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

Δυναμική Τοποθέτηση Εικονοποιημένων Δικτυακών Λειτουργιών Σε Πολυεπεξεργαστικά NUMA Συστήματα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΠΟΣΤΟΛΟΠΟΥΛΟΣ ΒΑΣΙΛΕΙΟΣ - ΝΙΚΟΛΑΟΣ

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

Αθήνα, Μάρτιος 2021

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

Δυναμική Τοποθέτηση Εικονοποιημένων Δικτυακών Λειτουργιών Σε Πολυεπεξεργαστικά NUMA Συστήματα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΠΟΣΤΟΛΟΠΟΥΛΟΣ ΒΑΣΙΛΕΙΟΣ - ΝΙΚΟΛΑΟΣ

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

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11η Μαρτίου 2021.

.....	.....	.....
Γεώργιος Γκούμας Αναπληρωτής Καθηγητής Ε.Μ.Π.	Νεκτάριος Κοζύρης Καθηγητής Ε.Μ.Π.	Διονύσιος Πνευματικάτος Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2021

Αποστολόπουλος Βασίλειος - Νικόλαος

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

Περίληψη

Η ανάπτυξη προϊόντων στον κλάδο των τηλεπικοινωνιών παραδοσιακά ακολουθούσε αυστηρά πρότυπα σταθερότητας, συμβατότητας και ποιότητας γεγονός που οδηγούσε σε μεγάλους κύκλους ανάπτυξης, αργούς ρυθμούς προόδου και εξάρτηση από ιδιόκτητο ή εξειδικευμένο υλικό (4). Για την επιτάχυνση του ρυθμού ανάπτυξης νέων τηλεπικοινωνιακών υπηρεσιών οι πάροχοι στράφηκαν προς την Εικονικοποίηση Δικτυακών Λειτουργιών (6), μία αρχιτεκτονική δικτύου που εξομοιώνει τα απαραίτητα δομικά συστατικά του δικτύου βασιζόμενη σε παραδοσιακές τεχνικές εικονικοποίησης. Έτσι μπορεί να δημιουργηθεί μια τηλεπικοινωνιακή υπηρεσία η οποία αποτελείται από επιμέρους εικονοποιημένες δικτυακές λειτουργίες. Η συγκεκριμένη διπλωματική εργασία εξετάζει διάφορα σενάρια διασύνδεσης τέτοιων λειτουργιών στα οποία εμφανίζεται μείωση της απόδοσης, λόγω μη αποδοτικής τοποθέτησης τους στο υλικό. Αυτό συμβαίνει διότι στα σύγχρονα πολυεπεξεργαστικά συστήματα ο χρόνος που η κάθε επεξεργαστική μονάδα επικοινωνεί με τις υπόλοιπες διαφέρει. Οι εικονικές λειτουργίες συνδεδεμένες η μία με την άλλη σχηματίζουν αλυσίδες έτσι ώστε κάθε αλυσίδα να αποτελεί μια τηλεπικοινωνιακή υπηρεσία και σε έναν εξυπηρετητή ενός τηλεπικοινωνιακού παρόχου ζουν πολλές τέτοιες αλυσίδες ταυτόχρονα. Το πρόβλημα εμφανίζεται όταν η τοποθέτηση αυτών των αλυσίδων συμβαίνει μία φορά, κατά την εκκίνηση τους χωρίς να υπάρχει η δυνατότητα αυτοματοποιημένης επανατοποθέτησης τους στον υπόλοιπο κύκλο της ζωής τους. Μπορεί η αρχική τοποθέτηση την χρονική στιγμή που συμβαίνει να είναι η αποδοτικότερη, όσον αφορά την επικοινωνία των λειτουργιών μεταξύ τους, κάτι που μεταφράζεται στο ότι βρίσκονται σε ”γειτονικούς” επεξεργαστές. Όμως η κίνηση μεταξύ των λειτουργιών είναι δυναμική και χωρίς δυναμική τοποθέτηση εύκολα καταλήγουμε σε περιπτώσεις όπου αλυσίδες με έντονη δικτυακή κίνηση έχουν τις επιμέρους λειτουργίες τους τοποθετημένες σε ”μη γειτονικούς” επεξεργαστές, ενώ άλλες αλυσίδες με λιγότερη ή και καθόλου δικτυακή κίνηση βρίσκονται σε ευνοϊκότερη τοποθέτηση. Όπως αποδεικνύεται απο τη συγκεκριμένη εργασία ένα τέτοιο σενάριο μπορεί να έχει μεγάλη επίπτωση στην απόδοση μιας τηλεπικοινωνιακής υπηρεσίας, ειδικά αν μιλάμε για υπηρεσία που ειναι latency critical. Σκοπός της συγκεκριμένης εργασίας είναι η ανάπτυξη ενός αλγορίθμου ο οποίος τοποθετεί τις αλυσίδες που βρίσκονται μέσα σε ένα σύστημα, δυναμικά, ανάλογα με την δικτυακή κίνηση που έχει η καθεμία, σε όλη τη διάρκεια της ζωής τους. Τα αποτελέσματα της πειραματικής αξιόλογησης αυτού του αλγορίθμου αποδεικύονται ιδιαίτερα ενδιαφέροντα, αφού παρουσιάζουν σημαντική μείωση της καθυστέρησης στην επικοινωνία αλλά και των drop rates μεταξύ των εικονικών λειτουργιών στα σενάρια όπου εμφανιζόταν πρόβλημα.

Λέξεις κλειδιά

Αλγόριθμος Δυναμικής Τοποθέτησης, Εικονικοποίηση Δικτυακών Λειτουργιών, Εικονικές Λειτουργίες Δικτύου, Πλαίσια Διαχείρισης και Ενορχήστρωσης της Εικονικοποίησης Δικτυακών Λειτουργιών, Μη Ομοιόμορφη Πρόσβαση στη Μνήμη, VPP, DPDK, TRex

Abstract

Product development within the telecommunication industry has traditionally followed strict standards for stability, compatibility and quality, leading to long product cycles, a slow pace of development and reliance on proprietary or specific hardware. To accelerate the pace of development of new telecommunication services, providers have turned to Network Function Virtualization, a network architecture that simulates the essential network components based on traditional virtualization techniques. Thus, a telecommunication service can be created which consists of individual virtual network functions. This dissertation examines various scenarios of interconnection of such functions in which performance decreases, due to their inefficient hardware placement. This happens because in modern multiprocessing systems cpu to cpu communication latency varies. Virtual network functions connect to each other forming chains, many of which, hosted simultaneously at a provider's server. Issues arise when the placement of these chains occurs once, at startup, without the possibility of automated re-positioning for the rest of their life cycle. The initial placement at the time of occurrence may be the most efficient in terms of inner function communication, which translates to the fact that they are placed in "neighboring" processors. However, traffic between the functions is dynamic, and without dynamic placement we easily end up in cases where chains with intense network traffic have their individual functions placed in "non-adjacent" processors, while others with less or no network traffic at all are in a more favorable placement. As shown by this work, such a scenario can have major impact on the performance of a telecommunication service, especially on a latency critical one. The purpose of this thesis is to develop an algorithm that places virtual network function chains within a system, dynamically, depending on the network traffic each of them has, throughout their life cycle. The results of the experimental evaluation of this algorithm prove to be particularly interesting, as they show a significant reduction in terms of communication latency and drop rates between virtual network functions in scenarios where such problems occurred.

Key words

Dynamic Scheduling Algorithm, Flow Aware Scheduling NFV, Network Function Virtualization, NFV, Network Function Virtualization Orchestrator, NFVO, Virtual Network Function, VNF, Non-uniform memory access, NUMA, Vector Packet Processing, VPP, Data Plane Development Kit, DPDK, TRex

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Γεώργιο Γκούμα για την επίβλεψη αυτής της διπλωματικής εργασίας και για την δυνατότητα που μου έδωσε να την εκπονήσω στο εργαστήριο Υπολογιστικών Συστημάτων του Εθνικού Μετσόβιου Πολυτεχνείου. Επίσης θα ήθελα να ευχαριστήσω ιδιαίτερα τον Νικόλαο Αναστόπουλο, την ομάδα του NFV στην Intracom Telecom καθώς και τους Κωνσταντίνο Νίκα και Βασίλειο Καρακώστα για την διαρκή καθοδήγηση και βοήθεια που μου παρείχαν όποτε τη χρειάστηκα. Ακόμη ευχαριστώ ιδιαίτερα την οικογένεια μου για τη στήριξη, τα εφόδια και τις ευκαιρίες που μου παρείχε μεγαλώνοντας. Τέλος, όλους τους φίλους μου και ιδιαίτερα την Α.Δ. χωρίς τους οποίους δεν θα μπορούσα σε καμία περίπτωση να φτάσω στο σημείο που βρίσκομαι σήμερα.

Αποστολόπουλος Βασίλειος - Νικόλαος,

Αθήνα, 11η Μαρτίου 2021

Περιεχόμενα

Κατάλογος πινάκων

Κατάλογος σχημάτων

Κεφάλαιο 1

Εισαγωγή

Οι πάροχοι τηλεπικοινωνιακών υπηρεσίων και δικτύων έρχονται αντιμέτωποι με έναν μεγάλο αριθμό διαφορετικών ιδιόκτητων συσκευών υλικού, κάθε μια απ' τις οποίες είναι ειδικά σχεδιασμένη για μία ή και παραπάνω τηλεπικοινωνιακές υπηρεσίες. Η δημουργία μιας νέας υπηρεσίας δικτύου συχνά ισοδυναμεί με μια ακόμη δημιουργία διαφορετικών συνδυασμών ή και νέων τέτοιων συσκευών. Η έυρεση του χώρου και της ενέργειας για την υποδοχή τους, σε συνδυασμό με το αυξανόμενο κόστος της ενέργειας, τις επενδυτικές προκλήσεις και τη σπανιότητα των δεξιοτήτων που απαιτούνται για το σχεδιασμό, την ενσωμάτωση και τη λειτουργία τους γίνεται ολοένα και πιο δύσκολη. Επιπλέον, οι συσκευές αυτές φτάνουν γρήγορα στο τέλος της ζωής τους, απαιτώντας μεγάλο μέρος του κύκλου procure-design-integrate-deploy να επαναληφθεί με ελάχιστο ή και καθόλου όφελος εσόδων. Ακόμη χειρότερα, οι κύκλοι ζωής του υλικού γίνονται ολοένα και συντομότεροι με την ραγδαία ανάπτυξη της τεχνολογίας, κάτι που εμποδίζει την ανάπτυξη νέων υπηρεσιών δικτύου και περιορίζει την καινοτομία σε έναν κόσμο άρρηκτα συνδεδεμένο με τα δίκτυα.

Ο ανταγωνισμός στο χώρο των επικοινωνιών από ευέλικτους οργανισμούς που δραστηριοποιούνται σε μεγάλη κλίμακα στο διαδίκτυο (όπως το Google Talk, το Skype, το Netflix) ανάγκασε τους παρόχους υπηρεσιών να αναζητήσουν τρόπους να αλλάξουν αυτή την κατάσταση. Τον Οκτώβριο του 2012, μια ομάδα τηλεπικοινωνιακών φορέων δημοσίευσε μια λευκή βίβλο (7) σε ένα συνέδριο στο Ντάρμσταντ της Γερμανίας, σχετικά με τη δικτύωση που καθορίζεται από λογισμικό (Software Defined Network - SDN) και το πρωτόκολλο OpenFlow (14). Η πρόσκληση για δράση που κατέληγε η λευκή βίβλος οδήγησε στη δημιουργία της ομάδας Network Functions Virtualization (NFV) Industry Specification Group (ISG) (3) στο πλαίσιο του Ευρωπαϊκού Ινστιτούτου Προτύπων Τηλεπικοινωνιών (European Telecommunications Standards Institute - ETSI).

Η Εικονικοποίηση Δικτυακών Λειτουργιών (NFV) στοχεύει στην αντιμετώπιση των προβλημάτων που περιγράφηκαν αξιοποιώντας παραδοσιακές τεχνικές εικονικοποίησης για την εξομοίωση ολόκληρων δικτυακών κόμβων σε δομικά στοιχεία που δημιουργούν τηλεπικοινωνιακές υπηρεσίες. Μια Εικονική Λειτουργία Δικτύου (Virtual Network Function - VNF) μπορεί να αποτελείται από μία ή περισσότερες εικονικές μηχανές ή containers που εκτελούν διαφορετικό λογισμικό, πάνω από τυπικούς διακομιστές μεγάλου όγκου, μεταγωγείς και συσκευές αποθήκευσης ή ακόμη και στο υπολογιστικό νέφος (cloud), αντί να έχει εξειδικευμένες συσκευές υλικού για κάθε λειτουργία της.

1.1 Αντικείμενο της διπλωματικής

Η εργασία αυτή ασχολείται με την δυναμική τοποθέτηση Εικονικών Λειτουργιών Δικτύου στις κεντρικές επεξεργαστικές μονάδες ενός συστήματος, λαμβάνοντας υπ' όψιν την δικτυακή κίνηση που υπάρχει σε αυτές. Τα Πλαίσια Διαχείρισης και Ενορχήστρωσης της Εικονικοποίησης Δικτυακών Λειτουργιών (NFV Orchestrators - NFVO) αναλαμβάνουν τη διασφάλιση των διαθέσιμων πόρων για την παροχή μιας υπηρεσίας δικτύου. Στα πλαίσια αυτού πραγματοποιούν μια αρχική τοποθέτηση των Εικονικών Λειτουργιών Δικτύου σε κεντρικές μονάδες επεξεργασίας χωρίς να λαμβάνουν υπ'οψιν οτι η δικτυακή τους κίνηση μεταβάλλεται δυναμικά, τη συνδεσμολογία τους, καθώς και οτι οι χρόνοι επικοινωνίας μεταξύ των κεντρικών μονάδων επεξεργασίας ενός συστήματος μπορεί να διαφέρουν. Στην παρούσα διπλωματική εργασία επιβεβαιώνεται πειραματικά οτι η μεταβολή της κίνησης σε στατικά τοποθετημένες αλυσίδες εικονικών λειτουργιών επηρεάζει την αποδοσή τους και αναπτύσσεται ένας αλγόριθμος δυναμικής τοποθέτησης αυτών.

1.2 Οργάνωση του τόμου

Η παρούσα εργασία είναι οργανωμένη στα παρακάτω κεφάλαια.

- Στο Κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο που κρίνεται απαραίτητο για την κατανόηση της διπλωματικής εργασίας καθώς και οι τεχνολογίες που χρησιμοποιήθηκαν σε αυτήν.
- Στο Κεφάλαιο 3 περιγράφεται αναλυτικά και επιβεβαιώνεται πειραματικά το πρόβλημα το οποίο έρχεται να καλύψει η συγκεκριμένη εργασία.
- Στο κεφάλαιο 4 αναλύεται ο αλγόριθμος που αναπτύχθηκε, για τη δυναμική τοποθέτηση εικονικών λειτουργιών δικτύου σε κεντρικές επεξεργαστικές μονάδες πολυεπεξεργαστικών συστημάτων και παρουσιάζεται η συμπεριφορά του σε διάφορα σενάρια.
- Στο κεφάλαιο 5 γίνεται η πειραματική αξιολόγηση του αλγορίθμου σε συστήματα με πραγματική δικτυακή κίνηση.
- Τέλος, στο κεφάλαιο 6 συνοψίζεται η διπλωματική εργασία και εξετάζονται οι πιθανές μελλοντικές προεκτάσεις της.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο - Τεχνολογίες

Στο παρόν κεφάλαιο θα καλυφθεί το θεωρητικό υπόβαθρο που κρίνεται απαραίτητο για την κατανόηση της διπλωματικής εργασίας και θα παρουσιαστούν οι τεχνολογίες που χρησιμοποιηθήκαν σε αυτήν.

