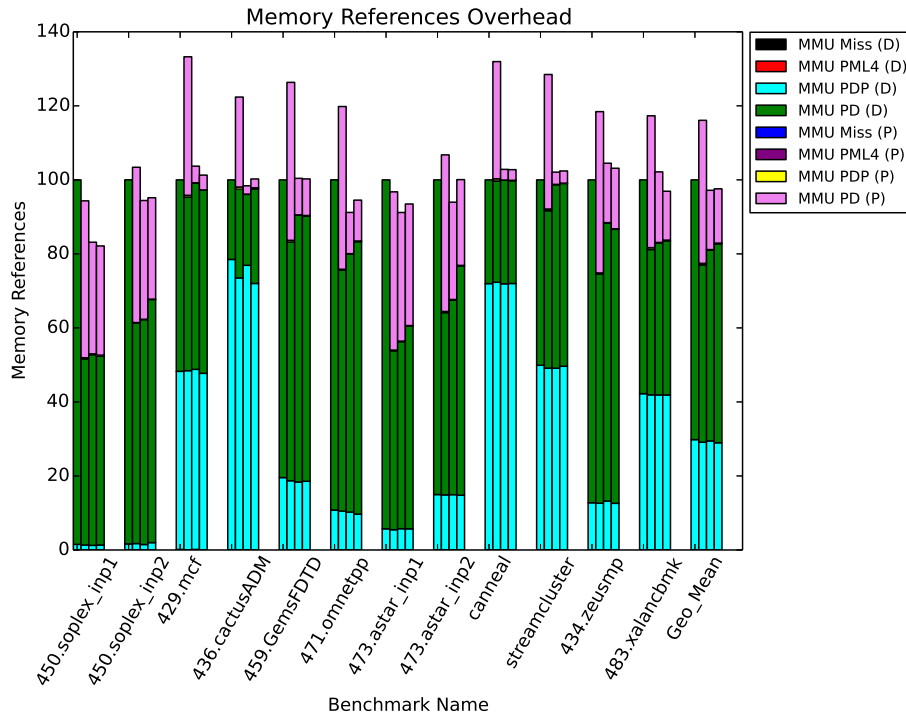


Σχήμα 5.6: Ποσοστό Ακρίβειας Προβλέψεων Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.



Σχήμα 5.7: Ποσοστό Προβλέψεων Μηδενικού Κόστους Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.

Το Σχήμα 5.6 δείχνει ότι ο Ακολουθιακός Προανακλητής παρουσιάζει ένα πολύ χαμηλό ποσοστό ακρίβειας προβλέψεων. Ωστόσο το Σχήμα 5.7 φανερώνει ότι το μεγαλύτερο ποσοστό των προβλέψεων αυτών είναι Μηδενικού Κόστους, γεγονός το οποίο σημαίνει ότι το Σχήμα 5.6 δίνει την λανθασμένη εντύπωση πως ο Ακολουθιακός Προανακλητής εισάγει ένα μεγάλο αριθμό από επιπρόσθετες λειτουργίες μνήμης για να κάνει τις προβλέψεις του. Το γεγονός αυτό ισχύει και για τους τρεις μηχανισμούς. Οι Μηχανισμοί Hit Interval Throttling και MMU Interval Throttling παρουσιάζουν μεγαλύτερο ποσοστό ευστοχίας λόγω των μειωμένων προβλέψεων που κάνουν.



**Σχήμα 5.8:** Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Ακολουθιακού Προανακλητή. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις.

Το Σχήμα 5.8 φανερώνει ότι ο Ακολουθιακός Προανακλητής δεν εισάγει μεγάλες επιβαρύνσεις στο σύστημα όσο αφορά τις λειτουργίες μνήμης, γεγονός αναμενόμενο λόγω του υψηλού ποσοστού των προβλέψεων Μηδενικού Κόστους. Οι μηχανισμοί Ελέγχου Προανάκλησης Hit Interval Throttling και MMU Interval Throttling προφανώς λόγω των μειωμένων προβλέψεων θα μειώσουν τις απαιτούμενες λειτουργίες μνήμης αλλά το αρνητικό είναι πως δεν καταφέρνουν να δώσουν τα ίδια αποτελέσματα απόδοσης με το Μηχανισμό No Throttling. Βέβαια καταφέρνουν να μειώσουν αρκετά το αρνητικό φαινόμενο της επιβάρυνσης του συστήματος από επιπρόσθετες λειτουργίες μνήμης στη περίπτωση που ο Μηχανισμός Προανάκλησης δεν μπορεί να προβλέψει σωστά τις μελλοντικές αστοχίες της TLB και κάνει συνέχεια άχρηστες προβλέψεις. Χαρακτηριστικά παραδείγματα αποτελούν τα μετροπρογράμματα 429.mcf, 436.cactusADM και canneal.

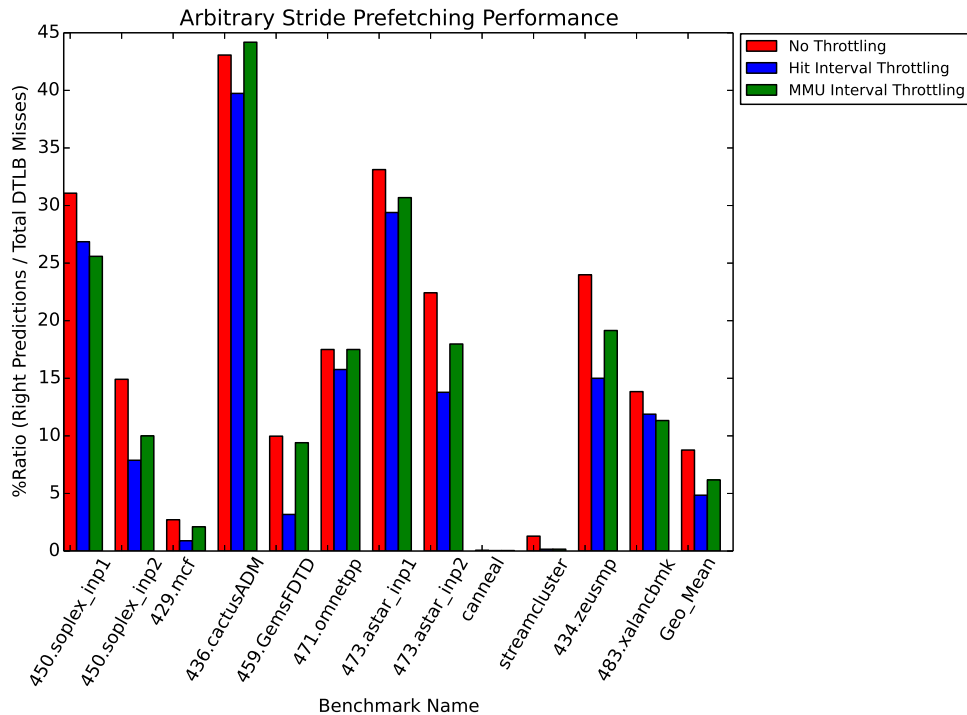
	perf	No Throttling	Hit Interval Throttling	MMU Interval Throttling
450.soplex_inp1	4.43%	2.31%	2.35%	2.35%
450.soplex_inp2	6.71%	4.40%	4.48%	4.86%
429.mcf	20.87%	19.40%	20.28%	20.39%
436.cactusADM	0.6%	0.60%	0.60%	0.60%
459.GemsFDTD	10.94%	8.97%	10.06%	10.09%
471.omnetpp	14.69%	10.94%	11.64%	12.23%
473.astar_inp1	4.93%	2.60%	2.71%	2.93%
473.astar_inp2	16.92%	10.34%	10.96%	12.67%
canneal	29.32%	29.11%	29.28%	29.28%
streamcluster	8.59%	7.69%	8.50%	8.50%
434.zeusmp	1.36%	0.99%	1.17%	1.17%
483.xalancbmk	8.41%	6.42%	6.60%	6.68%
Geo_Mean	7.02%	6.08%	6.54%	6.59%

**Πίνακας 5.2:** Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Ακολουθιακού Προανακλητή. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf.

Ο Πίνακας 5.2 επιβεβαιώνει το Σχήμα 5.5, πως δηλαδή η No Throttling εκδοχή πετυχαίνει καλύτερη απόδοση. Στη γενική περίπτωση πετυχαίνει μείωση του ποσοστού των κύκλων κατά 13%. Τα αποτελέσματα των Μηχανισμών Hit Interval Throttling και MMU Interval Throttling είναι παρόμοια, με τον πρώτο να πετυχαίνει ελάχιστα μεγαλύτερη μείωση.

## 5.4 Αποτελέσματα Προανακλητή Τυχαίου Βήματος

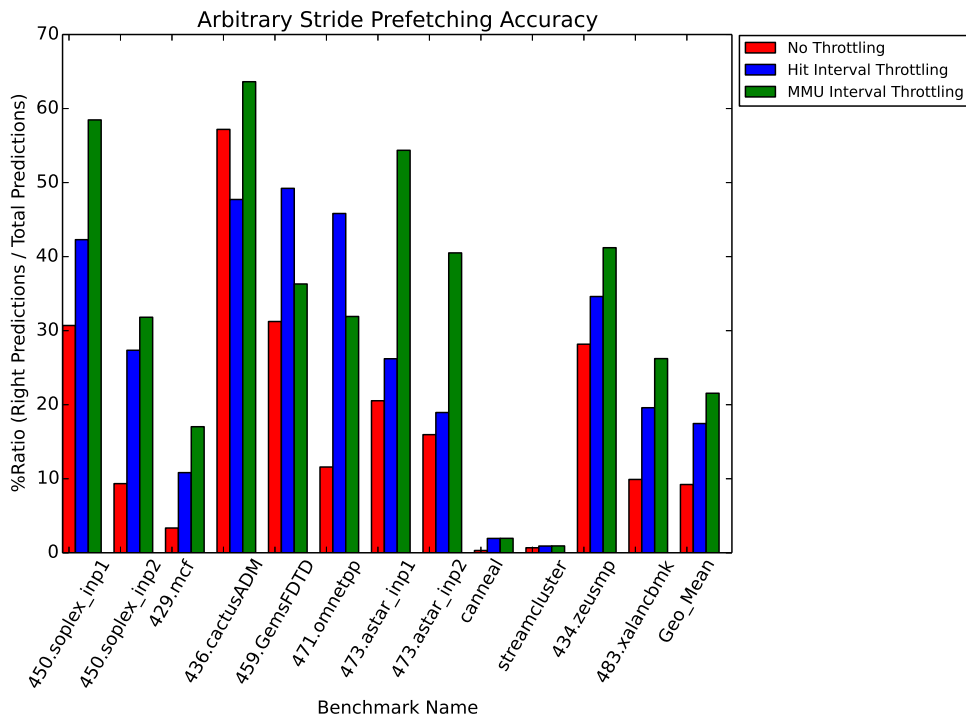
Στην ενότητα αυτή θα παρουσιαστούν τα αποτελέσματα του Προανακλητή Τυχαίου Βήματος για τις μετρικές που αναλύθηκαν στο υποκεφάλαιο 4.5. Θα παρουσιαστούν επίσης τα αποτελέσματα της εισαγωγής των Μηχανισμών Hit Interval Throttling και MMU Interval Throttling.



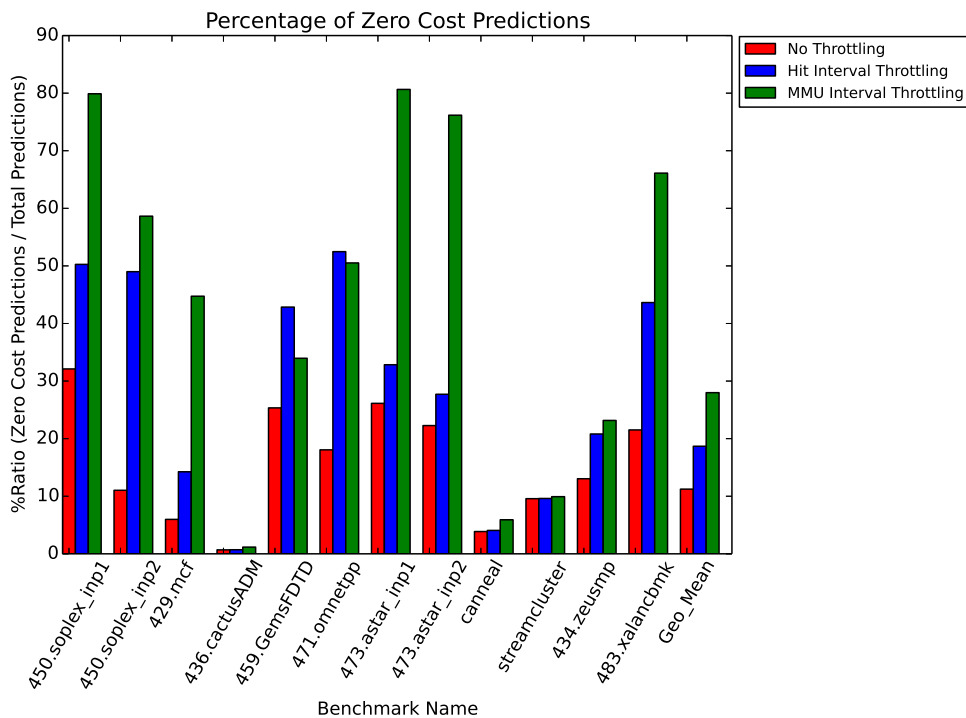
*Σχήμα 5.9: Ποσοστό Επιτυχημένων Προβλέψεων Προανακλητή Τυχαίου Βήματος για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.*

Το Σχήμα 5.9 δείχνει ότι ο No Throttling μηχανισμός του Προανακλητή Τυχαίου Βήματος δίνει τα καλύτερα αποτελέσματα απόδοσης για όλα τα μετροπρογράμματα εκτός από το 436.cactusADM, στο οποίο δίνει σχεδόν το ίδιο αποτέλεσμα με το μηχανισμό MMU Interval Throttling. Μπορεί ο μηχανισμός MMU Interval Throttling να λειτουργεί κάπως καλύτερα από τον Hit Interval Throttling αλλά όπως και στον Ακολουθιακό Προανακλητή δεν καταφέρνει κανένας από τους Μηχανισμούς Ελέγχου Προανάκλησης να δώσει αποτελέσματα ανάλογα με αυτά του No Throttling. Πολλές από τις προβλέψεις που έπρεπε να γίνουν απορρίπτονται λόγω των κατωφλίων που έχουν οριστεί. Επιλογή καταλληλότερων κατωφλίων θα έδινε αποτελέσματα κοντά σε αυτά που Μηχανισμού No Throttling.



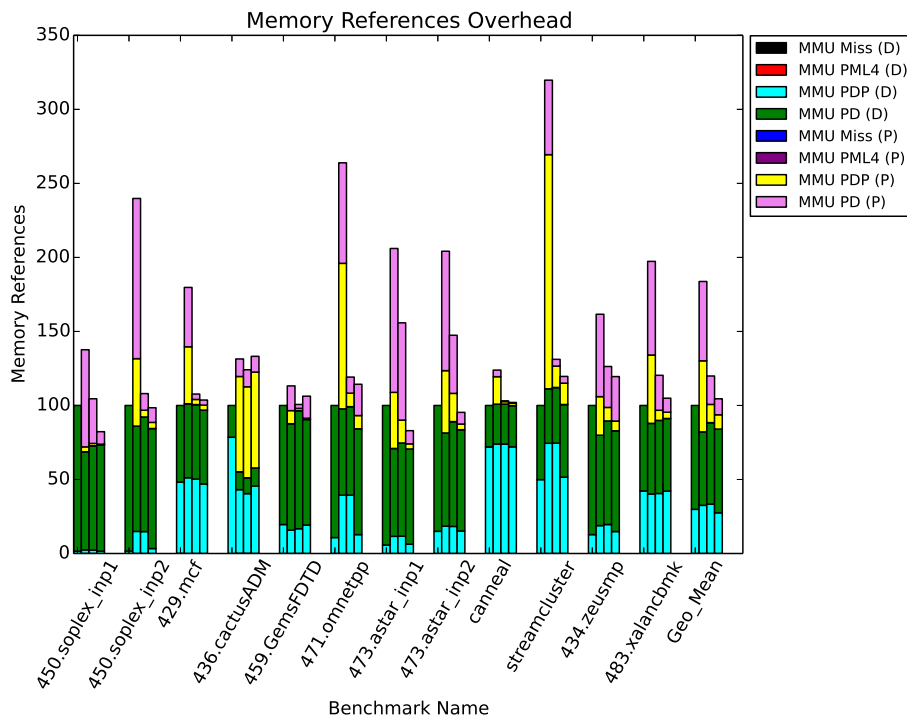


Σχήμα 5.10: Ποσοστό Ακρίβειας Προβλέψεων Προανακλητή Τυχαίου Βήματος για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.



