



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Βελτίωση Ποιότητας Υπηρεσίας μέσω Εξελιγμένων
Τεχνικών Κατάτμησης Κρυφής Μνήμης**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΛΕΥΘΕΡΙΑ ΜΑΠΠΟΥΡΑ

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

Αθήνα, Οκτώβριος 2020



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Βελτίωση Ποιότητας Υπηρεσίας μέσω Εξελιγμένων Τεχνικών Κατάτμησης Κρυφής Μνήμης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΛΕΥΘΕΡΙΑ ΜΑΠΠΟΥΡΑ

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

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 7^η Οκτωβρίου 2020.

(Υπογραφή)

.....

Γεώργιος Γκούμας
Επ. Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....

Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....

Διονύσιος Πνευματικάτος
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2020

(Υπογραφή)

.....
ΜΑΠΠΟΥΡΑ ΕΛΕΥΘΕΡΙΑ

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

Copyright © Ελευθερία Μαπούρα, 2020

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

Στα συστήματα μεγάλης κλίμακας, όπως οι υπηρεσίες νέφους και τα data centers, η εξασφάλιση ικανοποιητικής Ποιότητας Υπηρεσίας (Quality of Service, QoS), αποτελεί το κυριότερο ζήτημα. Λόγω του μεγάλου αριθμού εφαρμογών που συνεκτελούνται σε αυτές τις περιπτώσεις, η εξασφάλιση του QoS του συστήματος δεν μπορεί να είναι πάντοτε εφικτή. Πρωταρχικό μέλημα λοιπόν σε αυτές τις περιπτώσεις καθίσταται η διατήρηση του Service Level Objective (SLO) σε εφαρμογές με υψηλή προτεραιότητα.

Η παράλληλη εκτέλεση πολλών εφαρμογών, που επιτυγχάνεται λόγω των σύγχρονων πολυπύρηνων επεξεργαστικών συστημάτων, συχνά οδηγεί σε μείωση του QoS εξαιτίας του ανταγωνισμού για τους διαμοιραζόμενους πόρους του συστήματος. Ο ανταγωνισμός μπορεί να αντιμετωπιστεί με ποικίλους τρόπους. Ιδιαίτερο ενδιαφέρον παρουσιάζεται στον διαμερισμό των κοινόχρηστων πόρων μεταξύ των συνεκτελούμενων εφαρμογών και πιο συγκεκριμένα στον διαμερισμό της κρυφής μνήμης τελευταίου επιπέδου (Last Level Cache, LLC). Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός δυναμικού μηχανισμού για τον διαμοιρασμό της LLC μεταξύ των εφαρμογών, ανάλογα με τις ανάγκες της εφαρμογής με υψηλή προτεραιότητα, εξασφαλίζοντας έτσι το QoS και το SLO.

Ο μηχανισμός υλοποιήθηκε σε επεξεργαστή Intel® Xeon® E5-2630 v4 με 20 πυρήνες και 25MB LLC (20-way associative) και εφαρμόστηκε σε benchmarks από τη σουίτα SPEC CPU 2006. Για την επίβλεψη της επίδοσης των εφαρμογών και τον διαμερισμό της LLC χρησιμοποιήθηκαν οι τεχνολογίες του εργαλείου PQoS/Intel® RDT, Cache Monitoring Technology (CMT) και Cache Allocation Technology (CAT) αντίστοιχα. Οι εφαρμογές εκτελούνταν σε εικονικές μηχανές οι οποίες κατανέμονταν στους πυρήνες μέσω του διαχειριστή πόρων και ανιχνευτή ανταγωνισμού για υπηρεσίες νέφους, ACTiManager.

Λέξεις Κλειδιά: Ποιότητα Υπηρεσίας, Service Level Objective, Ανταγωνισμός για κοινόχρηστους πόρους, Διαμοιρασμός κρυφής μνήμης, Cache Monitoring Technology, Cache Allocation Technology

Abstract

In large-scale systems, such as cloud providers and data centers, ensuring satisfactory Quality of Service (QoS) remains the main matter. Due to the vast number of co-executed workloads in such systems, ensuring QoS is not always feasible. Thus, the primary concern in large-scale systems is to maintain the Service Level Objective (SLO) of high priority applications.

Workload co-execution is achieved by the modern multi-core processing systems and can usually lead to QoS degradation due to the competition for the system's shared resources. This phenomenon can be resolved in various ways. There is a particular interest in allocating the shared resources within the co-executed workloads, and more precisely in allocating the last level cache (LLC) amongst workloads. The scope of this thesis is to develop a dynamic mechanism that allocates the LLC amongst co-executed workloads, depending on the needs of the high priority workload, while ensuring the QoS and the SLO.

The mechanism was implemented on an Intel® Xeon® E5-2630 v4 processor which includes 20 cores and a 25MB LLC (20-way associative) and was tested on workloads from the benchmark suite SPEC CPU 2006. Performance monitoring and LLC allocation were achieved with the use of two technologies of PQoS/Intel® RDT tool, Cache Monitoring Technology (CMT) and Cache Allocation Technology (CAT) respectively. Workloads were executed on virtual machines and the virtual machines were allocated within the system's cores with the use of ACTiManager, an end-to-end interference-aware manager for cloud resources.

Key Words: Quality of Service, Service Level Objective, Contention for shared resources, Last level cache allocation, Cache Monitoring Technology, Cache Allocation Technology

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Δρα Γκούμα Γεώργιο ο οποίος μου έδωσε την ευκαιρία να ασχοληθώ με αυτό το θέμα. Ειδικότερα θα ήθελα να εκφράσω τις ευχαριστίες μου στο μεταδιδακτορικό ερευνητή Δρα Κωνσταντίνο Νίκα καθώς και τους υποψήφιους διδάκτορες κ. Ιωάννη Παπαδάκη και κ. Δημήτριο Σιακαβάρα για τη διαρκή καθοδήγησή τους, τις χρήσιμες προτάσεις και συμβουλές τους καθώς και την συνεχή υποστήριξή τους.

Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου, τα αδέρφια μου και τους φίλους μου, οι οποίοι βρίσκονταν δίπλα μου και με στήριζαν όλο αυτό το χρονικό διάστημα.

Περιεχόμενα

1. Εισαγωγή.....	1
1.1. Πολυπύρηνες Αρχιτεκτονικές	1
1.2. Ανταγωνισμός Συνεκτελούμενων Εφαρμογών	2
<i>1.2.1. Κρυφή Μνήμη Τελευταίου Επιπέδου.....</i>	<i>4</i>
1.3. Αντιμετώπιση του Ανταγωνισμού	5
<i>1.3.1. Profiling</i>	<i>6</i>
<i>1.3.2. Online Monitoring.....</i>	<i>7</i>
1.4. Αντικείμενο Διπλωματικής	8
<i>1.4.1. Συνεισφορά.....</i>	<i>8</i>
1.5. Οργάνωση Κειμένου	9
2. Μηχανισμοί Διαχείρισης Κρυφής Μνήμης	10
2.1. Υλοποιήσεις Βασισμένες σε Profiling/Online Monitoring	10
2.2. Υλοποιήσεις Συνδυασμού Profiling/Online Monitoring και Κατανομής της Cache	12
3. Τεχνολογίες Intel® CMT-CAT	15
3.1. CMT	15
3.2. CAT	17
4. Μελέτη Εφαρμογών	19
4.1. Πληροφορίες Συστήματος.....	19
<i>4.1.1. Επεξεργαστής.....</i>	<i>19</i>
<i>4.1.2. ACTiManager.....</i>	<i>20</i>
4.2. Πειραματική Διαδικασία.....	21
<i>4.2.1. Επίδοση Εφαρμογών σε απομονωμένη εκτέλεση</i>	<i>22</i>
<i>4.2.2. Επίδοση εφαρμογών σε σενάρια παράλληλης εκτέλεσης χωρίς κατανομή της LLC.....</i>	<i>23</i>

4.2.3. Επίδοση εφαρμογών σε σενάρια παράλληλης εκτέλεσης με στατική κατανομή της LLC.....	27
4.2.4. Συνολικό κέρδος στο σύστημα σε σενάρια παράλληλης εκτέλεσης με στατική κατανομή της LLC.....	33
4.3. Χαρακτηριστικά Εφαρμογών	39
5. Δυναμικός Μηχανισμός Διατήρησης του SLO με Κατανομή της LLC.....	41
5.1. Δομή και Λειτουργία.....	41
5.2. Αξιολόγηση Μηχανισμού.....	45
5.2.1. Επίδοση εφαρμογών	45
5.2.2. Συνολικό κέρδος στο σύστημα	50
6. Επίλογος	57
6.1. Σύνοψη και Συμπεράσματα	57
6.2. Μελλοντικές Επεκτάσεις	58
7. Βιβλιογραφία	59
8. Παράρτημα	61
8.1. Normalized IPC.....	61
8.2. Συνολικό Κέρδος Συστήματος.....	75

Ευρετήριο Γραφημάτων

Γράφημα 1-1: Παράδειγμα Normalized IPC.....	3
Γράφημα 1-2: Παράδειγμα LLC Misses	5
Γράφημα 4-1: Normalized IPC σε σενάρια συνεκτέλεσης χωρίς κατανομή της LLC (i)	25
Γράφημα 4-2: Normalized IPC σε σενάρια συνεκτέλεσης χωρίς κατανομή της LLC (ii)	25
Γράφημα 4-3: Normalized IPC σε σενάρια συνεκτέλεσης χωρίς κατανομή της LLC (iii).....	26
Γράφημα 4-4: Normalized IPC σε σενάρια συνεκτέλεσης χωρίς κατανομή της LLC (iv)	26
Γράφημα 4-5: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - astar	28
Γράφημα 4-6: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - bzip2	29
Γράφημα 4-7: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - mcf	29
Γράφημα 4-8: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - milc	30
Γράφημα 4-9: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - namd	30
Γράφημα 4-10: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - libquantum.....	31
Γράφημα 4-11: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - omnetpp.....	31
Γράφημα 4-12: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - soplex.....	32
Γράφημα 4-13: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - xalancbmk	32
Γράφημα 4-14: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – astar ως gold.....	34
Γράφημα 4-15: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – bzip2 ως gold.....	34
Γράφημα 4-16: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – mcf ως gold	35
Γράφημα 4-17: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – milc ως gold.....	35
Γράφημα 4-18: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – namd ως gold.....	36

Γράφημα 4-19: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – libquantum ως gold.....	36
Γράφημα 4-20: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – omnetpp ως gold.....	37
Γράφημα 4-21: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – soplex ως gold	37
Γράφημα 4-22: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – xalancbmk ως gold	38
Γράφημα 5-1: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - astar.....	45
Γράφημα 5-2: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - bzip2.....	46
Γράφημα 5-3: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - mcf	46
Γράφημα 5-4: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - milc	47
Γράφημα 5-5: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - namd.....	47
Γράφημα 5-6: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - libquantum	48
Γράφημα 5-7: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - omnetpp	48
Γράφημα 5-8: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - soplex	49
Γράφημα 5-9: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - xalancbmk.....	49
Γράφημα 5-10: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – astar ως gold.....	51
Γράφημα 5-11: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – bzip2 ως gold.....	51
Γράφημα 5-12: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – mcf ως gold	52
Γράφημα 5-13: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – milc ως gold	52
Γράφημα 5-14: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – namd ως gold.....	53

Γράφημα 5-15: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – libquantum ως gold	53
Γράφημα 5-16: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – omnetpp ως gold.....	54
Γράφημα 5-17: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – soplex ως gold.....	54
Γράφημα 5-18: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – xalancbmk ως gold.....	55

Ευρετήριο Πινάκων

Πίνακας 4-1: Χαρακτηριστικά του επεξεργαστή Intel® Xeon® E5-2630 v4.....	20
Πίνακας 4-2: Χαρακτηριστικά εφαρμογών σε απομονωμένη εκτέλεση.....	23
Πίνακας 4-3: Σενάρια συνεκτέλεσης 1-5	24
Πίνακας 4-4: Σενάρια συνεκτέλεσης 6-10	24
Πίνακας 4-5: Συμπεριφορά των benchmarks	40
Πίνακας 8-1: Normalized IPC - astar	61
Πίνακας 8-2: Normalized IPC - bzip2.....	62
Πίνακας 8-3: Normalized IPC - bwaves.....	62
Πίνακας 8-4: Normalized IPC - gamess.....	63
Πίνακας 8-5: Normalized IPC - mcf	63
Πίνακας 8-6: Normalized IPC - milc.....	64
Πίνακας 8-7: Normalized IPC - cactusADM	64
Πίνακας 8-8: Normalized IPC - leslie3d	65
Πίνακας 8-9: Normalized IPC - namd.....	65
Πίνακας 8-10: Normalized IPC - gobmk.....	66
Πίνακας 8-11: Normalized IPC - povray.....	66
Πίνακας 8-12: Normalized IPC - calculix	67
Πίνακας 8-13: Normalized IPC - GemsFDTD	67
Πίνακας 8-14: Normalized IPC - libquantum.....	68
Πίνακας 8-15: Normalized IPC - tonto.....	68
Πίνακας 8-16: Normalized IPC - omnetpp.....	69
Πίνακας 8-17: Normalized IPC - lbm	69
Πίνακας 8-18: Normalized IPC - soplex	70
Πίνακας 8-19: Normalized IPC - gcc	70
Πίνακας 8-20: Normalized IPC - zeusmp.....	71
Πίνακας 8-21: Normalized IPC - perlbench	71
Πίνακας 8-22: Normalized IPC - gromacs	72
Πίνακας 8-23: Normalized IPC - dealII	72
Πίνακας 8-24: Normalized IPC - hmmer	73
Πίνακας 8-25: Normalized IPC - sjeng	73
Πίνακας 8-26: Normalized IPC - h264ref	74
Πίνακας 8-27: Normalized IPC - sphinx3	74
Πίνακας 8-28: Normalized IPC - xalancbmk	75
Πίνακας 8-29: Συνολικό Κέρδος – astar ως gold	75
Πίνακας 8-30: Συνολικό Κέρδος - bzip2 ως gold	76

Πίνακας 8-31: Συνολικό Κέρδος – bwaves ως gold.....	76
Πίνακας 8-32: Συνολικό Κέρδος - gamess ως gold.....	77
Πίνακας 8-33: Συνολικό Κέρδος - mcf ως gold.....	77
Πίνακας 8-34: Συνολικό Κέρδος - milc ως gold.....	78
Πίνακας 8-35: Συνολικό Κέρδος - cactusADM ως gold.....	78
Πίνακας 8-36: Συνολικό Κέρδος - leslie3d ως gold.....	79
Πίνακας 8-37: Συνολικό Κέρδος - namd ως gold.....	79
Πίνακας 8-38: Συνολικό Κέρδος - gobmk ως gold.....	80
Πίνακας 8-39: Συνολικό Κέρδος - ponray ως gold.....	80
Πίνακας 8-40: Συνολικό Κέρδος - calculix ως gold.....	81
Πίνακας 8-41: Συνολικό Κέρδος - GemsFDTD ως gold.....	81
Πίνακας 8-42: Συνολικό Κέρδος - libquantum ως gold.....	82
Πίνακας 8-43: Συνολικό Κέρδος - tonto ως gold.....	82
Πίνακας 8-44: Συνολικό Κέρδος - omnetpp ως gold.....	83
Πίνακας 8-45: Συνολικό Κέρδος – lbm ως gold.....	83
Πίνακας 8-46: Συνολικό Κέρδος – soplex ως gold.....	84
Πίνακας 8-47: Συνολικό Κέρδος – gcc ως gold.....	84
Πίνακας 8-48: Συνολικό Κέρδος – zeusmp ως gold.....	85
Πίνακας 8-49: Συνολικό Κέρδος – perlbench ως gold.....	85
Πίνακας 8-50: Συνολικό Κέρδος – gromacs ως gold.....	86
Πίνακας 8-51: Συνολικό Κέρδος – dealII ως gold.....	86
Πίνακας 8-52: Συνολικό Κέρδος – hmmer ως gold.....	87
Πίνακας 8-53: Συνολικό Κέρδος – sjeng ως gold.....	87
Πίνακας 8-54: Συνολικό Κέρδος - h264ref ως gold.....	88
Πίνακας 8-55: Συνολικό Κέρδος - sphinx3 ως gold.....	88
Πίνακας 8-56: Συνολικό Κέρδος – xalancbmk ως gold.....	89