2.1 Εικονικοποίηση Δικτυακών Λειτουργιών

Στον κλάδο των τηλεπικοινωνιών η Εικονικοποίηση Δικτυακών Λειτουργιών (Network Function Virtualization, NFV) (16) είναι αρχιτεκτονική δικτύου που χρησιμοποιεί τις τεχνολογίες εικονικοποίησης για να εξομοιώσει λειτουργίες κόμβων δικτύων σε δομικά στοιχεία που μπορούν να συνδεθούν μαζί για να δημιουργήσουν υπηρεσίες τηλεπικοινωνιών.

Η εικονικοποίηση δικτυακών λειτουργιών βασίζεται σε παραδοσιακές τεχνικές εικονικοποίησης υπολογιστών, όπως αυτές χρησιμοποιούνται στην πληροφορική, αλλά εμφανίζει και διαφορές. Μια εικονική λειτουργία δικτύου (VNF) μπορεί να αποτελείται από μία ή περισσότερες εικονικές μηχανές που τρέχουν διαφορετικό λογισμικό και διεργασίες, πάνω σε κοινούς εξυπηρετητές, μεταγωγείς και συσκευές αποθήκευσης, ή ακόμα και σε υπολογιστικά νέφη, αντί να χρειάζεται εξειδικευμένες συσκευές για κάθε λειτουργία της.

Παραδείγματα δικτυακών λειτουργιών που εικονικοποιούνται είναι μεταγωγείς/δρομολογητές, εξισορροπητές φόρτου, τείχη προστασίας, συστήματα ανίχνευσης εισβολής και επιταχυντές διαδικτύου.

2.2 Τύποι δικτυακής κίνησης της NFV

Για λόγους ανάλυσης της επίδοσης, η δικτυακή κίνηση αναλύεται σε:

- **Data plane workloads**, ή κίνηση στο επίπεδο δεδομένων, η οποία περιλαμβάνει όλες τις εργασίες που αφορούν την επεξεργασία πακέτων απο end-to-end επικοινωνία μεταξύ απομακρυσμένων εφαρμογών. Οι συγκεκριμένες εργασίες πραγματοποιούν σημαντικά πολλά αιτήματα Εισόδου/Εξόδου καθώς και πολλές Αναγνώσεις/Εγγραφές από και πρός τη μνήμη.
  - Στην περίπτωση μιας edge λειτουργίας δικτύου όπως για παράδειγμα είναι ένα CDN cache node, η κίνηση αυτή περιλαμβάνει την εγκαθίδρυση καθώς και τον τερματισμό της συνεδρίας για τα επίπεδα L4-L7 καθώς και την εκπομπή και την λήψη των δεδομένων. Αυτό τυπικά σημαίνει μεγάλο αριθμό Αναγνώσεων/Εγγραφών από και πρός τη μνήμη καθώς και πολλά αιτήματα Εισόδου/Εξόδου που σχετίζονται με την εκπομπή και λήψη δεδομένων.

- Στην περίπτωση μιας ενδιάμεσης λειτουργίας δικτύου όπως για παράδειγμα είναι ένας δρομολογητής, οι εργασίες που πρέπει να γίνουν αφορούν στην αφαίρεση/πρόσθεση επικεφαλίδων, στην προώθηση πακέτων, στην τροποποίηση πεδίων στα πακέτα, στην καταγραφή της ροής/συνεδρίας κ.ο.κ. Αυτό σημαίνει μεγάλο αριθμό Αναγνώσεων/Εγγραφών από και πρός μνήμη καθώς και πολλά αιτήματα Εισόδου/Εξόδου που σχετίζονται με την προώθηση δεδομένων. Λόγω της σχετικής απλότητας των συγκεκριμένων εργασιών, σε κάποιες περιπτώσεις είναι εφικτό να παρακαμφθούν ορισμένες λειτουργίες του Λειτουργικού Συστήματος προκειμένου να αποφεφθούν ενδεχόμενα σημεία συμφόρησης.
- Στην περίπτωση μιας ενδιάμεσης λειτουργίας δικτύου με συναρτήσεις κρυπτογράφησης όπως για παράδειγμα ένας IPSec tunneller, πέραν της διαχείρησης των πακέτων που αναφέρεται παραπάνω, η κρυπτογράφηση ανά πακέτο γίνεται μία βασική λειτουργία, πράγμα το οποίο σημαίνει παραπάνω φόρτο εργασίας στους διαθέσιμους επεξεργαστικούς πόρους.

- **Control plane workloads**, ή κίνηση στο επίπεδο ελέγχου, η οποία περιλαμβάνει οποιαδήποτε άλλη επικοινωνία μεταξύ λειτουργιών δικτύου η οποία δεν σχετίζεται άμεσα με την end-to-end επικοινωνία δεδομένων μεταξύ απομακρυσμένων εφαρμογών. Αυτή η κατηγορία περιλαμβάνει την διαχείρηση της συνεδρίας, τη δρομολόγηση ή την αυθεντικοποίηση. Για παράδειγμα μία PPP (Point To Point Protocol) συνεδρία, μια δρομολόγηση BGP (Border Gateway Protocol) ή μία RADIUS (Remote Authentication Dial-In User Service) αυθεντικοποίηση σε μία BRAS (Broadband Remote Access Server) λειτουργία δικτύου είναι παραδείγματα κίνησης επιπέδου ελέγχου. Συγκρινόμενες με κινήσεις επιπέδου δεδομένων, οι κινήσεις επιπέδου ελέγχου είναι αισθητά λιγότερο έντονες σε όρους συναλλαγών ανά δευτερόλεπτο, ενώ η πολυπλοκότητα των συναλλαγών μπορεί να είναι μεγαλύτερη. Αυτό γενικά σημαίνει μικρότερο αριθμό Αναγνώσεων/Εγγραφών από και πρός τη μνήμη καθώς και λιγότερα αιτήματα Εισόδου/Εξόδου, και την ίδια στιγμή, πιθανόν μεγαλύτερο φόρτο εργασίας στους επεξεργαστικούς πόρους ανά πακέτο (παρόλο που ο ολικός φόρτος στις CPU αναμένεται να είναι μικρότερος καθώς και η αναλογία πακετών είναι μικρότερη).

2.3 NFV Ενορχήστρωση

Ένας πάροχος υπηρεσιών που χρησιμοποιεί την NFV αρχιτεκτονική μπορεί να προσομοιώσει με λογισμικό μία ή περισσότερες δικτυακές λειτουργίες. Μια εικονική δικτυακή λειτουργία από μόνη της δεν προσφέρει υποχρεωτικά κάποιο προϊόν ή υπηρεσία στους πελάτες του παρόχου. Για να δημιουργηθούν πιο περίπλοκες υπηρεσίες, χρησιμοποιείται η έννοια της αλυσίδας λειτουργιών, όπου πολλαπλές λειτουργίες χρησιμοποιούνται σε συνδυασμό για να δώσουν μια δικτυακή υπηρεσία.

Σε αυτό το σημείο εμφανίζεται μία σημαντική και αναγκαία πτυχή της NFV, που είναι η ενορχήστρωση (15). Για την κατασκευή αξιόπιστων και ολοκληρωμένων υπηρεσιών, η NFV απαιτεί ότι η υποδομή είναι σε θέση να ξεκινήσει εικονικές δικτυακές λειτουργίες, να τις παρακολουθεί, να τις επισκευάζει, και (το πιο σημαντικό για έναν πάροχο υπηρεσιών) να χρεώνει για τις υπηρεσίες που παρέχονται. Αυτά τα χαρακτηριστικά, που αναφέρονται ως ”επιπέδου-παρόχου” (carrier-grade)

(17) χαρακτηριστικά, ανατίθενται σε ένα επίπεδο ενορχήστρωσης, ώστε να παρέχει υψηλή διαθεσιμότητα και ασφάλεια, και χαμηλό κόστος λειτουργίας και συντήρησης.

Μερικά παραδείγματα ιδιαίτερα δημοφιλών ενορχηστροτών NFV είναι οι OSM (Open Source MANO (Management And Orchestration)) (9) και ONAP (Open Network Automation Platform) (8). Ο πρώτος πρόκειται για ένα έργο ανοιχτού κώδικα απο το Ευρωπαϊκο Ινστιτούτο Προτύπων Τηλεπικοινωνιών (European Telecommunications Standards Institute - ETSI) και ο δεύτερος έργο του Linux Foundation Networking (5). Και οι δύο είναι υπεύθυνοι για την επικοινωνία με την εικονική υποδομή και εξασφαλίζουν ότι οι αλυσίδες απο τις δικτυακές λειτουργίες είναι ικανές να ξεκινήσουν στο σύστημα, να επικοινωνούν επαρκώς, να ανακάμπτουν απο τυχών λάθη και άλλες λειτουργίες που ειναι απαραίτητες ώστε μια τηλεπικοινωνιακή υπηρεσία να λειτουργεί ομαλά. Όμως οι συγκεκριμένοι ενορχηστρωτές δεν λαμβάνουν υπ' όψιν τους χαρακτηριστικά που βρίσκονται πιο κοντά στο υλικό όπου φιλοξενεί την NFV όπως για παράδειγμα το cpu-to-cpu latency. Η συγκεκριμένη διπλωματική εργασία έρχεται να καλύψει αυτό ακριβώς το κενό λαμβάνοντας υπ' όψιν την καθυστέρηση μεταξύ των κεντρικών επεξεργαστικών μονάδων, την συνδεσμολογία μεταξύ των εικονικών λειτουργιών και την δικτυακή τους κίνηση.

2.4 Τεχνολογίες

Στην ενότητα αυτή αναφέρουμε τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της διπλωματικής εργασίας.

2.4.1 Data Plane Development Kit

Το Data Plane Development Kit (DPDK) (2) είναι ένα μεγάλο έργο ανοιχτού κώδικα που περιλαμβάνει βιβλιοθήκες και οδηγούς συσκευών για την υψηλή απόδοση σε λειτουργίες Εισόδου/Εξόδου και την ταχύτερη επεξεργασία πακέτων. Ξεκίνησε το 2010 απο την Intel και τον Απρίλιο του 2017 μεταφέρθηκε στο Linux Foundation. Αρχικά δημιουργήθηκε για τις ανάγκες τηλεπικοινωνιακών υποδομών, όμως πλέον χρησιμοποιείται παντού, στο cloud, σε data centers, σε συσκευές, σε containers και πολλά άλλα. Σήμερα θεωρείται ένα απο τα επικρατέστερα και σημαντικότερα έργα ανοιχτού κώδικα στο Linux.

Αδιαμφισβήτητα, στον δικτυακό κόσμο, η επίτευξη υψηλής ταχύτητας και επίδοσης είναι ένα απο τα σημαντικότερα χαρακτηριστικά που θέλουμε και προσπαθούμε να έχουμε. Ο κύριος τρόπος με τον οποίο το DPDK επιτυγχάνει αυτά τα δύο χαρακτηριστικά είναι μεσω των Poll Mode Drivers (PMD) (12). Η κύρια σχεδιαστική αρχή των PMD είναι η άμεση πρόσβαση στις ουρές λήψης (RX queue) και διάδοσης (TX queue) των καρτών δικτύου, χωρίς την εμπλοκή της στοίβας δικτύου του πυρήνα των Linux. Έτσι επιτυχγάνει την ταχύτατη λήψη, επεξεργασία και παράδοση των πακέτων στο χώρο χρήστη.

Ουσιαστικά, το DPDK εκτελεί έναν ατέρμονα βρόχο επεξεργασίας πακέτων πάνω σε αφιερωμένους επεξέργαστες ο οποίος περιλαμβάνει τα εξής βήματα:

- παραλαμβάνει μεσω των PMD τα πακέτα προς επεξεργασία απο την ουρά λήψης
- επεξεργάζεται ο ίδιος ή στέλνει μέσω ουρών λογισμικού τα πακέτα σε άλλους επεξεργαστές για την επεξεργασία

- στέλνει τα πακέτα προς αποστολή στην ουρά αποστολής μέσω των PMD.

Με αυτόν τον τρόπο οι PMD επιτυγχάνουν καλύτερη απόδοση απ' τους παραδοσιακούς οδηγούς συσκευών που λειτουργούν με διακοπές. Έχουν όμως το κόστος του οτι κρατούν συνεχώς ενεργό τον επεξεργαστή στην υψηλότερη δυνατή συχνότητα, καταναλώνοντας έτσι τη μέγιστη δυνατή ενέργεια, καθόλη την διάρκεια.

2.4.2 FD.io's Vector Packet Processing

Το VPP του FD.io (11) είναι μια επεκτάσιμη στοίβα δικτύου που προσφέρει υψηλής ποιότητας έτοιμες λειτουργίες μεταγωγέα/δρομολογητή. Πρόκειται για την έκδοση ανοιχτού κώδικα της τεχνολογίας επεξεργασίας πακέτων Vector Packet Processing της Cisco: μια στοίβα επεξεργασίας πακέτων υψηλής απόδοσης που μπορεί να τρέξει σε κοινούς επεξεργαστές του εμπορίου. Τρέχει στον χώρο χρήστη των Linux, σε διάφορες αρχιτεκτονικές και χρησιμοποιεί το Data Plane Development Kit (ενότητα 2.4.1). Ορισμένες κοινές χρήσεις των VPP είναι ως εικονικοί μεταγωγείς, εικονικοί δρομολογητές, πύλες, τείχη προστασίας, εξισορροπητές φόρτου κλπ. Τα πλεονεκτήματα του VPP είναι η υψηλή απόδοσή του, η αποδεδειγμένη τεχνολογία του, η αρθρωτή μορφή και ευελιξία του καθως και το πλούσιο σύνολο χαρακτηριστικών που προσφέρει.

Η πλατφόρμα χρησιμοποιεί ένα διάνυσμα (vector) πακέτων προς επεξεργασία, σε αντίθεση με την παραδοσιακή προσέγγιση, της επεξεργασίας δηλαδή ενός πακέτου τη φορα. Αυτού του είδους η τεχνική είναι συχνή σε εφαρμογές που αποσκοπούν στην υψηλή απόδοση όσον αφορά την επεξεργασία πακέτων. Μία στοίβα δικτύου που χρησιμοποιεί την παραδοσιακή προσέγγιση επεξεργάζεται ένα πακέτο τη φορά: μια συνάρτηση χειρισμού διακοπής παίρνει ένα πακέτο από την κάρτα δικτύου το οποίο στη συνέχεια επεξεργάζεται μέσα από ένα σύνολο συναρτήσεων: η συνάρτηση A καλεί την συνάρτηση B η οποία με την σειρά της καλεί την συνάρτηση C και ούτω καθεξής. Η παραπάνω διαδικασία είναι απλή αλλά εγείρει τα εξής προβλήματα:

- στην I-cache παρατηρείται φαινόμενο thrashing (1)
- κάθε πακέτο προκαλεί πανομοιότυπα misses στην I-cache
- η μόνη λύση στα παραπάνω είναι οι μεγαλύτερες κρυφές μνήμες

Εν αντιθέσει, η λογική του διανύσματος πακέτων επιτρέπει την επεξεργασία περισσότερων του ενός πακέτου κάθε φορά. Ένα απο τα πλεονεκτήματα της προσέγγισης αυτής είναι οτι διορθώνει το φαινόμενο thrashing στην I-cache.

Ακόμη μειώνει το χρόνο που χάνεται λόγω εξαρτημένων αναγνώσεων από τη μνήμη μέσω prefetching και διορθώνει ζητήματα που αφορούν το βάθος της στίβας, αστοχίες της D-cache δηλαδή σε διευθύνσεις στοίβας.