Σχήμα 5.11: Ποσοστό Προβλέψεων Μηδενικού Κόστους Προανακλητή Τυχαίου Βήματος για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.

Το Σχήμα 5.10 δείχνει ότι ο Ακολουθιακός Προανακλητής παρουσιάζει ένα πολύ χαμηλό ποσοστό ακρίβειας προβλέψεων και το Σχήμα 5.11 φανερώνει ότι μόνο ένα μικρό ποσοστό των προβλέψεων αυτών είναι Μηδενικού Κόστους. Τα αποτελέσματα αυτά είναι λογικά αν αναλογιστεί κανείς ότι ο Προανακλητής Τυχαίου Βήματος στη παραγωγή των προβλέψεων του λαμβάνει υπόψη του αποστάσεις. Το χαμηλό ποσοστό ακρίβειας προβλέψεων και προβλέψεων Μηδενικού Κόστους οδηγεί στο συμπέρασμα ότι ο Μηχανισμός αυτός στη No Throttling εκδοχή του θα εισάγει ένα μεγάλο αριθμό από επιπρόσθετες λειτουργίες μνήμης. Τα Σχήματα 5.10 και 5.11 δείχνουν ότι τα παραπάνω ποσοστά αυξάνονται αρκετά για τους Hit Interval Throttling και MMU Interval Throttling Μηχανισμούς λόγω των μειωμένων προβλέψεων. Για το λόγο αυτό αναμένεται μεγάλη μείωση της επιβάρυνσης που προκαλεί ο No Throttling Μηχανισμός.



**Σχήμα 5.12:** Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Προανακλητή Τυχαίου Βήματος. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Προανακλητή Τυχαίου Βήματος για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις.

Το Σχήμα 5.12 δείχνει πως ο Προανακλητής Τυχαίου Βήματος στη No Throttling εκδοχή του, εισάγει μεγάλη επιβάρυνση στο σύστημα όσο αφορά τις λειτουργίες μνήμης, γεγονός αναμενόμενο λόγω του χαμηλού ποσοστού των προβλέψεων Μηδενικού Κόστους και ακρίβειας. Οι Μηχανισμοί Ελέγχου Προανάκλησης Hit Interval Throttling και MMU Interval Throttling καταφέρνουν να μειώσουν σε πολύ μεγάλο βαθμό αυτή την επιβάρυνση, ειδικά ο MMU Interval Throttling Μηχανισμός. Βέβαια δεν καταφέρνουν να δώσουν τα ίδια αποτελέσματα απόδοσης με το Μηχανισμό No Throttling. Επίσης στις περιπτώσεις που ο Μηχανισμός Προανάκλησης δεν μπορεί να προβλέψει σωστά τις μελλοντικές αστοχίες της TLB καταφέρνουν να απενεργοποιούν το Μηχανισμό Προανάκλησης Μεταφράσεων ώστε να μην επιβαρύνουν στο σύστημα με άχρηστες προβλέψεις.

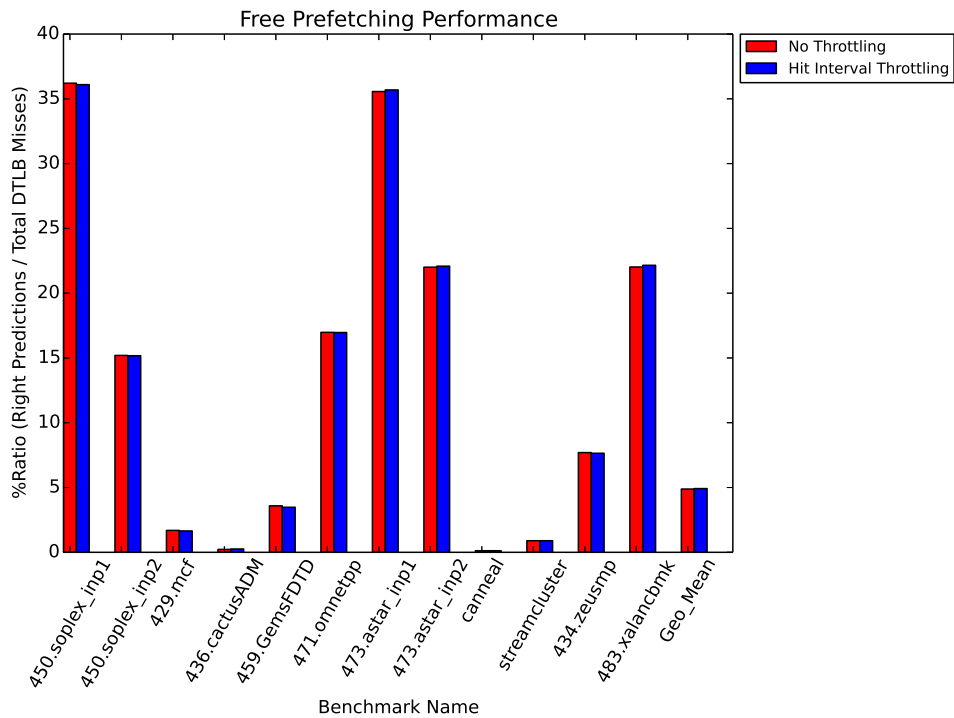
	perf	No Throttling	Hit Interval Throttling	MMU Interval Throttling
450.soplex_inp1	4.43%	3.05%	3.24%	3.30%
450.soplex_inp2	6.71%	5.71%	6.18%	6.04%
429.mcf	20.87%	20.30%	20.68%	20.43%
436.cactusADM	0.60%	0.34%	0.36%	0.33%
459.GemsFDTD	10.94%	9.85%	10.59%	9.91%
471.omnetpp	14.69%	12.12%	12.37%	12.12%
473.astar_inp1	4.93%	3.30%	3.48%	3.42%
473.astar_inp2	16.92%	13.13%	14.59%	13.88%
canneal	29.32%	29.30%	29.31%	29.31%
streamcluster	8.59%	8.48%	8.58%	8.58%
434.zeusmp	1.36%	1.03%	1.16%	1.10%
483.xalancbmk	8.41%	7.25%	7.41%	7.46%
Geo_Mean	7.02%	6.41%	6.68%	6.59%

**Πίνακας 5.3:** Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Προανακλητή Τυχαίου Βήματος. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf.

Ο Πίνακας 5.3 επιβεβαιώνει το Σχήμα 5.9, πως δηλαδή η No Throttling εκδοχή πετυχαίνει καλύτερη απόδοση. Στη γενική περίπτωση πετυχαίνει μείωση του ποσοστού των κύκλων κατά 8.5%. Τα αποτελέσματα των Μηχανισμών Hit Interval Throttling και MMU Interval Throttling είναι παρόμοια, με τον δεύτερο να πετυχαίνει ελάχιστα μεγαλύτερη μείωση.

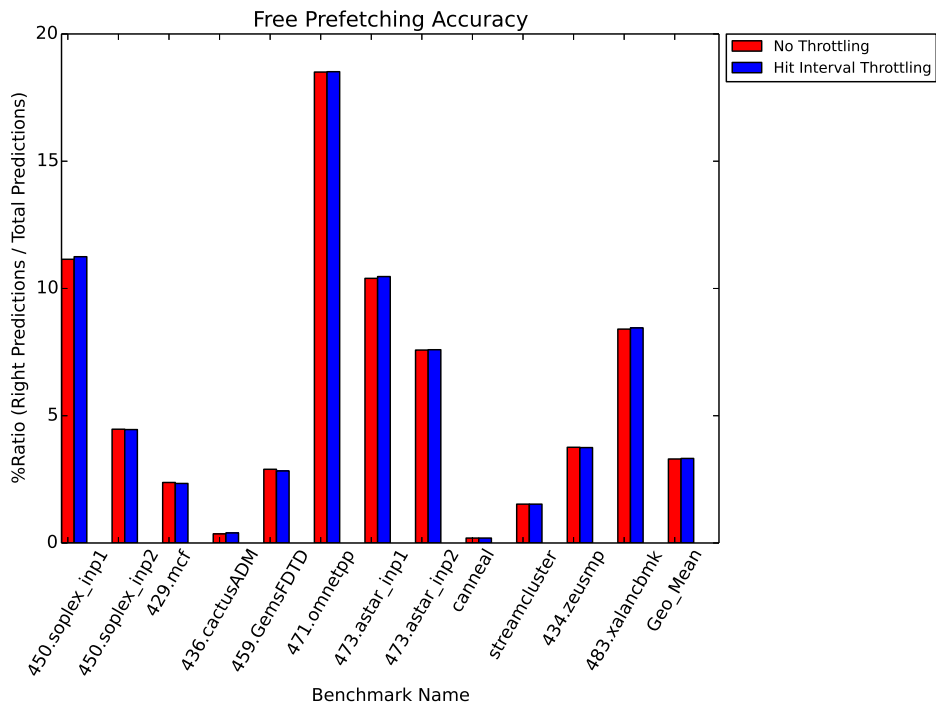
## 5.5 Αποτελέσματα Προανακλητή Μηδενικού Κόστους

Στην ενότητα αυτή θα παρουσιαστούν τα αποτελέσματα του Προανακλητή Μηδενικού Κόστους για τις μετρικές που αναλύθηκαν στο υποκεφάλαιο 4.5. Θα παρουσιαστούν επίσης τα αποτελέσματα της εισαγωγής του Μηχανισμού Hit Interval Throttling. Ο Μηχανισμός MMU Interval Throttling δεν υλοποιείται για τον Προανακλητή αυτόν αφού όλες οι προβλέψεις που κάνει είναι Μηδενικού Κόστους.

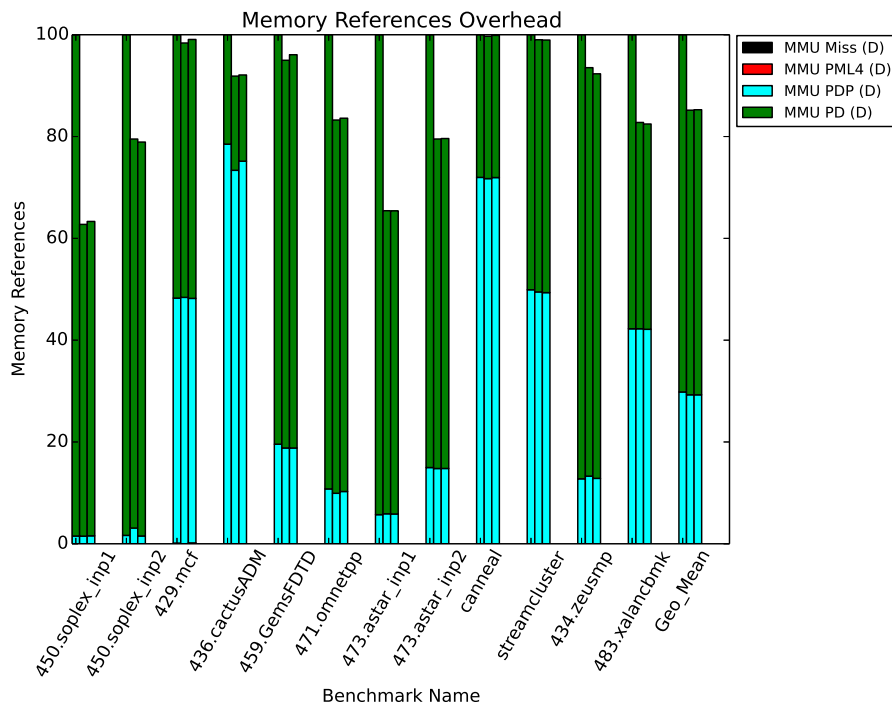


*Σχήμα 5.13: Ποσοστό Επιτυχημένων Προβλέψεων Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.*

Το Σχήμα 5.13 δείχνει ότι ο Προανακλητής Μηδενικού Κόστους τόσο με No Throttling όσο και με Hit Interval Throttling δίνει ίδια αποτελέσματα απόδοσης. Τα ποσοστά απόδοσης που πετυχαίνει δεν είναι ιδιαίτερα υψηλά αλλά είναι ικανοποιητικά αν αναλογιστεί κανείς ότι δεν προκαλεί καμία επιπλέον λειτουργία μνήμης για να κάνει τις προβλέψεις του.



Σχήμα 5.14: Ποσοστό Ακρίβειας Προβλέψεων Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.



Σχήμα 5.15: Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Προανακλητή Μηδενικού Κόστους. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις.

Τα Σχήματα 5.14 και 5.15 δείχνουν και αυτά ότι δεν υπάρχει κάτι που να διαχωρίζει τους Μηχανισμούς No Throttling και Hit Interval Throttling. Το Σχήμα 5.14 παρουσιάζει χαμηλά ποσοστά ευστοχίας αλλά στη πράξη δεν έχει μεγάλη σημαντικότητα σαν μετρική για το Προανακλητή Μηδενικού Κόστους αφού όλες οι προβλέψεις του είναι μηδενικού κόστους. Τα χαμηλά ποσοστά ευστοχίας οφείλονται στην επιθετικότητα του Μηχανισμού αφού σε κάθε αστοχία της TLB παράγει εφτά νέες προβλέψεις. Το Σχήμα 5.15 φανερώνει και στη πράξη την απουσία επιπρόσθετων λειτουργιών μνήμης για προβλέψεις.

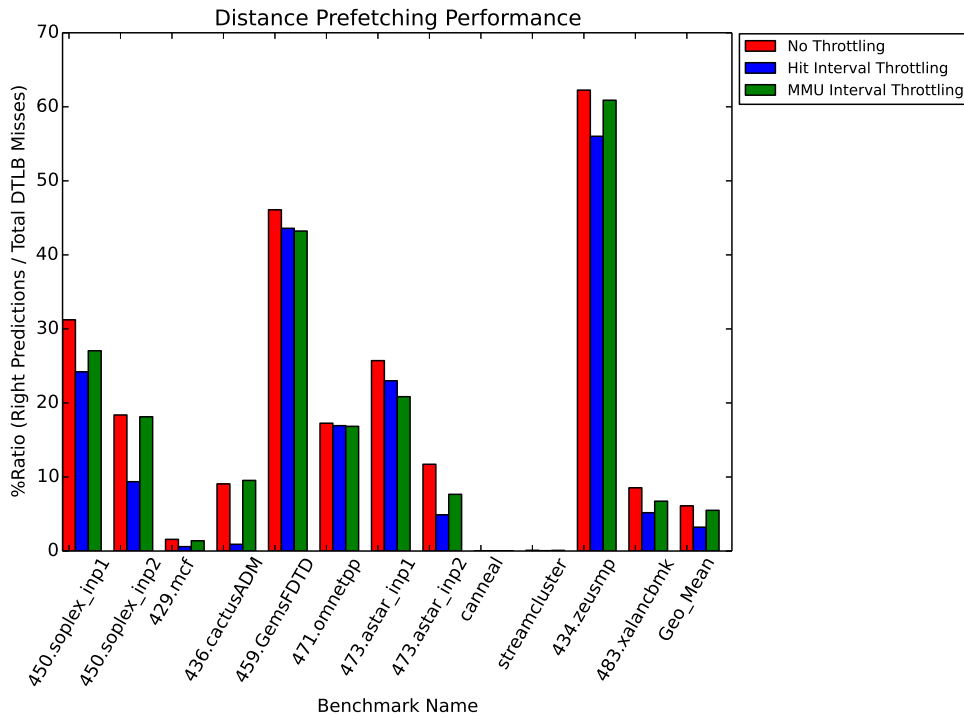
	perf	No Throttling	Hit Interval Throttling	MMU Interval Throttling
450.soplex_inp1	4.43%	2.83%	2.83%	-
450.soplex_inp2	6.71%	5.69%	5.69%	-
429.mcf	20.87%	20.52%	20.53%	-
436.cactusADM	0.60%	0.60%	0.60%	-
459.GemsFDTD	10.94%	10.55%	10.56%	-
471.omnetpp	14.69%	12.20%	12.20%	-
473.astar_inp1	4.93%	3.18%	3.17%	-
473.astar_inp2	16.92%	13.20%	13.18%	-
canneal	29.32%	29.28%	29.28%	-
streamcluster	8.59%	8.51%	8.51%	-
434.zeusmp	1.36%	1.26%	1.26%	-
483.xalancbmk	8.41%	6.56%	6.55%	-
Geo_Mean	7.02%	6.68%	6.68%	-

**Πίνακας 5.4:** Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Προανακλητή Μηδενικού Κόστους. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf.

Ο Πίνακας 5.4 επιβεβαιώνει το Σχήμα 5.13, πως δηλαδή οι εκδοχές No Throttling Hit Interval Throttling δεν διαφοροποιούνται μεταξύ τους και πετυχαίνουν μείωση κοντά στο 5% όσο αφορά το ποσοστό των κύκλων που ξοδεύονται σε TLB αστοχίες. Γενικά ο Προανακλητής Μηδενικού Κόστους δεν καταφέρνει να κάνει σημαντική μείωση των κύκλων.