Ευρετήριο Εικόνων

Εικόνα 1-1: Ιεραρχία κρυφή μνήμης στα πολυπύρρηνα επεξεργαστικά συστήματα.....	2
Εικόνα 3-1: Αντιστοίχιση των RMIDs.....	16
Εικόνα 3-2: Ο καταχωρητής IA32_PQR_ASSOC (PQR) MSR	16
Εικόνα 3-3: Ο καταχωρητής IA32_QM_EVTSEL MSR.....	17
Εικόνα 3-4: Ο καταχωρητής IA32_QM_CTR MSR.....	17
Εικόνα 3-5: Αντιστοίχιση σε COS	17
Εικόνα 3-6: Παραδείγματα ορισμού μασκών στα COS σε μια 8-way cache.....	18
Εικόνα 5-1: Διάγραμμα ροής του μηχανισμού.....	44

1

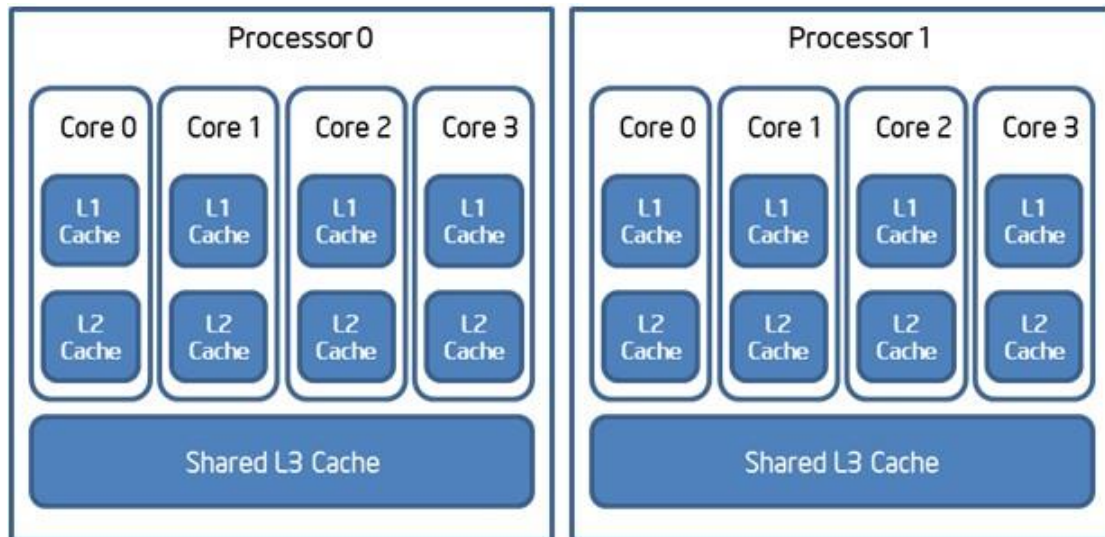
Εισαγωγή

1.1 Πολυπύρηνες Αρχιτεκτονικές

Για χρόνια οι επεξεργαστές αποτελούνταν από ένα και μόνο πυρήνα. Προκειμένου να αυξηθεί η επίδοση των μονοπύρηνων επεξεργαστών, τα chips μειώνονταν συνεχώς σε μέγεθος ενώ αυξανόταν ο αριθμός των transistors που περιλάμβαναν. Οι προσπάθειες των κατασκευαστών για συρρίκνωση των transistors, ώστε τα chips να περιλαμβάνουν όσο το δυνατό περισσότερα, οδήγησαν σε τρία πολύ σημαντικά προβλήματα τα οποία επιδρούσαν αρνητικά στην Ποιότητα Υπηρεσίας (Quality of Service, QoS) που παρείχαν οι επεξεργαστές. Τα προβλήματα αυτά ήταν η υπερβολική κατανάλωση ενέργειας και παραγωγή θερμότητας καθώς και η διαφορά της ταχύτητας μεταξύ τού επεξεργαστή και της μνήμης. Με σκοπό την αποφυγή τους οδηγηθήκαμε στην κατασκευή των πολυπύρηνων επεξεργαστικών συστημάτων [1].

Σήμερα λοιπόν, αναφερόμαστε σε πολυπύρηνες αρχιτεκτονικές οι οποίες συναντώνται παντού, από τους προσωπικούς υπολογιστές μέχρι και σε υπερυπολογιστές, data centers και υπηρεσίες νέφους (cloud services). Πλέον, σε μια πλακέτα (chip) περιλαμβάνονται περισσότεροι από ένας πυρήνες (Chip Multiprocessors, CMPs). Στην εικόνα 1-1 παρουσιάζεται η πιο συνηθισμένη, αν και όχι μοναδική, περίπτωση ιεραρχίας της κρυφής μνήμης που συναντάμε σε έναν επεξεργαστή. Κάθε πυρήνας

αποτελείται από 2 ιδιωτικές κρυφές μνήμες, επιπέδου 1 (Level 1 Cache, L1) και επιπέδου 2 (Level 2 Cache, L2), ενώ οι υπόλοιποι πόροι της πλακέτας όπως η κρυφή μνήμη τελευταίου επιπέδου (Last Level Cache, LLC), ο διάυλος της μνήμης, οι memory controllers και άλλα στοιχεία, είναι κοινοί και προσβάσιμοι από όλους τους πυρήνες.



Εικόνα 1-1: Ιεραρχία κρυφής μνήμης στα πολυπύρηννα επεξεργαστικά συστήματα

Σε αντίθεση με τα μονοπύρηννα επεξεργαστικά συστήματα, οι πολυπύρηννοι επεξεργαστές έχουν την ικανότητα να μοιράζουν το φόρτο εργασίας σε πολλές μονάδες και έτσι επιτυγχάνουν την παράλληλη εκτέλεση εφαρμογών. Όμως, παρά τις βελτιώσεις που έχουν επιφέρει οι πολυπύρηννες αρχιτεκτονικές, γίνονται αναφορές για ένα φαινόμενο που επηρεάζει σημαντικά την επίδοση των προγραμμάτων που εκτελούνται παράλληλα. Ο ανταγωνισμός ανάμεσα στις συνεκτελούμενες διεργασίες για τους διαμοιραζόμενους πόρους έχει αποδειχθεί πως μπορεί να μειώσει την επίδοσή τους και εν τέλη το QoS όλου του συστήματος [2].

1.2 Ανταγωνισμός Συνεκτελούμενων Εφαρμογών

Κανείς θα περίμενε η λειτουργία των πυρήνων σε ένα πολυπύρηννο επεξεργαστικό σύστημα να είναι εντελώς ανεξάρτητη. Πολλοί ερευνητές εξέφρασαν την άποψη και απέδειξαν πως ο ανταγωνισμός για τους διαμοιραζόμενους πόρους του συστήματος μπορεί να δράσει αρνητικά στην επίδοση των διεργασιών που τρέχουν παράλληλα ακόμη και σε διαφορετικούς πυρήνες, υποβαθμίζοντας έτσι το QoS ολόκληρου του

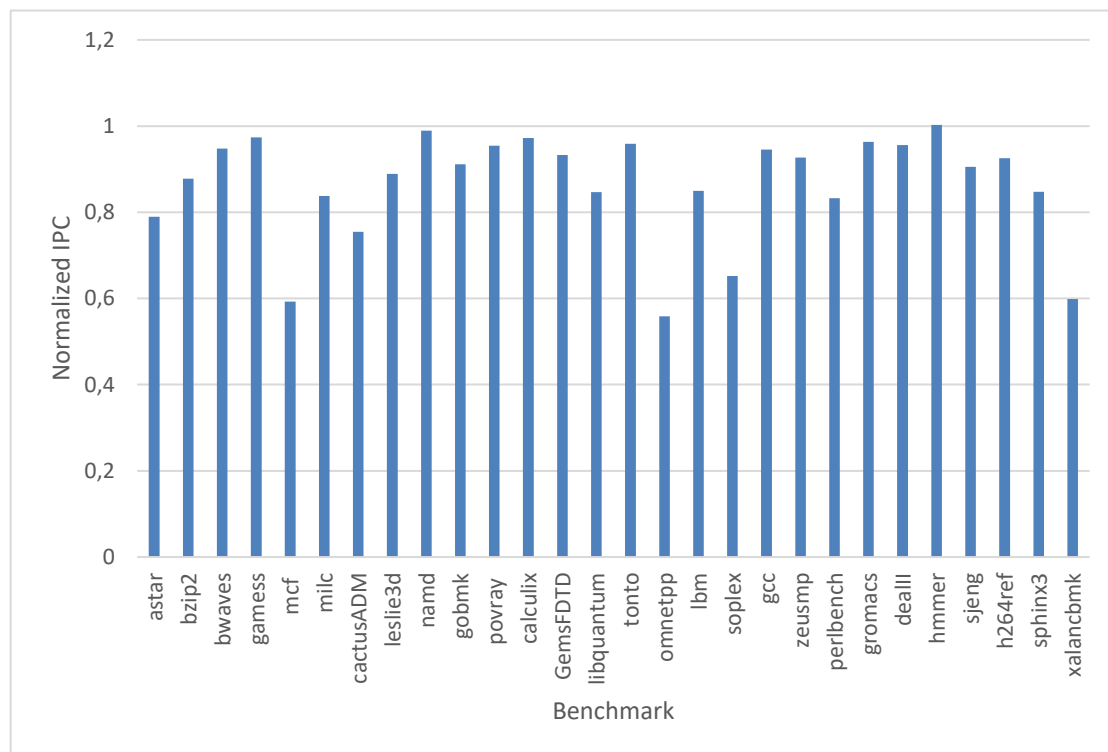
συστήματος [3], [4], [5]. Οι εφαρμογές μπορεί να οδηγηθούν ακόμη και σε λιμοκτονία (starvation) λόγω της άπληστης συμπεριφοράς των απαιτητικών διεργασιών, οι οποίες αξιοποιούν το μεγαλύτερο μέρος των διαμοιραζόμενων πόρων [6]. Έτσι μειώνεται η χρησιμοποίηση (utilization) του συστήματος και το λειτουργικό κόστος αυξάνεται.

Η επίδοση μιας διεργασίας μπορεί να περιγραφεί από τον αριθμό των εντολών που εκτελεί σε ένα κύκλο (Instructions per cycle, IPC). Στο πιο κάτω γράφημα απεικονίζεται το κανονικοποιημένο IPC (Normalized IPC) διάφορων εφαρμογών όταν η κάθε εφαρμογή εκτελείται ταυτόχρονα με άλλες εννέα τυχαία επιλεγμένες διεργασίες, σε διαφορετικό πυρήνα η κάθε μια. Το Normalized IPC υπολογίζεται ως :

$$\text{Normalized IPC} = \frac{\text{IPC}_{\text{co-execution}}}{\text{IPC}_{\text{isolation}}},$$

όπου:

- $\text{IPC}_{\text{co-execution}}$: Το IPC της εφαρμογής όταν εκτελείται παράλληλα με άλλες εφαρμογές
- $\text{IPC}_{\text{isolation}}$: Το IPC της εφαρμογής σε απομονωμένη εκτέλεση



Γράφημα 1-1: Παράδειγμα Normalized IPC

Σε όλες σχεδόν τις περιπτώσεις το κανονικοποιημένο IPC είναι μικρότερο της μονάδας. Άρα το $IPC_{co-execution}$ είναι μικρότερο από το $IPC_{isolation}$. Φαίνεται δηλαδή πως όταν εφαρμογές τρέχουν παράλληλα σε ένα επεξεργαστικό σύστημα, έστω και σε διαφορετικούς πυρήνες, τείνουν να εμφανίζουν μειωμένο IPC. Αρκετές διεργασίες επηρεάζονται σε μεγάλο βαθμό ενώ άλλες δεν φαίνεται να ενοχλούνται ιδιαίτερα από την ταυτόχρονη εκτέλεση.

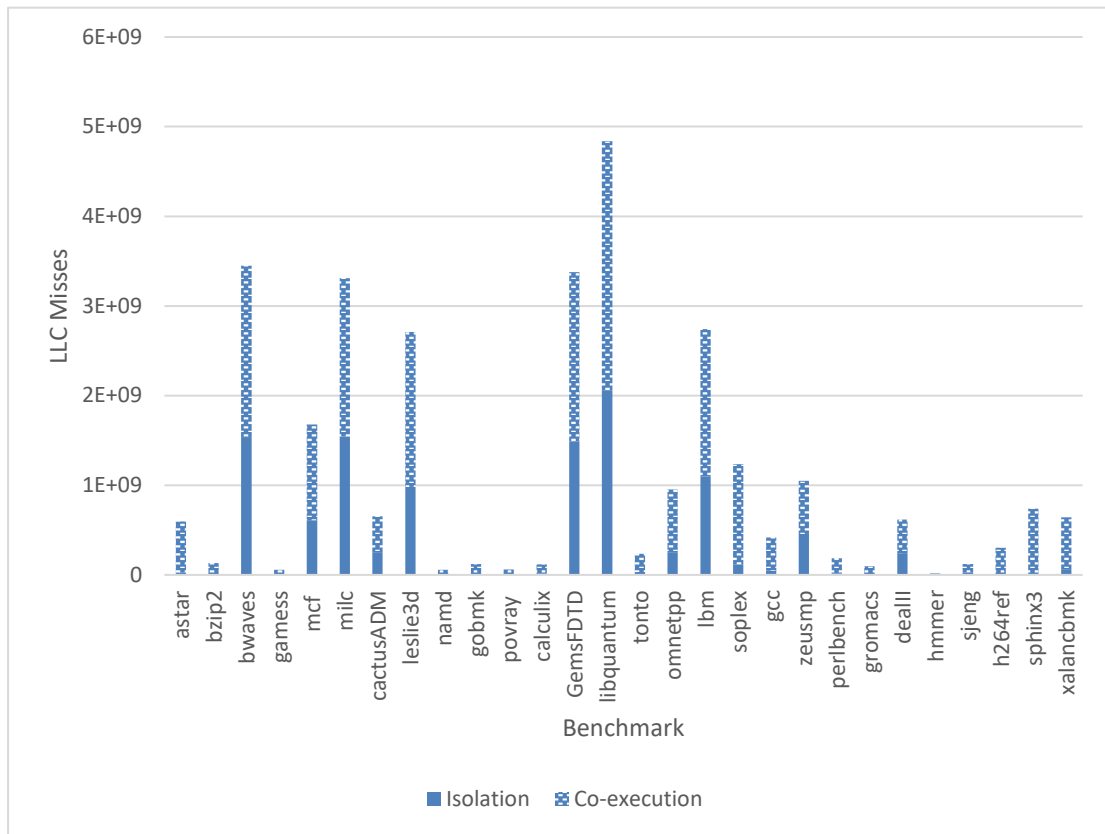
1.2.1 Κρυφή Μνήμη Τελευταίου Επιπέδου

Πολλές έρευνες έδειξαν πως η κρυφή μνήμη αποτελεί μια από τις σημαντικότερες αιτίες ανταγωνισμού μεταξύ των εφαρμογών που εκτελούνται παράλληλα στους πυρήνες, λόγω των αποτυχημένων προσπελάσεων τους στη cache, προτείνοντας παράλληλα διάφορους τρόπους λύσης του φαινομένου αυτού [6] - [8], [18]–[20] .