Τέλος μειώνει το συνολικό χρόνο της διαδικάσιας, το "μάζεμα" δηλαδή των διαθέσιμων πακέτων από την ουρά λήψης (RX queue) της κάρτας δικτύου, την τοποθέτηση τους σε διάνυσμα, την επεξεργασία του διανύσματος μέσω του κατευθυνόμενου γράφου και την επιστροφή στην κάρτα δικτύου. Καθώς η επεξεργασία των πακέτων προχωράει ο συνολικός χρόνος φτάνει ένα σταθερό ισοζύγιο που βασίζεται στο μέγεθος του φόρτου. Καθώς το μέγεθος του διανύσματος αυξάνει, η επεξεργασία ανά πακέτο μειώνεται, επειδή οι αστοχίες στην I-cache αποσβένονται λόγω μεγάλου αριθμού N.

Η επεξεργασία των πακέτων γίνεται μέσω ενός κατευθυνόμενου γράφου. Αυτή η προσέγγιση σημαίνει ότι ο καθένας μπορεί πολύ εύκολα να προσθέσει νέους κόμβους στο γράφο,ανάλογα με τις λειτουργίες που θέλει να εκτελέσει στα πακέτα. Μάλιστα κάτι τέτοιο γίνεται χωρίς να χρειάζεται να αλλάξει ή προσθέσει κάτι στον κώδικα πυρήνα, μιας και η πλατφόρμα τρέχει σε χώρο χρήστη, πράγμα που καθιστά την επεκτασιμότητα της εφαρμογής εξαιρετικά απλή.

2.4.3 TRex: Realistic Traffic Generator

Ο TRex (10) είναι ένα έργο ανοικτού κώδικα που δημιουργήθηκε απο τη Cisco και χρησιμοποιείται για την παραγωγή δικτυακής κίνησης στρώματος 3 και πάνω. Χρησιμοποιεί το DPDK (ενότητα 2.4.1), υποστηρίζει stateless και stateful λειτουργίες και μπορεί να παράξει κίνηση εώς και 200Gb/sec σε έναν μόνο σέρβερ.

Κεφάλαιο 3

Κίνητρο - Περιγραφή Προβλήματος

Ένας σημαντικός τομέας που παραβλέπεται κατά την χρονοδρομολόγηση εικονικών λειτουργιών δικτύου σε πολυεπεξεργαστικά συστήματα είναι η κίνηση που υπάρχει μεταξύ τους. Η πραγματική κίνηση των πακέτων, ένας παράγοντας κρίσιμος για την αποδοτική τοποθέτηση των εικονικών λειτουργιών δικτύου στις κεντρικές μονάδες επεξεργασίας ώστε να ελαχιστοποιηθούν οι καθυστερήσεις, γίνεται γνωστή στο runtime. Ωστόσο η τοποθέτηση των εικονικών λειτουργιών δικτύου στις κεντρικές μονάδες επεξεργασίας γίνεται πολύ πριν την έναρξη της δικτυακής κίνησης, στο στάδιο του deployment. Τέλος, η κίνηση αυτή μεταβάλλεται δυναμικά κατά τον κύκλο ζωής μιάς εικονικής λειτουργίας δικτύου, λόγω της κίνησης του δικτύου γενικότερα, των διαφορετικών απαιτήσεων που έχουν υπηρεσίες on demand κ.λ.π.

Οι εικονικές λειτουργίες συνδεδεμένες η μία με την άλλη σχηματίζουν αλυσίδες έτσι ώστε κάθε αλυσίδα να αποτελεί μια τηλεπικοινωνιακή υπηρεσία και σε έναν εξυπηρετητή ενός τηλεπικοινωνιακού παρόχου ζουν πολλές τέτοιες αλυσίδες ταυτόχρονα. Το πρόβλημα εμφανίζεται όταν η τοποθέτηση αυτών των αλυσίδων συμβαίνει μία φορά, κατά την εκκίνηση τους χωρίς να υπάρχει η δυνατότητα αυτοματοποιημένης επανατοποθέτησης τους στον υπόλοιπο κύκλο της ζωής τους. Μπορεί η αρχική τοποθέτηση την χρονική στιγμή που συμβαίνει να είναι η αποδοτικότερη,όσον αφορά την επικοινωνία των λειτουργιών μεταξύ τους, κάτι που μεταφράζεται στο ότι βρίσκονται σε ”γειτονικούς” επεξεργαστές. Όμως η κίνηση μεταξύ των λειτουργιών είναι δυναμική και χωρίς δυναμική τοποθέτηση εύκολα καταλήγουμε σε περιπτώσεις όπου αλυσίδες με έντονη δικτυακή κίνηση έχουν τις επιμέρους λειτουργίες τους τοποθετημένες σε ”μη γειτονικούς” επεξεργαστές, ενώ άλλες αλυσίδες με λιγότερη ή και καθόλου δικτυακή κίνηση βρίσκονται σε ευνοϊκότερη τοποθέτηση. Όπως αποδεικνύεται απο τη συγκεκριμένη εργασία ένα τέτοιο σενάριο μπορεί να έχει μεγάλη επίπτωση στην απόδοση μιας τηλεπικοινωνιακής υπηρεσίας, ειδικά αν μιλάμε για υπηρεσία που ειναι latency critical.

3.1 Καθυστερήσεις μεταξύ κόμβων NUMA

Όπως γνωρίζουμε σε ένα σύστημα με μη ομοιόμορφη πρόσβαση στη μνήμη (13) κάθε κεντρική επεξεργαστική μονάδα έχει διαφορετικό χρόνο προσπέλασης της μνήμης ανάλογα με την θέση στην οποία βρίσκεται.

3.1.1 Μετρήσεις

Στο σημείο αυτό παρουσιάζονται μερικές πειραματικές μετρήσεις που έγιναν σε συστήματα με αρχιτεκτονική μνήμης NUMA.

Σύστημα **Broady2**

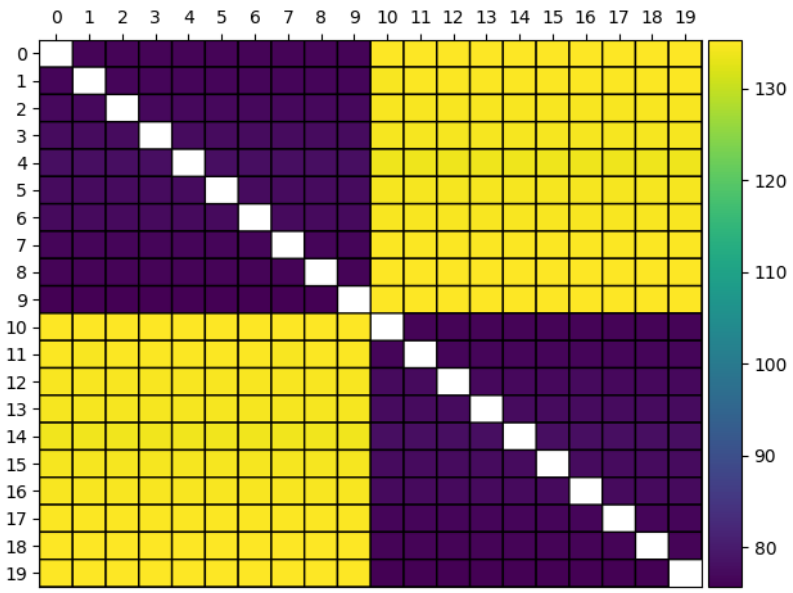
- Επεξεργαστής Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz
- 10 πυρήνες σε κάθε socket, 2 socket, 20 ΚΜΕ συνολικά στο σύστημα
- 251 GB RAM
- Κάρτα δικτύου Intel Corporation Ethernet Controller 10-Gigabit X540-AT2
- 2 κόμβοι NUMA
- ΚΜΕ που βρίσκονται στον κόμβο NUMA 0: 0 - 9
- ΚΜΕ που βρίσκονται στον κόμβο NUMA 1: 10 - 19
- Η κάρτα δικτύου βρίσκεται στον κομβο NUMA 0.

node	0	1
0	10	21
1	21	10

Πίνακας 3.1: Σχετικές αποστάσεις Broady2

node	0	1
0	76.6 ns	134.8 ns
1	135 ns	76.5 ns

Πίνακας 3.2: Μέση καθυστέρηση Broady2



Σχήμα 3.1: Μέση καθυστέρηση (ns) μεταξύ ΚΜΕ Broady2

Σύστημα **Gold2**

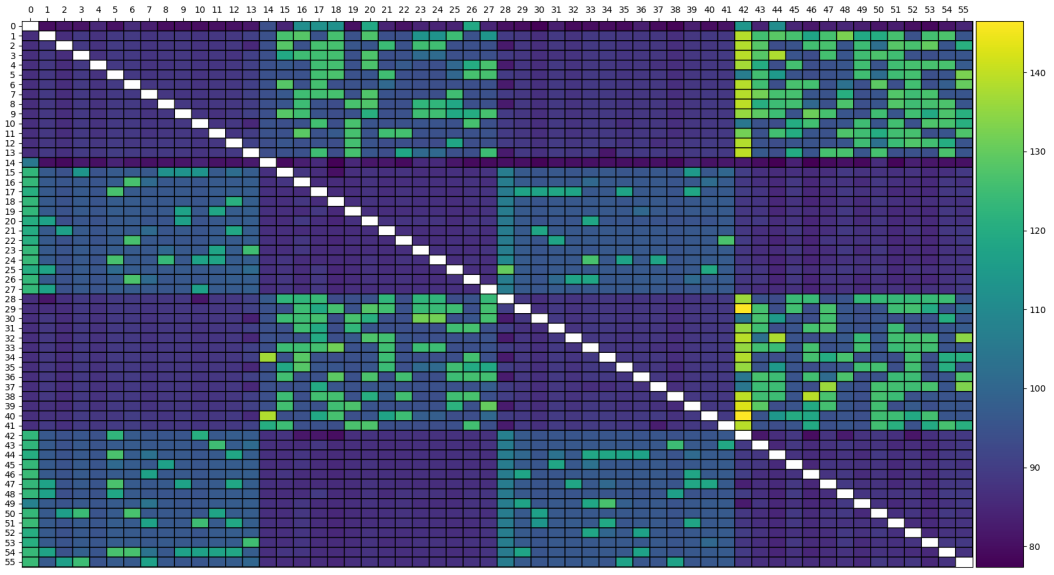
- Επεξεργαστής Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz
- 14 πυρήνες σε κάθε socket, hyperthreaded, 2 socket, 56 ΚΜΕ συνολικά στο σύστημα
- 251 GB RAM
- Κάρτα δικτύου Intel Corporation Ethernet Controller 10-Gigabit X540-AT2
- 2 κόμβοι NUMA
- ΚΜΕ που βρίσκονται στον κόμβο NUMA 0: 0 - 13, 28 - 41
- ΚΜΕ που βρίσκονται στον κόμβο NUMA 1: 14 - 27, 42 - 55
- Η κάρτα δικτύου βρίσκεται στον κομβο NUMA 0.

node	0	1
0	10	21
1	21	10

Πίνακας 3.3: Σχετικές αποστάσεις Gold2

node	0	1
0	81.3 ns	128.4 ns
1	128.6 ns	81.1 ns

Πίνακας 3.4: Μέση καθυστέρηση Gold2



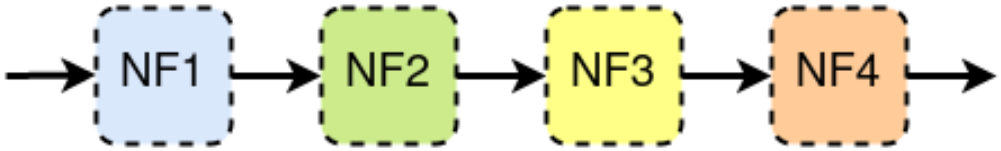
Σχήμα 3.2: Μέση καθυστέρηση (ns) μεταξύ ΚΜΕ Gold2

Στα παραπάνω σχήματα φαίνεται η αλλαγή στην καθυστέρηση επικοινωνίας μεταξύ των ΚΜΕ του εκάστοτε συστήματος. Παρατηρούμε οτι ΚΜΕ που βρίσκονται στον ίδιο κόμβο NUMA επικοινωνούν ταχύτερα μεταξύ τους απ' οτι με ΚΜΕ που βρίσκονται σε άλλον κόμβο NUMA. Στην περίπτωση του Broady2 η διαφορά αυτή είναι ιδιαίτερα διακριτή εφόσον πρόκειται για ένα σύστημα

τεχνολογίας Skylake. Στην περίπτωση του Gold2, μιλάμε για νεότερη τεχνολογία, Cascadelake, και παρατηρείται διακύμανση μεταξύ της καθυστέρησης απο επεξεργαστή σε επεξεργαστή.

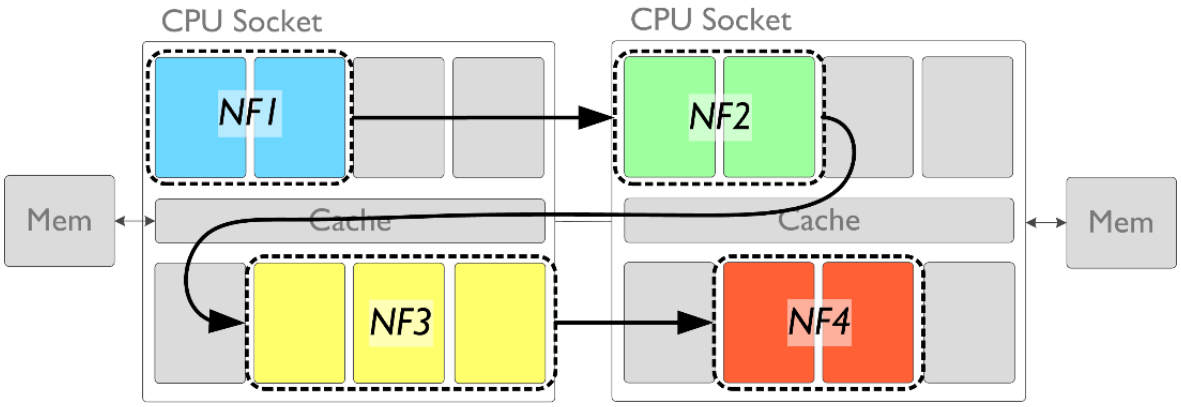
3.1.2 Σενάριο μη αποδοτικής τοποθέτησης Εικονικών Λειτουργιών Δικτύου

Στο Σχήμα 3.3 φαίνεται μια τηλεπικοινωνιακή υπηρεσία η οποία αποτελείται απο 4 εικονικές λειτουργίες δικτύου συνδεδεμένες σε αλυσίδα.



Σχήμα 3.3: Αλυσίδα εικονικών λειτουργιών δικτύου

Με βάση τα όσα προκύπτουν από την ενότητα 3.1.1 στο Σχήμα 3.4 βλέπουμε ένα μη αποδοτικό σενάριο τοποθέτησης των εικονικών λειτουργιών δικτύου στις κεντρικές μονάδες επεξεργασίας του συστήματος. Παρατηρείται μια συνεχής "μεταπήδηση" μεταξύ ΚΜΕ οι οποίες ανήκουν σε διαφορετικούς κόμβους NUMA. Το σενάριο αυτό έχει **σοβαρές επιπτώσεις στην απόδοση** της τηλεπικοινωνιακής υπηρεσίας, ιδιαίτερα αν πρόκειται για υπηρεσία η οποία είναι latency-critical.