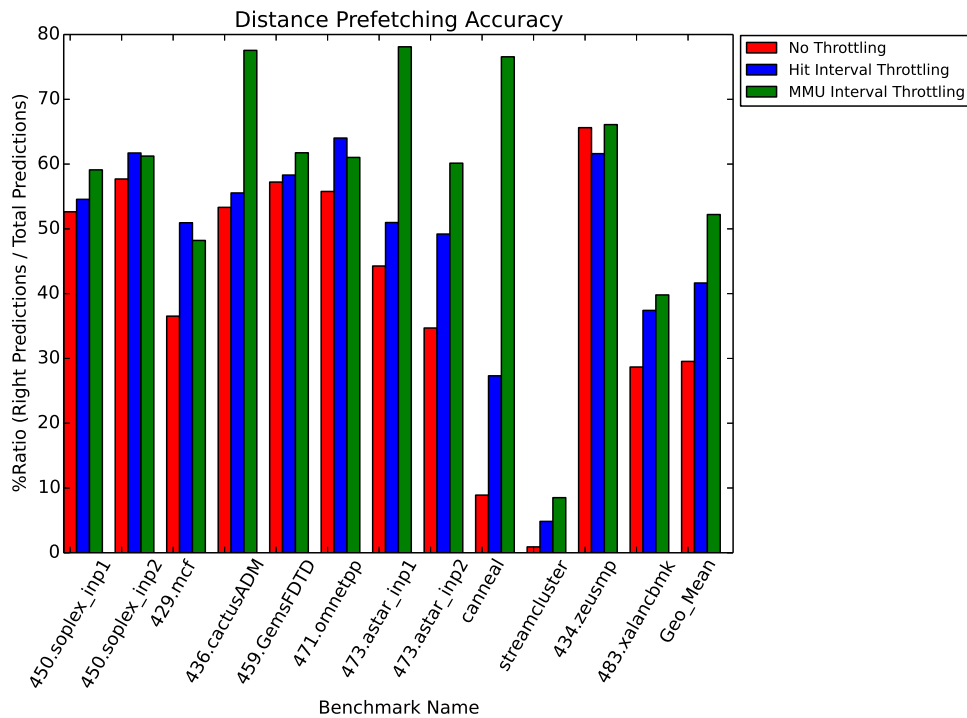
## 5.6 Αποτελέσματα Προανακλητή Απόστασης

Στην ενότητα αυτή θα παρουσιαστούν τα αποτελέσματα του Προανακλητή Απόστασης για τις μετρικές που αναλύθηκαν στο υποκεφάλαιο 4.5. Θα παρουσιαστούν επίσης τα αποτελέσματα της εισαγωγής των Μηχανισμών Hit Interval Throttling και MMU Interval Throttling.

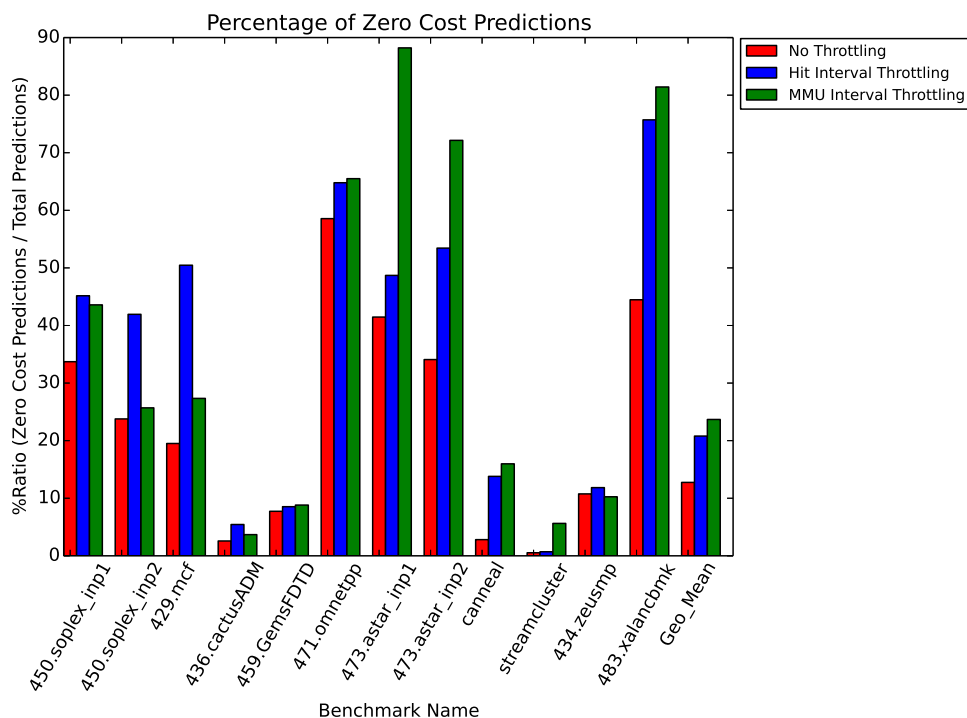


**Σχήμα 5.16:** Ποσοστό Επιτυχημένων Προβλέψεων Προανακλητή Απόστασης για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.

Το Σχήμα 5.16 δείχνει ότι ο Προανακλητής Απόστασης δεν καταφέρνει υψηλή απόδοση στη γενική περίπτωση. Ωστόσο, καταφέρνει να έχει υψηλή απόδοση στα μετροπρογράμματα 459.GemsFDTD και 483.xalancbmk, στα οποία δεν καταφέρνει κανένας άλλος απλός Μηχανισμός Πρόβλεψης τέτοια απόδοση. Ο No Throttling μηχανισμός δίνει τα καλύτερα αποτελέσματα απόδοσης για όλα τα μετροπρογράμματα. Σε κάποια από αυτά ο Μηχανισμός MMU Interval Throttling έχει απόδοση αρκετά κοντά στο No Throttling και είναι φανερό ότι λειτουργεί καλύτερα από τον Hit Interval Throttling. Ωστόσο κανένας από τους Μηχανισμούς Ελέγχου Προανάκλησης δεν καταφέρνει να δώσει αποτελέσματα ανάλογα με αυτά του No Throttling. Πολλές από τις προβλέψεις που έπρεπε να γίνουν απορρίπτονται λόγω των κατωφλίων που έχουν οριστεί. Επιλογή καταλληλότερων κατωφλίων θα έδινε αποτελέσματα κοντά σε αυτά που Μηχανισμού No Throttling.



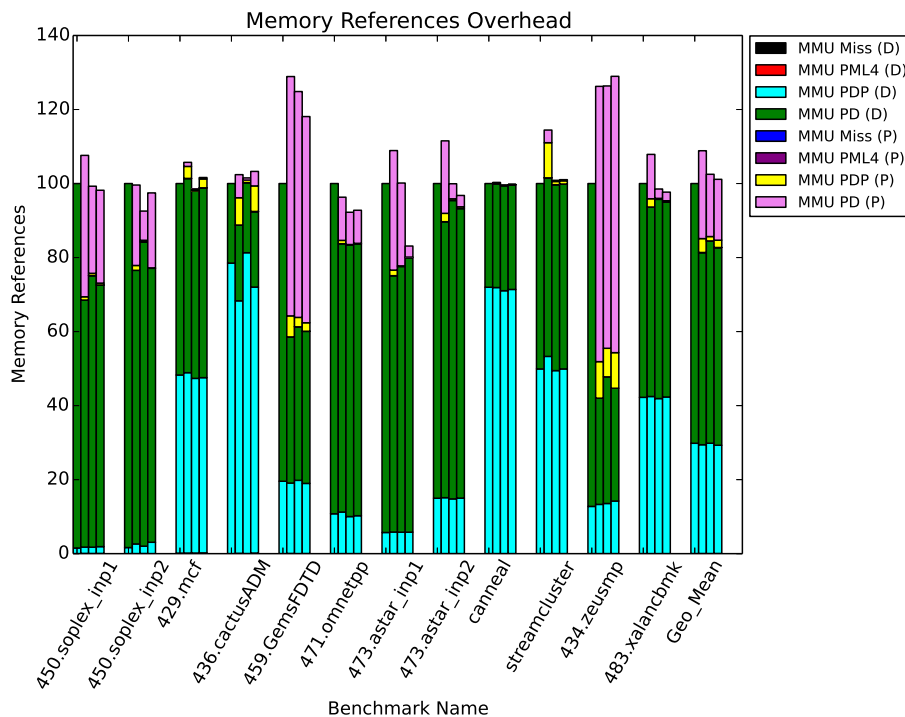
*Σχήμα 5.17: Ποσοστό Ακρίβειας Προβλέψεων Προανακλητή Απόστασης για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.*



*Σχήμα 5.18: Ποσοστό Προβλέψεων Μηδενικού Κόστους Προανακλητή Απόστασης για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.*



Το Σχήμα 5.17 δείχνει ότι ο Προανακλητής Απόστασης παρά το γεγονός ότι ασχολείται με απόστασεις για να κάνει τις προβλέψεις του, πετυχαίνει αρκετά καλά ποσοστά ακριβείας αν εξαιρέσει κανείς το μετροπρόγραμμα streamcluster. Αυτό οφείλεται στο γεγονός ότι δεν κάνει συχνά προβλέψεις αφού ανανεώνει το Πίνακα Αποστάσεων σε κάθε TLB αστοχία. Έτσι οι περισσότερες από τις προβλέψεις του οδηγούν σε επιτυχίες. Οι Μηχανισμοί Hit Interval Throttling και MMU Interval Throttling έχουν ακόμα μεγαλύτερα ποσοστά ακριβείας, λόγω της μείωσης των προβλέψεων. Προβλέψεις Μηδενικού Κόστους δεν κάνει πολλές κανέναν από τους τρεις Μηχανισμούς, όπως φαίνεται και στο Σχήμα 5.18. Ο MMU Interval Throttling ωστόσο ξεχωρίζει σε κάποιες περιπτώσεις. Ο Προανακλητής Απόστασης παρά το χαμηλό ποσοστό προβλέψεων Μηδενικού Κόστους αναμένεται να μην εισάγει υψηλή επιβάρυνση στο σύστημα για την πραγματοποίηση των προβλέψεων του λόγω της ακριβείας του.



**Σχήμα 5.19:** Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Προανακλητή Απόστασης. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Προανακλητή Απόστασης για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις.

Πράγματι, το Σχήμα 5.19 δείχνει πως ο Προανακλητής Απόστασης δεν εισάγει υψηλή επιβάρυνση όσο αφορά τις λειτουργίες μνήμης. Μόνο για τα μετροπρογράμματα 459.GemsFDTD και 483.xalanbmk εισάγει επιβάρυνση σχετικά υψηλή αλλά στα μετροπρογράμματα αυτά πετυχαίνει τη καλύτερη απόδοση ανάμεσα στους διάφορους Μηχανισμούς Προανάκλησης. Οι Μηχανισμοί Ελέγχου Προανάκλησης Hit Interval Throttling και MMU Interval Throttling μειώνουν την επιβάρυνση, ιδιαίτερα εκεί που ο Προανακλητής Απόστασης δεν καταφέρνει καλή απόδοση και αναλώνεται σε άχρηστες προβλέψεις. Βέβαια δεν καταφέρνουν να δώσουν τα ίδια αποτελέσματα απόδοσης με το Μηχανισμό No Throttling, το οποίο ήταν και το ζητούμενο.

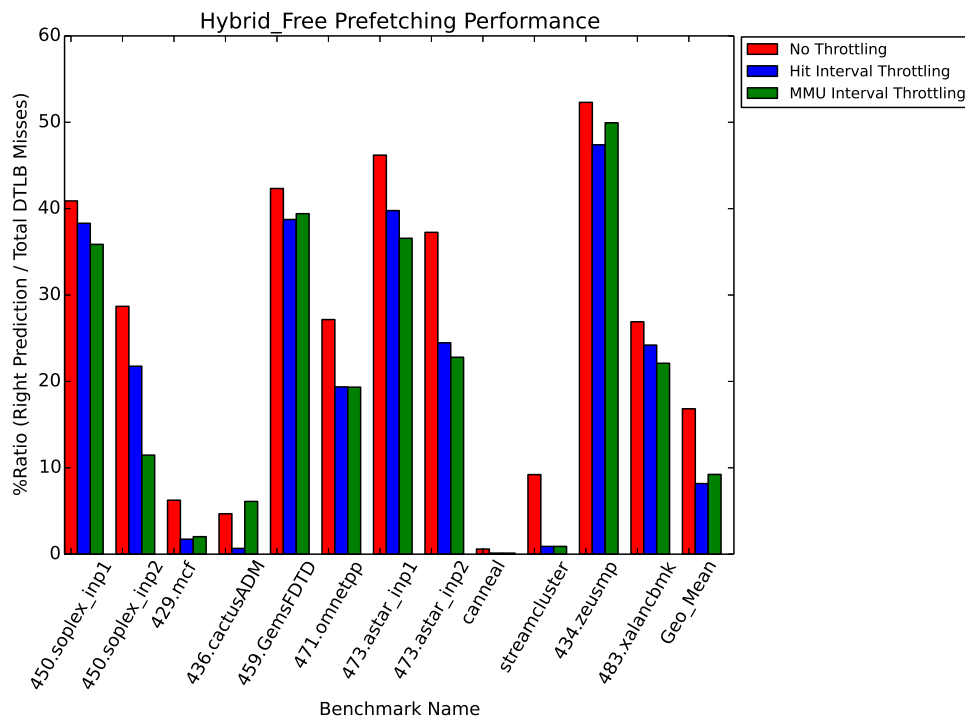
	perf	No Throttling	Hit Interval Throttling	MMU Interval Throttling
450.soplex_inp1	4.43%	3.05%	3.36%	3.23%
450.soplex_inp2	6.71%	5.48%	6.08%	5.49%
429.mcf	20.87%	20.54%	20.75%	20.58%
436.cactusADM	0.60%	0.55%	0.59%	0.54%
459.GemsFDTD	10.94%	5.90%	6.17%	6.21%
471.omnetpp	14.69%	12.15%	12.20%	12.22%
473.astar_inp1	4.93%	3.66%	3.80%	3.90%
473.astar_inp2	16.92%	14.94%	16.09%	15.62%
canneal	29.32%	29.31%	29.32%	29.31%
streamcluster	8.59%	8.58%	8.59%	8.58%
434.zeusmp	1.36%	0.51%	0.60%	0.53%
483.xalancbmk	8.41%	7.69%	7.97%	7.84%
Geo_Mean	7.02%	6.59%	6.80%	6.64%

**Πίνακας 5.5:** Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Προανακλητή Απόστασης. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf.

Ο Πίνακας 5.5 επιβεβαιώνει το Σχήμα 5.16, πως δηλαδή η No Throttling εκδοχή πετυχαίνει καλύτερη απόδοση. Στη γενική περίπτωση πετυχαίνει μείωση του ποσοστού των κύκλων κατά 6.5%. Τα αποτελέσματα του Μηχανισμού MMU Interval Throttling είναι πολύ κοντά σε αυτά του No Throttling, ενώ ο Hit Interval Throttling έχει αρκετά χειρότερα αποτελέσματα. Γενικά ο Προανακλητής Απόστασης δεν καταφέρνει να κάνει σημαντική μείωση των κύκλων.

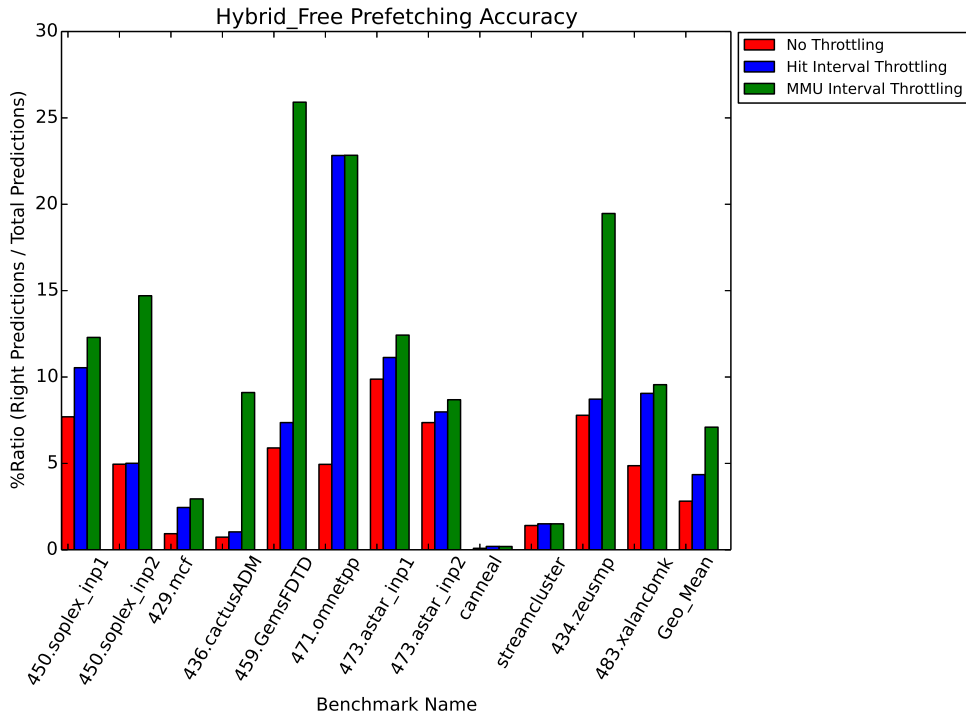
## 5.7 Αποτελέσματα Υβριδικού Προανακλητή Μηδενικού Κόστους

Στην ενότητα αυτή θα παρουσιαστούν τα αποτελέσματα του Υβριδικού Προανακλητή Μηδενικού Κόστους για τις μετρικές που αναλύθηκαν στο υποκεφάλαιο 4.5. Θα παρουσιαστούν επίσης τα αποτελέσματα της εισαγωγής των Μηχανισμών Hit Interval Throttling και MMU Interval Throttling.