Μια διεργασία όταν τρέχει σε ένα πυρήνα, αποθηκεύει τα δεδομένα της σε μπλοκ της κρυφής μνήμης για να έχει γρήγορη πρόσβαση σε αυτά όταν θα τα χρειαστεί ξανά στο μέλλον. Η κρυφή μνήμη δεν επαρκεί για να φέρουν όλες οι εφαρμογές τα δεδομένα τους. Για το λόγο αυτό, όταν μια διεργασία δεν βρίσκει ελεύθερα μπλοκ για να αποθηκεύσει τα δικά της δεδομένα, εκτοπίζει αυτά άλλων διεργασιών, ανάλογα με την πολιτική αντικατάστασης που εφαρμόζεται. Έτσι, όταν μια διεργασία της οποίας τα δεδομένα έχουν εκτοπιστεί θα καταλήξει σε αποτυχημένη προσπέλαση (miss). Τότε αναγκάζεται να φέρει ξανά τα δεδομένα της και να ψάξει για ελεύθερα μπλοκ στη κρυφή μνήμη. Εάν στη κρυφή μνήμη δεν υπάρχουν άδεια μπλοκ, με τη σειρά της η εφαρμογή εκτοπίζει τα δεδομένα άλλων διεργασιών.

Είναι προφανές πως μια εφαρμογή σε ένα πολυπύρηνο επεξεργαστικό σύστημα, θα επαναλαμβάνει συχνά την παραπάνω διαδικασία. Κάθε αποτυχημένη προσπέλαση μνήμης καθυστερεί την εκτέλεση της εντολής που χρειάζεται τα εκάστοτε δεδομένα και εν τέλη αυξάνεται ο συνολικός χρόνος εκτέλεσης των διεργασιών αφού εκτελούνται λιγότερες εντολές ανά κύκλο, δικαιολογώντας έτσι και το μειωμένο IPC στη περίπτωση της συνεκτέλεσης.

Στο πιο κάτω γράφημα απεικονίζονται οι αστοχίες στη κρυφή μνήμη τελευταίου επιπέδου (LLC misses) που αντιμετωπίζουν οι εφαρμογές της περίπτωσης της ενότητας 1.2. Φαίνεται πως κατά την συνεκτέλεση των διεργασιών, στις πλείστες περιπτώσεις, τα LLC Misses κατά την συνεκτέλεση (Co-execution) είναι πολύ περισσότερα από αυτά κατά την απομονωμένη εκτέλεση (Isolation).



Γράφημα 1-2: Παράδειγμα LLC Misses

1.3 Αντιμετώπιση του Ανταγωνισμού

Λόγω των σοβαρών επιπτώσεων που προκαλεί ο ανταγωνισμός για τους κοινόχρηστους πόρους τόσο στις διεργασίες, όσο και σε ολόκληρο το σύστημα, η επιστημονική κοινότητα έχει προτείνει ένα μεγάλο αριθμό από διαφορετικούς τρόπους για την αντιμετώπιση του φαινομένου αυτού [11], [12], [16]-[19].

Σε γενικές γραμμές, οι μηχανισμοί που έχουν προταθεί ακολουθούν μια παρόμοια αλληλουχία από βήματα για κάθε καινούργια εφαρμογή που εισέρχεται στο σύστημα.

Τα βήματα που εκτελούνται είναι τα εξής:

1. Δημιουργία προφίλ εκτός σύνδεσης (offline profiling), στη περίπτωση που αφίχθηκε μια καινούρια εφαρμογή για την οποία δεν υπάρχουν ήδη πληροφορίες.
2. Χρονοδρομολόγηση και έναρξη λειτουργίας της εφαρμογής:

- a. Εάν έχει γίνει offline profiling η εφαρμογή χρονοδρομολογείται στον επεξεργαστή και έπειτα στον πυρήνα που αποφαίνεται περισσότερο ωφέλιμος τόσο για την ίδια όσο και για τις διεργασίες που θα συνεκτελούνται μαζί της, με πιθανή κατανομή των πόρων.
 - b. Εάν δεν έχει γίνει offline profiling η εφαρμογή τοποθετείται τυχαία σε κάποιο επεξεργαστή και πυρήνα.
3. Δημιουργία προφίλ εντός σύνδεσης (online profiling):
- a. Ανάλογα με τις πληροφορίες που συλλέγονται από το online profiling, γίνονται οι κατάλληλες κατανομές των πόρων ή και η εκ νέου χρονοδρομολόγηση της εφαρμογής, στη περίπτωση που θεωρείται αναγκαίο.
4. Συνεχής παρακολούθηση της επίδοσης εφαρμογής κατά την εκτέλεσή της (online monitoring):
- a. Εάν παρατηρηθεί μείωση στην επίδοση της εφαρμογής τότε γίνεται ξανά είτε κατανομή των πόρων, είτε χρονοδρομολόγηση της εφαρμογής σε άλλο επεξεργαστή ή πυρήνα.

Τα βήματα 1,3,4 είναι προαιρετικά και δεν συναντώνται σε όλους τους μηχανισμούς. Συνήθως εκτελείται ένα από τα τρία βήματα.

1.3.1 Profiling

Το profiling είναι η μέθοδος κατά την οποία αναλύουμε την συμπεριφορά μιας εφαρμογής όσο αυτή εκτελείται, είτε απομονωμένα, είτε μαζί με άλλες εφαρμογές ή micro-benchmarks. Μέσω του profiling περιγράφεται το πως εκμεταλλεύεται η κάθε διεργασία τους κοινόχρηστους πόρους (χρήση CPU, cache, συσκευές I/O κ.α.) του συστήματος τόσο κατά την απομονωμένη όσο και κατά την παράλληλη εκτέλεση. Επίσης γίνεται έλεγχος ως προς το πώς επηρεάζεται η κάθε διεργασία από τον ανταγωνισμό για τους διαμοιραζόμενους πόρους, όταν συνεκτελείται με άλλες εφαρμογές, αλλά και ως προς το πώς μπορεί η ίδια η εφαρμογή να επηρεάσει την επίδοση άλλων διεργασιών στα σενάρια παράλληλης εκτέλεσης [8]–[15]. Έχοντας αποθηκεύσει ένα προφίλ στο σύστημα για κάθε είδος εφαρμογής, γίνεται πιο εύκολη η πρόβλεψη της συμπεριφοράς κάθε νέας διεργασίας που ξεκινά την εκτέλεσή της. Έτσι ο μηχανισμός διαχείρισης πόρων μπορεί να αποφανθεί εξ αρχής ως προς την κατανομή των πόρων που πρέπει να εφαρμοσθεί και ως προς τον πιο ωφέλιμο

επεξεργαστή/πυρήνα, στον οποίο θα πρέπει να χρονοδρομολογηθεί η εκάστοτε διεργασία, τόσο για την ίδια όσο και για τις διεργασίες που θα συνεκτελούνται μαζί της.

Στα συστήματα μεγάλης κλίμακας το profiling φαίνεται να βελτιώνει σημαντικά το QoS που παρέχουν. Έτσι επωφελείται τόσο ο πάροχος όσο και ο πελάτης. Όσο πιο πολύ ικανοποιείται το SLO της κάθε εφαρμογής τόσο αυξάνεται και το χρηματικό κέρδος που θα επιφέρει στο σύστημα. Παρόλα αυτά, το profiling απαιτεί επιπλέον υλικό (για παράδειγμα έναν απομονωμένο server) για να εφαρμοσθεί. Αυτό έχει αρνητικό αντίκτυπο στην υπολογιστική δύναμη του συστήματος αφού εν τέλει μειώνονται οι διαθέσιμοι πόροι. Ακόμη, σε συστήματα μεγάλης κλίμακας, όπου τρέχουν παράλληλα πολλές διεργασίες, η εύρεση της εφαρμογής που προκαλεί τον μεγαλύτερο ανταγωνισμό, δηλαδή της εφαρμογής που επηρεάζει περισσότερο την επίδοση των υπολοίπων, αποτελεί μια πολύ δύσκολη και χρονοβόρα διαδικασία.

Η συγκεκριμένη μέθοδος μπορεί να εφαρμοσθεί online, δηλαδή σε πραγματικό χρόνο εκτέλεσης, αλλά και offline. Στη παρούσα εργασία εφαρμόστηκε το offline profiling διότι για την εκτέλεση του τελικού μηχανισμού χρειάστηκαν offline πληροφορίες σχετικά με την ευαισθησία της κάθε εφαρμογής ως προς την LLC.

1.3.2 Online Monitoring

Το online monitoring αποσκοπεί στην παρακολούθηση της επίδοσης των διεργασιών κατά την πραγματική εκτέλεση τους. Για να εφαρμοσθεί δεν απαιτούνται κατ' ανάγκη εκ των προτέρων πληροφορίες για τις εφαρμογές [20], [21]. Αυτό όμως δεν σημαίνει πως αποκλείεται η χρήση κάποιου μηχανισμού profiling πριν ή κατά την εφαρμογή του online monitoring [14].

Σε αντίθεση με το profiling, οι μηχανισμοί που χρησιμοποιούν το online monitoring δεν αποσκοπούν στο να προλάβουν και να προβλέψουν τον ανταγωνισμό πριν την χρονοδρομολόγηση των εφαρμογών. Σκοπός τους είναι να ανιχνεύουν τη μείωση στην επίδοση των διεργασιών κατά την πραγματική τους εκτέλεση. Στη περίπτωση που ο μηχανισμός διακρίνει σημαντική μείωση στην επίδοση μιας διεργασίας τότε αποφαινεται για τυχόν αλλαγές που πρέπει να γίνουν ως προς την ανακατανομή των πόρων ή και την εκ νέου χρονοδρομολόγηση των εφαρμογών.

1.4 Αντικείμενο Διπλωματικής

Όπως αναφέρθηκε προηγουμένως, οι αστοχίες που αντιμετωπίζει μια διεργασία στην LLC μπορούν να αποβούν καταστροφικές για την επίδοσή της και το QoS του συστήματος. Με μια πιο προσεκτική διαχείριση της cache, αυτό το φαινόμενο μπορεί να αποφευχθεί.

Στα σύστημα μεγάλης κλίμακας, πρωταρχικό μέλημα των παρόχων είναι η εξασφάλιση του SLO των εφαρμογών με την μεγαλύτερη προτεραιότητα. Όπως είναι λογικό, τέτοιες εφαρμογές επιφέρουν και το μεγαλύτερο χρηματικό κέρδος στα συστήματα. Έτσι, η διπλωματική έχει σκοπό την υλοποίηση ενός δυναμικού αλγορίθμου που θα επιτηρεί την επίδοση της εφαρμογής με προτεραιότητα και θα κατανέμει την cache με τέτοιο τρόπο, ώστε να εξασφαλίζει το SLO που έχει συμφωνηθεί για αυτές τις διεργασίες, αυξάνοντας εν τέλη το συνολικό κέρδος του συστήματος. Για τους σκοπούς της διπλωματικής χρησιμοποιήθηκαν δύο τεχνολογίες του εργαλείου PQoS/Intel® Intel Resource Director Technology (RDT), Cache Allocation Technology (CAT) και Cache Monitoring Technology (CMT), για τον καταμερισμό της cache και τη παρακολούθηση της επίδοσης των εφαρμογών αντίστοιχα [22]. Επίσης για την κατανομή των εικονικών μηχανών στους πυρήνες, έγινε χρήση του διαχειριστή πόρων και ανιχνευτή ανταγωνισμού για cloud υπηρεσίες - ACTiManager [23].

Για την υλοποίηση του δυναμικού μηχανισμού χρειάστηκαν, για την κάθε εφαρμογή, offline πληροφορίες σε απομονωμένη και παράλληλη εκτέλεση. Η συλλογή των πληροφοριών πραγματοποιήθηκε κατά τις πειραματικές διαδικασίες μελέτης των χαρακτηριστικών της κάθε εφαρμογής. Κατά την παράλληλη εκτέλεση των εφαρμογών εξετάστηκαν ποικίλα σενάρια ως προς τις διεργασίες που συνεκτελούνταν μαζί με την εφαρμογή με προτεραιότητα και ως προς τη κατανομή της LLC.

1.4.1 Συνεισφορά

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Μελέτη επίδοσης των εφαρμογών σε απομονωμένη εκτέλεση.
2. Μελέτη επίδοσης της εφαρμογής με προτεραιότητα και του συνολικού κέρδους στο σύστημα σε σενάρια παράλληλης εκτέλεσης, χωρίς καταμερισμό της LLC.

3. Μελέτη επίδοσης της εφαρμογής με προτεραιότητα και του συνολικού κέρδους στο σύστημα σε σενάρια παράλληλης εκτέλεσης σε διαφορετικές περιπτώσεις καταμερισμού της LLC.
4. Εξαγωγή χαρακτηριστικών για κάθε εφαρμογή.
5. Ανάπτυξη αλγορίθμου παρακολούθησης της επίδοσης της εφαρμογής με προτεραιότητα και καταμερισμού της LLC ανάλογα με τις ανάγκες της με σκοπό την τήρηση του SLO και την αύξηση του συνολικού κέρδους στο σύστημα
6. Μελέτη επίδοσης της εφαρμογής με προτεραιότητα και του συνολικού κέρδους στο σύστημα με τη χρήση του αλγορίθμου που αναπτύχθηκε.
7. Τελικά Συμπεράσματα και μελλοντικές επεκτάσεις.

1.5 Οργάνωση Κειμένου

Στο δεύτερο κεφάλαιο περιγράφονται έρευνες στις οποίες αναπτύχθηκαν μηχανισμοί καταμερισμού της cache, πάνω στις οποίες βασίστηκε η παρούσα διπλωματική εργασία.

Ο τρόπος λειτουργίας των τεχνολογιών Intel® CMT και CAT που χρησιμοποιήθηκαν στον τελικό μηχανισμό, αναπτύσσεται πλήρως στο δεύτερο κεφάλαιο.

Στη συνέχεια στο τέταρτο κεφάλαιο περιγράφονται λεπτομερώς τα πειράματα που διεξήχθησαν για την εξαγωγή των χαρακτηριστικών της κάθε εφαρμογής. Στο συγκεκριμένο κεφάλαιο παρουσιάζονται σε μορφή διαγραμμάτων και αναλύονται τα αποτελέσματα που εξαγάγαμε, ενώ δίνονται και σε πίνακα τα χαρακτηριστικά της κάθε διεργασίας.

Ο τελικός μηχανισμός που αναπτύχθηκε αναλύεται και εξετάζεται στο πέμπτο κεφάλαιο.

Τέλος, στο έκτο κεφάλαιο παρουσιάζονται τα τελικά συμπεράσματα στα οποία καταλήξαμε, με βάση τα αρχικά πειράματα και τον τελικό μηχανισμό, καθώς και ιδέες για μελλοντικές επεκτάσεις.