Σχήμα 3.4: Μη αποδοτική τοποθέτηση εικονικών λειτουργιών δικτύου σε ΚΜΕ

3.2 Στατικές πολιτικές τοποθέτησης Εικονικών Λειτουργιών Δικτύου

Στην πράξη τα Πλαίσια Διαχείρισης και Ενορχήστρωσης της Εικονικοποίησης Δικτυακών Λειτουργιών (NFV Orchestrators - NFVO) λειτουργούν με τον εξής τρόπο. Κάθε φορά που μία κανούρια εικονική λειτουργία δικτύου δημιουργείται στο σύστημα, ο ενορχηστρωτής αναλαμβάνει να την τοποθετήσει σε μια κεντρική μονάδα επεξεργασίας, συνήθως στην πρώτη ελεύθερη που θα βρεί. Ας δούμε ένα παράδειγμα οπού αυτο αποτελεί πρόβλημα όσον αφορά τις καθυστερήσεις μεταξύ εικονικών λειτουργιών δικτύου.

Έστω οτι έχουμε 2 εικονικές λειτουργίες δικτύου, την VNF1 και την VNF2, οι οποίες συνδεόμενες αποτελούν μια τηλεπικοινωνιακή υπηρεσία. Έστω επίσης οτι βρισκόμαστε σε σύστημα όπου έχει 4 ΚΜΕ στον κόμβο NUMA 0 και άλλες 4 ΚΜΕ στον κόμβο NUMA 1. Ο ενορχηστρωτής αποφασίζει να βάλει την VNF1 και την VNF2 σε ΚΜΕ του κόμβου 0, αφήνοντας πλέον 2 ελεύθερες ΚΜΕ στον κόμβο 0 ενώ ο κόμβος 1 έχει 4 ελεύθερες ΚΜΕ. Έστω τώρα οτι στο σύστημα δημιουργείται μια δεύτερη τηλεπικοινωνιακή υπηρεσία αποτελούμενη από τις συνδεδεμένες μεταξύ τους εικονικές λειτουργίες δικτύου VNF3 και VNF4. Παραδοσιακά ο ενορχηστρωτής θα τοποθετήσει της νέες εικονικές λειτουργίες δικτύου στο κόμβο 0 αφήνοντας ελεύθερες 0 ΚΜΕ στον συγκεκριμένο κόμβο πλέον. Το πρόβλημα δημιουργείται στην περίπτωση που η πρώτη τηλεπικοινωνιακή υπηρεσία αποφασίσει να επεκταθεί (scale out) και να δημιουργήσει για παράδειγμα άλλες 2 εικονικές λειτουργίες δικτύου, την VNF5 και την VNF6, οι οποίες συνδέονται σε σειρά με τις VNF1 και VNF2. Λόγω της πολιτικής που χρησιμοποιεί ο ενορχηστρωτής θα τις τοποθετήσει στον κόμβο 1, εφόσον ο κόμβος 0 δεν έχει διαθέσιμες ελεύθερες ΚΜΕ. Πλέον στην πρώτη τηλεπικοινωνιακή υπηρεσία εισάγεται μια καθυστέρηση η οποία έγκειται στην επικοινωνία των ΚΜΕ του κόμβου 0 με αυτές του κόμβου 1, η οποία θα μπορούσε να αποφευχθεί εφόσον και οι 4 εικονικές λειτουργίες δικτύου της θα μπορούσαν να βρίσκονται στον κόμβο 0.

3.3 Η ανάγκη για δυναμικές πολιτικές τοποθέτησης

Στην παρούσα ενότητα αναλύεται μέσω πειραματικών μετρήσεων ο λόγος που υπάρχει η ανάγκη για δυναμικές πολιτικές τοποθέτησης εικονικών λειτουργιών δικτύου.

3.3.1 Περιγραφή συστημάτων

Με την χρήση των τεχνολογιών που αναλύονται στις ενότητες 2.4.1, 2.4.2 και 2.4.3 διεξήχθησαν πειράματα σε σύστημα ”Cascadelake” της Intel με 2 κόμβους NUMA, μετρώντας από σύστημα ”Skylake” της Intel.

Χαρακτηριστικά συστήματος **Cascadelake**:

- 2 επεξεργαστές Intel(R) Xeon(R) Gold 6252 CPU @ 2.10GHz
- 24 πυρήνες ο καθένας, hyperthreaded, 96 ΚΜΕ συνολικά στο σύστημα
- 187 GB RAM
- Καρτα δικτύου Intel XXV710-DA2 10/25GbE SFP28 2-Port PCIe Ethernet Adapter
- 2 κόμβοι NUMA
- ΚΜΕ που βρίσκονται στον κόμβο NUMA 0: 0 - 23, 48 - 71
- ΚΜΕ που βρίσκονται στον κόμβο NUMA 1: 24 - 47, 72 - 95
- Η κάρτα δικτύου βρίσκεται στον κομβο NUMA 0.

node	0	1
0	10	21
1	21	10

Πίνακας 3.5: Σχετικές αποστάσεις Cascadelake

Πιο συγκεκριμένα δημιουργήθηκαν εικονικές λειτουργίες δικτύου στο Cascadelake σύστημα χρησιμοποιώντας την τεχνολογία VPP του FD.io για την προσομοίωση κάθε εικονικής λειτουργίας δικτύου. Στο σύστημα Skylake μέσω του TRex της Cisco παράχθηκε δικτυακή κίνηση **1000 Mbps** η οποία εστάλη στο Cascadelake. Μετρήθηκε η μέση καθυστέρηση για την επεξεργασία και αποστολή των πακέτων απο τις εικονικές λειτουργίες δικτύου που βρίσκονταν στο Cascadelake.

3.3.2 Δεδομένα

Τα πειράματα που διεξήχθησαν έγιναν με τα εξής δεδομένα:

- Η κάρτα δικτύου που δέχεται/στέλνει δικτυακή κίνηση είναι τοπική σε συγκεκριμένο κόμβο NUMA, στο συγκεκριμένο σύστημα στον κόμβο 0.
- Κάθε εικονική λειτουργία δικτύου είναι τοποθετημένη σε διαφορετική ΚΜΕ.
- Δεν χρησιμοποιήθηκε υπερνηματική επεξεργασία (hyperthreading).

3.3.3 Σενάρια τοποθέτησης

Χρησιμοποιήθηκαν τα εξής σενάρια τοποθέτησης εικονικών λειτουργιών δικτύου:

1. Όλες οι εικονικές λειτουργίες δικτύου βρίσκονται σε πυρήνες του κόμβου NUMA 0.

   Πρόκειται για το ιδανικότερο σενάριο, με την μικρότερη καθυστέρηση, τόσο μεταξύ των εικονικών λειτουργιών δικτύου όσο και μεταξύ της αλυσίδας των εικονικών λειτουργιών δικτύου και της κάρτας δικτύου.

2. Όλες οι εικονικές λειτουργίες δικτύου βρίσκονται σε πυρήνες του κόμβου NUMA 1.

   Εδώ έχουμε ιδανικό σενάριο καθυστέρησης μεταξύ των εικονικών λειτουργιών δικτύου αλλά όχι μεταξύ της αλυσίδας των εικονικών λειτουργιών δικτύου και της κάρτας δικτύου.

3. Οι πρώτες μισές εικονικές λειτουργίες δικτύου της αλυσίδας βρίσκονται σε πυρήνες του κόμβου NUMA 0 και οι υπόλοιπες σε πυρήνες του κόμβου NUMA 1.

   Πρόκειται για ένα ενδιάμεσο σενάριο.

4. Οι πρώτες μισές εικονικές λειτουργίες δικτύου της αλυσίδας βρίσκονται σε πυρήνες του κόμβου NUMA 1 και οι υπόλοιπες σε πυρήνες του κόμβου NUMA 0.

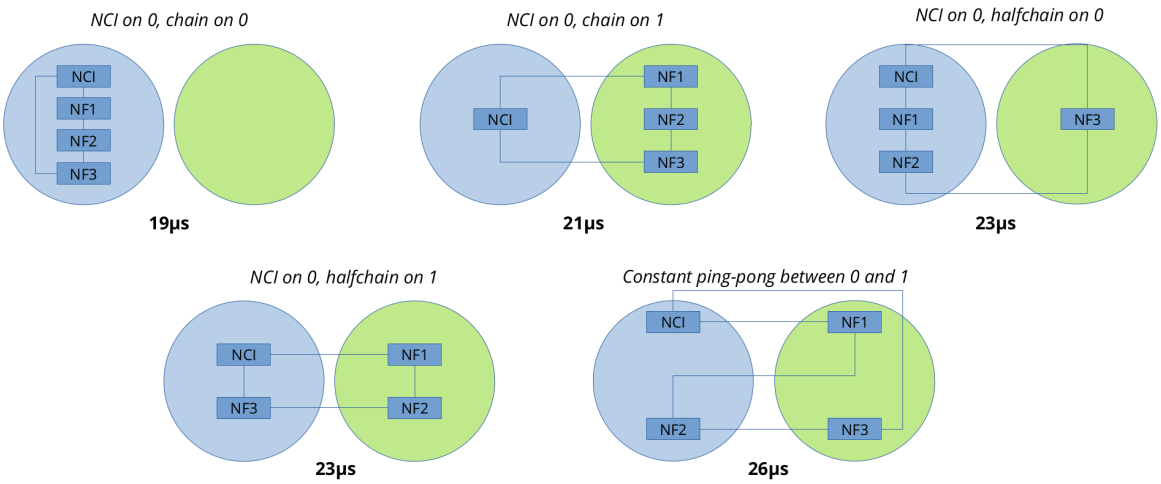
   Πρόκειται για ένα ενδιάμεσο σενάριο.

5. Οι εικονικές λειτουργίες δικτύου είναι τοποθετημένες έτσι ώστε τα πακέτα να "μεταπηδούν" συνεχώς μεταξύ των δύο κόμβων NUMA, με την πρώτη εικονική λειτουργία δικτύου της αλυσίδας να βρίσκεται στο κόμβο NUMA 1.

Πρόκειται για το χειρότερο σενάριο, με την μεγαλύτερη καθυστέρηση, τόσο μεταξύ των εικονικών λειτουργιών δικτύου όσο και μεταξύ της αλυσίδας των εικονικών λειτουργιών δικτύου και της κάρτας δικτύου.

Τρείς εικονικές λειτουργίες δικτύου.

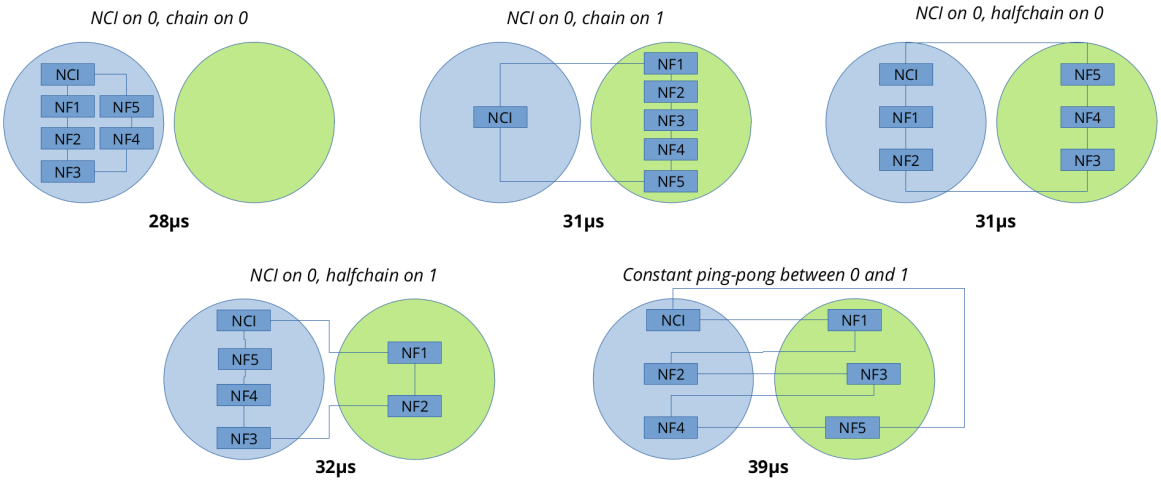
Αποτελέσματα πειραμάτων για 3 εικονικές λειτουργίες δικτύου συνδεδεμένες στη σειρά :



Σχήμα 3.5: Πειραματικά αποτελέσματα για 3 εικονικές λειτουργίες δικτύου

Πέντε εικονικές λειτουργίες δικτύου.

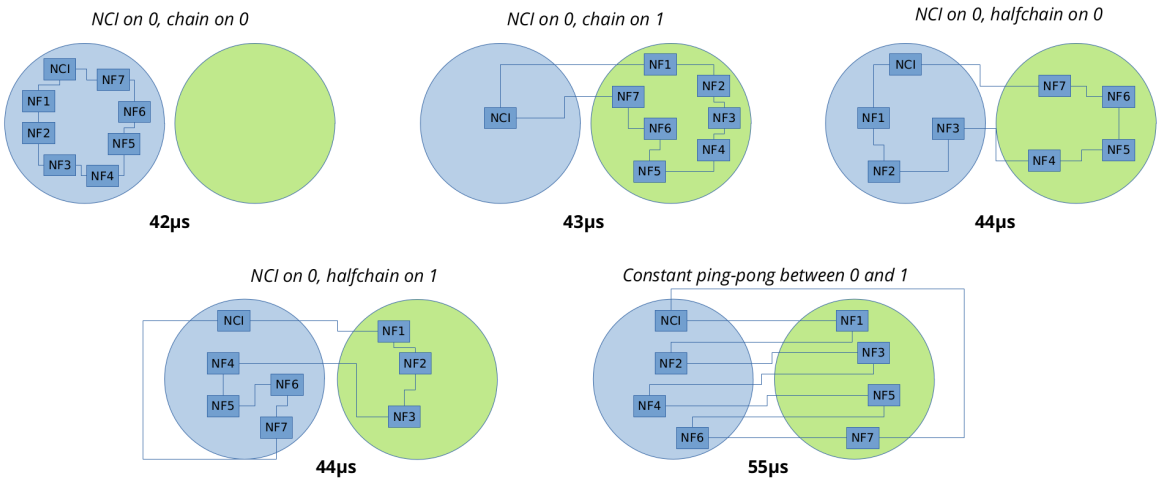
Αποτελέσματα πειραμάτων για 5 εικονικές λειτουργίες δικτύου συνδεδεμένες στη σειρά :



Σχήμα 3.6: Πειραματικά αποτελέσματα για 5 εικονικές λειτουργίες δικτύου

Επτά εικονικές λειτουργίες δικτύου.

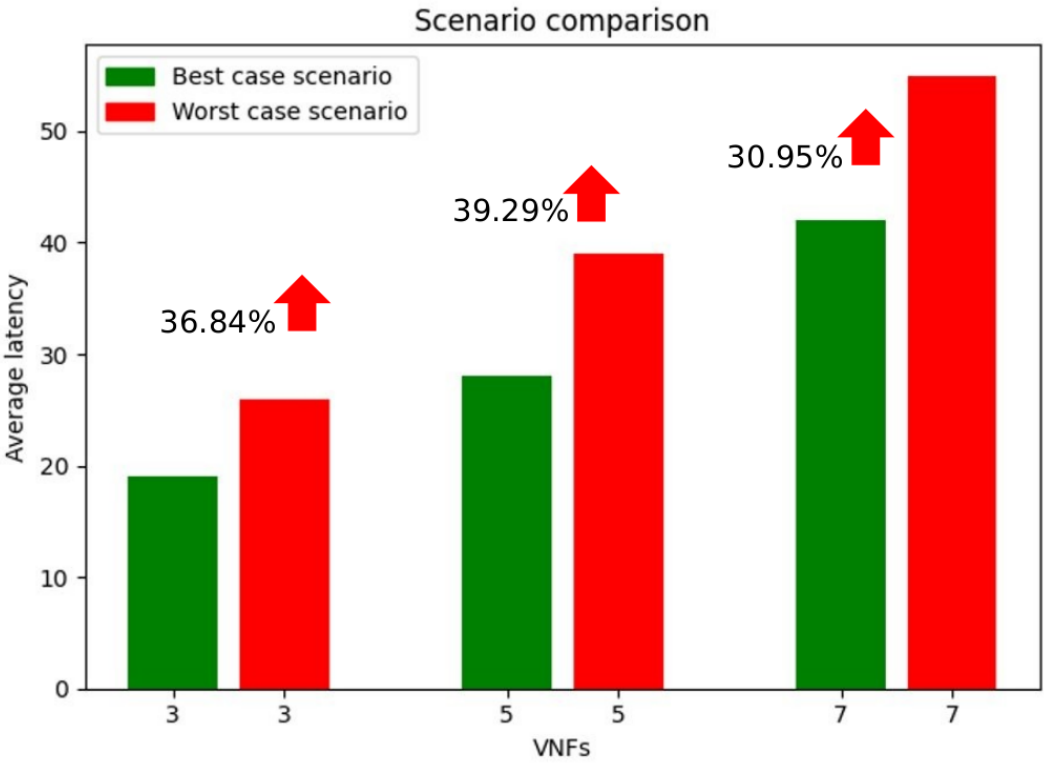
Αποτελέσματα πειραμάτων για 7 εικονικές λειτουργίες δικτύου συνδεδεμένες στη σειρά :



Σχήμα 3.7: Πειραματικά αποτελέσματα για 7 εικονικές λειτουργίες δικτύου

Τα αποτελέσματα των πειραμάτων αποδεικνύουν μεγάλη διαφοροποίηση της επίδοσης τηλεπικοινωνιακών υπηρεσιών ανάλογα με την τοποθέτηση των εικονικών λειτουργιών δικτύου απο τις οποίες αποτελούνται σε διάφορα σενάρια.