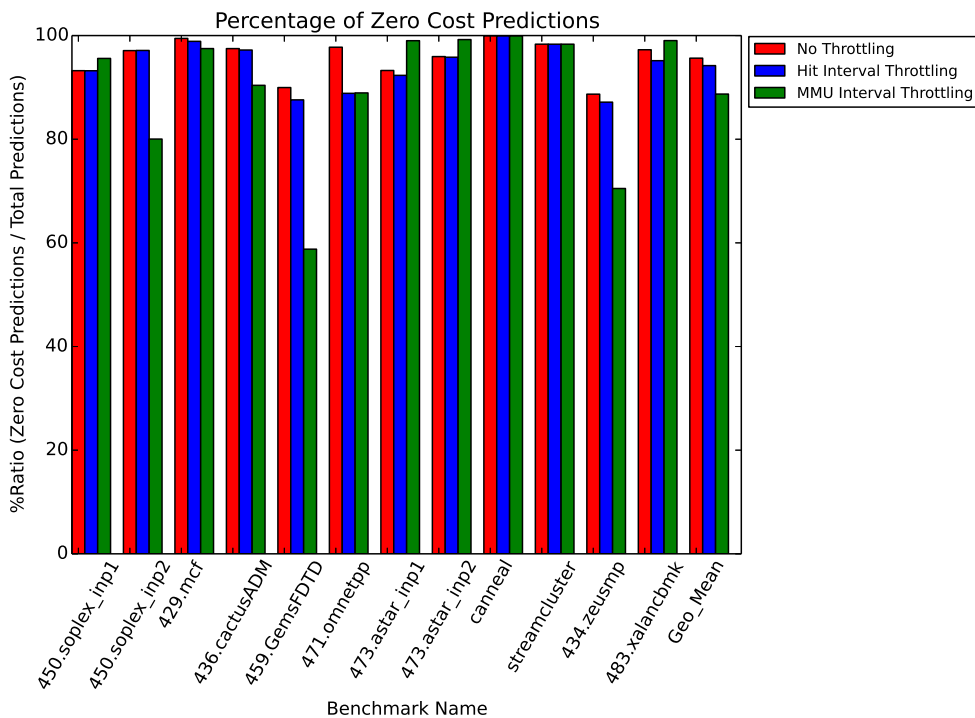


**Σχήμα 5.20:** Ποσοστό Επιτυχημένων Προβλέψεων Υβριδικού Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.

Το Σχήμα 5.20 δείχνει πως ο Υβριδικός Προανακλητής Μηδενικού Κόστους πετυχαίνει σχετικά καλή απόδοση. Ωστόσο, όπως αναλύθηκε και στο υποκεφάλαιο 5.2, καταφέρνει να συνδυάζει καλά τους δύο μηχανισμούς που τον αποτελούν μόνο όταν και οι δύο παρουσιάζουν μία σχετικά καλή απόδοση ατομικά. Σε αντίθετη περίπτωση, η κακή απόδοση τους ενός εμποδίζει τον άλλον να αποδοσει αυτό που είναι ικανός. Ο No Throttling μηχανισμός δίνει τα καλύτερα αποτελέσματα απόδοσης για όλα τα μετροπρογράμματα. Κανένας από τους Μηχανισμούς Ελέγχου Προανάκλησης δεν καταφέρνει να δώσει αποτελέσματα ανάλογα με αυτά του No Throttling. Πολλές από τις προβλέψεις που έπρεπε να γίνουν απορρίπτονται λόγω των κατωφλίων που έχουν οριστεί. Επιλογή καταλληλότερων κατωφλίων θα έδινε αποτελέσματα κοντά σε αυτά που Μηχανισμού No Throttling.

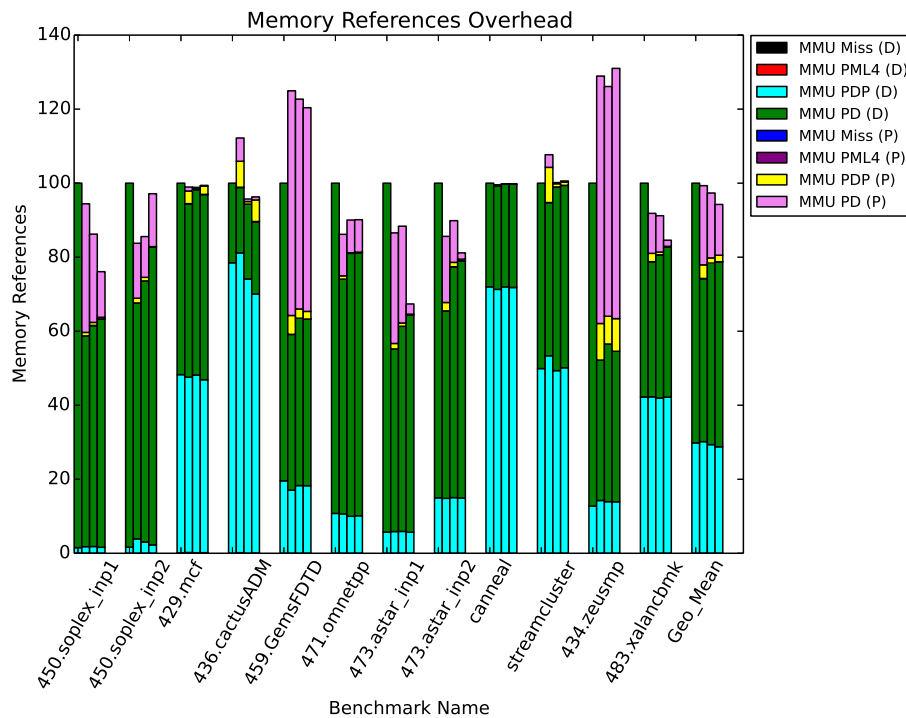


Σχήμα 5.21: Ποσοστό Ακρίβειας Προβλέψεων Υβριδικού Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.



Σχήμα 5.22: Ποσοστό Προβλέψεων Μηδενικού Κόστους Υβριδικού Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.

Ο Υβριδικός Προανακλητής Μηδενικού Κόστους αποτελείται από το Προανακλητή Απόστασης και το Προβλετή Μηδενικού Κόστους. Ο πρώτος είναι ένας αρκετά εύστοχος μηχανισμός, ο οποίος όμως δεν κάνει συχνά προβλέψεις αλλά μόνο όταν είναι αρκετά σίγουρος. Ο δεύτερος είναι ένας πολύ επιθετικός Μηχανισμός Προανάκλησης Μεταφράσεων, ο οποίος επειδή οι προβλέψεις του είναι Μηδενικού Κόστους δεν ενδιαφέρεται για τη μετρική της ακρίβειας. Αυτός είναι ο λόγος για τον οποίο το 5.21 έχει χαμηλά ποσοστά ακρίβειας και το Σχήμα 5.22 έχει εξαιρετικά υψηλά ποσοστά προβλέψεων Μηδενικού Κόστους. Οι Μηχανισμοί Ελέγχου Προανάκλησης Hit Interval Throttling και MMU Interval Throttling διαφοροποιούνται στην ακρίβεια κατά ένα μικρό ποσοστό λόγω των μειωμένων προβλέψεων που κάνουν. Ο Μηχανισμός MMU Interval Throttling λόγω των ισχυρών κατωφλίων που έχει πολλές φορές κόβει και Μηδενικού Κόστους Προβλέψεις, γι' αυτό και κάποια μετροπρογράμματα παρουσιάζουν χαμηλότερο ποσοστά.



**Σχήμα 5.23:** Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Υβριδικού Προανακλητή Μηδενικού Κόστους. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Υβριδικού Προανακλητή Μηδενικού Κόστους για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις.

Μπορεί ο Υβριδικός Προανακλητής Μηδενικού Κόστους να απολαμβάνει πολύ υψηλά ποσοστά προβλέψεων Μηδενικού Κόστους αλλά στη πράξη αυτό γίνεται επειδή το πλήθος των προβλέψεων που κάνει ο Προβλετής Μηδενικού Κόστους είναι εξαιρετικά μεγαλύτερο (περίπου επτά φορές μεγαλύτερο) από αυτό του Προανακλητή Απόστασης. Ωστόσο εκεί που ο Προανακλητής Απόστασης αρχίζει να παράγει προβλέψεις με μεγαλύτερη συχνότητα εμφανίζονται ακριβώς οι ίδιες επιβαρύνσεις σε λειτουργίες μνήμης όπως γινόταν όταν αξιολογήθηκε μόνος του. Το Σχήμα 5.23 επιβεβαιώνει το γεγονός αυτό. Οι Μηχανισμοί Ελέγχου Προανάκλησης Hit Interval Throttling και MMU Interval Throttling μειώνουν την επιβάρυνση, ιδιαίτερα εκεί που ο Προανακλητής Απόστασης δεν καταφέρνει καλή απόδοση και αναλώνεται σε άχρηστες προβλέψεις. Βέβαια δεν καταφέρνουν να δώσουν τα ίδια αποτελέσματα απόδοσης με το Μηχανισμό No Throttling, το οποίο ήταν και το ζητούμενο.

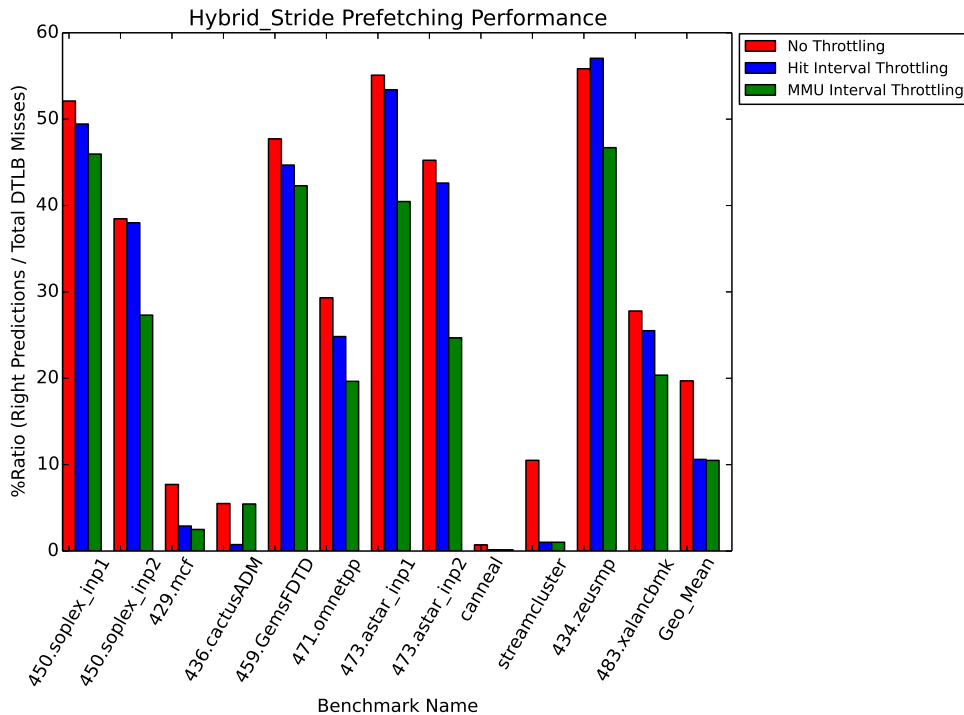
	perf	No Throttling	Hit Interval Throttling	MMU Interval Throttling
450.soplex_inp1	4.43%	2.62%	2.73%	2.84%
450.soplex_inp2	6.71%	4.78%	5.25%	5.94%
429.mcf	20.87%	19.57%	20.51%	20.45%
436.cactusADM	0.60%	0.57%	0.60%	0.56%
459.GemsFDTD	10.94%	6.31%	6.70%	6.63%
471.omnetpp	14.69%	10.70%	11.85%	11.85%
473.astar_inp1	4.93%	2.65%	2.97%	3.13%
473.astar_inp2	16.92%	10.62%	12.78%	13.06%
canneal	29.32%	29.15%	29.28%	29.28%
streamcluster	8.59%	7.80%	8.51%	8.51%
434.zeusmp	1.36%	0.65%	0.72%	0.68%
483.xalancbmk	8.41%	6.15%	6.37%	6.55%
Geo_Mean	7.02%	5.84%	6.45%	6.37%

**Πίνακας 5.6:** Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Υβριδικού Προανακλητή Μηδενικού Κόστους. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf.

Ο Πίνακας 5.6 επιβεβαιώνει το Σχήμα 5.20, πως δηλαδή η No Throttling εκδοχή πετυχαίνει καλύτερη απόδοση. Στη γενική περίπτωση πετυχαίνει μείωση του ποσοστού των κύκλων κατά 16%, η οποία είναι αρκετά ικανοποιητική. Τα αποτελέσματα των Μηχανισμών Hit Interval Throttling και MMU Interval Throttling είναι παρόμοια και απέχουν πολύ από εκείνα της No Throttling εκδοχής.

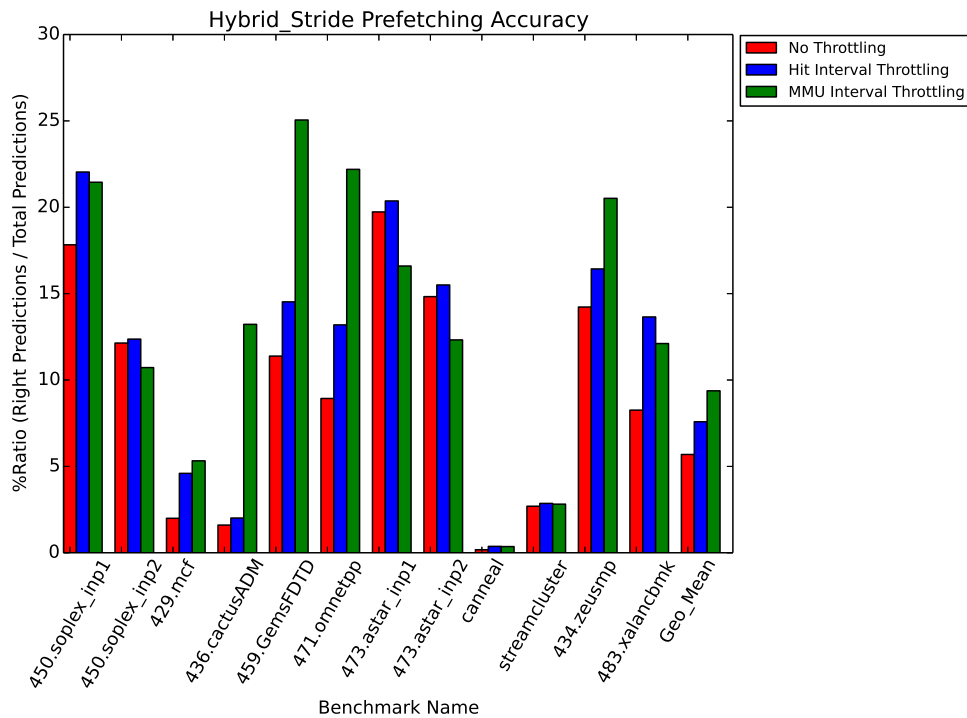
## 5.8 Αποτελέσματα Υβριδικού Ακολουθιακού Προανακλητή

Στην ενότητα αυτή θα παρουσιαστούν τα αποτελέσματα του Υβριδικού Ακολουθιακού Προβλέπτη για τις μετρικές που αναλύθηκαν στο υποκεφάλαιο 4.5. Θα παρουσιαστούν επίσης τα αποτελέσματα της εισαγωγής των Μηχανισμών Hit Interval Throttling και MMU Interval Throttling.