Στο παράρτημα δίνονται σε μορφή πίνακα οι πληροφορίες σχετικά με την επίδοση και το κέρδος στο σύστημα που επιφέρουν οι εφαρμογές κατά την πειραματική μελέτη των εφαρμογών (κεφάλαιο 4) και κατά την εφαρμογή του τελικού μηχανισμού (κεφάλαιο 5)

2

Μηχανισμοί Διαχείρισης Κρυφής Μνήμης

Τα προβλήματα που συζητήθηκαν στην ενότητα 1.2.1 σχετικά με τον ανταγωνισμό για την LLC που εμφανίζουν οι συνεκτελούμενες εφαρμογές, οδήγησαν την επιστημονική κοινότητα στο να αναπτύξει και να υλοποιήσει μηχανισμούς διαχείρισης της κρυφής μνήμης. Οι μηχανισμοί αυτοί ποικίλουν ως προς τον τρόπο υλοποίησης. Αρκετές υλοποιήσεις βασίζονται μόνο σε μηχανισμούς profiling ή online monitoring (ενότητα 1.3). Οι πιο πρόσφατοι μηχανισμοί όμως, εκμεταλλευόμενοι τα πλεονεκτήματα του profiling και του online monitoring καθώς και τις δυνατότητες του υλικού (hardware), ανέπτυξαν μηχανισμούς οι οποίοι διαχειρίζονται και κατανέμουν την κρυφή μνήμη ανάλογα με τις ανάγκες που παρουσιάζουν οι εφαρμογές για την LLC. Στο παρόν κεφάλαιο περιγράφονται διάφοροι μηχανισμοί που έχουν προταθεί με σκοπό να αντιμετωπιστεί το φαινόμενο του ανταγωνισμού για την cache βασιζόμενοι είτε μόνο στο profiling ή online monitoring είτε στο συνδυασμό ενός από τους δύο μηχανισμούς με τεχνικές κατανομής της cache.

2.1 Υλοποιήσεις Βασισμένες σε Profiling/Online Monitoring

Οι μηχανισμοί που κατατάσσονται σε αυτή τη κατηγορία, δεν αποσκοπούν στον καταμερισμό της cache. Αντιθέτως, χρησιμοποιώντας το profiling, επιδιώκουν στο να ομαδοποιήσουν της εφαρμογές με τέτοιο τρόπο έτσι ώστε η εκτέλεση της κάθε ομάδας να παρουσιάζει όσο το δυνατόν λιγότερο ανταγωνισμό. Οι μηχανισμοί που επιλέγουν το online monitoring στοχεύουν στην αντιμετώπιση του ανταγωνισμού κατά τον πραγματικό χρόνο εκτέλεσης. Έτσι, και στις δύο περιπτώσεις δεν γίνεται χρήση του υλικού αλλά του λογισμικού. Το λογισμικό αν και δεν έχει την ικανότητα να

διαχειρίζεται την LLC ή άλλους πόρους μπορεί να αξιοποιήσει τον χρονοδρομολογητή για την κατάλληλη χρονοδρομολόγηση των εφαρμογών.

Οι J. Mars, L. Tang, R. Hundt, K. Skadron, και M. Lou Soffa πρότειναν το Bubble-Up [19], μια μέθοδο κατά την οποία εκτελείται ένας μηχανισμός profiling για να συλλεχθούν πληροφορίες σχετικά με την ευαισθησία που παρουσιάζει η κάθε εφαρμογή, όσο αφορά την LLC. Το profiling εφαρμόζεται με σκοπό να διερευνηθεί πόσο επηρεάζεται η κάθε διεργασία από τη συνεκτέλεση αλλά και πώς επηρεάζει η ίδια τις συνεκτελούμενες εφαρμογές. Με αυτές τις πληροφορίες το σύστημα μπορεί να αποφανθεί ως προς τα αποδοτικότερα σενάρια ομαδοποίησης των εφαρμογών για συνεκτέλεση, κατά τα οποία θα εμφανίζεται λιγότερος ανταγωνισμός για την LLC, διατηρώντας έτσι το QoS. Η πειραματική μελέτη του Bubble-Up σε datacenters της Google, όπου έτρεχαν πραγματικές εφαρμογές μεγάλης κλίμακας, απέδειξε πως ο μηχανισμός αυτός μπορεί να αυξήσει την χρησιμοποίηση των επεξεργαστών από 50% μέχρι και 90%.

Το Pythia [16] αποτελεί ένα χρονοδρομολογητή εφαρμογών σε datacenters. Ο μηχανισμός βασίζεται σε πληροφορίες που συλλέγονται μέσω του profiling και ενός μαθηματικού μοντέλου πρόβλεψης ανταγωνισμού μεταξύ των εφαρμογών. Αρχικά για κάθε νέα παρτίδα εφαρμογών (batch workloads) που καταφθάνει στο σύστημα εκτελείται ένα είδος profiling για να καταγραφεί η ανταγωνιστικότητα που παρουσιάζουν απέναντι σε μια εφαρμογή με υψηλές απαιτήσεις για ελαχιστοποίηση της καθυστέρησης (Latency Critical, LC). Έπειτα, για τους ελέγχους που θα ακολουθήσουν, εξαιρεί τις batch εφαρμογές οι οποίες είναι τόσο ανταγωνιστικές με αποτέλεσμα να μην καθίσταται δυνατό να συνεκτελεστούν ούτε και με άλλες batch εφαρμογές. Από τις εφαρμογές που έχουν μείνει γίνονται μερικοί συνδυασμοί παράλληλων εκτελέσεων πολλαπλών batch εφαρμογών με μια LC εφαρμογή. Παράλληλα συλλέγονται πληροφορίες σχετικά με το IPC της LC διεργασίας για να ελεγχθούν τα επίπεδα του ανταγωνισμού στα σενάρια αυτά. Τέλος κάνοντας χρήση των πληροφοριών που συλλέχθηκαν από τους μηχανισμούς profiling και του μαθηματικού μοντέλου πρόβλεψης του ανταγωνισμού που έχουν αναπτύξει οι R. Xu κ.ά., επιλέγονται οι καταλληλότεροι εξυπηρετητές στους οποίους μπορεί να χρονοδρομολογηθεί η εφαρμογή.

Για την αντιμετώπιση του φαινομένου του ανταγωνισμού για την LLC, μια ακόμη λύση προτάθηκε αυτή τη φορά από τους A. Fedorova, M. Seltzer και M. D. Smith. Πιο

συγκεκριμένα ανατίθεται ένα προκαθορισμένο χρονικό διάστημα για το οποίο μπορεί να εκτελείται μια εφαρμογή σε ένα πυρήνα χωρίς αυτή να εκτοπίζεται από νέες διεργασίες. Όσο οι εφαρμογές εκτελούνται υπολογίζεται το IPC που εμφανίζουν και ανάλογα με το εάν ξεπερνά ένα όριο ή είναι πολύ πιο χαμηλό από ένα άλλο επιθυμητό όριο, το χρονικό διάστημα εκτέλεσης της εκάστοτε εφαρμογής μειώνεται ή αυξάνεται αντιστοίχως [24]. Η αύξηση του χρονικού διαστήματος εκτέλεσης γίνεται με στόχο να μην εκτοπίζεται μια εφαρμογή με μειωμένη επίδοση προκειμένου να αντιμετωπίσει όσο το δυνατόν λιγότερα LLC misses.

2.2 Υλοποιήσεις Συνδυασμού Profiling/Online Monitoring και Κατανομής της Cache

Οι υλοποιήσεις των δύο προηγούμενων ενοτήτων αν και βελτιώνουν σημαντικά το QoS του συστήματος, μπορεί να καταλήξουν σε μειωμένη χρησιμοποίηση τη cache. Έτσι, αναπτύχθηκαν μηχανισμοί οι οποίοι εκμεταλλεύονται την δυνατότητα της cache να «σπάει» σε τμήματα. Οι υλοποιήσεις των μηχανισμών συνδυάζουν τις τεχνικές του profiling ή του online monitoring με μηχανισμούς του υλικού οι οποίοι έχουν την ικανότητα να κατανέμουν την cache μεταξύ των συνεκτελούμενων εφαρμογών.

Ένα πρώτο παράδειγμα αποτελεί το PARTIES, ένας ελεγκτής και κατανεμητής πόρων για υπηρεσίες νέφους ο οποίος επιτρέπει σε πολλαπλές LC υπηρεσίες να ικανοποιήσουν τους περιορισμούς για το QoS τους. Κατά την λειτουργία του γίνεται χρήση μηχανισμών απομόνωσης πόρων σε επίπεδο υλικού και λογισμικού ενώ για τις διεργασίες που καταφτάνουν στο σύστημα δεν απαιτείται καμία εκ των προτέρων γνώση. Το πρώτο στοιχείο που απαρτίζει το PARTIES είναι ένας μηχανισμός παρακολούθησης της χωρητικότητας της μνήμης, της χρήσης του εύρους ζώνης του δικτύου και της καθυστέρηση της ουράς για κάθε εφαρμογή που βρίσκεται στο σύστημα. Το δεύτερο στοιχείο είναι ένας κατανεμητής πόρων ο οποίος, ανάλογα με τις μετρήσεις που έχει λάβει ο προαναφερθείς μηχανισμός παρακολούθησης, κατανέμει αναλόγως τους πόρους του συστήματος (πυρήνες, χώρος κρυφής μνήμης, συχνότητα, χωρητικότητα μνήμης, εύρος ζώνης δίσκου) με την χρήση των μηχανισμών απομόνωσης. Όταν παρατηρηθεί μια μικρή μείωση στην επίδοση των εφαρμογών ανατίθεται κάθε φορά, τυχαία, ένας πόρος στην LC εφαρμογή με την μικρότερη χαλάρωση στην επίδοσή της. Κάθε ένα προκαθορισμένο κβάντο χρόνου οι αλλαγές που

έγιναν, σχετικά με την κατανομή των πόρων, εξετάζονται από τον μηχανισμό παρακολούθησης και ανάλογα με τις μετρήσεις που λαμβάνονται αποφασίζεται εάν πρέπει να αποδευμευτεί κάποιος πόρος από μια LC εφαρμογή, λόγω πολύ αυξημένης επίδοσης, ή αν πρέπει να κατανεμηθεί εκ νέου κάποιος άλλος πόρος σε μια εφαρμογή, λόγω μειωμένης επίδοσης. Στην ακραία περίπτωση που μια LC διεργασία δεν καταφέρει μετά από αρκετό χρόνο να τηρήσει το SLO της, επιλέγεται η εφαρμογή που θα προκαλέσει το λιγότερο κόστος κατά την μετακίνησή της και χρονοδρομολογείται σε άλλο μηχανήμα.

Με την χρήση online αλλά και offline profiling στο KPart [12] ορίζεται η ευαισθησία των εφαρμογών ως προς την cache που τους παραχωρείται μετρώντας τα LLC misses, το IPC και το bandwidth της μνήμης. Έτσι, ανάλογα με την συμβατότητα που παρουσιάζουν οι εφαρμογές ως προς την κοινή χρήση της cache χωρίζονται σε ομάδες. Με σκοπό να κατανεμηθεί η cache μεταξύ των ομάδων που έχουν δημιουργηθεί, εκτελείται ακόμη ένα profiling. Πιο συγκεκριμένα εξετάζεται και πάλι η ευαισθησία που παρουσιάζει συνολικά η κάθε ομάδα ως προς την κατανομή της cache. Έτσι ο μηχανισμός αποφαινεται ως προς το μέγεθος της cache που θα παραχωρηθεί σε κάθε ομάδα. Η αξιολόγηση του KPart σε ένα πραγματικό σύστημα έδειξε ότι μπορεί να βελτιώσει την απόδοση κατά 24% κατά μέσο όρο.

Το Ubik προτάθηκε από τους H. Kasture και D. Sanchez οι οποίοι υλοποίησαν ένα δυναμικό μηχανισμό καταμερισμού της cache ο οποίος προβλέπει την συμπεριφορά των LC εφαρμογών μεγιστοποιώντας ταυτόχρονα τον διαθέσιμο χώρο της cache για τις υπόλοιπες εφαρμογές. Οι H. Kasture και D. Sanchez υλοποίησαν το Ubik με δύο τρόπους. Ο πρώτος ονομάζεται Strict Ubik και παρέχει αυστηρές εγγυήσεις για την επίδοση των LC εφαρμογών κάνοντας συντηρητικές παραδοχές ενώ ο δεύτερος ονομάζεται Ubik with slack στον οποίο οι παραδοχές γίνονται με μεγαλύτερη χαλαρότητα. Κατά την πρώτη υλοποίηση (Strict Ubik) στην LC εφαρμογή δίνεται τόση cache όση έχει φανεί από το profiling πως χρειάζεται για να λειτουργεί με ικανοποιητική επίδοση. Όταν η LC εφαρμογή γίνει αδρανής, προκειμένου να μεγιστοποιηθεί η χρησιμοποίηση της cache, της αφαιρείται ένα μέρος της LLC και δίνεται στις υπόλοιπες εφαρμογές που συνεκτελούνται μαζί της. Όταν η LC εφαρμογή γίνει ξανά ενεργή τότε της παραχωρείται ένα μεγαλύτερο μέρος της cache από ότι είχε αρχικά, το οποίο αφαιρείται από τις υπόλοιπες εφαρμογές, με σκοπό να ανακάμψει η LC διεργασία. Με το που ανακάμπτει η LC εφαρμογή, της αφαιρείται ξανά ένα μέρος

της LLC, το οποίο παραχωρείται και πάλι στις υπόλοιπες εφαρμογές, έτσι ώστε η LC εφαρμογή να κατέχει το μέρος της cache που κατείχε αρχικά. Η παραχώρηση ενός μεγαλύτερου μέρους της cache από ότι έχουν ανάγκη οι LC εφαρμογές κατά την μετάβασή τους από την αδρανή στην ενεργή κατάσταση, έδειξε πως σε μερικές περιπτώσεις δεν επιφέρει κάποια σημαντική αύξηση στην επίδοση των εφαρμογών. Έτσι στη δεύτερη υλοποίηση (Ubik with slack) υπολογίζεται ένα χαλαρότερο όριο, επιτρέποντας μεγαλύτερες υποβαθμίσεις στην επίδοση της LC εφαρμογής. Με τον τρόπο αυτό μειώνεται και η συχνότητα με την οποία αλλάζουν οι κατανομές της cache μεταξύ της LC διεργασίας και των υπολοίπων εφαρμογών. Ταυτόχρονα βελτιώνεται και η επίδοση των εφαρμογών που συνεκτελούνται με την LC διεργασία, αφού τους κατανέμεται ένα μεγαλύτερο μέρος της cache.

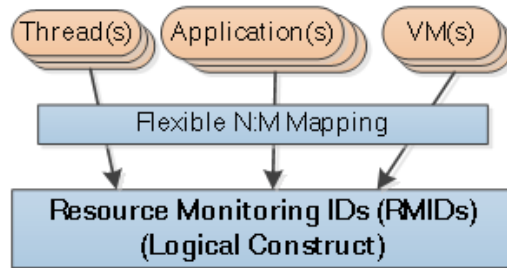
3

Τεχνολογίες Intel® CMT-CAT

Η Intel ανέπτυξε την τεχνολογία Intel® RDT [22] η οποία αποτελεί ένα μηχανισμό διαχείρισης της cache σε επίπεδο υλικού. Υποστηρίζεται από τους επεξεργαστές που ανήκουν στην οικογένεια Intel® Xeon® και χρησιμοποιείται για σκοπούς επιτήρησης και κατανομής των διαμοιραζόμενων πόρων του συστήματος. Αποτελείται από τις τεχνολογίες CMT και CAT. Το CMT επιτρέπει την παρακολούθηση της χρήσης της LLC από τις εφαρμογές που εκτελούνται στο σύστημα και το CAT καθιστά δυνατή την κατανομή της cache ανάμεσα στις εφαρμογές. Στους επεξεργαστές της οικογένειας Intel® Xeon® E5 v4 το RDT υποστηρίζει και τις τεχνολογίες Code and Data Prioritization (CDP) και Memory Bandwidth Monitoring (MBM). Η πρώτη επιτρέπει την ξεχωριστή τοποθέτηση του κώδικα και των δεδομένων στην LLC ενώ δευτέρα επιβλέπει τη χρήση του εύρους ζώνης της μνήμης.