Αναλυτικότερα όπως φαίνεται και στο Σχήμα 3.8 παρατηρείται αύξηση στις καθυστερήσεις, και κατα συνέπεια **μείωση της απόδοσης** της υπηρεσίας, της τάξεως του **30 με 40 τοις εκατό** απο το καλύτερο σενάριο τοποθέτησης στο χειρότερο. Καλύτερο σενάριο θεωρείται το σενάριο 1 και χειρότερο το σενάριο 5.



Σχήμα 3.8: Σύγκριση καλύτερου/χειρότερου σεναρίου τοποθέτησης VNFs

Τα αποτελέσματα αυτά κάνουν σαφή την ανάγκη ανάπτυξης δυναμικών πολιτικών τοποθέτησης εικονικών λειτουργιών δικτύων στις κεντρικές επεξεργαστικές μονάδες πολυεπεξεργαστικών συστημάτων.

Ειδικότερα, αν αναλογιστεί κανείς το μικρό μέγεθος της κίνησης, το γεγονός οτι οι εικονικές λειτουργίες που προσομοιώθηκαν ήταν εικονικοί δρομολογητές -δεν λάμβανε χώρα κάποια χρονοβόρα επεξεργασία- καθώς και το μικρό πλήθος των εικονικών λειτουργιών, αναμένει οτι σε περιβάλλον πραγματικών τηλεπικοινωνιακών υπηρεσιών και δικτυακής κίνησης τα νούμερα αυτά είναι αρκετά αισιόδοξα.

Κεφάλαιο 4

Μηχανισμός Δυναμικής Τοποθέτησης Εικονικών Λειτουργιών Δικτύου

Στο παρόν κεφάλαιο θα αναλυθεί ο μηχανισμός που αναπτύχθηκε, στα πλαίσια της διπλωματικής εργασίας, για τη δυναμική τοποθέτηση εικονικών λειτουργιών δικτύου σε κεντρικές επεξεργαστικές μονάδες πολυεπεξεργαστικών συστημάτων. Το κεφάλαιο αναλύεται σε 2 ενότητες.

Η πρώτη ενότητα αποτελείται απο τον αλγόριθμο που αναπτύχθηκε για δυναμικό online χρονοπρογραμματισμό και τοποθέτηση των εικονικών λειτουργιών, στις διαθέσιμες κεντρικές επεξεργαστικές μονάδες του συστήματος.

Στη δέυτερη ενότητα παρουσιάζεται η συμπεριφορά του αλγορίθμου σε διάφορα σενάρια.

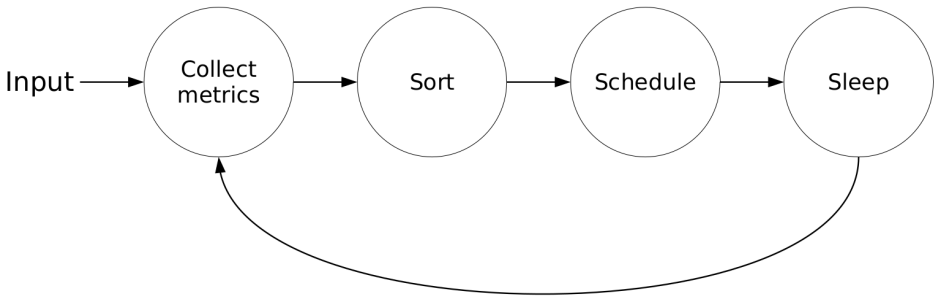
4.1 Αλγόριθμος δυναμικής τοποθέτησης

Πριν εκτελέσουμε τον αλγόριθμο θα πρέπει να ενημερώσουμε σωστά 3 πεδία στον κώδικα του:

1. Τις κεντρικές επεξεργαστικές μονάδες τις οποίες θέλουμε να βλέπει ο αλγόριθμος ως διαθέσιμες.

2. 'Εναν πίνακα σαν τους πίνακες 3.1 και 3.3 για τις σχετικές αποστάσεις των κόμβων μη ομοιόμορφης πρόσβασης στη μνήμη.

3. Τον κόμβο μη ομοιόμορφης πρόσβασης στη μνήμη στον οποίο βρίσκεται η κάρτα δικτύου.



Σχήμα 4.1: Finite State Machine του αλγορίθμου

Ο αλγόριθμος δέχεται σαν είσοδο μια λίστα απο λίστες, με τις εσωτερικές λίστες να αναπαριστούν τις αλυσίδες των εικονικών λειτουργιών πάνω στις οποίες θα εφαρμόσουμε τον αλγόριθμο. Στην συνέχεια συγκεντρώνει τις μετρικές της κίνησης για κάθε αλυσίδα και ταξιμομεί την εξωτερική λίστα σε φθίνουσα σειρά. Τοποθετεί τις αλυσίδες βάση της διαδικασίας που αναλύεται παρακάτω, κοιμάται για κάποιο διάστημα και στη συνέχεια επαναλαμβάνει την διαδικασία απο το στάδιο της συλλογής των μετρικών.

Τα **βήματα** του αλγορίθμου για την τοποθέτηση των αλυσίδων **σε ψευδογλώσσα**:

```
for every chain in sorted list:
    if part of the chain is on specific numa:
        if the rest of the chain fits there:
            place the rest of the chain in that numa
        else:
            start filling numas from the numa_distance_table until chain is fully scheduled
    else:
        for every numa node in numa_distance_table:
            if whole chain fits in numa:
                place chain in numa
                break
        if chain is not placed:
            start filling numas from the numa_distance_table until chain is fully scheduled
```

Σχήμα 4.2: Στάδιο τοποθέτησης αλυσίδων σε ψευδογλώσσα

Σε αυτό το σημείο αξίζει να σημειωθεί πως η επικοινωνία μετα μεταξύ των διεργασιών που προσομοιώνουν τις εικονικές δικτυακές λειτουργίες, καθώς και η συλλογή των μετρικών τους έγινε μεσω **socket files** του Linux.

4.2 Συμπεριφορά του αλγορίθμου

Στην ενότητα αυτήν παρουσιάζουμε τη συμπεριφορά του αλγορίθμου, σε διάφορα σενάρια, με διαφορετικές αλυσίδες εικονικών λειτουργιών δικτύου και δικτυακές κινήσεις μεταξύ τους.

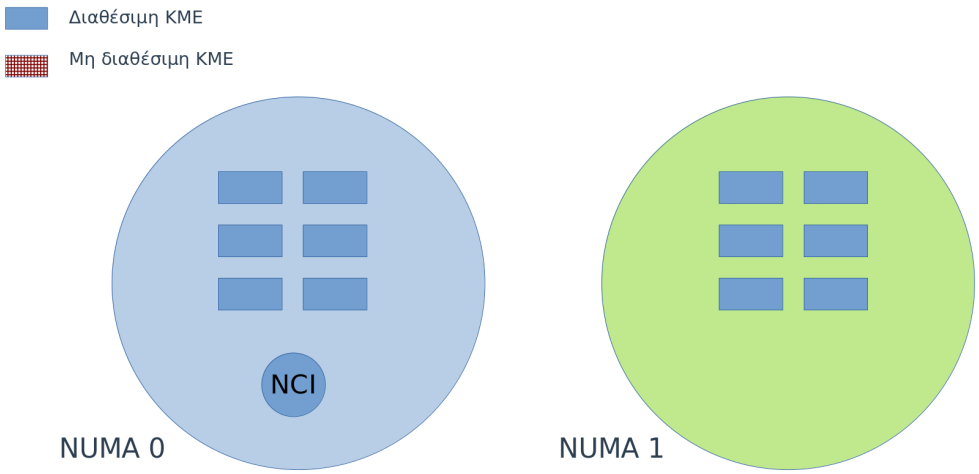
4.2.1 Χωρίς αλλαγή κίνησης στις αλυσίδες

Εδώ θα δούμε τη συμπεριφορά του αλγορίθμου σε διάφορα στατικά σενάρια όπου η κίνηση των αλυσίδων δεν μεταβάλλεται με το χρόνο.

Σενάριο 1

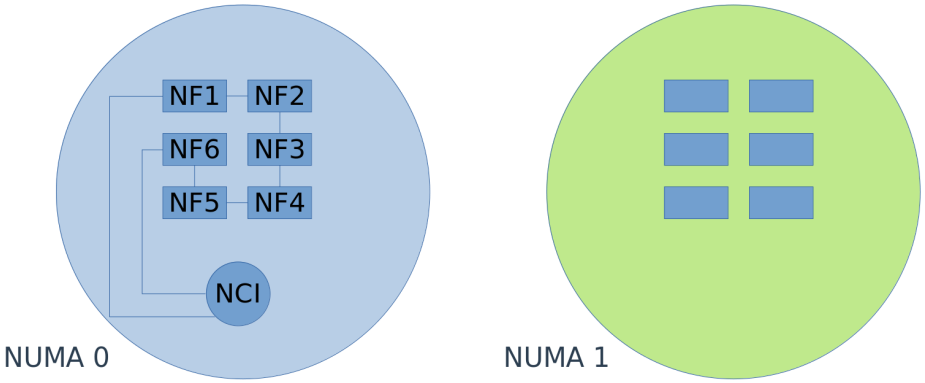
Στο παρακάτω σενάριο έχουμε σύστημα με 2 κόμβους NUMA απο 6 ΚΜΕ επεξεργασίας ο καθένας και την κάρτα δικτύου να βρίσκεται στον κόμβο NUMA 0. Ως λίστα εισόδου έχουμε 1 αλυσίδα αποτελούμενη απο 7 εικονικές λειτουργίες δικτύου.

Λίστα εισόδου: [[vpp1, vpp2, vpp3, vpp4, vpp5, vpp6]].



Σχήμα 4.3: Εικόνα ΚΜΕ σεναρίου 1 πριν την τοποθέτηση

Ταξινομημένη λίστα: [[vpp1, vpp2, vpp3, vpp4, vpp5, vpp6]].



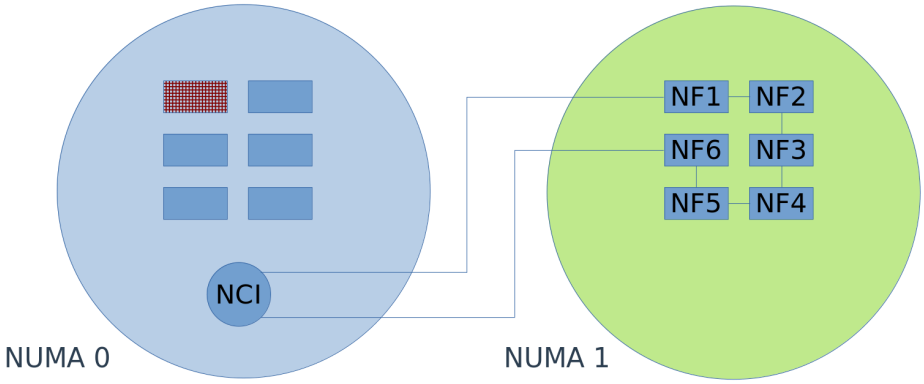
Σχήμα 4.4: Εικόνα ΚΜΕ σεναρίου 1 μετά την τοποθέτηση

Σενάριο 2

Το σενάριο 2 πρόκειται για το σενάριο 1 με την μόνη αλλαγή οτι πλέον ο κόμβος NUMA 0 έχει ελεύθερες 5 ΚΜΕ αντί για 6.

Λίστα εισόδου: [[vpp1, vpp2, vpp3, vpp4, vpp5, vpp6]].

Ταξινομημένη λίστα: [[vpp1, vpp2, vpp3, vpp4, vpp5, vpp6]].



Σχήμα 4.5: Εικόνα ΚΜΕ σεναρίου 2 μετά την τοποθέτηση

Σενάριο 3

Στο σενάριο 3 έχουμε την εικόνα του συστήματος όπως είναι και στο σενάριο 2 με την διαφορά οτι πλέον έχουμε 3 διαφορετικές αλυσίδες αντι για μία, αποτελούμενες απο 3, 3 και 2 εικονικές λειτουργίες αντίστοιχα.

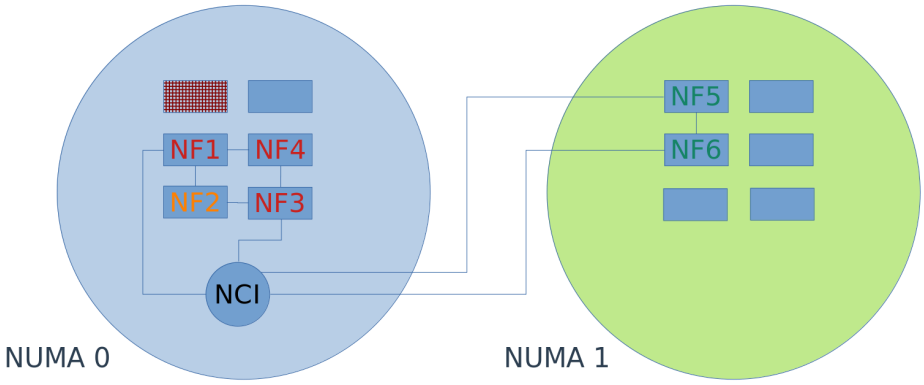
Λίστα εισόδου: [[vpp1, vpp2, vpp3], [vpp1, vpp4, vpp3], [vpp5, vpp6]].

Κίνηση αλυσίδας [vpp1, vpp2, vpp3]: μέτρια.

Κίνηση αλυσίδας [vpp1, vpp4, vpp3]: έντονη.

Κίνηση αλυσίδας [vpp5, vpp6]: ήπια.

Ταξινομημένη λίστα: [[vpp1, vpp4, vpp3], [vpp1, vpp2, vpp3], [vpp5, vpp6]].



Σχήμα 4.6: Εικόνα ΚΜΕ σεναρίου 3 μετά την τοποθέτηση

Σενάριο 4

Το σενάριο 4 πρόκειται για το σενάριο 3 με την μόνη αλλαγή οτι πλέον ο κόμβος NUMA 0 έχει ελεύθερες 3 ΚΜΕ αντί για 5.