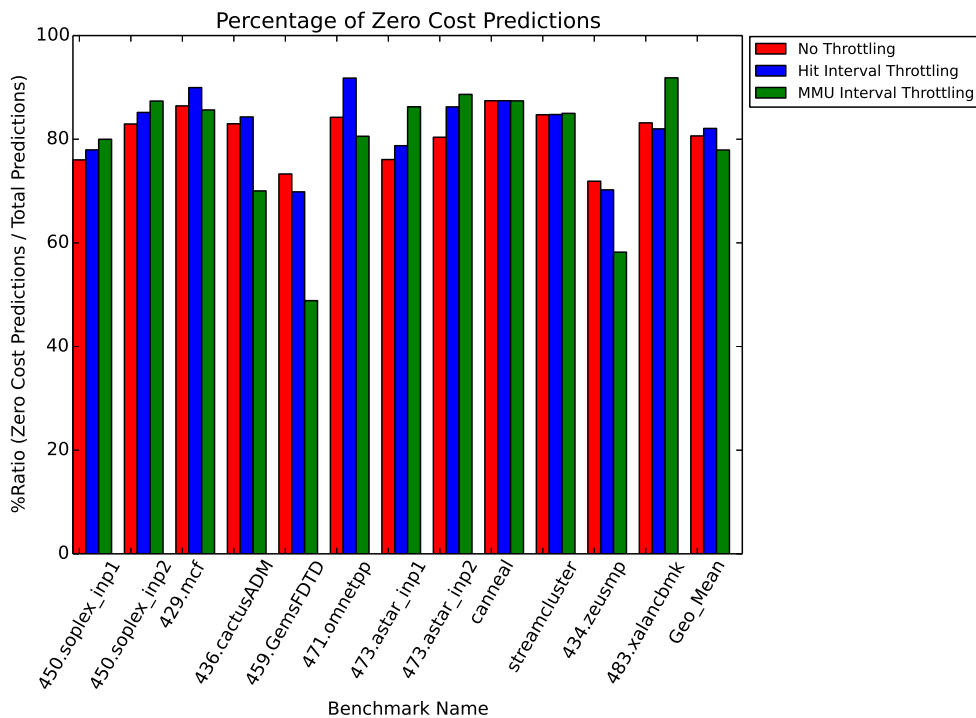


**Σχήμα 5.24:** Ποσοστό Επιτυχημένων Προβλέψεων Υβριδικού Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.

Το Σχήμα 5.24 δείχνει πως ο Υβριδικός Προανακλητής Μηδενικού Κόστους έχει τα υψηλότερα ποσοστά απόδοσης από τους μηχανισμούς που έχουν παρουσιαστεί έως τώρα. Όπως αναλύθηκε και στο υποκεφάλαιο 5.2, φαίνεται να συνδυάζει καλά τους μηχανισμούς που τον αποτελούν ακόμα και όταν κάποιος από τους δύο δεν παρουσιάζει καλή απόδοση. Ο No Throttling μηχανισμός δίνει τα καλύτερα αποτελέσματα απόδοσης για όλα τα μετροπρογράμματα εκτός από το 434.zeusmp. Σε αρκετές περιπτώσεις τα αποτελέσματα απόδοσης του Μηχανισμού Hit Interval Throttling είναι πολύ κοντά σε αυτά του No Throttling. Ωστόσο κανένας από τους Μηχανισμούς Κόστους δεν καταφέρνει να δώσει αποτελέσματα ανάλογα με αυτά του No Throttling για όλα τα μετροπρογράμματα. Πολλές από τις προβλέψεις που έπρεπε να γίνουν απορρίπτονται λόγω των κατωφλίων που έχουν οριστεί. Επιλογή καταλληλότερων κατωφλίων θα έδινε αποτελέσματα κοντά σε αυτά που Μηχανισμού No Throttling.



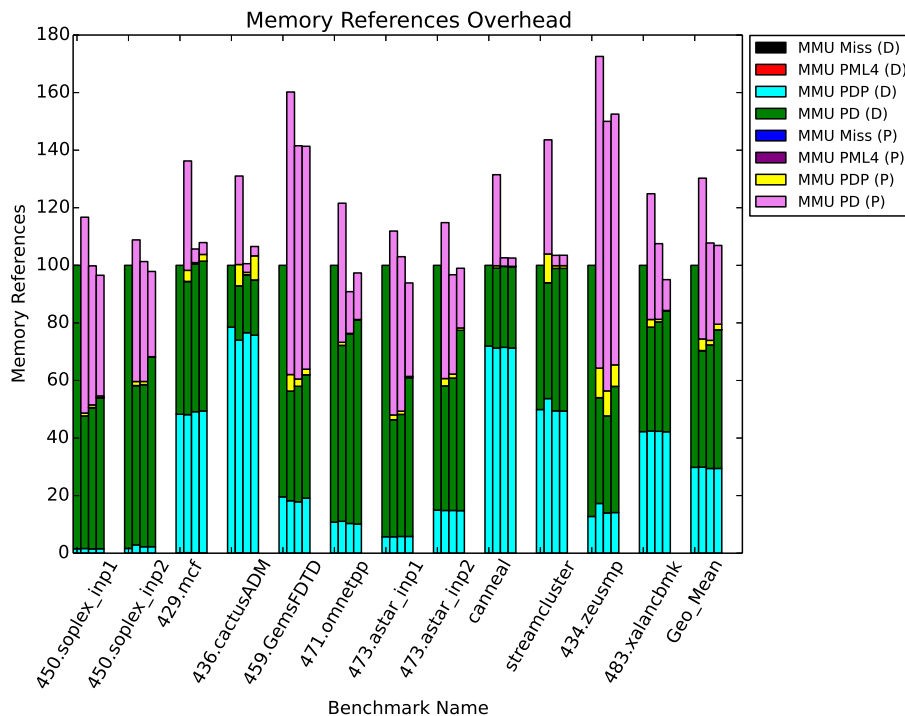
Σχήμα 5.25: Ποσοστό Ακρίβειας Προβλέψεων Υβριδικού Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.



Σχήμα 5.26: Ποσοστό Προβλέψεων Μηδενικού Κόστους Υβριδικού Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.



Ο Υβριδικός Ακολουθιακός Προανακλητής αποτελείται από τον Προανακλητή Απόστασης και τον Ακολουθιακό Προβλετή. Ο πρώτος είναι ένας αρκετά εύστοχος μηχανισμός, ο οποίος όμως δεν κάνει συχνά προβλέψεις αλλά μόνο όταν είναι αρκετά σίγουρος. Ο δεύτερος είναι ένας αρκετά επιθετικός Μηχανισμός Προανάκλησης Μεταφράσεων, του οποίου οι προβλέψεις είναι με μεγάλη πιθανότητα Μηδενικού Κόστους. Αυτός είναι ο λόγος για τον οποίο το 5.25 έχει χαμηλά ποσοστά ευστοχίας και το Σχήμα 5.26 έχει υψηλά ποσοστά προβλέψεων Μηδενικού Κόστους. Οι Μηχανισμοί Ελέγχου Προανάκλησης Hit Interval Throttling και MMU Interval Throttling διαφοροποιούνται στην ευστοχία κατά ένα μικρό ποσοστό λόγω των μειωμένων προβλέψεων που κάνουν. Ο Μηχανισμός MMU Interval Throttling λόγω των ισχυρών κατωφλίων που έχει πολλές φορές κόβει και Μηδενικού Κόστους Προβλέψεις, γι' αυτό και κάποια μετροπρογράμματα παρουσιάζουν χαμηλότερο ποσοστά.



**Σχήμα 5.27:** Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Υβριδικού Ακολουθιακού Προανακλητή. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Υβριδικού Ακολουθιακού Προανακλητή για τις περιπτώσεις No Throttling, Hit Interval Throttling, MMU Interval Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις.

Το Σχήμα 5.23 δείχνει ότι ο Μηχανισμός του Υβριδικού Ακολουθιακού Προανακλητή στην εκδοχή του No Throttling παρουσιάζει αρκετά υψηλή επιβάρυνση σε λειτουργίες μνήμης. Το φαινόμενο αυτό είναι λογικό καθώς οι δύο Μηχανισμοί Προανάκλησης που τον αποτελούν ενεργοποιούνται σε κάθε TLB αστοχία και δεν γίνεται επιλογή του καλύτερου με κάποιο κριτήριο. Την υψηλή αυτή επιβάρυνση καταφέρνουν οι Μηχανισμοί Hit Interval Throttling και MMU Interval Throttling να την περιορίσουν σε μεγάλο βαθμό. Ειδικά για μετροπρογράμματα όπως τα 429.mcf και 436.cactusADM η μείωση είναι αρκετά μεγάλη. Η υψηλή μείωση των λειτουργιών μνήμης δεν συνδυάζεται με τη διατήρηση απόδοσης ίδιας με τη No Throttling εκδοχή, το οποίο ήταν και το ζητούμενο.

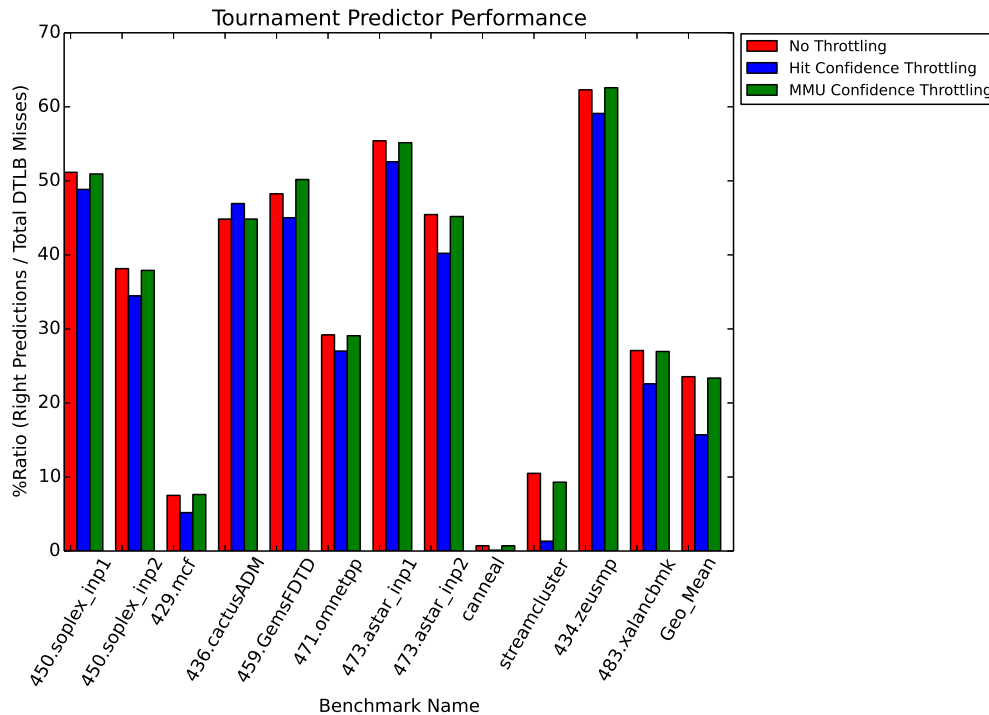
	perf	No Throttling	Hit Interval Throttling	MMU Interval Throttling
450.soplex_inp1	4.43%	2.12%	2.24%	2.39%
450.soplex_inp2	6.71%	4.13%	4.16%	4.88%
429.mcf	20.87%	19.26%	20.27%	20.35%
436.cactusADM	0.60%	0.57%	0.60%	0.57%
459.GemsFDTD	10.94%	5.72%	6.05%	6.31%
471.omnetpp	14.69%	10.38%	11.04%	11.80%
473.astar_inp1	4.93%	2.21%	2.30%	2.94%
473.astar_inp2	16.92%	9.27%	9.71%	12.75%
canneal	29.32%	29.11%	29.28%	29.28%
streamcluster	8.59%	7.69%	8.50%	8.50%
434.zeusmp	1.36%	0.60%	0.58%	0.72%
483.xalancbmk	8.41%	6.07%	6.26%	6.70%
Geo_Mean	7.02%	5.64%	6.28%	6.29%

**Πίνακας 5.7:** Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Υβριδικού Ακολουθιακού Προανακλητή. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf.

Ο Πίνακας 5.7 επιβεβαιώνει το Σχήμα 5.24, πως δηλαδή η No Throttling εκδοχή πετυχαίνει καλύτερη απόδοση. Στη γενική περίπτωση πετυχαίνει μείωση του ποσοστού των κύκλων κατά 19.5%, η οποία είναι αρκετά ικανοποιητική. Τα αποτελέσματα των Μηχανισμών Hit Interval Throttling και MMU Interval Throttling είναι παρόμοια και απέχουν πολύ από εκείνα της No Throttling εκδοχής.

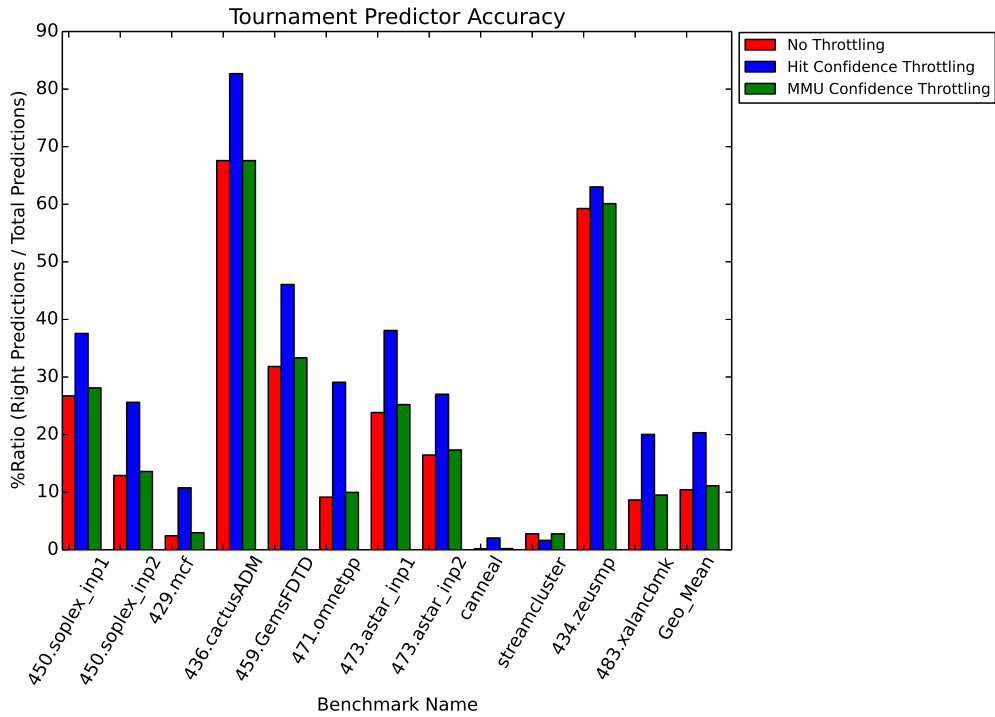
## 5.9 Αποτελέσματα Tournament Προβλέπτη

Στην ενότητα αυτή θα παρουσιαστούν τα αποτελέσματα του Tournament Προβλέπτη για τις μετρικές που αναλύθηκαν στο υποκεφάλαιο 4.5. Θα παρουσιαστούν επίσης τα αποτελέσματα της εισαγωγής των Μηχανισμών Hit Confidence Throttling και MMU Confidence Throttling.

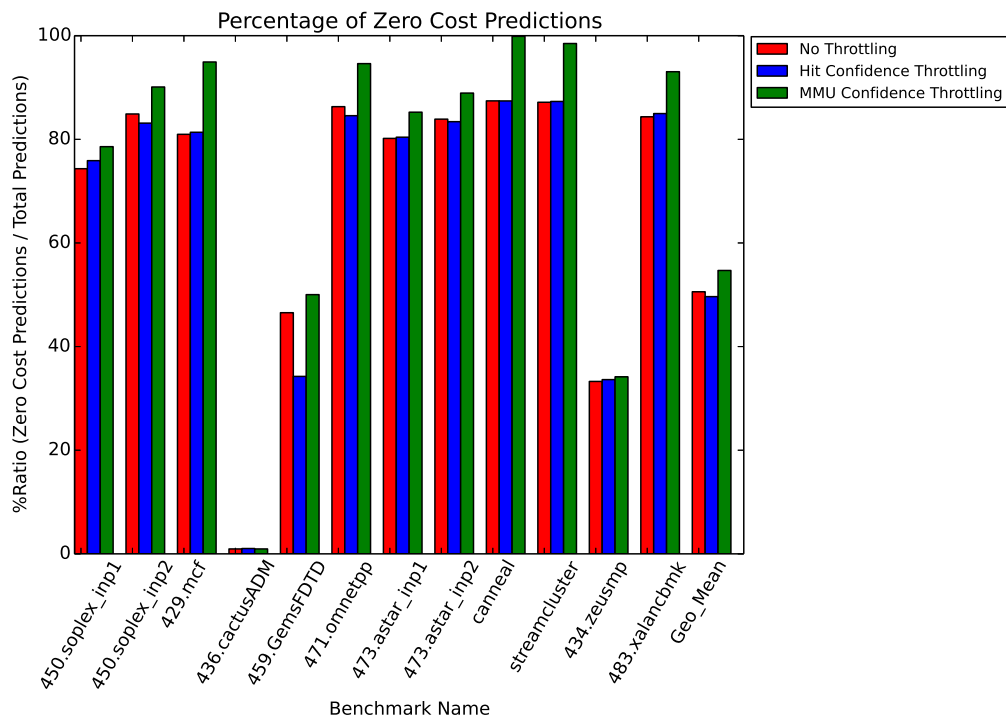


**Σχήμα 5.28:** Ποσοστό Επιτυχημένων Προβλέψεων Tournament Προβλέπτη για τις περιπτώσεις No Throttling, Hit Confidence Throttling, MMU Confidence Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.