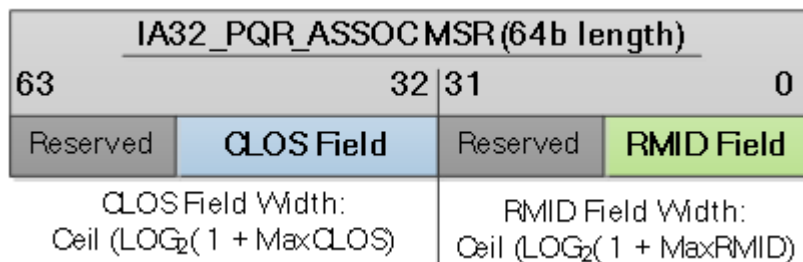
3.1 CMT

Όπως έχουμε αναφέρει η τεχνολογία CMT χρησιμοποιείται για την παρακολούθηση των διαμοιραζόμενων πόρων. Για να επιτευχθεί η συγκεκριμένη λειτουργία προστέθηκε ένα επίπεδο αφαίρεσης μεταξύ των εφαρμογών που εκτελούνται στο σύστημα και των πυρήνων, γνωστό ως Resource Monitoring IDs (RMIDs). Ένα RMID μπορεί να αντιστοιχεί σε ένα μόνο πυρήνα ή και σε πολλούς για σκοπούς ομαδοποίησης τους (π.χ. σε περιπτώσεις πολυνηματικών εφαρμογών.) Τα RMIDs όμως δεν συσχετίζονται μόνο με πυρήνες. Υπάρχει επίσης η δυνατότητα αντιστοίχισής τους σε νήματα (threads), εφαρμογές (applications) ή εικονικές μηχανές (virtual machines, VMs), όπως φαίνεται και στην εικόνα 3-1. Κάθε πυρήνας αποτελείται από ένα αριθμό RMIDs και κάθε στιγμή μόνο ένα RMID μπορεί να είναι ενεργό.



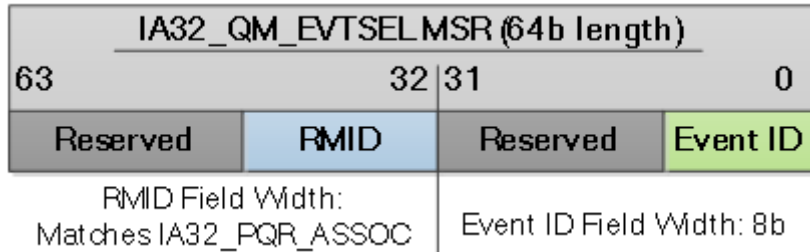
Εικόνα 3-1: Αντιστοίχιση των RMIDs

Ο καταχωρητής IA32_PQR_ASSOC (PQR) MSR (Model Specific Register) που αναπαρίσταται στην εικόνα 3-2, συναντάται σε κάθε πυρήνα και περιέχει πάντα το RMID που είναι ενεργό κάθε δεδομένη στιγμή. Όταν μια νέα εφαρμογή καταφθάνει για εκτέλεση σε ένα πυρήνα, τότε ο PQR ενημερώνεται και το πεδίο του RMID (RMID Field) αποκτά τιμή ίση με το RMID της εκάστοτε εφαρμογής. Με το τρόπο αυτό συσχετίζεται μια εφαρμογή/νήμα/VM με ένα RMID και οι μηχανισμοί παρακολούθησης είναι σε θέση να γνωρίζουν ποια εφαρμογή επιτηρείται.



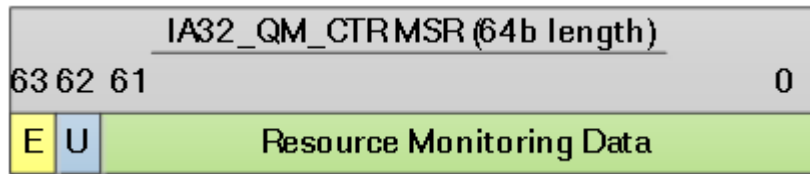
Εικόνα 3-2: Ο καταχωρητής IA32_PQR_ASSOC (PQR) MSR

Οι μηχανισμοί επιτήρησης πρέπει όμως να γνωρίζουν και για ποια δεδομένα επιθυμεί ο χρήστης να λαμβάνει πληροφορίες. Έτσι, ο καταχωρητής IA32_QM_EVTSEL MSR που φαίνεται και στην εικόνα 3-3 συνδυάζει το RMID με ένα EventID. Το EventID αντικατοπτρίζει το είδος των δεδομένων για τα οποία θα λαμβάνονται πληροφορίες ενώ όπως και προηγούμενος το RMID συμβολίζει την εκάστοτε εφαρμογή. Ως αποτέλεσμα, ο μηχανισμός ενημερώνεται και ως προς τα δεδομένα που πρέπει να επιτηρεί για την κάθε εφαρμογή.



Εικόνα 3-3: Ο καταχωρητής IA32_QM_EVTSEL MSR

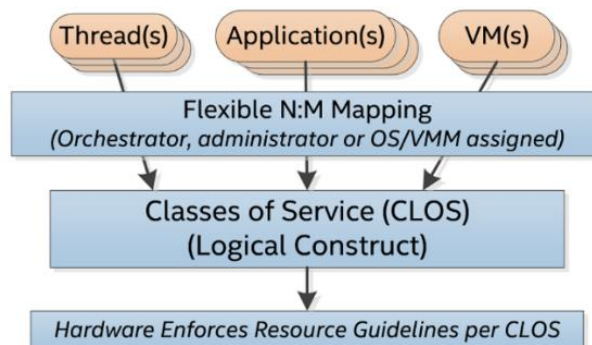
Τέλος, τα δεδομένα καταχωρούνται σε ένα άλλο καταχωρητή, τον IA32_QM_CTR MSR, ο οποίος περιέχει και δύο bits για τον έλεγχο της εγκυρότητας των δεδομένων.



Εικόνα 3-4: Ο καταχωρητής IA32_QM_CTR MSR

3.2 CAT

Όπως και στο CMT, έτσι και για τη λειτουργία του CAT έχει προστεθεί ένα επίπεδο αφαίρεσης, γνωστό ως Class of Service (COS ή CLOS). Τα COS λειτουργούν και αυτά ως ετικέτες για την ομαδοποίηση των εφαρμογών, νημάτων ή των VMs, ορίζοντας τα cache ways στα οποία μπορούν να έχουν πρόσβαση τα «μέλη» της κάθε ομάδας. Το COS αποθηκεύεται και αυτό στον καταχωρητή IA32_PQR_ASSOC (PQR) MSR (Εικόνα 3-2) και είναι εντελώς ανεξάρτητο από το RMID.



Εικόνα 3-5: Αντιστοίχιση σε COS

Πιο συγκεκριμένα όσο αφορά την LLC (L3 cache), για κάθε COS μπορούμε να καθορίσουμε μια μάσκα. Οι μάσκες ορίζονται σε ένα μπλοκ από MSRs, το IA32_L3_QOS_MASK_n, όπου το n συμβολίζει τον αριθμό του εκάστοτε COS. Σκοπός της μάσκας είναι να καθορίσει την ποσότητα της LLC που είναι διαθέσιμη για κάθε COS. Όπως φαίνεται και στην εικόνα 3-6, κάθε μάσκα αποτελείται από χαρακτήρες «0» (κενό κουτί) και «1» (κουτί με χαρακτήρα «A») και το άθροισμα των χαρακτήρων είναι ίσο με τα cache ways στα οποία χωρίζεται η LLC. Το «0» αναπαριστά την απαγόρευση της χρήσης του εκάστοτε way από το COS ενώ το 1 συμβολίζει την άδεια χρήσης του. Είναι σημαντικό να τονιστεί πως επιτρέπονται μόνο συνεχόμενοι συνδυασμοί του «1». Δηλαδή ένα COS μπορεί να έχει πρόσβαση μόνο σε συνεχόμενα ways της cache. Επίσης, σε ένα σύστημα αρχικά όλοι οι πυρήνες είναι ορισμένοι στο COS0, με σκοπό όλοι οι πυρήνες να έχουν κοινή πρόσβαση σε όλη τη cache. Η μάσκα του COS0 αναπαρίσταται στη πρώτη περίπτωση (Default Bitmask) της εικόνας 3-6. Η δεύτερη περίπτωση (Overlapped Bitmask) αναπαριστά σενάριο κατά το οποίο διαφορετικά COS μπορούν να έχουν πρόσβαση σε κοινά cache ways ενώ η τρίτη περίπτωση (Isolated Bitmask) παρουσιάζει ένα σενάριο κατά το οποίο σε κάθε COS δίνεται απομονωμένη πρόσβαση ενός σημείου της cache. Έτσι, εν τέλει, ορίζοντας μια μάσκα για κάθε COS και ένα COS στον κάθε πυρήνα μπορούμε να ορίζουμε το μέρος της LLC στο οποίο θα έχει πρόσβαση κάθε εφαρμογή.

	M7	M6	M5	M4	M3	M2	M1	M0	
COS0	A	A	A	A	A	A	A	A	Default Bitmask
COS1	A	A	A	A	A	A	A	A	
COS2	A	A	A	A	A	A	A	A	
COS3	A	A	A	A	A	A	A	A	

	M7	M6	M5	M4	M3	M2	M1	M0	
COS0	A	A	A	A	A	A	A	A	Overlapped Bitmask
COS1					A	A	A	A	
COS2							A	A	
COS3								A	

	M7	M6	M5	M4	M3	M2	M1	M0	
COS0	A	A	A	A					Isolated Bitmask
COS1					A	A			
COS2							A		
COS3								A	

Εικόνα 3-6: Παραδείγματα ορισμού μαस्कόν στα COS σε μια 8-way cache

4

Μελέτη Εφαρμογών

Στο παρόν κεφάλαιο εξετάζεται η συμπεριφορά των benchmarks ως προς την απομονωμένη και παράλληλη εκτέλεση. Επίσης, στα σενάρια παράλληλης εκτέλεσης γίνεται έλεγχος ως προς την επιρροή που έχει η στατική κατανομή της LLC στην επίδοση των εφαρμογών και στο κέρδος του συστήματος.

4.1 Πληροφορίες Συστήματος

4.1.1 Επεξεργαστής

Τόσο η μελέτη των εφαρμογών όσο και ο τελικός μηχανισμός, που θα αναλυθεί στο επόμενο κεφάλαιο, εκτελέστηκαν σε επεξεργαστή Intel® Xeon® E5-2630 v4 του οποίου τα χαρακτηριστικά δίνονται αναλυτικά στον πιο κάτω πίνακα (Πίνακας 4-1: Χαρακτηριστικά του επεξεργαστή Intel® Xeon® E5-2630 v4).

Ο επεξεργαστής διαθέτει την τεχνολογία Intel® RDT η οποία δίνει την δυνατότητα διαχείρισης των διαμοιραζόμενων πόρων και παρακολούθησης των μετρικών απόδοσης. Η συγκεκριμένη τεχνολογία αναλύθηκε στο κεφάλαιο 3.

Όνομα μοντέλου	Intel® Xeon® E5-2630 v4
Βασική συχνότητα επεξεργαστή	2.20 GHz
CPUs	20
Αριθμός νημάτων ανά πυρήνα	1
Πυρήνες ανά socket	10
Αριθμός sockets	2
NUMA nodes	2
L1d cache	32KB
L1i cache	32KB
L2 cache	256KB
L3 cache	25600KB, 20-ways

Πίνακας 4-1: Χαρακτηριστικά του επεξεργαστή Intel® Xeon® E5-2630 v4

4.1.2 ACTiManager

Προκειμένου να προσομοιωθεί μια υπηρεσία νέφους, χρησιμοποιήθηκε ο ACTiManager, ένας διαχειριστής πόρων σε υπηρεσίες νέφους με την ικανότητα ανίχνευσης του ανταγωνισμού για τους διαμοιραζόμενους πόρους (interference-aware). Κάθε benchmark εκτελείται σε ένα VM και στις πειραματικές διαδικασίες κάθε VM αποτελείται από μια εικονική κεντρική μονάδα επεξεργασίας (virtual central processing unit, vcpu). Στον ACTiManager τα VMs κατατάσσονται σε δύο ομάδες, στα gold και στα silver. Τα gold VMs αποτελούν εφαρμογές με προτεραιότητα και για το λόγο αυτό υπακούν σε πιο αυστηρά επίπεδα απόδοσης με αποτέλεσμα να επιφέρουν και μεγαλύτερο χρηματικό κέρδος σε σχέση με τα silver VMs. Πιο συγκεκριμένα στη δική μας περίπτωση τέθηκε ως SLO, για τα gold VMs, Slowdown μικρότερο και ίσο του 1.20 ενώ για τα silver VMs, Slowdown μικρότερο και ίσο του 2.4. Το Slowdown υπολογίζεται ως εξής:

$$Slowdown = \frac{1}{Normalized\ IPC} = \frac{IPC_{isolation}}{IPC_{co-execution}}$$

Για να επιφέρει κέρδος ένα VM στο σύστημα, είτε αυτό είναι gold είτε silver, πρέπει να υπακούει στο SLO που έχει συμφωνηθεί μεταξύ του παρόχου και του πελάτη. Στη

παρούσα εργασία το μοντέλο χρέωσης για τους δύο τύπους των VM που υποστηρίζει το σύστημα υπολογίζεται ως :

$$price_{gold} = \begin{cases} 0 & \text{για } Slowdown > 1.20 \\ 10.0 \times \frac{runtime}{Slowdown^2 \times executions} & \text{για } Slowdown \leq 1.20 \end{cases}$$

$$price_{silver} = \begin{cases} 0 & \text{για } Slowdown > 2.40 \\ 1.0 \times \frac{runtime}{Slowdown^2 \times executions} & \text{για } Slowdown \leq 2.40 \end{cases}$$

όπου:

- runtime: ο συνολικός χρόνος ζωής του VM
- executions: ο αριθμός των εκτελέσεων του ίδιου benchmark στο VM

Από τον ACTiManager χρησιμοποιήθηκε ο ACTiManager.internal ο οποίος εκτελείται μέσα στους servers και αναθέτει τα vcpus σε πυρήνες. Η πολιτική που επιλέχθηκε να ακολουθηθεί είναι η GNO (Gold Not Over Subscribed), η οποία δεν είναι interference-aware. Επίσης, έγινε μια αλλαγή στην πολιτική GNO ώστε να αναθέτει ένα πυρήνα σε κάθε vcpu είτε αυτή προέρχεται από gold VM είτε από silver VM [22].

4.2 Πειραματική Διαδικασία

Συνολικά εξετάστηκαν 28 benchmarks από τη σουίτα SPEC CPU 2006 [24] και για κάθε benchmark εκτελέστηκαν 51 πειράματα. Πάντα στο σύστημα υπήρχε μόνο ένα gold VM στο οποίο έτρεχε το benchmark που εξεταζόταν. Πιο συγκεκριμένα για κάθε benchmark ακολουθήθηκε η εξής πειραματική διαδικασία:

1. Απομονωμένη εκτέλεση του benchmark.
2. Παράλληλη εκτέλεση του gold benchmark με άλλα εννέα silver benchmarks χωρίς κατανομή της LLC για μια ώρα.
3. Παράλληλη εκτέλεση του gold benchmark με άλλα εννέα silver benchmarks σε τέσσερα διαφορετικά σενάρια κατανομής της LLC για μια ώρα.