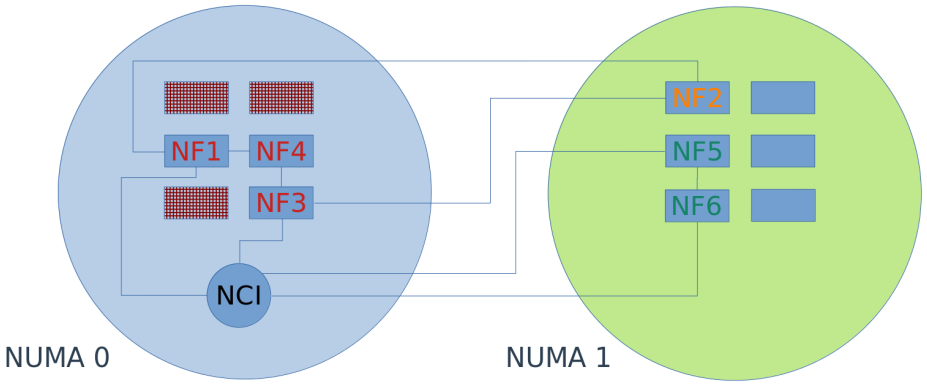
Λίστα εισόδου: [[vpp1, vpp2, vpp3], [vpp1, vpp4, vpp3], [vpp5, vpp6]].

Κίνηση αλυσίδας [vpp1, vpp2, vpp3]: μέτρια.

Κίνηση αλυσίδας [vpp1, vpp4, vpp3]: έντονη.

Κίνηση αλυσίδας [vpp5, vpp6]: ήπια.

Ταξινομημένη λίστα: [[vpp1, vpp4, vpp3], [vpp1, vpp2, vpp3], [vpp5, vpp6]].



Σχήμα 4.7: Εικόνα ΚΜΕ σεναρίου 4 μετά την τοποθέτηση

Σενάριο 5

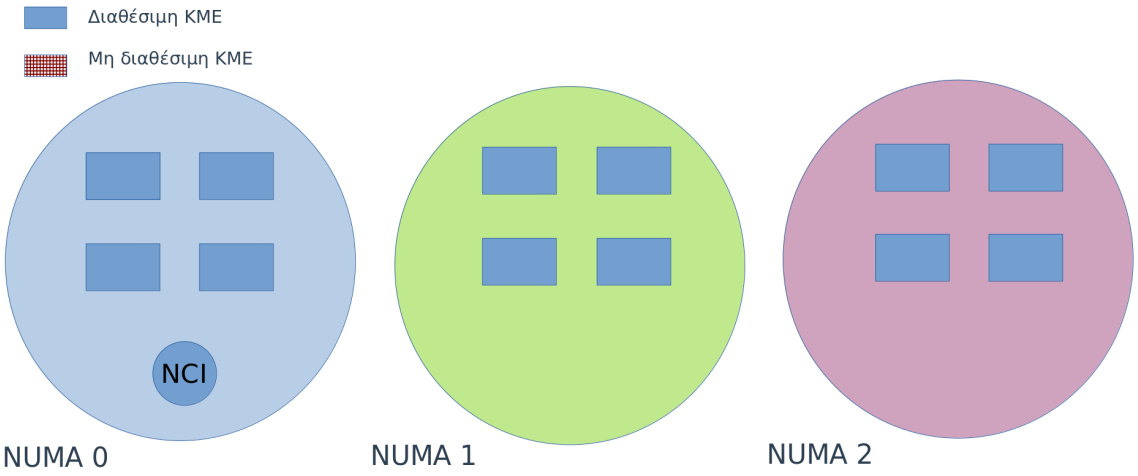
Στο παρακάτω σενάριο έχουμε σύστημα με 3 κόμβους NUMA απο 4 ΚΜΕ επεξεργασίας ο καθένας και την κάρτα δικτύου να βρίσκεται στον κόμβο NUMA 0. Ως λίστα εισόδου έχουμε 3 αλυσίδες αποτελούμενες απο 3 εικονικές λειτουργίες δικτύου η κάθεμια.

Λίστα εισόδου: [[vpp1, vpp2, vpp3], [vpp4, vpp5, vpp6], [vpp7, vpp8, vpp9]].

Κίνηση αλυσίδας [vpp1, vpp2, vpp3]: μέτρια.

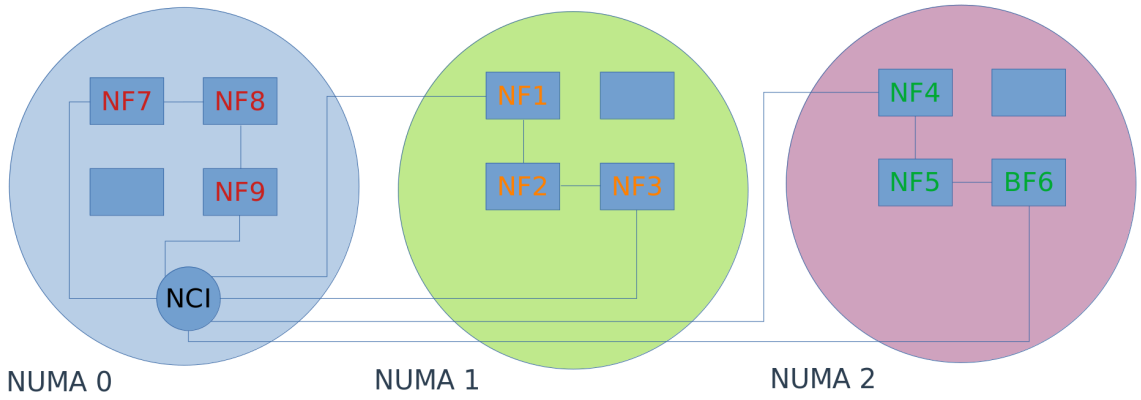
Κίνηση αλυσίδας [vpp4, vpp5, vpp6]: ήπια.

Κίνηση αλυσίδας [vpp7, vpp8, vpp9]: έντονη.



Σχήμα 4.8: Εικόνα ΚΜΕ σεναρίου 5 πριν την τοποθέτηση

Ταξινομημένη λίστα: [[vpp7, vpp8, vpp9], [vpp1, vpp2, vpp3], [vpp4, vpp5, vpp6]].



Σχήμα 4.9: Εικόνα ΚΜΕ σεναρίου 5 μετά την τοποθέτηση

Σενάριο 6

Το σενάριο 6 πρόκειται για το σενάριο 5 με την μόνη αλλαγή οτι πλέον ο κόμβος NUMA 2 έχει ελεύθερη 1 ΚΜΕ αντί για 4.

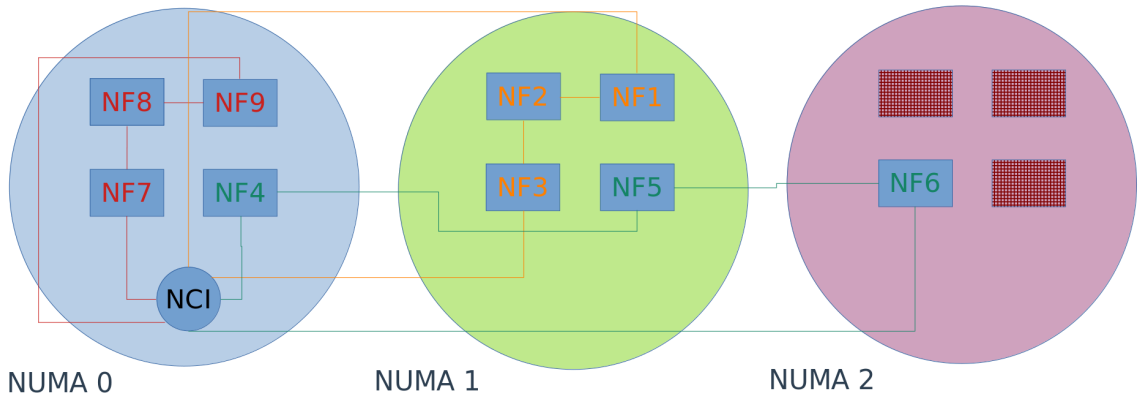
Λίστα εισόδου: [[vpp1, vpp2, vpp3], [vpp4, vpp5, vpp6], [vpp7, vpp8, vpp9]].

Κίνηση αλυσίδας [vpp1, vpp2, vpp3]: μέτρια.

Κίνηση αλυσίδας [vpp4, vpp5, vpp6]: ήπια.

Κίνηση αλυσίδας [vpp7, vpp8, vpp9]: έντονη.

Ταξινομημένη λίστα: [[vpp7, vpp8, vpp9], [vpp1, vpp2, vpp3], [vpp4, vpp5, vpp6]].



Σχήμα 4.10: Εικόνα ΚΜΕ σεναρίου 6 μετά την τοποθέτηση

4.2.2 Με αλλαγή κίνησης στις αλυσίδες

Εδώ θα δούμε τη συμπεριφορά του αλγορίθμου σε ένα δυναμικό σενάριο όπου η κίνηση των αλυσίδων μεταβάλλεται με το χρόνο.

Δυναμικό σενάριο

Θεωρούμε σαν αρχικό σενάριο το σενάριο 3. Θυμίζουμε τα δεδομένα καθώς και την τοποθέτηση που πραγαμτοποιεί ο αλγόριθμος.

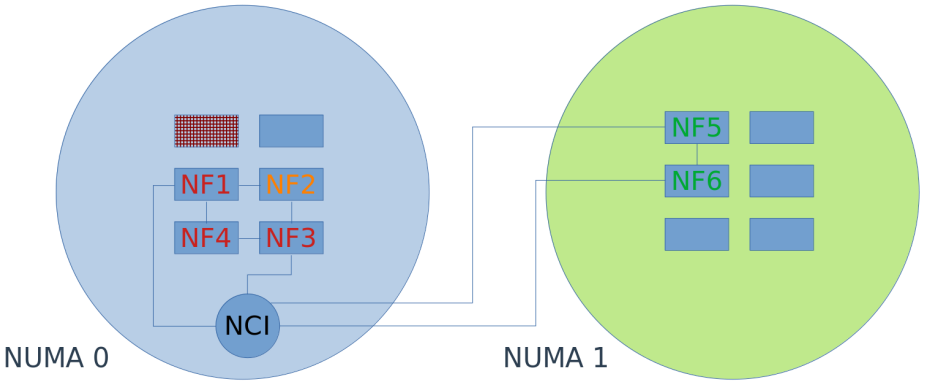
Λίστα εισόδου: [[vpp1, vpp2, vpp3], [vpp1, vpp4, vpp3], [vpp5, vpp6]].

Κίνηση αλυσίδας [vpp1, vpp2, vpp3] την t1: μέτρια.

Κίνηση αλυσίδας [vpp1, vpp4, vpp3] την t1: έντονη.

Κίνηση αλυσίδας [vpp5, vpp6] την t1 : ήπια.

Ταξινομημένη λίστα την t1: [[vpp1, vpp4, vpp3], [vpp1, vpp2, vpp3], [vpp5, vpp6]].



Σχήμα 4.11: Εικόνα ΚΜΕ δυναμικού σεναρίου μετά τοποθέτηση για την t1

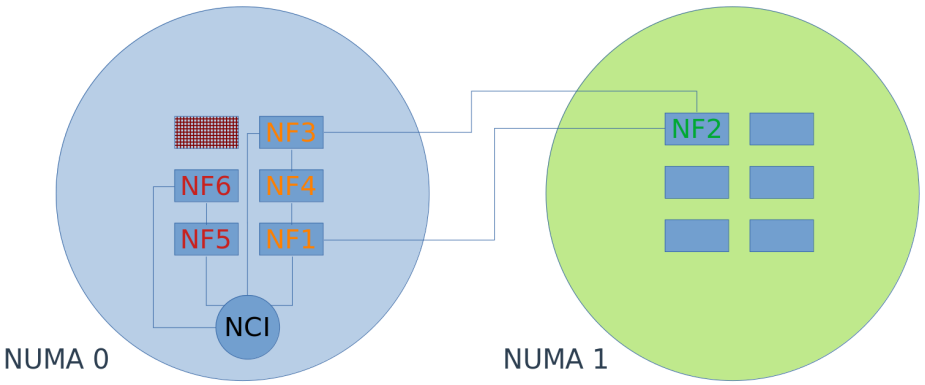
Κατά τη χρονική στιγμή t2 η κίνηση στις αλυσίδες μεταβάλλεται με συνέπεια να αλλάξει η ταξινομημένη λίστα και ο αλγόριθμος να πραγματοποιήσει την εξής τοποθέτηση.

Κίνηση αλυσίδας [vpp1, vpp2, vpp3] την t2: ήπια.

Κίνηση αλυσίδας [vpp1, vpp4, vpp3] την t2: μέτρια.

Κίνηση αλυσίδας [vpp5, vpp6] την t2: έντονη.

Ταξινομημένη λίστα την t2: [[vpp5, vpp6], [vpp1, vpp4, vpp3], [vpp1, vpp2, vpp3]].



Σχήμα 4.12: Εικόνα ΚΜΕ δυναμικού σεναρίου μετά τοποθέτηση για την t2

Κεφάλαιο 5

Πειραματική Αξιολόγηση

Στο παρόν κεφάλαιο παρουσιάζονται αποτελέσματα μετρήσεων που έγιναν σε συστήματα με πραγματική δικτυακή κίνηση. Πιο συγκεκριμένα αναλύεται η καθυστέρηση που καταγράφηκε, στην επικοινωνία μεταξύ των λειτουργιών, με τη χρήση και χωρίς τη χρήση του μηχανισμού δυναμικής τοποθέτησης εικονικών λειτουργιών δικτύου που περιγράφεται στο Κεφάλαιο 4.

5.1 Περιγραφή συστημάτων

Για την διεξαγωγή των πειραμάτων χρησιμοποιήθηκαν 2 πανομοιότητα συστήματα ”Skylake” της Intel. Στο πρώτο σύστημα δημιουργήθηκαν εικονικές λειτουργίες δικτύου χρησιμοποιώντας την τεχνολογία VPP του FD.io για την προσομοίωση κάθε εικονικής λειτουργίας δικτύου. Στο δεύτερο σύστημα παράχθηκε δικτυακή κίνηση, μέσω του TRex της Cisco, και στάλθηκε στο πρώτο.

Χαρακτηριστικά συστήματος **Skylake**:

- 2 επεξεργαστές Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
- 24 πυρήνες ο καθένας, hyperthreaded, 96 ΚΜΕ συνολικά στο σύστημα
- 376 GB RAM
- Καρτα δικτύου Intel XXV710-DA2 10/25GbE SFP28 2-Port PCIe Ethernet Adapter
- 2 κόμβοι NUMA
- ΚΜΕ που βρίσκονται στον κόμβο NUMA 0: 0 - 23, 48 - 71
- ΚΜΕ που βρίσκονται στον κόμβο NUMA 1: 24 - 47, 72 - 95
- Η κάρτα δικτύου βρίσκεται στον κομβο NUMA 1.

node	0	1
0	10	21
1	21	10

Πίνακας 5.1: Σχετικές αποστάσεις Skylake

node	0	1
0	81.8 ns	163.0 ns
1	162.7 ns	81.6 ns

Πίνακας 5.2: Μέση καθυστέρηση Skylake

5.2 Δεδομένα

Τα πειράματα που διεξήχθησαν έγιναν με τα εξής δεδομένα:

- Η κάρτα δικτύου που δέχεται/στέλνει δικτυακή κίνηση είναι τοπική σε συγκεκριμένο κόμβο NUMA, στο συγκεκριμένο σύστημα στον κόμβο 1.
- Κάθε εικονική λειτουργία δικτύου είναι τοποθετημένη σε διαφορετική ΚΜΕ.
- Δεν χρησιμοποιήθηκε υπερνηματική επεξεργασία (hyperthreading).

5.3 Μετρήσεις

Μετρήθηκε η μέση καθυστέρηση για την επεξεργασία και αποστολή των πακέτων απο τις εικονικές λειτουργίες δικτύου που βρίσκονται σύστημα που εξετάζουμε.