Το Σχήμα 5.28 παρουσιάζει δύο πολύ σημαντικά πράγματα. Πρώτον, δείχνει ότι ο Tournament Προβλέπτης στη No Throttling εκδοχή του έχει τη καλύτερη δυνατή απόδοση για κάθε μετροπρόγραμμα, αφού καταφέρνει να ενεργοποιεί το καταλληλότερο Μηχανισμό Προανάκλησης για κάθε φάση του κάθε μετροπρογράμματος. Ουσιαστικά καταφέρνει να έχει απόδοση τουλάχιστον ίση με τη μεγαλύτερη απόδοση από τα τρία Σχήματα Προανάκλησης που συνδυάζει, για κάθε μετροπρόγραμμα. Το φαινόμενο αυτό αναφέρθηκε και στο υποκεφάλαιο 5.2. Δεύτερον, παρουσιάζει εξαιρετικό ενδιαφέρον η απόδοση που πετυχαίνει με χρήση του Μηχανισμού Κόστους MMU Confidence Throttling. Φαίνεται ότι ο μηχανισμός αυτός καταφέρνει να κόψει τις άχρηστες προβλέψεις της No Throttling εκδοχής, αφήνοντας αναλλοίωτη την απόδοσή της. Ο Μηχανισμός Ελέγχου Προανάκλησης Hit Confidence Throttling αφού η απόδοσή του διαφέρει από εκείνη του No Throttling Μηχανισμού.

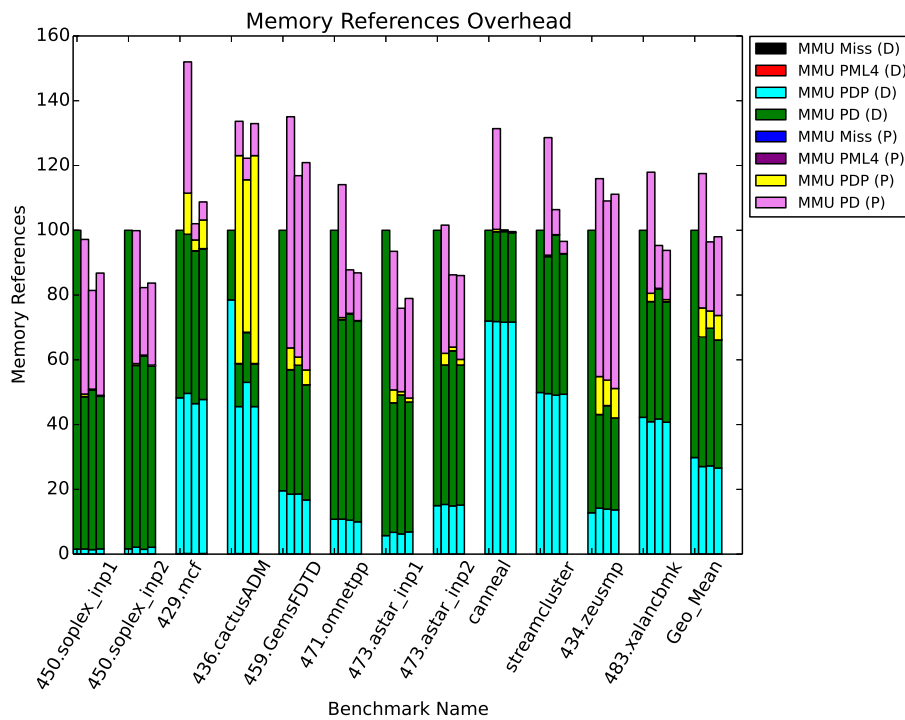


Σχήμα 5.29: Ποσοστό Ακρίβειας Προβλέψεων Tournament Προβλέπτη για τις περιπτώσεις No Throttling, Hit Confidence Throttling, MMU Confidence Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.



Σχήμα 5.30: Ποσοστό Προβλέψεων Μηδενικού Κόστους Tournament Προβλέπτη για τις περιπτώσεις No Throttling, Hit Confidence Throttling, MMU Confidence Throttling αντίστοιχα. Στο τέλος φαίνεται και ο γεωμετρικός μέσος όρων των αποτελεσμάτων.

Το Σχήμα 5.29 δείχνει πως ο Tournament Προβλέπτης στη No Throttling εκδοχή του, άλλοτε παρουσιάζει υψηλά ποσοστά ευστοχίας και άλλοτε χαμηλά. Το φαινόμενο αυτό είναι λογικό, αφού κατά τη εκτέλεση κάποιου μετροπρογράμματος γίνεται δυναμική επιλογή του καταλληλότερου Σχήματος Προανάκλησης. Χαρακτηριστικά παραδείγματα είναι τα μετροπρογράμματα 436.cactusADM και 434.zeusmp. Στο πρώτο καταφέρνει να έχει αυτό το ποσοστό ευστοχίας διότι στη μεγαλύτερη διάρκεια της εκτέλεσης του επιλέγει τον Προανακλητή Τυχαίου Βήματος, ο οποίος κάνει εύστοχες προβλέψεις για το 436.cactusADM, όπως παρουσιάστηκε στο Σχήμα 5.10. Στο δεύτερο καταφέρνει να έχει αυτό το ποσοστό ευστοχίας διότι στη μεγαλύτερη διάρκεια της εκτέλεσης του επιλέγει τον Προανακλητή Απόστασης, ο οποίος κάνει εύστοχες προβλέψεις για το 434.zeusmp, όπως παρουσιάστηκε στο Σχήμα 5.17. Το Σχήμα 5.30 κάνει περισσότερο εμφανές το γεγονός ότι ο Tournament Προβλέπτης ακολουθεί κάθε φορά τον καλύτερο Προανακλητή από εκείνους που τον αποτελούν. Βέβαια πολλές φορές μέχρι να βρει τον καλύτερο, ή σε φάσεις που μπορεί να δουλεύουν παραπάνω από ένας καλά, ενεργοποιεί και κάποιους από τους άλλους για να τους εκμεταλλευτεί. Αυτό οδηγεί για παράδειγμα σε μικρή αύξηση του ποσοστού των προβλέψεων Μηδενικού Κόστους για το 434.zeusmp. Ο Μηχανισμός Ελέγχου Προανάκλησης Hit Confidence Throttling έχει το μεγαλύτερο ποσοστό ευστοχίας, το οποίο είναι και λογικό αν αναλογιστεί κανείς ότι απορρίπτει πολύ περισσότερες προβλέψεις. Ο Μηχανισμός Ελέγχου Προανάκλησης MMU Confidence Throttling έχει μεγαλύτερα ποσοστά προβλέψεων Μηδενικού Κόστους αφού σε σχέση με τη No Throttling εκδοχή κάνει λιγότερες προβλέψεις και σε σχέση με την Hit Throttling εκδοχή δίνει μεγαλύτερη ελευθερία στη πραγματοποίηση Μηδενικού Κόστους προβλέψεων.



**Σχήμα 5.31:** Ποσοστό Επιβάρυνσης Λειτουργιών Μνήμης Tournament Προβλέπτη. Για κάθε μετροπρόγραμμα η πρώτη μπάρα παρουσιάζει τις απαιτούμενες λειτουργίες μνήμης αν δεν χρησιμοποιείται κανένα Σχήμα Προανάκλησης Μεταφράσεων. Οι επόμενες μπάρες παρουσιάζουν τα αποτελέσματα του Tournament Προβλέπτη για τις περιπτώσεις No Throttling, Hit Confidence Throttling, MMU Confidence Throttling αντίστοιχα. Το (D) αφορά τις TLB αστοχίες και το (P) τις προβλέψεις.

Η πεποίθηση πως ο Μηχανισμός MMU Confidence Throttling θα μειώσει τις απαιτούμενες λειτουργίες μνήμης, απαλοφώντας ένα μεγάλο ποσοστό των άχρηστων προβλέψεων με ταυτόχρονη διατήρηση της απόδοσης επιβεβαιώνεται πλήρως από το Σχήμα 5.31. Μπορεί ο Μηχανισμός Hit

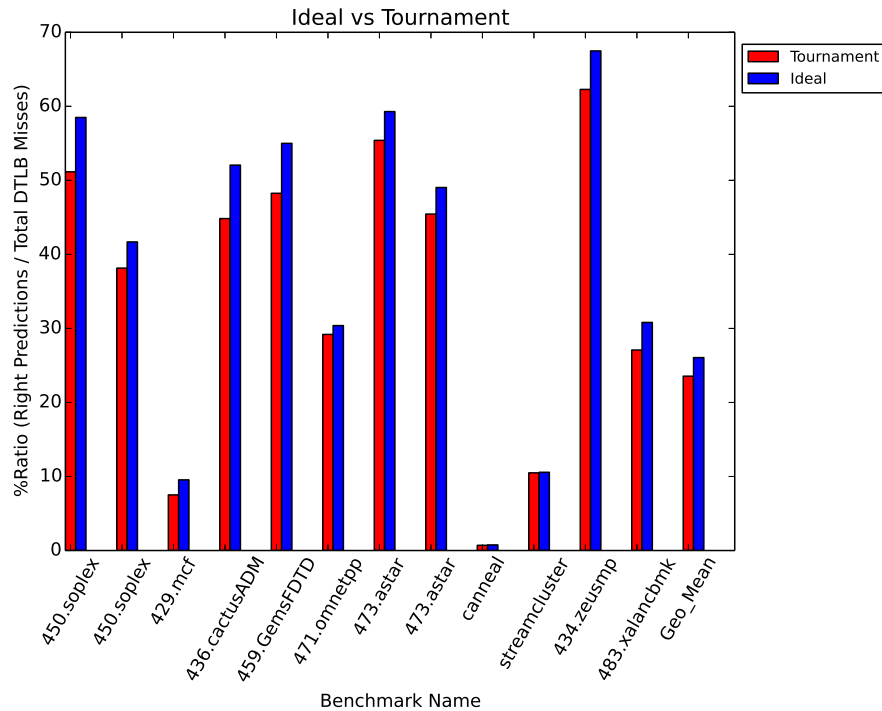
Confidence Throttling να παρουσιάζει μερικές φορές μεγαλύτερη μείωση αλλά μειώνει ταυτόχρονα και την απόδοση. Ο MMU Confidence Throttling Μηχανισμός διατηρεί την απόδοση στα επίπεδα της No Throttling εκδοχής και μειώνει σε όλα τα μετροπρογράμματα τις λειτουργίες μνήμης. Χαρακτηριστικά παραδείγματα μεγάλης μείωσης με ταυτόχρονη διατήρηση της απόδοσης είναι τα 429.mcf, streamcluster και 483.xalancbmk. Ακόμα και για το canneal που δεν καταφέρνει κάποια βελτίωση στην απόδοση καταφέρνει να μην επιβαρύνει το σύστημα με λειτουργίες μνήμης που δεν έχουν κανένα όφελος.

	perf	No Throttling	Hit Confidence Throttling	MMU Confidence Throttling
450.soplex_inp1	4.43%	2.16%	2.27%	2.17%
450.soplex_inp2	6.71%	4.15%	4.40%	4.17%
429.mcf	20.87%	19.30%	19.79%	19.28%
436.cactusADM	0.60%	0.33%	0.32%	0.33%
459.GemsFDTD	10.94%	5.66%	6.02%	5.45%
471.omnetpp	14.69%	10.40%	10.72%	10.42%
473.astar_inp1	4.93%	2.20%	2.34%	2.21%
473.astar_inp2	16.92%	9.23%	10.12%	9.28%
canneal	29.32%	29.11%	29.29%	29.11%
streamcluster	8.59%	7.69%	8.48%	7.79%
434.zeusmp	1.36%	0.51%	0.56%	0.51%
483.xalancbmk	8.41%	6.13%	6.51%	6.14%
Geo_Mean	7.02%	5.37%	5.92%	5.38%

**Πίνακας 5.8:** Αποτελέσματα γραμμικού μοντέλου επιτάχυνσης Tournament Προβλέπτη. Η στήλη perf περιλαμβάνει στο ποσοστό των κύκλων που ξοδεύει κάθε benchmark για τις TLB αστοχίες, σύμφωνα με το εργαλείο perf.

Ο Πίνακας 5.8 επιβεβαιώνει το Σχήμα 5.28, πως δηλαδή η No Throttling εκδοχή και η MMU Confidence Throttling εκδοχή πετυχαίνουν εξίσου καλά αποτελέσματα. Στη γενική περίπτωση πετυχαίνει μείωση του ποσοστού των κύκλων κατά 23.5%, η οποία είναι και η μεγαλύτερη μείωση ανάμεσα σε όλους τους Μηχανισμούς Προανάκλησης Μεταφράσεων που υλοποιήθηκαν στη μελέτη αυτή. Τα αποτελέσματα του Μηχανισμού Hit Confidence Throttling απέχουν πολύ από εκείνα των άλλων δύο μηχανισμών.

Όλα τα γραφήματα και οι σχολιασμοί που προηγήθηκαν ανέδειξαν ότι ο Tournament Προβλέπτης, και συγκεκριμένα η εκδοχή του MMU Confidence Throttling, είναι ο καλύτερος Μηχανισμός Προανάκλησης της εν λόγω μελέτης. Το Σχήμα 5.32 δείχνει την απόκλιση όσο αφορά την απόδοση που πετυχαίνει ο Tournament Προβλέπτης από την ιδανική περίπτωση, στην οποία κάθε Μηχανισμός Πρόβλεψης που τον αποτελεί έχει τον δικό του Πίνακα Προβλέψεων (Prefetch Buffer) 16 καταχωρήσεων και παράγει αυτόνομα προβλέψεις. Θα πρέπει να τονιστεί ότι επιτυχής πρόβλεψη κάποιας TLB αστοχίας από κάποιο μηχανισμό αγνοεί πιθανή επιτυχημένη πρόβλεψη κάποιου από τους υπολοίπους.

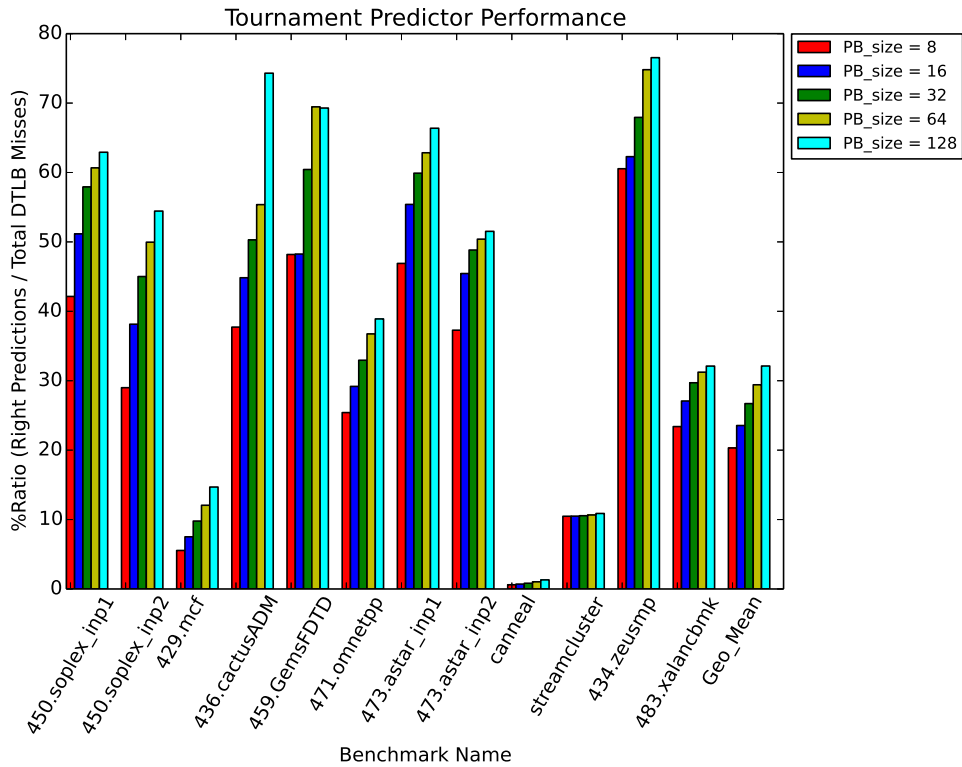


Σχήμα 5.32: Σύγκριση απόδοσης Tournament Προβλέπτη με την ιδανική περίπτωση απόδοσης.