Στο βήμα 1 το κάθε benchmark εκτελείτο μια φορά και άρα η διάρκεια του βήματος εξαρτιόταν από τον χρόνο εκτέλεσης του εκάστοτε benchmark. Αντιθέτως, στα βήματα 2 και 3 ο χρόνος εκτέλεσης ήταν σταθερός στη μια ώρα ενώ τα benchmarks μπορούσαν να εκτελεστούν περισσότερες από μια φορά σε ένα VM. Εάν κάποιο VM, είτε gold είτε

silver, τελειώνει την εκτέλεσή του πριν την μια ώρα τότε το εκάστοτε VM διαγραφόταν και στο σύστημα δημιουργούταν νέο VM στο οποίο έτρεχε το benchmark που μόλις είχε τελειώσει. Τα βήματα 2 και 3 επαναλήφθηκαν από δέκα φορές το κάθε ένα και σε κάθε επανάληψη εκτελείτο, παράλληλα με το gold VM, διαφορετική ομάδα από εννέα silver benchmarks. Δηλαδή το βήμα 2 εκτελέστηκε δέκα φορές και το βήμα 3 εκτελέστηκε συνολικά σαράντα φορές (4 σενάρια κατανομής της LLC). Εν τέλει όλα τα benchmarks μελετήθηκαν, ως gold, σε παράλληλη εκτέλεση και με τις δέκα εννιάδες. Σε κάθε σενάριο συνεκτέλεσης και κατανομής της LLC χρειάστηκαν οι εξής μετρήσεις για κάθε benchmark που εκτελείτο σε ένα VM:

- $IPC_{isolation}$
- $IPC_{co-execution}$
- Συνολικός Χρόνος Εκτέλεσης
- Αριθμός Εκτελέσεων του benchmark

4.2.1 Επίδοση Εφαρμογών σε απομονωμένη εκτέλεση

Μια εφαρμογή κατά την απομονωμένη εκτέλεση έχει στη διάθεσή της όλους τους πόρους του συστήματος και είναι αδύνατον να αντιμετωπίσει κάποια μείωση στην επίδοσή της ή καθυστέρηση στο χρόνο εκτέλεσής της λόγω ανταγωνισμού. Άρα οι μετρήσεις που προκύπτουν για το IPC και τον χρόνο εκτέλεσης του κάθε benchmark μπορούν να θεωρηθούν ως οι βέλτιστες. Στον πιο κάτω πίνακα (Πίνακας 4-2: Χαρακτηριστικά εφαρμογών σε απομονωμένη εκτέλεση) δίνονται ο χρόνος που χρειάστηκε το κάθε benchmark για να ολοκληρώσει μια εκτέλεση καθώς και το IPC του.

Benchmark	Χρόνος Εκτέλεσης (sec)	IPC _{isolation}
astar	172	1.01
bzip2	125	1.49
bwaves	419	2.38
gamess	62	2.73
mcf	277	0.51
milc	470	1.17
cactusADM	669	1.87
leslie3d	330	2.06
namd	458	2.23
gobmk	82	1.24
povray	184	2.36
calculix	894	3.07
GemsFDTD	384	1.72
libquantum	356	2.05
tonto	625	2.24
omnetpp	301	0.85
lbm	411	1.40
soplex	136	1.12
gcc	23	1.44
zeusmp	422	1.88
perlbench	195	2.46
gromacs	392	2.41
dealII	372	2.33
hmmmer	153	2.60
sjeng	628	1.60
h264ref	79	2.86
sphinx3	629	2.26
xalancbmk	236	1.86

Πίνακας 4-2: Χαρακτηριστικά εφαρμογών σε απομονωμένη εκτέλεση

4.2.2 Επίδοση εφαρμογών σε σενάρια παράλληλης εκτέλεσης χωρίς κατανομή της LLC

Στο δεύτερο στάδιο των μετρήσεων εξετάστηκε η επιρροή που είχε στην επίδοση των εφαρμογών με προτεραιότητα η παράλληλη εκτέλεση. Έτσι σε αυτό το στάδιο δεν εφαρμόστηκε κατανομή της cache και όλα τα VMs είχαν απεριόριστη πρόσβαση στην LLC. Κάθε ένα από τα 28 benchmarks εξετάστηκε σε δέκα σενάρια παράλληλης εκτέλεσης. Σε κάθε σενάριο το gold VM συνεκτελείτο με άλλα εννέα silver VMs, έχοντας το κάθε VM σε ξεχωριστό πυρήνα στον ίδιο NUMA κόμβο. Στους πίνακες 4-3

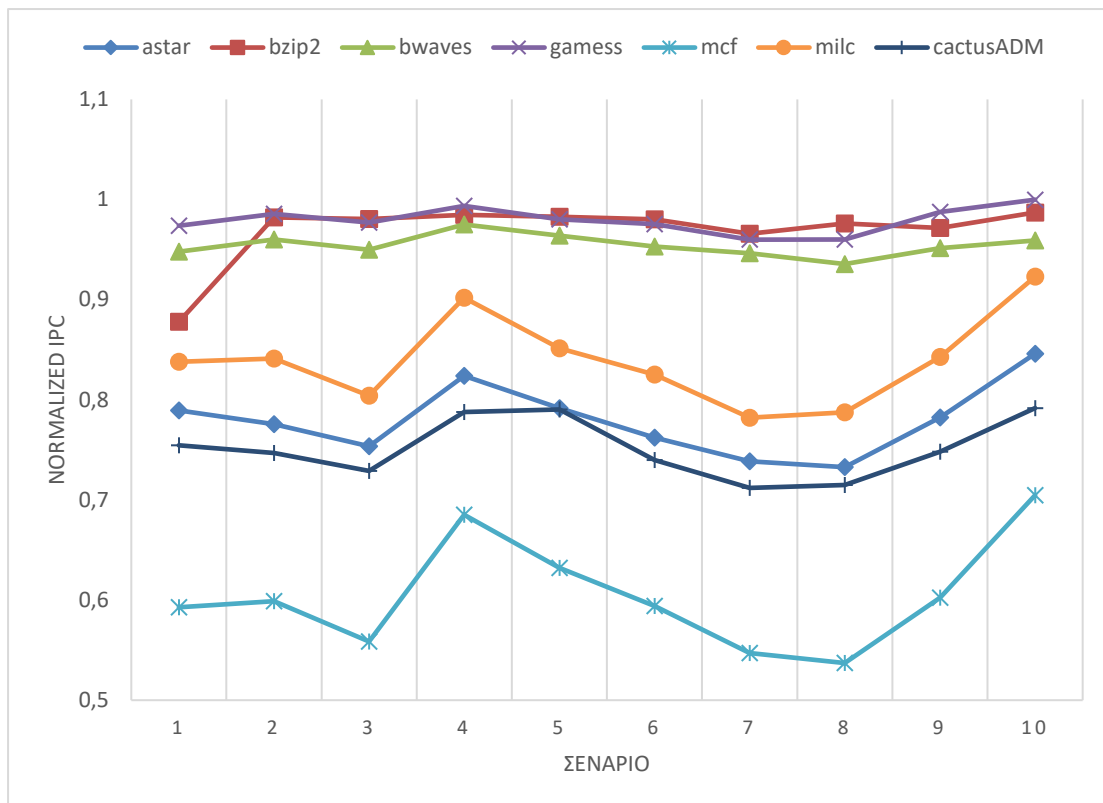
και 4-4 δίνονται οι δέκα διαφορετικές εννιάδες των silver VMs απέναντι στα οποία συνεκτελείται και μελετάται το κάθε gold VM. Στη συνέχεια, στις πιο κάτω γραφικές αναπαρίσταται, για όλα τα πιο πάνω σενάρια, το Normalized IPC του κάθε benchmark όταν αυτό έτρεχε σε gold VM.

	Σενάριο 1	Σενάριο 2	Σενάριο 3	Σενάριο 4	Σενάριο 5
VM1	tonto	bwaves	mcf	cactusADM	cactusADM
VM2	gamess	calculix	GemsFDTD	mcf	mcf
VM3	bwaves	omnetpp	leslie3d	gamess	mcf
VM4	leslie3d	gobmk	namd	astar	mcf
VM5	GemsFDTD	gamess	omnetpp	namd	gobmk
VM6	gamess	tonto	bzip2	gamess	calculix
VM7	tonto	cactusADM	bwaves	povray	GemsFDTD
VM8	omnetpp	leslie3d	omnetpp	astar	namd
VM9	povray	milc	tonto	milc	milc

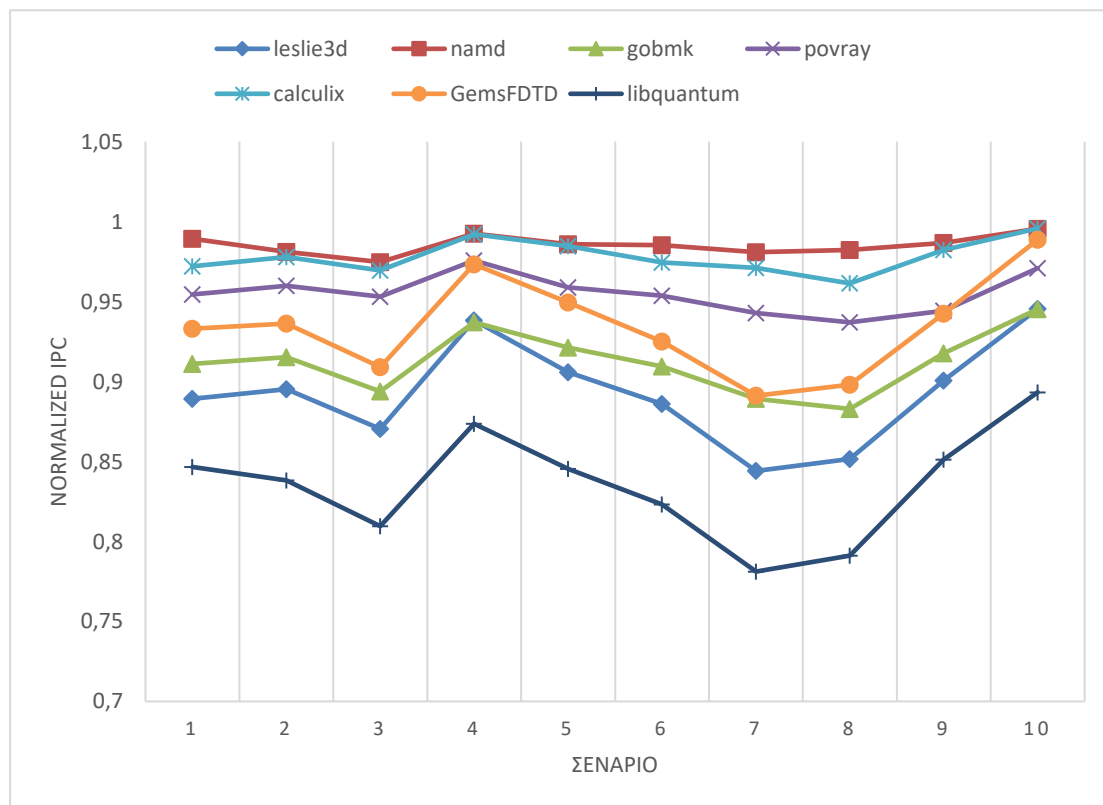
Πίνακας 4-3: Σενάρια συνεκτέλεσης 1-5

	Σενάριο 6	Σενάριο 7	Σενάριο 8	Σενάριο 9	Σενάριο 10
VM1	bzip2	bwaves	bwaves	mcf	gobmk
VM2	mcf	mcf	mcf	mcf	mcf
VM3	namd	leslie3d	bwaves	namd	povray
VM4	astar	bwaves	astar	calculix	namd
VM5	milc	cactusADM	bzip2	milc	milc
VM6	milc	tonto	libquantum	gamess	calculix
VM7	GemsFDTD	namd	bzip2	povray	leslie3d
VM8	omnetpp	GemsFDTD	mcf	gobmk	bwaves
VM9	milc	milc	gobmk	bwaves	cactusADM

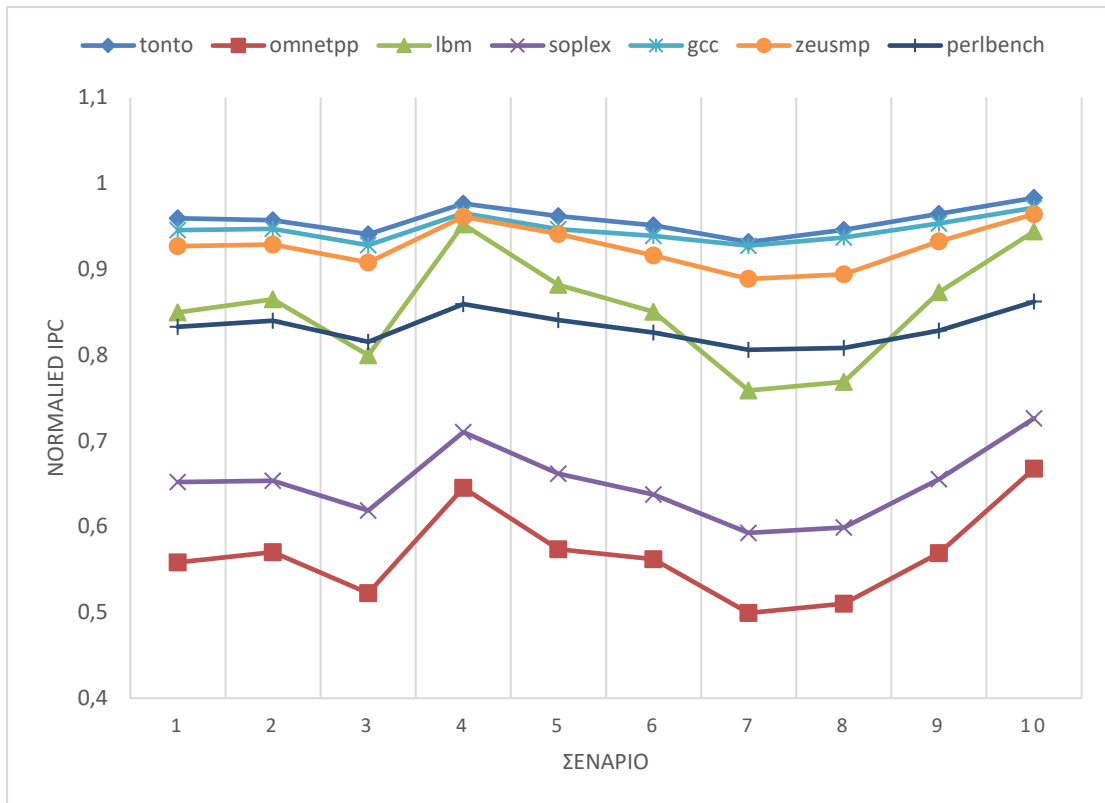
Πίνακας 4-4: Σενάρια συνεκτέλεσης 6-10



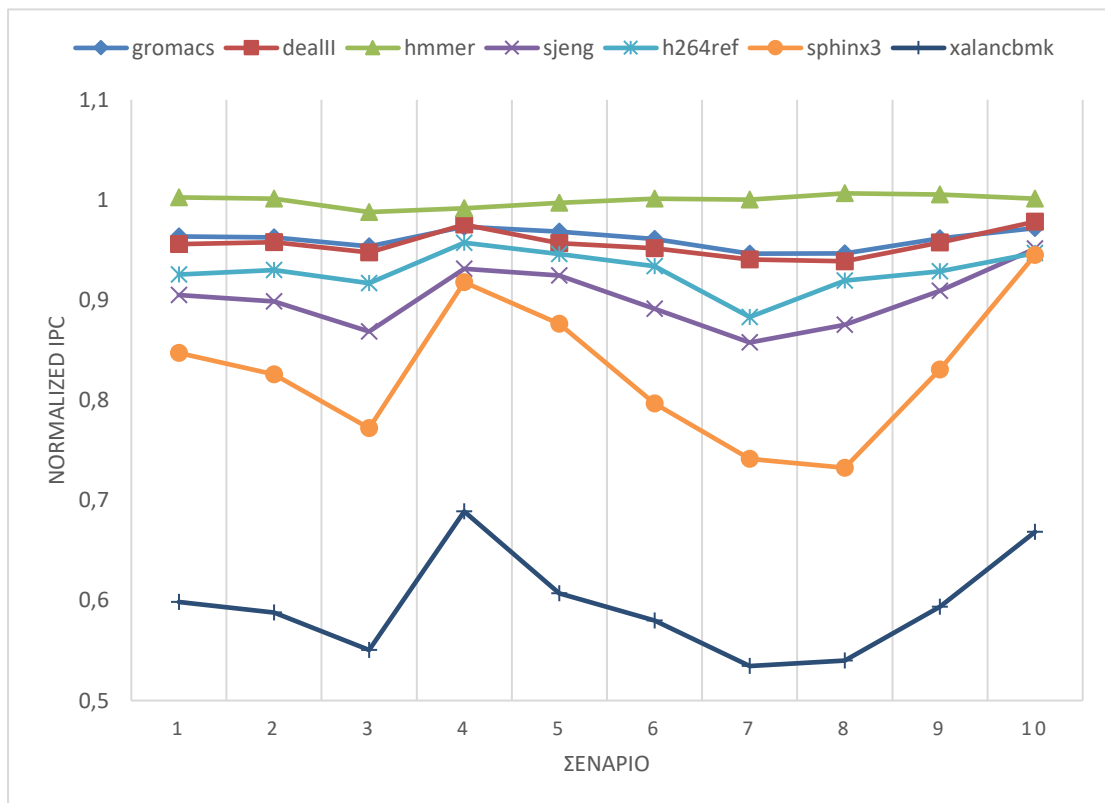
Γράφημα 4-1: Normalized IPC σε σενάρια συνεκτέλεσης χωρίς κατανομή της LLC (i)