Απο εδώ και στο εξής θα αναφερόμαστε στις σε σειρά συνδέδεμενες εικονικές λειτουργίες δικτύου, ως αλυσίδες/υπηρεσίες.

Για το σκοπό των πειραμάτων δημιουργούμε 3 υπηρεσίες, την υπηρεσία 1 αποτελούμενη απο 3, την υπηρεσία 2 αποτελούμενη απο 5 και την υπηρεσία 3 αποτελούμενη απο 7 εικονικές λειτουργίες δικτύου αντίστοιχα.

5.3.1 Κίνηση σε μία μόνο αλυσίδα

Μια κατηγορία πειραμάτων που έχει αξία να εξεταστεί είναι το πώς επηρεάζεται η συνολική καθυστέρηση, με και χωρίς τη χρήση του μηχανισμού δυναμικής τοποθέτησης, όταν ένας χρήστης χρησιμοποιεί **ξεχωριστά** κάθε υπηρεσία/αλυσίδα.

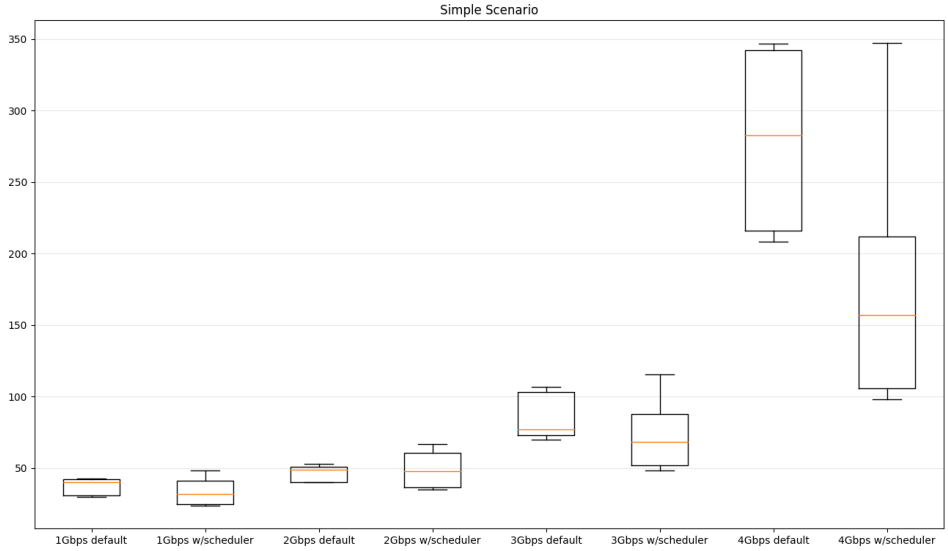
Για το σκοπό αυτό θεωρούμε οτι ένας χρήστης χρησιμοποιεί διαδοχικά, και για το ίδιο χρονικό διάστημα την κάθε μία, τρείς διαφορετικές υπηρεσίες και μετά το πέρας του πειράματος μετράει την συνολική καθυστέρηση.

Δεδομένα:

- Διάρκεια πειράματος 1800 δευτερόλεπτα
- Αλλαγή κίνησης κάθε 10 δευτερόλεπτα
- Καταγραφή μέσης καθυστέρησης κάθε 1 δευτερόλεπτο
- Απόφαση μηχανισμού για επανατοποθέτηση υπηρεσιών κάθε 1 δευτερόλεπτο
- Συνολικά 1800 μετρήσεις καθυστέρησης
- Υπηρεσία 1: vpp1, vpp2, vpp3
- Υπηρεσία 2: vpp4, vpp5, vpp6, vpp7, vpp8
- Υπηρεσία 3: vpp9, vpp10, vpp11, vpp12, vpp13, vpp14, vpp15
- 8 ελεύθερες ΚΜΕ στον κόμβο NUMA 0
- 8 ελεύθερες ΚΜΕ στον κόμβο NUMA 1

5.3.1.1 Ιδανικό Σενάριο

Στο σενάριο αυτό κάθε αλυσίδα βρίσκεται **ολόκληρη** σε κάποιον NUMA κόμβο. Για παράδειγμα οι αλυσίδες 1 και 2 στον κόμβο 0 και η αλυσίδα 3 στον κόμβο 1.



Σχήμα 5.1: Ιδανικό σενάριο

Στις περιοχές σχετικά μικρής κίνησης, 1 και 2 Gbps, γίνεται εμφανές απο την γραφική παράσταση το overhead του συνεχούς scheduling του αλγορίθμου. Παρόλο που η μέση καθυστέρηση παρουσιάζει μικρή μείωση, οι τιμές τις καθυστέρησης ”απλώνονται” περισσότερο. Αυτό συμβαίνει διότι ο αλγόριθμός μας επανατοποθετεί συνεχώς τις εικονικές λειτουργίες δικτύου και ο χρόνος που χάνεται η επικοινωνία των λειτουργίων σε αυτό το στάδιο είναι συγκρίσιμος με το χρόνο της καθυστέρησης λόγω κόμβων NUMA.

Στις περιοχές μεγαλύτερης κίνησης ωστόσο και ιδιαίτερα σε αυτή των 4 Gbps, παρατηρούμε σημαντικές μειώσεις στην καθυστέρηση, σχεδόν στο μισό, αφού το overhead της επανατοποθέτησης λειτουργιών δικτύου έχει πλέον αμελητέα επίδραση στην καθυστέρηση που οφείλεται λόγω NUMA.

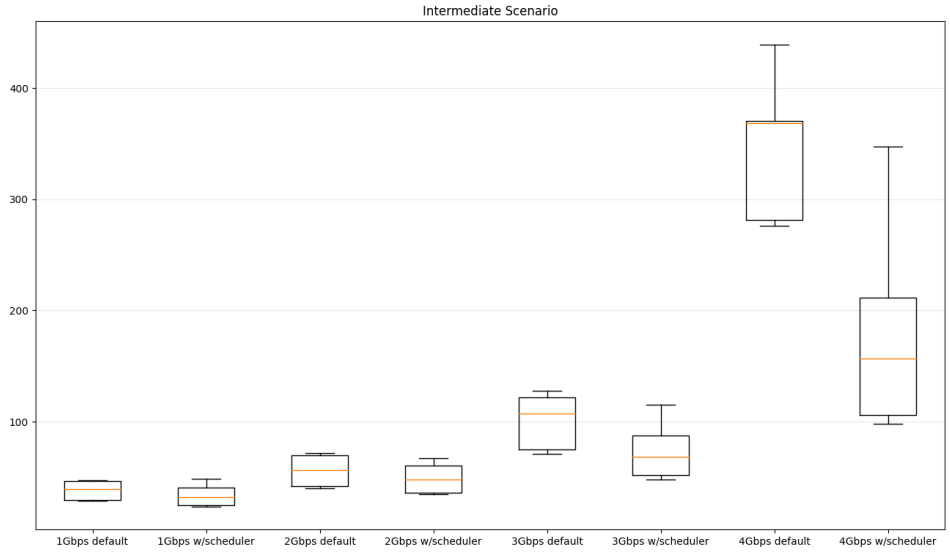
	3500 Mbps	4000 Mbps	4500 Mbps	5000 Mbps
Με αλγόριθμο	-	-	100 - 300 Mbps περιστασιακό στην επανατοποθέτηση	500 Mbps
Χωρίς αλγόριθμο	100 - 300 Mbps περιστασιακό όταν κίνηση στον κόμβο NUMA 0	500 - 600 Mbps	> 800 Mbps	> 1000 Mbps

Πίνακας 5.3: Drop rates ιδανικού σεναρίου

Επίσης ένα πολύ σημαντικό πλεονέκτημα στην περίπτωση που εφαρμόζεται ο αλγόριθμος είναι αυτο του περιορισμού και της μείωσης των drop rates. Συγκεκριμένα χωρίς τον αλγόριθμο εμφανίζεται περιστασιακό drop rate της τάξεως 100 - 300 Mbps όταν η κίνηση στέλνεται στον κόμβο NUMA 0 στα 3500 Mbps ενώ για κίνηση 4000 Mbps και άνω το drop rate είναι συνεχές. Κάτι τέτοιο περιορίζεται στην περίπτωση που στο πείραμα εφαρμόζεται ο αλγόριθμος με εμφάνιση περιστασιακού drop rate κατά την επανατοποθέτηση στα 4500 Mbps. Επιτυγχάνεται λοιπόν μια αντοχή στο σύστημα για 1Gbps παραπάνω κίνηση χωρίς droprate.

5.3.1.2 Μέσο Σενάριο

Στο σενάριο αυτό κάθε αλυσίδα βρίσκεται **μισή** σε κάποιον κόμβο NUMA (πχ στον κόμβο 0) **και μισή** στον άλλον (κόμβο 1).



Σχήμα 5.2: Μέσο σενάριο

Πλέον λόγω της τοποθέτησης παρατηρείται διασπορά της καθυστέρησης και στην περίπτωση του πειράματος χωρίς τον αλγόριθμο στην περιοχή των μικρών κινήσεων. Η διαφορά στην καθυστέρηση, με και χωρίς τον αλγόριμθο, μεγαλώνει ακόμα περισσότερο σε κίνηση 3Gbps και άνω σε σχέση με το προηγούμενο σενάριο.

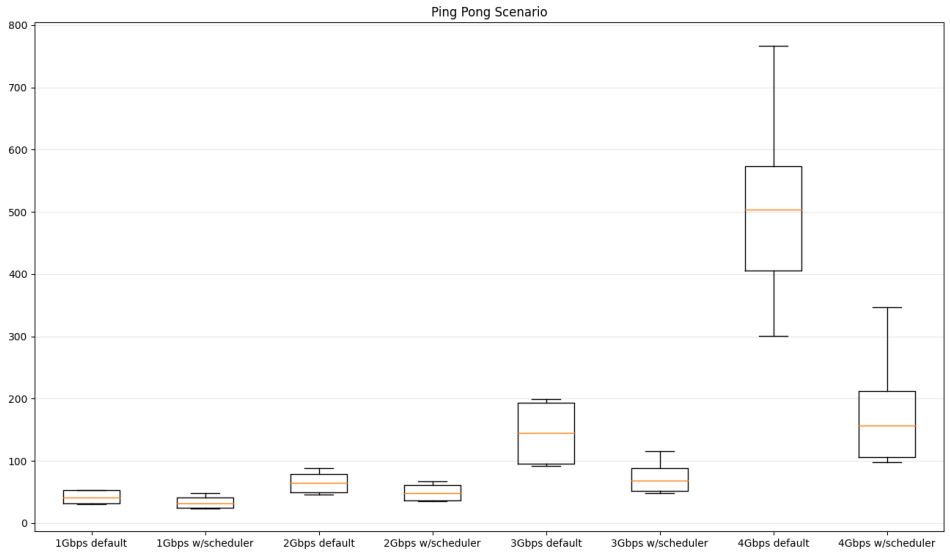
	3500 Mbps	4000 Mbps	4500 Mbps	5000 Mbps
Με αλγόριθμο	-	-	100 - 300 Mbps περιστασιακό στην επανατοποθέτηση	500 Mbps
Χωρίς αλγόριθμο	200 - 400 Mbps περιστασιακό όταν κίνηση στον κόμβο NUMA 0	500 - 800 Mbps	> 800 Mbps	> 1000 Mbps

Πίνακας 5.4: Drop rates μέσου σεναρίου

Προκύπτουν αντίστοιχα αποτελέσματα με αυτά του πίνακα για τα drop rates του ιδανικού σεναρίου με τη διαφορά οτι το μέγεθος του drop rate παρουσιάζει μικρή άνοδο.

5.3.1.3 Χειρότερο Σενάριο

Στο σενάριο αυτό **κάθε διαδοχικός** κόμβος κάθε αλυσίδας βρίσκεται σε **διαφορετικό** κόμβο NUMA (πχ για την αλυσίδα 1: vpp1 στον NUMA 0, vpp2 στον NUMA 1 και vpp3 στον NUMA 0).



Σχήμα 5.3: Χειρότερο σενάριο

Τα αποτελέσματα είναι αντίστοιχα με αυτά των γραφικών παραστάεων των δύο προηγούμενων σεναρίων με τον αλγόριθμο να επιτυγχάνει και μικρότερη διασπορά πλέον σε μικρή κίνηση. Επίσης η διαφορά στην καθυστέρηση σε μεγάλη κίνηση οξύνεται ακόμα περισσότερο. Χαρακτηριστικά, σε κίνηση 4Gbps έχουμε σχεδόν την τριπλάσια μέση καθυστέρηση στην επικοινωνία, στο τρέξιμο χωρίς τον αλγόριθμο απ' οτι στο τρέξιμο με τον αλγόριθμο.

	3500 Mbps	4000 Mbps	4500 Mbps	5000 Mbps
Με αλγόριθμο	-	-	100 - 300 Mbps περιστασιακό στην επανατοποθέτηση	500 Mbps
Χωρίς αλγόριθμο	400 - 600 Mbps περιστασιακό όταν κίνηση στον κόμβο NUMA 0	700 - 800 Mbps	> 800 Mbps	> 1000 Mbps

Πίνακας 5.5: Drop rates χειρότερου σεναρίου

Αντίστοιχα αποτελέσματα με αυτά των πινάκων για τα drop rates των δύο προηγούμενων σεναρίων με το μέγεθος του drop rate να μεγαλώνει περισσότερο.

5.3.2 Κίνηση σε όλες τις αλυσίδες ταυτόχρονα

Μια άλλη κατηγορία πειραμάτων που έχει αξία να εξεταστεί είναι το πώς επηρεάζεται η συνολική καθυστέρηση, με και χωρίς τη χρήση του μηχανισμού δυναμικής τοποθέτησης, όταν ένας χρήστης χρησιμοποιεί ταυτόχρονα υπηρεσίες/αλυσίδες.

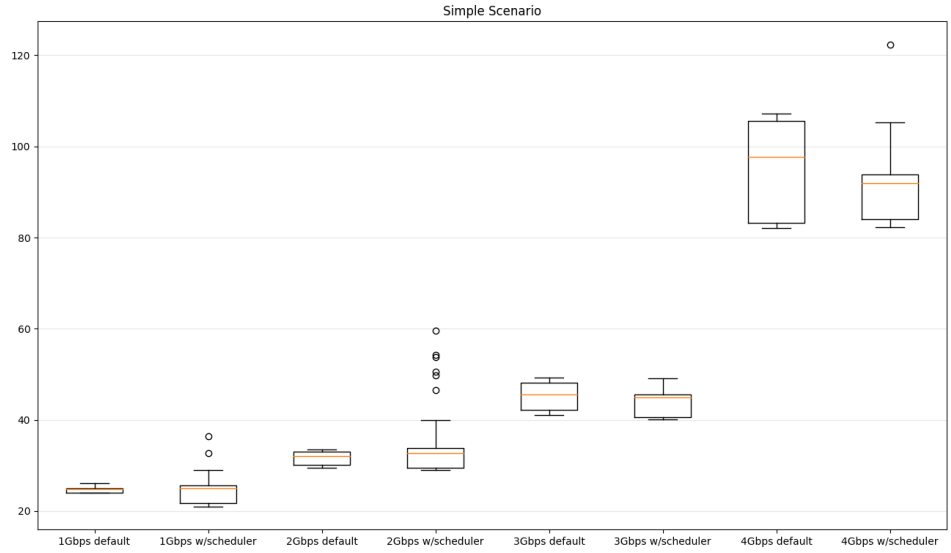
Για το σκοπό αυτό θεωρούμε οτι ένας χρήστης χρησιμοποιεί **ταυτόχρονα**, και τις 3 διαφορετικές υπηρεσίες, αλλάζοντας την κίνηση που δέχεται η κάθε μία περιοδικά.