Απο το Σχήμα 5.32 παρατηρούμε ότι ο Tournament Προβλέπτης δεν απέχει πολύ από την ιδανική περίπτωση. Αυτό σημαίνει ότι ο Tournament Προβλέπτης καταφέρνει να παίρνει εξαιρετικά εύστοχες αποφάσεις, ειδικά στην εκδοχή MMU Confidence Throttling στην οποία αποφεύγει και πολλές άχρηστες προβλέψεις. Αν αναλογιστεί κανείς πως στην ιδανική περίπτωση λειτουργούν τρία διαφορετικά Σχήματα Προανάκλησης Μεταφράσεων αυτόνομα και ταυτόχρονα, αντιλαμβάνεται πως παρά το γεγονός ότι καταφέρνει λίγο καλύτερα αποτελέσματα απόδοσης, η επιβάρυνση που εισάγει στο σύστημα είναι πολύ μεγάλη, αφού για κάθε benchmark η επιβάρυνση είναι σε μεγάλο βαθμό ίση με το άθροισμα των επιβαρύνσεων που φαίνονται στα Σχήματα 5.8, 5.12 και 5.19.

## 5.10 Μελέτη Παραμέτρων

Σε όλα τα αποτελέσματα μέχρι στιγμής θεωρούμε ότι ο Prefetch Buffer έχει πάντα 16 καταχωρήσεις. Στην ενότητα αυτή θα παρουσιαστεί ο τρόπος με τον οποίο επηρεάζονται τα αποτελέσματα απόδοσης του Tournament Προβλέπτη για διαφορετικά μεγέθη Prefetch Buffer. Θα πρέπει να τονιστεί ότι παράλληλα με το μέγεθος του πραγματικού Prefetch Buffer επηρεάζονται και τα μεγέθη των Fake Prefetch Buffers με ανάλογο τρόπο. Το Σχήμα 5.33 παρουσιάζει τα αποτελέσματα.



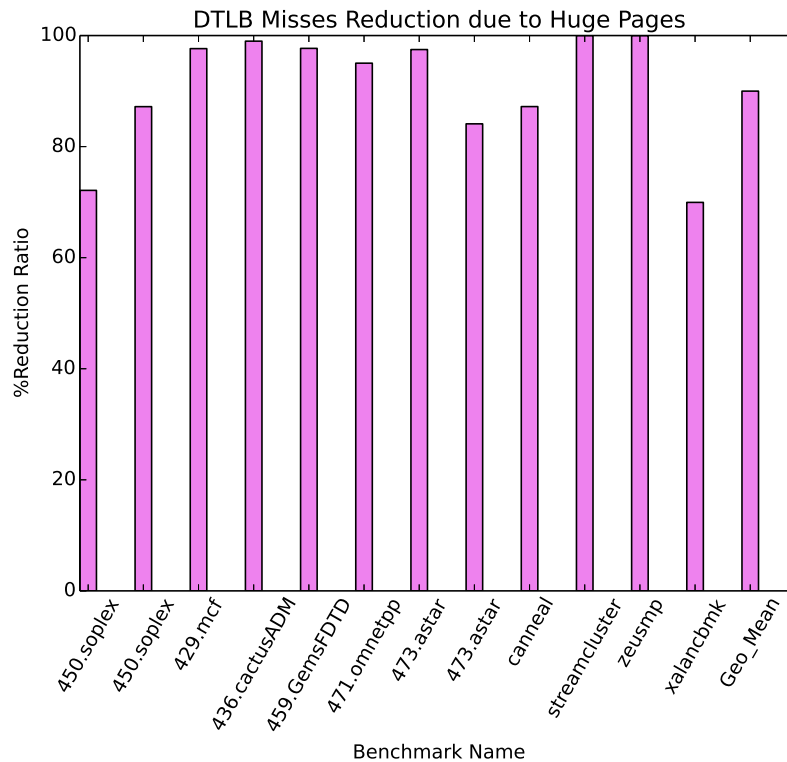
Σχήμα 5.33: Σύγκριση απόδοσης Tournament Προβλέπτη με την ιδανική περίπτωση απόδοσης.

Από τα αποτελέσματα του Σχήματος 5.33 φαίνεται ότι αύξηση του μεγέθους του Prefetch Buffer οδηγεί σε αύξηση της απόδοσης. Το γεγονός αυτό είναι λογικό αφού πλέον μπορούν να γίνουν περισσότερες προβλέψεις χωρίς να χρειαστεί να απομακρυνθεί κάποια άλλη λόγω μειωμένης χωρητικότητας. Αν εξαιρεθεί το benchmark 436.cactusADM, τα υπόλοιπα δεν φαίνεται να αυξάνουν κατά πολύ την απόδοση τους όταν το μέγεθος του Prefetch Buffer αλλάξει από 64 σε 128, παρά το μεγάλο μέγεθος του. Τα αποτελέσματα απόδοσης που παρουσιάζονται όταν ο Prefetch Buffer είναι 64 καταχωρήσεων είναι αρκετά ικανοποιητικά και παράλληλα το μέγεθος του δεν είναι συγκρίσιμο με αυτό της TLB του μηχανήματος που γίνονται τα πειράματα. Βέβαια το αρνητικό στοιχείο του Σχήματος 5.33 είναι πως όσο μεγαλώνει το μέγεθος του Prefetch Buffer, η απόδοση των benchmarks canneal και streamcluster δεν αλλάζει, γεγονός το οποίο σημαίνει πως οι Μηχανισμοί Πρόβλεψης που χρησιμοποιεί ο Tournament Προβλέπτης δεν είναι ικανοί να κάνουν εύστοχες προβλέψεις γι' αυτά, ανεξαρτήτως του μεγέθους του Prefetch Buffer.



## 5.11 Huge Pages

Στην ενότητα αυτή θα γίνει παρουσίαση της επίδρασης του μηχανισμού των Huge Pages όσο αφορά τον αριθμό των TLB αστοχιών. Το Σχήμα 5.34 παρουσιάζει τα αποτελέσματα.



Σχήμα 5.34: Ποσοστό μείωσης TLB αστοχιών λόγω ενεργοποίησης του μηχανισμού των Huge Pages.

Το Σχήμα 5.34 παρουσιάζει εξαιρετικά μεγάλη μείωση των αστοχιών της TLB για τα benchmarks που εξετάζονται στη μελέτη αυτή όταν ενεργοποιείται ο μηχανισμός των Huge Pages. Ωστόσο τα Huge Pages δεν λειτουργούν τόσο καλά σε όλες τις περιπτώσεις. Αν για παράδειγμα η μνήμη του υπολογιστικού συστήματος είχε υποστεί κατακερματισμό (fragmentation) τότε τα διάφορα αιτήματα για δέσμευση Huge Pages δεν θα μπορούσαν να ικανοποιηθούν και θα ήταν μονόδρομος η χρησιμοποίηση των 4KB pages. Το φαινόμενο του fragmentation είναι πολύ συχνό, ειδικά για μηχανήματα που λειτουργούν για πολύ καιρό ή υπό μεγάλο φόρτο εργασίας. Ωστόσο, αυτό δεν είναι το μόνο ενδεχόμενο τα Huge Pages να μην λειτουργούν καλά. Υπάρχουν εφαρμογές οι οποίες έχουν πολύ μεγαλύτερες απαιτήσεις τόσο για μνήμη όσο και για ενεργή μνήμη από τα benchmarks που χρησιμοποιούνται στη μελέτη αυτή. Οι εφαρμογές αυτές καταφέρνουν να χαλάσουν σε μεγάλο βαθμό και την απόδοση των Huge Pages και εμφανίζουν εκ νέου το πρόβλημα της επιβάρυνσης που εισάγεται σε ένα υπολογιστικό σύστημα λόγω πολλαπλών TLB αστοχιών.



## Κεφάλαιο 6

### Σχετική Βιβλιογραφία

Στο κεφάλαιο 3, αναλύθηκαν τα Σχήματα Προανάκλησης Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης (TLB) τα οποία υλοποιήθηκαν για τις ανάγκες της μελέτης αυτής. Πρόκειται για Σχήματα Προανάκλησης Μεταφράσεων που θεωρούν υπολογιστικό μηχάνημα με ένα μόνο πυρήνα, χωρίς να ενδιαφέρονται και να επηρεάζονται από την ύπαρξη περισσότερων. Στο κεφάλαιο αυτό θα γίνει παρουσίαση και άλλων Σχημάτων Προανάκλησης Μεταφράσεων που έχουν προταθεί τόσο για μονοπύρηνα όσο και για πολυπύρηνα υπολογιστικά μηχανήματα. Τέλος θα αναφερθούν μηχανισμοί που προσπαθούν να βελτιώσουν την απόδοση της TLB αλλά δεν χρησιμοποιούν Σχήματα Προανάκλησης Μεταφράσεων.

#### 6.1 Μηχανισμοί Προανάκλησης Μεταφράσεων για Μονοπύρηνα Συστήματα

Στην ενότητα αυτή θα γίνει παρουσίαση διάφορων Μηχανισμών Προανάκλησης Μεταφράσεων για Μονοπύρηνα Συστήματα που δεν επιλέχθηκαν να υλοποιηθούν στην εν λόγω μελέτη.

1. *Μαρκοβιανός Μηχανισμός Προανάκλησης.* Ο Μαρκοβιανός Μηχανισμός Προανάκλησης (Markov Prefetching), δημιουργεί δυναμικά ένα Markov διάγραμμα κατάστασης για να ανακαλύψει πιθανή ιστορία στις αστοχίες της TLB. Χρησιμοποιεί ένα πίνακα, ο οποίος δεικτοδοτείται από τη σελίδα στην οποία οφείλεται η TLB αστοχία. Περισσότερες λεπτομέριες για το μηχανισμό αυτό παρέχονται στο [30].
2. *Μηχανισμός Προανάκλησης βασισμένος στην Επικαιρότητα.* Ο συγκεκριμένος Μηχανισμός Προανάκλησης (Recency Based Mechanism), στηρίζεται στην ιδέα ότι σελίδες που αναφέρθηκαν στο παρελθόν σε κοντινές χρονικές στιγμές θα επαναληφθεί και στο μέλλον. Για να το καταφέρει αυτό, προσθέτει δύο επιπλέον πεδία σε κάθε καταχώρηση του Πίνακα Σελίδων (Page Table), τα οποία είναι δείκτες σε κάποια άλλη καταχώρηση. Με το τρόπο αυτό χτίζει δυναμικά μία στοίβα (stack). Περισσότερες λεπτομέριες για το μηχανισμό αυτό παρέχονται στο [39].
3. *Μηχανισμός Προανάκλησης Τυχαίου Βήματος με Χρήση του Μετρητή Προγράμματος.* Ο Μηχανισμός Προανάκλησης Τυχαίου Βήματος (Arbitrary Stride Prefetching) με Χρήση του Μετρητή Προγράμματος (PC) προσπαθεί να ανακαλύψει κανονικότητα όσο αφορά τις αποστάσεις μεταξύ των σελίδων, των οποίων η αναζήτηση μετάφρασης στη TLB οδήγησε σε αποτυχία. Για να πετύχει το σκοπό του χρησιμοποιεί ένα Πίνακα Αναφορών δεικτοδοτούμενο από το PC και μία Μηχανή Πεπερασμένης Κατάστασης τεσσάρων καταστάσεων. Περισσότερες λεπτομέριες για το μηχανισμό αυτό παρέχονται στο [30].
4. *Ακολουθιακός Μηχανισμός Προανάκλησης.* Πρόκειται για τον πλέον απλό μηχανισμό προανάκλησης. Η υλοποίηση του είναι πολύ εύκολη, αφού το μόνο που κάνει είναι να προβλέπει σε κάθε TLB αστοχία την ακριβώς επόμενη σελίδα από εκείνη που ευθύνεται για την αστοχία. Πρόκειται για ένα μηχανισμό που αποτελεί υποσύνολο του μηχανισμού που αναλύθηκε στο υποκεφάλαιο 3.3.1. Περισσότερες λεπτομέριες για το μηχανισμό αυτό παρέχονται στο [30].

## 6.2 Μηχανισμοί Προανάκλησης Μεταφράσεων για Πολυπύρρηνα Συστήματα

Στην ενότητα αυτή θα γίνει παρουσίαση διάφορων Μηχανισμών Προανάκλησης Μεταφράσεων για Πολυπύρρηνα Συστήματα.

1. *Μηχανισμός Προανάκλησης Αρχηγού-Ακόλουθου*. Ο Μηχανισμός Προανάκλησης Αρχηγού-Ακόλουθου (Leader-Follower prefetching) στηρίζεται στην ιδέα ότι αν υπάρξει αστοχία εύρεσης κάποιας μετάφρασης στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης κάποιου από τους πυρήνες του συστήματος (Αρχηγός), τότε πρόκειται να συμβεί η ίδια αστοχία και στους υπόλοιπους πυρήνες (Ακόλουθοι). Με βάση αυτή την ιδέα, ο Αρχηγός είναι υπεύθυνος να μεταφέρει τη μετάφραση και στις TLBs των άλλων πυρήνων ώστε να αποφύγουν την ίδια αστοχία. Περισσότερες λεπτομέρειες για το μηχανισμό αυτό παρέχονται στο [18].
2. *Μηχανισμός Προανάκλησης βασισμένος στην Απόσταση*. Ο Μηχανισμός Προανάκλησης βασισμένος στην Απόσταση (Distance-Based Cross-Core Prefetching) βασίζεται στην ιδέα που υλοποιείται στο [30]. Διατηρεί την ακολουθία των αποστάσεων διαδοχικών αστοχιών της TLB για καθένα πυρήνα και τις ομαδοποιεί. Προσπαθεί να παράγει σωστές προβλέψεις ακομά και στη περίπτωση που οι πυρήνες έχουν αστοχίες σε διαφορετικές διευθύνσεις, αφού χρησιμοποιεί σαν στοιχείο σύγκρισης την ακολουθία διαδοχικών αποστάσεων. Περισσότερες λεπτομέρειες για το μηχανισμό αυτό παρέχονται στο [18].

## 6.3 Μηχανισμοί χωρίς Προανάκληση Μεταφράσεων

Στην ενότητα αυτή θα γίνει παρουσίαση διάφορων μηχανισμών βελτίωσης της απόδοσης της TLB, οι οποίοι δεν χρησιμοποιούν Σχήματα Προανάκλησης Μεταφράσεων στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης.

Διάφορες υλοποιήσεις στηρίζονται στη μετάφραση πολλαπλών σελίδων με μία μόνο καταχώρηση της TLB (Multipage mappings), όπως οι [20, 31, 35, 36, 37, 41]. Στη συνέχεια αναλύονται δύο από τις ερευνητικές προσπάθειες που έγιναν για Multipage mappings.

1. *Επαρκείς Αντιστοιχίσεις Μνήμης (Redundant Memory Mappings)*. Ο μηχανισμός αυτός εισάγει την έννοια του εύρους σελίδων, το οποίο είναι ένα υποσύνολο των σελίδων κάποιας εφαρμογής, το οποίο έχει ως χαρακτηριστικό ότι είναι συνεχές τόσο στο φυσικό όσο και στον εικονικό χώρο διευθύνσεων. Υλοποιεί ένα Πίνακα Εύρους Σελίδων και μία Κρυφή Μνήμη Αναζήτησης Εύρους Σελίδων για να εκμεταλλευτεί την ιδέα αυτή. Περισσότερες λεπτομέρειες για το μηχανισμό αυτό παρέχονται στο [31].
2. *Συγχωνευμένα TLB Μεγάλης Εμβέλειας (Coalesced Large-Reach TLBs)*. Ο μηχανισμός αυτός προσπαθεί να εκμεταλλευτεί τη συνέχεια που παρουσιάζεται μεταξύ ορισμένων φυσικών και εικονικών σελίδων. Στη πράξη συγχωνεύει σε κάθε καταχώρηση της TLB, πολλαπλές μεταφράσεις εικονικών σε φυσικές σελίδες. Περισσότερες λεπτομέρειες για το μηχανισμό αυτό παρέχονται στο [37].

Τα Direct Segments ήταν άλλη μία ιδέα η οποία χρησιμοποιήθηκε σε ερευνητικές μελέτες για τη βελτίωση της απόδοσης της TLB. Χαρακτηριστικά παραδείγματα αποτελούν οι προτάσεις των [16, 22].