Γράφημα 4-2: Normalized IPC σε σενάρια συνεκτέλεσης χωρίς κατανομή της LLC (ii)



Γράφημα 4-3: Normalized IPC σε σενάρια συνεκτέλεσης χωρίς κατανομή της LLC (iii)



Γράφημα 4-4: Normalized IPC σε σενάρια συνεκτέλεσης χωρίς κατανομή της LLC (iv)

Από τις πιο πάνω γραφικές παραστάσεις φαίνεται πως αρκετά από τα benchmarks (π.χ. bzip2, games, namd, calculix, tonto, hmmer κ.α.) δεν επηρεάζονται ιδιαίτερα από την συνεκτέλεσή τους με άλλες εφαρμογές αφού παρουσιάζουν σε όλα τα σενάρια Normalized IPC μεγαλύτερο του 0.90. Υπάρχουν όμως και benchmarks (π.χ. mcf, libquantum, omnetpp, soplex, xalancbmk κ.α.) των οποίων το Normalized IPC πέφτει κάτω από το 0.84, παρουσιάζουν δηλαδή Slowdown μεγαλύτερο του 1.20, είτε στα πλείστα είτε και σε όλα τα σενάρια παράλληλης εκτέλεσης που εξετάστηκαν. Αυτό σημαίνει πως τα συγκεκριμένα benchmarks δεν καταφέρνουν να ικανοποιήσουν το SLO που συμφωνήθηκε. Από τα γραφήματα μπορεί κανείς επίσης να διακρίνει το γεγονός πως όλα τα benchmarks παρουσιάζουν παρόμοια συμπεριφορά όσο αφορά το κάθε σενάριο συνεκτέλεσης. Στα σενάρια 4 και 10 φαίνεται πως όλες οι εφαρμογές αποδίδουν καλύτερα από τις υπόλοιπες περιπτώσεις, ενώ στα σενάρια 3, 7 και 8 η επίδοσή τους παρουσιάζει μεγάλη πτώση σε σχέση με τα υπόλοιπα σενάρια.

4.2.3 Επίδοση εφαρμογών σε σενάρια παράλληλης εκτέλεσης με

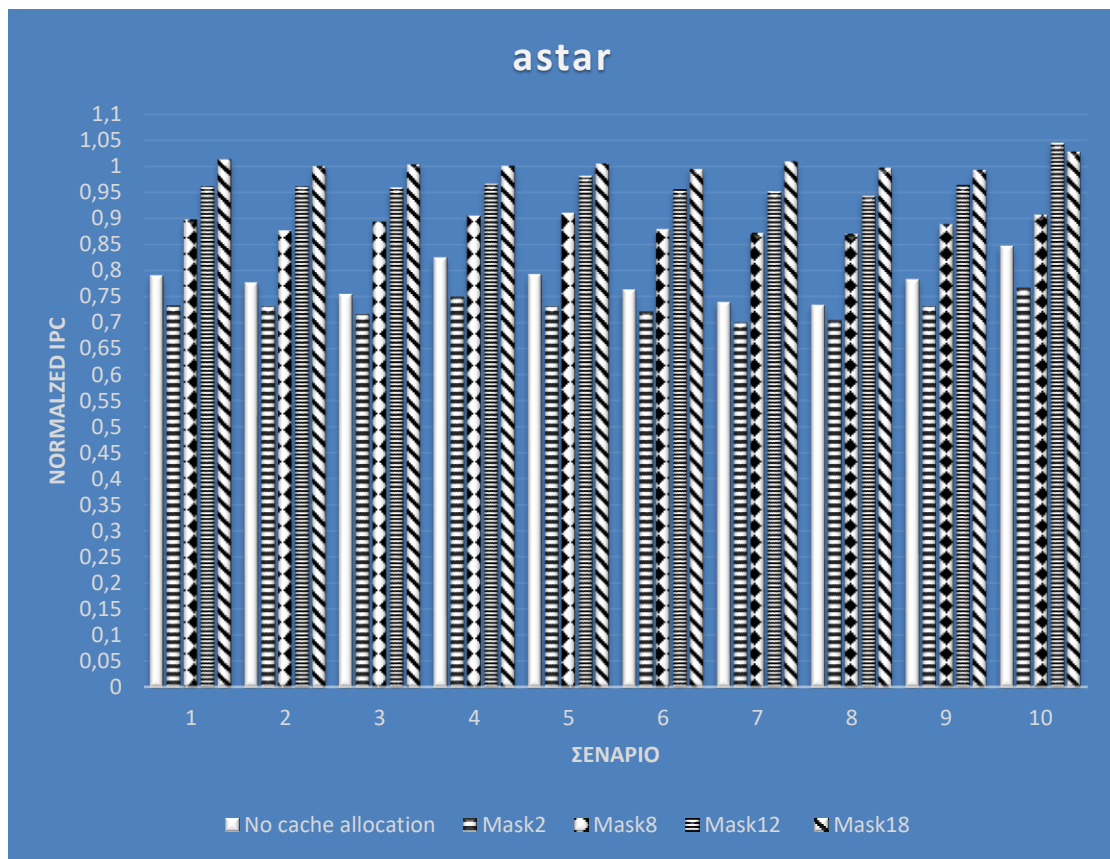
στατική κατανομή της LLC

Σε αυτό το στάδιο της μελέτης των εφαρμογών μας ενδιαφέρει να εξετάσουμε την επίδοση που θα εμφανίσουν τα benchmarks με προτεραιότητα, στα πιο πάνω σενάρια παράλληλης εκτέλεσης, κατανέμοντας ταυτόχρονα και την LLC. Όπως φαίνεται στον πίνακα 4-1, η LLC μοιράζεται σε 20 cache ways. Το CAT μας δίνει την δυνατότητα σε ένα και μόνο πυρήνα να κατανέμουμε μέχρι και τα 19 cache ways και το 1 εναπομείναν cache way στους υπόλοιπους πυρήνες. Πιο συγκεκριμένα, σε αυτή τη φάση κάθε ένα από τα πιο πάνω δέκα σενάρια παράλληλης εκτέλεσης εξετάζεται σε τέσσερις περιπτώσεις κατανομής της LLC:

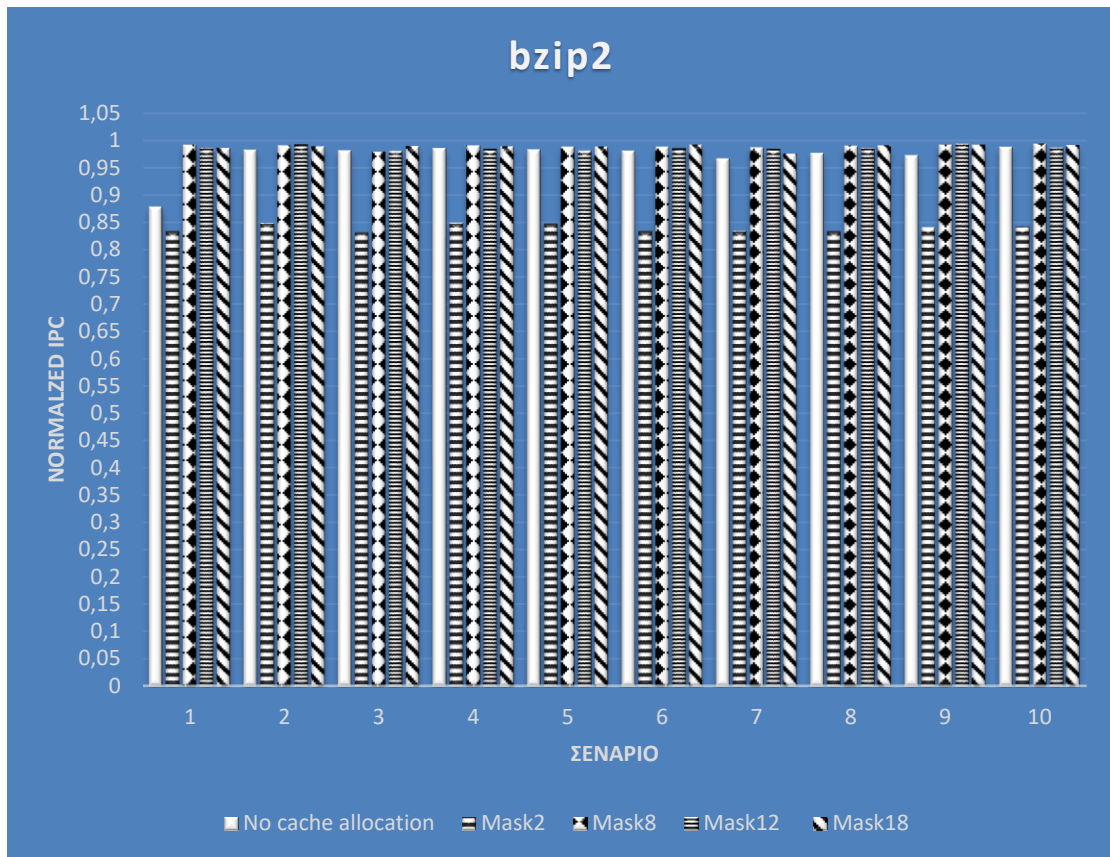
- Mask2: Κατανομή των 2 cache ways στο πυρήνα με το gold VM και κατανομή των εναπομεινάντων 18 cache ways στους υπόλοιπους εννέα πυρήνες.
- Mask8: Κατανομή των 8 cache ways στο πυρήνα με το gold VM και κατανομή των εναπομεινάντων 12 cache ways στους υπόλοιπους εννέα πυρήνες.
- Mask12: Κατανομή των 12 cache ways στο πυρήνα με το gold VM και κατανομή των εναπομεινάντων 8 cache ways στους υπόλοιπους εννέα πυρήνες.

- Mask18: Κατανομή των 18 cache ways στο πυρήνα με το gold VM και κατανομή των εναπομεινάντων 2 cache ways στους υπόλοιπους εννέα πυρήνες.

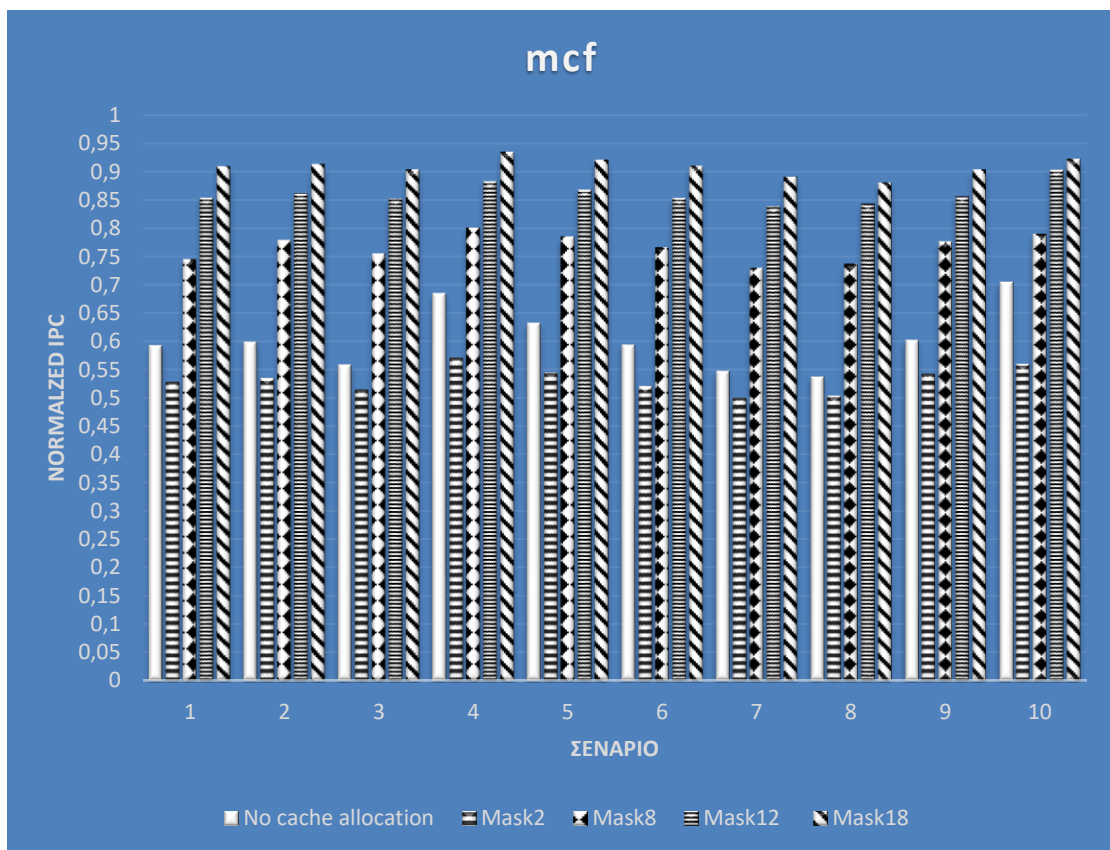
Τα πιο κάτω γραφήματα αναπαριστούν το Normalized IPC που μετρήθηκε για την κάθε εφαρμογή με προτεραιότητα στις τέσσερις πιο πάνω περιπτώσεις σε κάθε σενάριο συνεκτέλεσης. Στα γραφήματα δίνονται και οι μετρήσεις της ενότητας 4.2.1 με λεζάντα «No cache allocation» για να γίνει μια συνολική σύγκριση όλων των περιπτώσεων. Παρουσιάζονται ενδεικτικά κάποια από τα benchmarks ενώ η επίδοση όλων των benchmarks καταγράφεται σε μορφή πινάκων στην ενότητα 8.1 του παραρτήματος.