Δεδομένα:

- Διάρκεια πειράματος 1800 δευτερόλεπτα
- Αλλαγή κίνησης κάθε 10 δευτερόλεπτα
- Καταγραφή μέσης καθυστέρησης κάθε 1 δευτερόλεπτο
- Απόφαση μηχανισμού για επανατοποθέτηση υπηρεσιών κάθε 1 δευτερόλεπτο
- Συνολικά 1800 μετρήσεις καθυστέρησης
- Υπηρεσία 1: vpp1, vpp2, vpp3
- Υπηρεσία 2: vpp4, vpp5, vpp6, vpp7, vpp8
- Υπηρεσία 3: vpp9, vpp10, vpp11, vpp12, vpp13, vpp14, vpp15
- 8 ελεύθερες ΚΜΕ στον κόμβο NUMA 0
- 8 ελεύθερες ΚΜΕ στον κόμβο NUMA 1

5.3.2.1 Ιδανικό Σενάριο

Στο σενάριο αυτό κάθε αλυσίδα βρίσκεται **ολόκληρη** σε κάποιον NUMA κόμβο. Για παράδειγμα οι αλυσίδες 1 και 2 στον κόμβο 0 και η αλυσίδα 3 στον κόμβο 1.



Σχήμα 5.4: Ιδανικό σενάριο

Συγκρίνοντας την γραφική παράσταση με αυτήν του Σχ.5.1 προκύπτουν παρόμοια συμπεράσματα. Η διαφορά στην επικοινωνία εδώ δεν είναι τόσο έντονη, δεδομένου ότι η κίνηση υπάρχει πλέον σε όλες της αλυσίδες ταυτόχρονα και όχι μόνο σε μία.

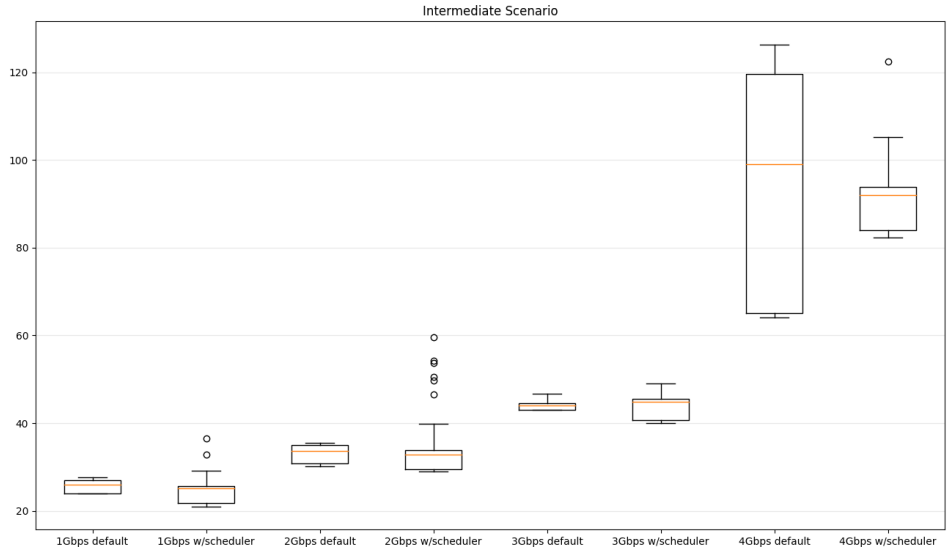
	4000 Mbps	4500 Mbps	5000 Mbps
Με αλγόριθμο	-	100 - 300 Mbps περιστασιακό στην επανατοποθέτηση	500 Mbps
Χωρίς αλγόριθμο	100 - 300 Mbps	> 500 Mbps	> 800 Mbps

Πίνακας 5.6: Drop rates ιδανικού σεναρίου

Από τον συγκεκριμένο πίνακα των drop rates παρατηρείται εμφάνιση περιστασιακού drop rate λίγο αργότερα απ' ότι στα προηγούμενα σενάρια (500 Mbps μετά) πράγμα που επίσης οφείλεται στο ότι η κίνηση υπάρχει πλέον σε όλες τις αλυσίδες ταυτόχρονα. Και πάλι ο αλγόριθμος μας προσδίδει ανθεκτικότητα στο σύστημα, εδώ 500 Mbps.

5.3.2.2 Μέσο Σενάριο

Στο σενάριο αυτό κάθε αλυσίδα βρίσκεται **μισή** σε κάποιον κόμβο NUMA (πχ στον κόμβο 0) **και μισή** στον άλλον (κόμβο 1).



Σχήμα 5.5: Μέσο σενάριο

Παρόμοια συμπεράσματα με το προηγούμενο σενάριο με όξυνση της διασποράς, στην περίπτωση του πειράματος χωρίς τον αλγόριθμο, σε κίνηση 4Gbps.

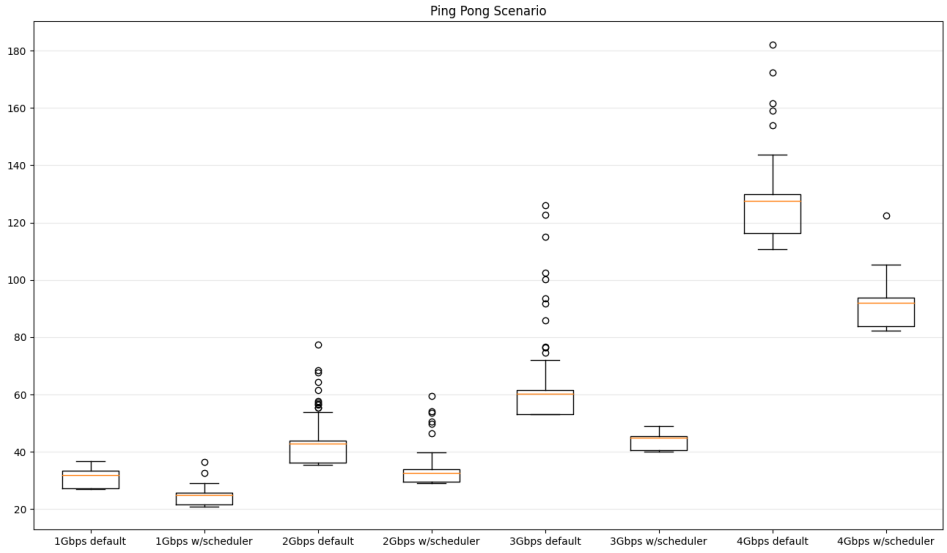
	4000 Mbps	4500 Mbps	5000 Mbps
Με αλγόριθμο	-	100 - 300 Mbps περιστασιακό στην επανατοποθέτηση	500 Mbps
Χωρίς αλγόριθμο	200 - 400 Mbps	> 600 Mbps	> 800 Mbps

Πίνακας 5.7: Drop rates μέσου σεναρίου

Αντίστοιχα αποτελέσματα με αυτά του πίνακα για τα drop rates του ιδανικού σεναρίου με τη διαφορά οτι το μέγεθος του drop rate παρουσιάζει μικρή άνοδο.

5.3.2.3 Χειρότερο Σενάριο

Στο σενάριο αυτό **κάθε διαδοχικός** κόμβος κάθε αλυσίδας βρίσκεται σε **διαφορετικό** κόμβο NUMA (πχ για την αλυσίδα 1: vpp1 στον NUMA 0, vpp2 στον NUMA 1 και vpp3 στον NUMA 0).



Σχήμα 5.6: Χειρότερο σενάριο

Προκύπτουν αντίστοιχα συμπεράσματα με τις γραφικές των 2 προηγούμενων σεναρίων. Διαφορές που παρατηρούνται είναι η όξυνση της διαφοράς της καθυστέρησης μεταξύ των δυο πειραμάτων σε όλα τα μεγέθη κίνησης.

	4000 Mbps	4500 Mbps	5000 Mbps
Με αλγόριθμο	-	100 - 300 Mbps περιστασιακό στην επανατοποθέτηση	500 Mbps
Χωρίς αλγόριθμο	200 - 500 Mbps	> 800 Mbps	> 800 Mbps

Πίνακας 5.8: Drop rates χειρότερου σεναρίου

Αντίστοιχα αποτελέσματα με αυτά των πινάκων για τα drop rates των δύο προηγούμενων σεναρίων με το μέγεθος του drop rate να μεγαλώνει περισσότερο.

Κεφάλαιο 6

Επίλογος - Μελλοντικές Επεκτάσεις

Στο συγκεκριμένο κεφάλαιο παρουσιάζονται συνοπτικά το κίνητρο, οι ενέργειες και τα αποτελέσματα της συγκεκριμένης εργασίας και εξετάζονται οι πιθανές μελλοντικές προεκτάσεις του αλγορίθμου δυναμικής τοποθέτησης εικονικών λειτουργιών δικτύου.

6.1 Επίλογος

Η συγκεκριμένη διπλωματική εργασία ξεκίνησε ως μελέτη της απόδοσης αλυσίδων απο εικονικές λειτουργίες δικύου και πώς οι στατικές πολιτικές που υπάρχουν μέχρι σήμερα για την τοποθέτηση τους σε πολυεπεξεργαστικά συστήματα την επηρεάζουν. Μέσα απο πειραματικές μετρήσεις αποδείχθηκε οτι ακόμη και για σχετικά μικρή κίνηση, 1000 Mbps, η μείωση της απόδοσης μιας τηλεπικοινωνιακής υπηρεσίας λόγω κακής τοποθέτησης μπορεί να αγγίξει και το 40 τοις εκατό. Στην συνέχεια αναπτύχθηκε αλγόριθμος που διασφαλίζει την δυναμική τοποθέτηση των αλυσίδων στις επεξεργαστικές μονάδες βάσει της κίνησης τους.

Ο αλγόριθμος αυτός αξιολογήθηκε με δεδομένα πραγματικής δικτυακής κίνησης και προέκυψαν τα εξής συμπεράσματα. Σε περιοχές σχετικά μικρής κίνησης, 1 και 2 Gbps, γίνεται εμφανές το overhead του συνεχούς scheduling του αλγορίθμου. Παρόλο που η μέση καθυστέρηση παρουσιάζει μικρή μείωση, οι τιμές τις καθυστέρησης ”απλώνονται” περισσότερο λόγω του ότι ο αλγόριθμός μας επανατοποθετεί συνεχώς τις εικονικές λειτουργίες δικτύου. Αυτό συμβαίνει διότι ο χρόνος που χάνεται η επικοινωνία των λειτουργίων κατά την συγκεκριμένη διαδικασία είναι συγκρίσιμος με το χρόνο της καθυστέρησης λόγω NUMA. Στις περιοχές μεγαλύτερης κίνησης απο την άλλη, παρατηρούμε σημαντικές μειώσεις στην καθυστέρηση αφού το overhead της επανατοποθέτησης λειτουργιών δικτύου έχει πλέον αμελητέα επίδραση στην καθυστέρηση λόγω NUMA. Τέλος, μια πολύ σημαντική συνέπεια του αλγορίθμου που αξίζει να σημειωθεί είναι η μείωση του droprate. Σε ορισμένα πειράματα το σύστημα μπορεί να δεχτεί μέχρι και 1 Gbps παραπάνω κίνηση μέχρι να εμφανισθεί droprate κάτι που το καθιστά περισσότερο ανθεκτικό.

6.2 Μελλοντικές επεκτάσεις

Μπορούμε να αναλύσουμε τις μελλοντικές προεκτάσεις του αλγορίθμου σε 2 κατευθύνσεις, μια προς την βελτίωση του αλγορίθμου όσον αφορά παραμέτρους του συστήματος στο οποίο εκτελείται και μια προς την κατεύθυνση των εικονικών λειτουργιών δικτύου.

Ως προς τις παραμέτρους του συστήματος στο οποίο εκτελείται, μερικές βελτιώσεις είναι η αυτοματοποίηση ορισμένων ενεργειών που είναι αναγκαίες για να τρέξει ο αλγόριθμος. Η ανα-

κάλυψη δηλαδή των διαθέσιμων κεντρικών επεξεργαστικών μονάδων, η δημιουργία του πίνακα για τις σχετικές αποστάσεις κόμβων NUMA καθώς και η έυρεση της θέσης της κάρτας δικτύου σε σχέση με του κόμβους NUMA. Στην σημερινή μορφή του αλγορίθμου και οι τρείς αυτοί παράμετροι πρέπει να εισάγονται χειροκίνητα πριν το τρέξιμο, στον πηγαίο του κώδικα.

Ως προς την κατεύθυση των εικονικών λειτουργιών δικτύου μια απ' τις μελλοντικές προεκτάσεις του αλγορίθμου είναι η αυτόματη ανακάλυψη των εικονικών λειτουργιών που βρίσκονται στο σύστημα καθώς και η συνδεσμολογία τους. Η πληροφορία αυτή μέχρι στιγμής είναι η είσοδος του αλγορίθμου που εισάγει ο χρήστης στην αρχή της εκτέλεσης. Μια άλλη πιθανή βελτίωση είναι η απόφαση του αλγορίθμου για επανατοποθέτηση των λειτουργιών να μην γίνεται κάθε συγκεκριμένο χρονικό διάστημα αλλά όποτε υπάρχει αλλαγή κίνησης στις αλυσίδες κάτι το οποίο θα πρέπει ο μηχανισμός μας να ανακαλύπτει. Τέλος για λόγους επαλήθευσης της αρχής, η συγκεκριμένη διπλωματική εργασία χρησιμοποιεί ένα πολύ συγκεκριμένο είδος εικονικών λειτουργιών δικτύου, το VPP. Μία πολύ χρήσιμη επέκταση θα ήταν η απεμπλοκή του απο ένα συγκεκριμένο είδος και η δυνατότητα να χρησιμοποιηθεί ανεξάρτητα με κάθε πιθανή εικονική λειτουργία δικτύου.

Βιβλιογραφία

[1] Cache thrashing. https://en.wikipedia.org/wiki/Thrashing_(computer_science).

[2] Data plane development kit. https://www.dpdk.org/.

[3] Etsi - standards for nfv - network functions virtualisation | nfv solutions. https://www.etsi.org/technologies/nfv.

[4] How low-cost telecom killed five 9s in cloud computing. https://www.wired.com/insights/2013/03/how-low-cost-telecom-killed-five-9s-in-cloud-computing/.

[5] Linux foundation networking. https://www.lfnetworking.org/.

[6] Network function virtualization. https://en.wikipedia.org/wiki/Network_function_virtualization.

[7] Network functions virtualization - introductory white paper. https://portal.etsi.org/NFV/NFV_White_Paper.pdf.

[8] Open network automation platform. https://www.onap.org/.

[9] Open source mano. https://osm.etsi.org/.

[10] Trex - realistic traffic generator. https://trex-tgn.cisco.com/.

[11] Vector packet processing. https://fd.io/.

[12] P. Emmerich, M. Pudelko, S. Bauer, S. Huber, T. Zwickl, and G. Carle. User space network drivers. In *2019 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, pages 1–12, 2019.

[13] C. Lameter. An overview of non-uniform memory access. *Commun. ACM*, 56(9):59–54, Sept. 2013.

[14] A. Lara, A. Kolasani, and B. Ramamurthy. Network innovation using openflow: A survey. *IEEE Communications Surveys Tutorials*, 16(1):493–512, 2014.

[15] C. Makaya, D. Freimuth, D. Wood, and S. Calo. Policy-based nfv management and orchestration. In *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, pages 128–134, 2015.

[16] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys Tutorials*, 18(1):236–262, 2016.

[17] D. Staessens, S. Sharma, D. Colle, M. Pickavet, and P. Demeester. Software defined networking: Meeting carrier grade requirements. In *2011 18th IEEE Workshop on Local Metropolitan Area Networks (LANMAN)*, pages 1–6, 2011.