Επιπρόσθετες ερευνητικές δουλειές πάνω στη βελτίωση της απόδοσης της TLB είναι οι [15] και [43].

## Κεφάλαιο 7

### Συμπεράσματα και Μελλοντικές Επεκτάσεις

Σκοπός της μελέτης αυτής ήταν αρχικά η εύρεση της επιβάρυνσης που εισάγεται σε ένα υπολογιστικό σύστημα, το οποίο χρησιμοποιεί τη τεχνική της Εικονικής Μνήμης, από εφαρμογές που προκαλούν πολλαπλές αστοχίες στη Κρυφή Μνήμη Αναζήτησης Μετάφρασης (TLB). Στη συνέχεια χρησιμοποιείται η Τεχνική της Προανάκλησης Μεταφράσεων στη TLB ώστε να μειωθεί το μέγεθος της επιβάρυνσης αυτής. Υλοποιήθηκαν και αξιολογήθηκαν διαφορετικά Σχήματα Προανάκλησης Μεταφράσεων στη TLB, αναλυτική επεξήγηση και περιγραφή των οποίων παρέχεται στο Κεφάλαιο 3. Τέλος ενσωματώθηκαν στα Σχήματα αυτά τέσσερις διαφορετικές υλοποιήσεις κόστους για την αποφυγή προβλέψεων που δεν προσφέρουν τίποτα στη βελτίωση της απόδοσης της TLB, μειώνοντας παράλληλα και το κόστος σε λειτουργίες μνήμης που απαιτούνται για τις προβλέψεις.

#### 7.1 Συμπεράσματα

Τα αποτελέσματα της πειραματικής αξιολόγησης για καθένα από τα Σχήματα TLB Prefetching και τους Μηχανισμούς Ελέγχου Προανάκλησης, τα οποία παρουσιάστηκαν στο Κεφάλαιο 5 έδειξαν ότι ο καλύτερος Μηχανισμός TLB Prefetching της μελέτης αυτής είναι ο Tournament Προβλέπτης με MMU Confidence Throttling. Καταφέρνει να εκμεταλλεύεται και τους τρεις μηχανισμούς που τον αποτελούν με το καλύτερο δυνατό τρόπο, όπως φάνηκε και από τη σύγκριση του με τα ιδανικά αποτελέσματα απόδοσης. Ταυτόχρονα αποκόπτει πολλές από τις άχρηστες προβλέψεις που κάνει η No Throttling εκδοχή του διατηρώντας την ίδια απόδοση. Έτσι, καταφέρνει να μειώσει τις απαιτούμενες λειτουργίες μνήμης για την πραγματοποίηση των προβλέψεων του, γλιτώνοντας επιπρόσθετη επιβάρυνση του υπολογιστικού συστήματος. Ένα ακόμα θετικό του Tournament Προβλέπτη με MMU Confidence Throttling είναι πως χρησιμοποιεί λίγες δομές δεδομένων μικρού μεγέθους, των οποίων η προσπέλαση μπορεί να γίνει παράλληλα από το hardware και το access time κάθε δομής είναι μικρό λόγω του μεγέθους τους. Ο τρόπος υλοποίησης του είναι τέτοιος ώστε η εισαγωγή νέων Μηχανισμών Προανάκλησης ή η αντικατάσταση κάποιων από τους υπάρχοντες να απαιτεί ελάχιστη δουλειά και τροποποιήσεις. Ένας επίσης αξιοσημείωτος Μηχανισμός Προανάκλησης είναι και ο Προανακλητής Μηδενικού Κόστους (Free Prefetcher), ο οποίος παρά τη σχετικά χαμηλή απόδοση που καταφέρνει στα περισσότερα μετροπρογράμματα, καταφέρνει να μειώσει κατά ένα ποσοστό τις λειτουργίες μνήμης που θα υπήρχαν στη περίπτωση που δεν υπήρχε Σχήμα Προανάκλησης Μεταφράσεων και ταυτόχρονα δεν εισάγει καμία επιπρόσθετη επιβάρυνση για τις προβλέψεις του. Ίσως για Πίνακες Προβλέψεων μεγαλύτερου μεγέθους από 16 καταχωρήσεις, η εισαγωγή του Free Prefetcher στον Tournament Προβλέπτη να είχε πρακτικό ενδιαφέρον. Για Πίνακα Προβλέψεων 16 καταχωρήσεων δεν έχει αφού ο Free Prefetcher είναι ένα πολύ επιθετικό Σχήμα Προανάκλησης Μεταφράσεων.

## 7.2 Μελλοντικές Επεκτάσεις

Ο μηχανισμός των Huge Pages έδειξε πως λειτουργεί πολύ καλά για τα benchmarks της μελέτης αυτής. Ήδη στο υποκεφάλαιο 5.11 παρουσιάστηκαν περιπτώσεις στις οποίες ο μηχανισμός των Huge Pages δεν έχει θετικά αποτελέσματα. Είναι πολύ πιθανό πως ο μηχανισμός των Huge Pages σε σύντομο χρονικό διάστημα δεν θα είναι αρκετός για να διατηρεί τις αστοχίες της TLB σε χαμηλά επίπεδα, για τη πλειοψηφία των απαιτητικών εφαρμογών. Για το λόγο αυτό, προτείνεται για έρευνα η εισαγωγή Μηχανισμών Προανάκλησης Μεταφράσεων και για 2MB pages. Συνύπαρξη Μηχανισμών Προανάκλησης Μεταφράσεων τόσο για 4KB όσο και για 2MB pages κρίνεται πως θα είχε ιδιαίτερο ερευνητικό ενδιαφέρον αφού θα μπορούσε να αντιμετωπίσει εκτός από τις απαιτητικές εφαρμογές που προκαλούν πολλές TLB αστοχίες και το πρόβλημα του κατακερματισμού της μνήμης (fragmentation). Έντονο ενδιαφέρον έχει επίσης η εισαγωγή Μηχανισμών Ελέγχου Προανάκλησης στους Μηχανισμούς Προανάκλησης Μεταφράσεων τόσο για 2MB όσο και για 4KB pages, όπως αυτός του MMU Confidence Throttling που προτείνεται στη μελέτη αυτή, για την αποφυγή άχρηστων προβλέψεων που εισάγουν επιπρόσθετες λειτουργίες μνήμης και τον εντοπισμό εφαρμογών των οποίων οι προσβάσεις στη μνήμη δεν μπορούν να προβλεφθούν σωστά.

Μία ακόμα ιδέα προς υλοποίηση είναι η ανάλυση της ενέργειας που καταναλώνουν τα διάφορα Σχήματα Προανάκλησης Μεταφράσεων που αναλύθηκαν παραπάνω, στη προσπάθεια τους να παράγουν χρήσιμες προβλέψεις. Επίσης σημαντική κρίνεται και η ανάπτυξη όλων αυτών των Μηχανισμών Προανάκλησης και για πολυπύρηννα συστήματα για την εκμετάλλευση των ομοιοτήτων που παρουσιάζουν οι TLB αστοχίες σε κάθε πυρήνα ενός συστήματος. Τέλος, εκτός από τους Μηχανισμούς Προανάκλησης Μεταφράσεων στους οποίους είναι ενσωματωμένοι Μηχανισμοί Κόστους, ερευνητικό ενδιαφέρον θα είχε η συνύπαρξη τους με Μηχανισμούς όπως ο TEMPO (translation-enabled memory prefetching optimizations) [14], ο οποίος αγνοεί τις TLB αστοχίες αλλά προσπαθεί να γλιτώσει την επιβάρυνση που εισάγει η φάση στην οποία έχει γίνει η μετάφραση της διεύθυνσης και γίνεται η εξερεύνηση της ιεραρχίας μνήμης πλέον χρησιμοποιώντας τη φυσική διεύθυνση.

## Βιβλιογραφία

- [1] BadgerTrap: A Tool to Instrument TLB Misses. <http://research.cs.wisc.edu/multifacet/BadgerTrap/>. [Online].
- [2] BadgerTrap Documentation. [http://research.cs.wisc.edu/multifacet/papers/can14\\_badgertrap.pdf](http://research.cs.wisc.edu/multifacet/papers/can14_badgertrap.pdf). [Online].
- [3] Combining Branch Predictors. <http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-TN-36.pdf>. online.
- [4] Huge Pages Introduction. <https://lwn.net/Articles/374424/>. [Online].
- [5] Intel®64 and IA-32 Architectures Developer's Manual: Vol. 3B. <https://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-software-developer-vol-3b-part-2-manual.html>. Cited Table 19-3 p.299.
- [6] OMNeTT++: Discrete Event Simulator. <https://www.omnetpp.org/>. [Online].
- [7] Parsec CSWiki. <http://wiki.cs.princeton.edu/index.php/PARSEC>. [Online].
- [8] Parsec Suite. <http://parsec.cs.princeton.edu/>. [Online].
- [9] perf: Linux profiling with performance counters. [https://perf.wiki.kernel.org/index.php/Main\\_Page](https://perf.wiki.kernel.org/index.php/Main_Page). [Online].
- [10] SPEC CPU2006 Documentation. <https://www.spec.org/cpu2006/Docs/>. [Online].
- [11] SPEC: Standard Performance Evaluation Corporation. <https://www.spec.org/>. [Online].
- [12] Transparent Huge Pages. <https://lwn.net/Articles/423584/>. [Online].
- [13] ANDERSON, T. E., LEVY, H. M., BERSHAD, B. N., AND LAZOWSKA, E. D. The interaction of architecture and operating system design. *SIGARCH Comput. Archit. News* 19, 2 (Apr. 1991), 108–120.
- [14] BARR, T. W., COX, A. L., AND RIXNER, S. Translation caching: Skip, don't walk (the page table). *SIGARCH Comput. Archit. News* 38, 3 (June 2010), 48–59.
- [15] BARR, T. W., COX, A. L., AND RIXNER, S. Spectlb: A mechanism for speculative address translation. *SIGARCH Comput. Archit. News* 39, 3 (June 2011), 307–318.
- [16] BASU, A., GANDHI, J., CHANG, J., HILL, M. D., AND SWIFT, M. M. Efficient virtual memory for big memory servers. *SIGARCH Comput. Archit. News* 41, 3 (June 2013), 237–248.
- [17] BHATTACHARJEE, A. Large-reach Memory Management Unit Caches. In *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture* (New York, NY, USA, 2013), MICRO-46, ACM, pp. 383–394.
- [18] BHATTACHARJEE, A., AND MARTONOSI, M. Inter-core cooperative tlb for chip multiprocessors. *SIGARCH Comput. Archit. News* 38, 1 (Mar. 2010), 359–370.

- [19] CLARK, D. W., AND EMER, J. S. Performance of the vax-11/780 translation buffer: Simulation and measurement. *ACM Trans. Comput. Syst.* 3, 1 (Feb. 1985), 31–62.
- [20] COX, G., AND BHATTACHARJEE, A. Efficient address translation for architectures with multiple page sizes. *SIGPLAN Not.* 52, 4 (Apr. 2017), 435–448.
- [21] FAN, D., TANG, Z., HUANG, H., AND GAO, G. R. An energy efficient tlb design methodology. In *Proceedings of the 2005 International Symposium on Low Power Electronics and Design* (New York, NY, USA, 2005), ISLPED '05, ACM, pp. 351–356.
- [22] GANDHI, J., BASU, A., HILL, M. D., AND SWIFT, M. M. Efficient memory virtualization: Reducing dimensionality of nested page walks. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture* (Washington, DC, USA, 2014), MICRO-47, IEEE Computer Society, pp. 178–189.
- [23] HUCK, J., AND HAYS, J. Architectural support for translation table management in large address space machines. *SIGARCH Comput. Archit. News* 21, 2 (May 1993), 39–50.
- [24] JACOB, B., AND MUDGE, T. Virtual memory in contemporary microprocessors. *IEEE Micro* 18, 4 (July 1998), 60–75.
- [25] JACOB, B., AND MUDGE, T. Virtual memory: Issues of implementation. *Computer* 31, 6 (June 1998), 33–43.
- [26] JACOB, B. L., AND MUDGE, T. N. A look at several memory management units, tlb-refill mechanisms, and page table organizations. *SIGPLAN Not.* 33, 11 (Oct. 1998), 295–306.
- [27] JUAN, T., LANG, T., AND NAVARRO, J. J. Reducing tlb power requirements. In *Proceedings of the 1997 International Symposium on Low Power Electronics and Design* (New York, NY, USA, 1997), ISLPED '97, ACM, pp. 196–201.
- [28] KADAYIF, I., NATH, P., KANDEMIR, M. T., AND SIVASUBRAMANIAM, A. Compiler-directed physical address generation for reducing dtlb power. In *ISPASS* (2004).
- [29] KADAYIF, I., SIVASUBRAMANIAM, A., KANDEMIR, M., KANDIRAJU, G., AND CHEN, G. Generating physical addresses directly for saving instruction tlb energy. In *Proceedings of the 35th Annual ACM/IEEE International Symposium on Microarchitecture* (Los Alamitos, CA, USA, 2002), MICRO 35, IEEE Computer Society Press, pp. 185–196.
- [30] KANDIRAJU, G. B., AND SIVASUBRAMANIAM, A. Going the distance for tlb prefetching: An application-driven study. *SIGARCH Comput. Archit. News* 30, 2 (May 2002), 195–206.
- [31] KARAKOSTAS, V., GANDHI, J., AYAR, F., CRISTAL, A., HILL, M. D., MCKINLEY, K. S., NEMIROVSKY, M., SWIFT, M. M., AND ÜNSAL, O. Redundant memory mappings for fast access to large memories. *SIGARCH Comput. Archit. News* 43, 3 (June 2015), 66–78.
- [32] KARAKOSTAS, V., ÜNSAL, O. S., NEMIROVSKY, M., CRISTAL, A., AND SWIFT, M. M. Performance analysis of the memory management unit under scale-out workloads. In *Proceedings of the 2014 IEEE International Symposium on Workload Characterization, IISWC 2014, Raleigh, NC, USA, October 26-28, 2014* (2014), IEEE Computer Society, pp. 1–12.
- [33] MCCURDY, C., VETTER, J., AND COXA, A. L. Investigating the tlb behavior of high-end scientific applications on commodity microprocessors. In *Proceedings of the ISPASS 2008 - IEEE International Symposium on Performance Analysis of Systems and software(ISPASS)* (04 2008), vol. 00, pp. 95–104.



- [34] NAGLE, D., UHLIG, R., STANLEY, T., SECHREST, S., MUDGE, T., AND BROWN, R. Design tradeoffs for software-managed tlbs. *SIGARCH Comput. Archit. News* 21, 2 (May 1993), 27–38.
- [35] PARK, C. H., HEO, T., JEONG, J., AND HUH, J. Hybrid tlb coalescing: Improving tlb translation coverage under diverse fragmented memory allocations. *SIGARCH Comput. Archit. News* 45, 2 (June 2017), 444–456.
- [36] PHAM, B., BHATTACHARJEE, A., ECKERT, Y., AND LOH, G. H. Increasing TLB reach by exploiting clustering in page translations. In *Proceedings of the 20th IEEE International Symposium on High Performance Computer Architecture, HPCA 2014, Orlando, FL, USA, February 15-19, 2014* (2014), IEEE Computer Society, pp. 558–567.
- [37] PHAM, B., VAIDYANATHAN, V., JALEEL, A., AND BHATTACHARJEE, A. Colt: Coalesced Large-Reach TLBs. In *Proceedings of the 45th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2012, Vancouver, BC, Canada, December 1-5, 2012* (2012), IEEE Computer Society, pp. 258–269.
- [38] ROSENBLUM, M., BUGNION, E., DEVINE, S., AND HERROD, S. A. Using the simos machine simulator to study complex computer systems. *ACM Trans. Model. Comput. Simul.* 7, 1 (Jan. 1997), 78–103.
- [39] SAULSBURY, A., DAHLGREN, F., AND STENSTRÖM, P. Recency-based tlb preloading. *SIGARCH Comput. Archit. News* 28, 2 (May 2000), 117–127.
- [40] SWANSON, M., STOLLER, L., AND CARTER, J. Increasing tlb reach using superpages backed by shadow memory. *SIGARCH Comput. Archit. News* 26, 3 (Apr. 1998), 204–213.
- [41] TALLURI, M., AND HILL, M. D. Surpassing the tlb performance of superpages with less operating system support. *SIGPLAN Not.* 29, 11 (Nov. 1994), 171–182.
- [42] VANDERWIEL, S. P., AND LILJA, D. J. Data prefetch mechanisms. *ACM Comput. Surv.* 32, 2 (June 2000), 174–199.
- [43] YANIV, I., AND TSAFRIR, D. Hash, don’t cache (the page table). *SIGMETRICS Perform. Eval. Rev.* 44, 1 (June 2016), 337–350.