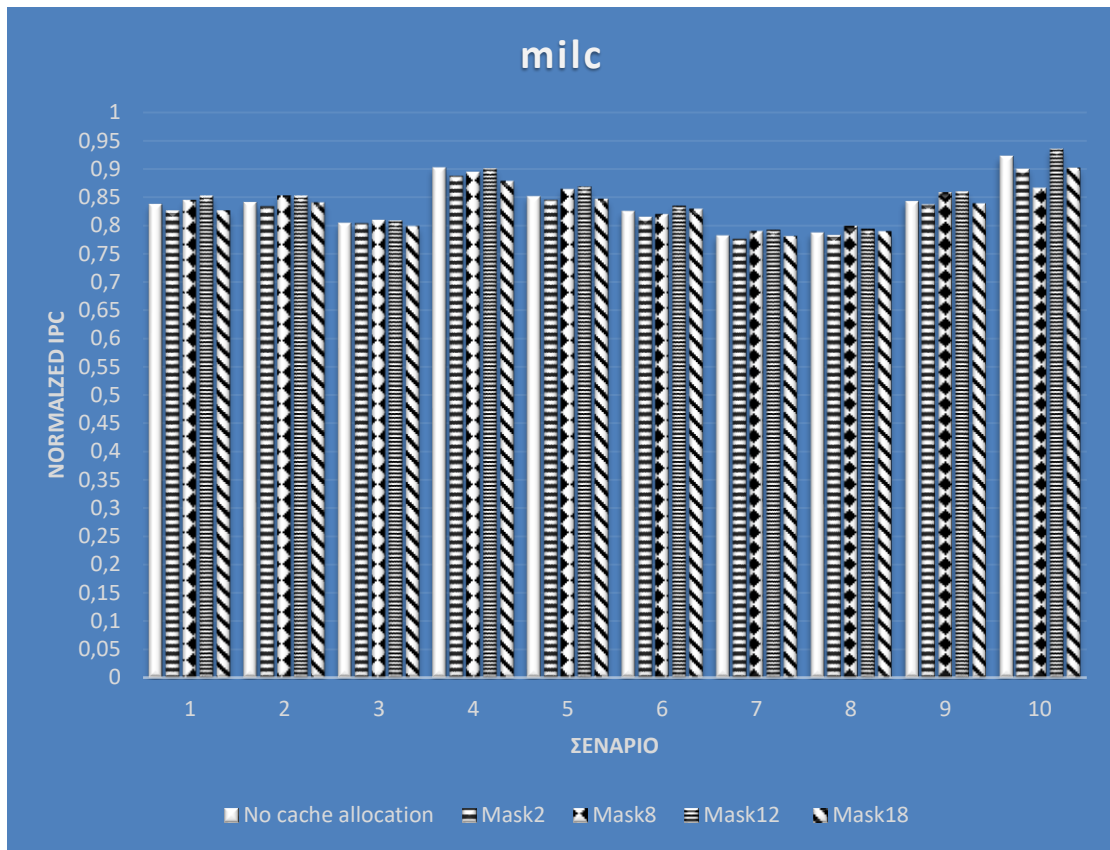
Γράφημα 4-5: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - astar



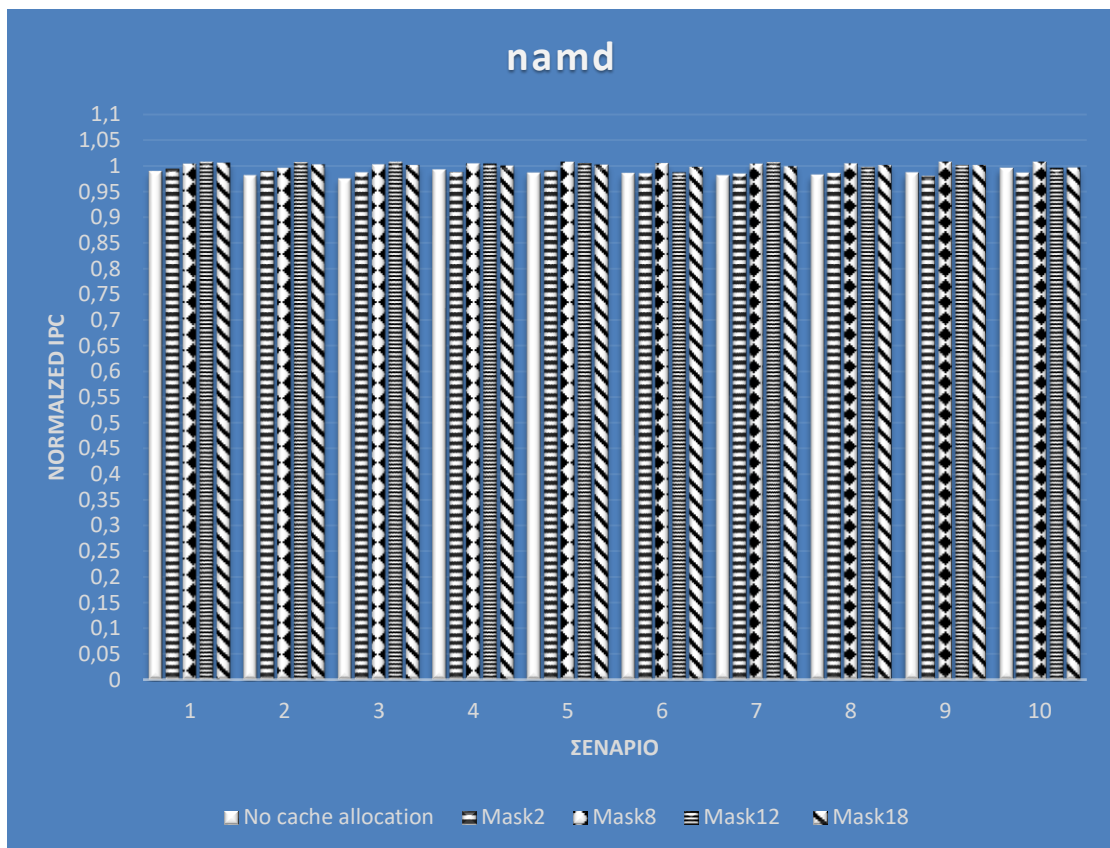
Γράφημα 4-6: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – bzip2



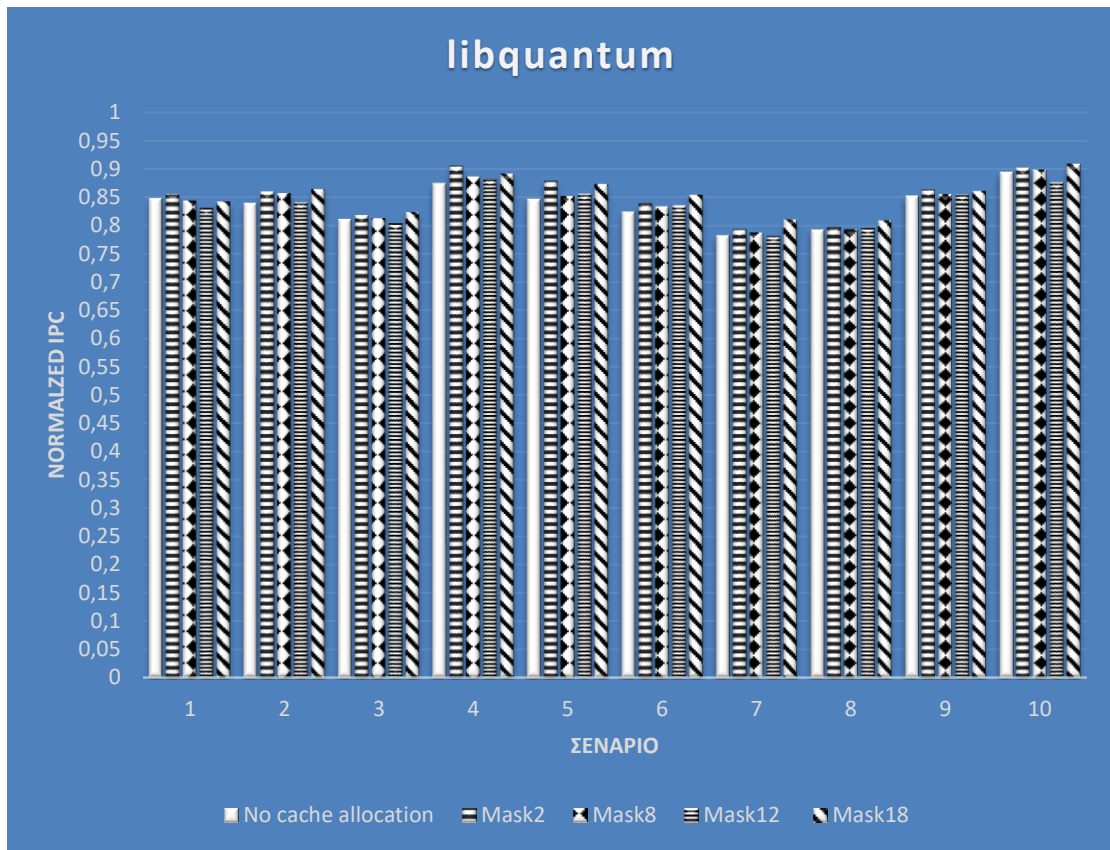
Γράφημα 4-7: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - mcf



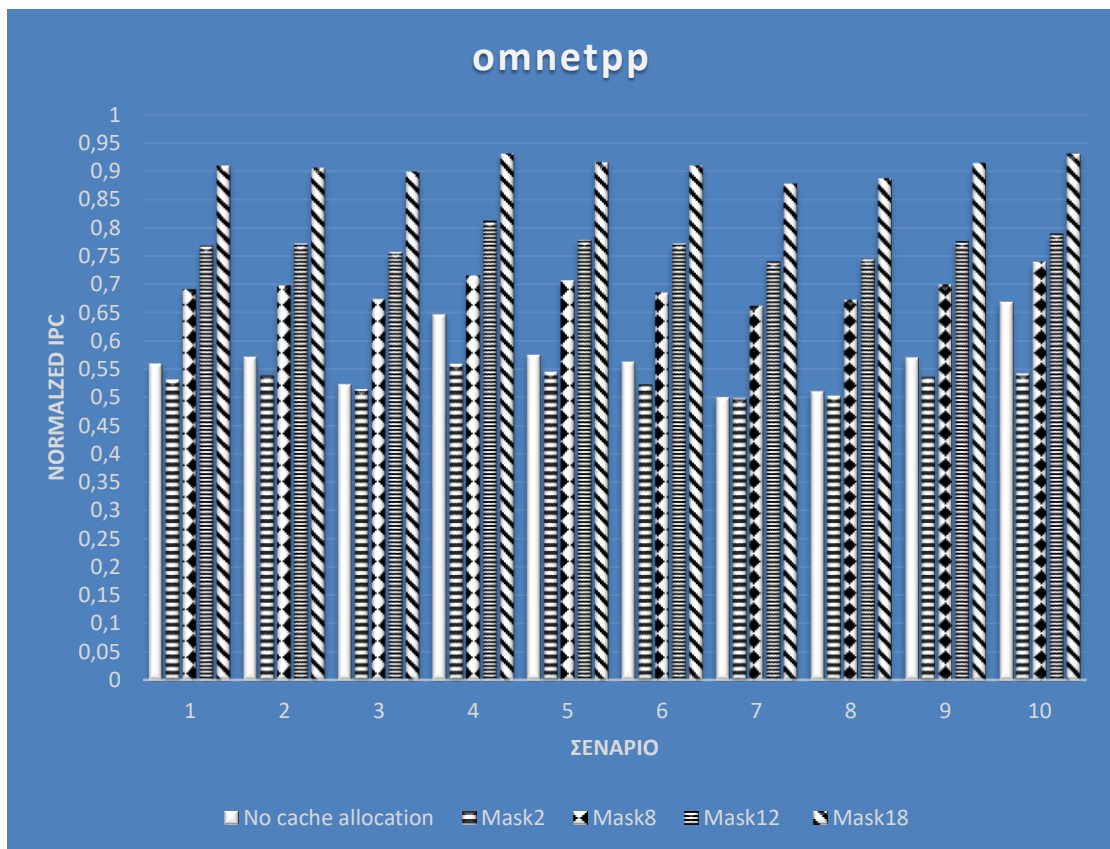
Γράφημα 4-8: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - milc



Γράφημα 4-9: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - namd



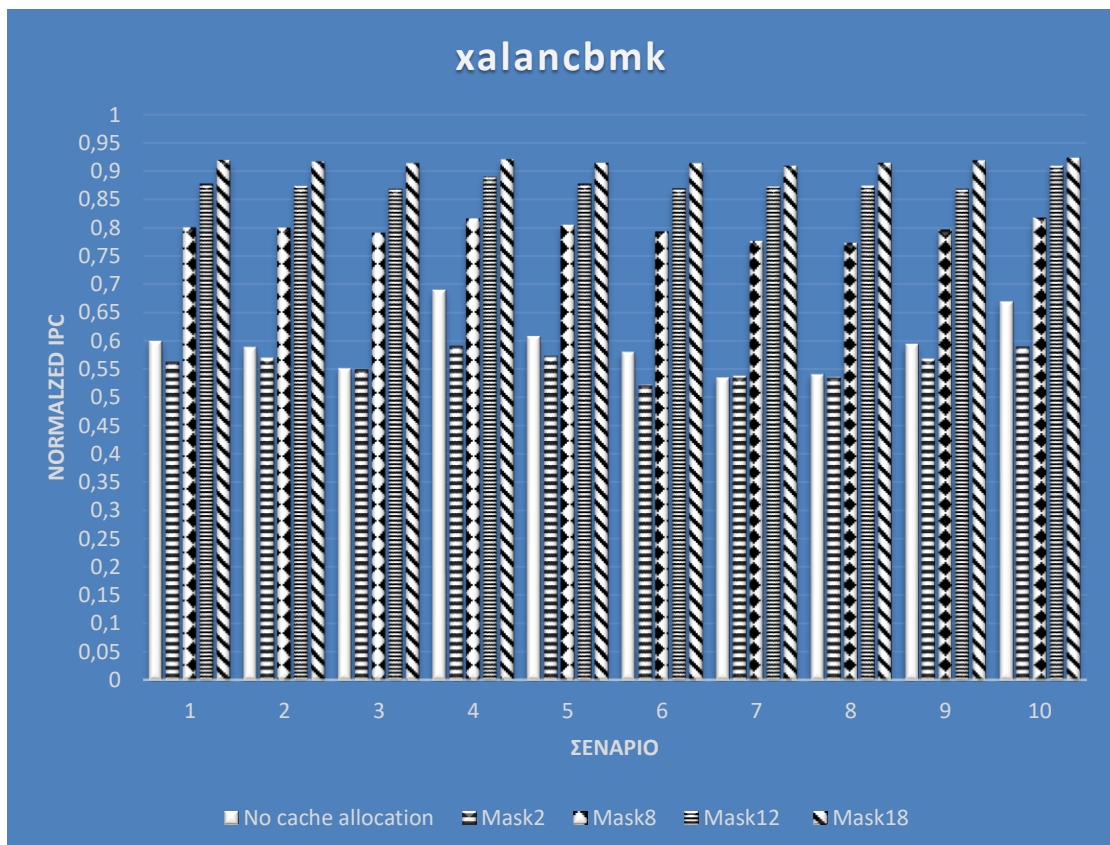
Γράφημα 4-10: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - libquantum



Γράφημα 4-11: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - omnetpp



Γράφημα 4-12: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - soplex

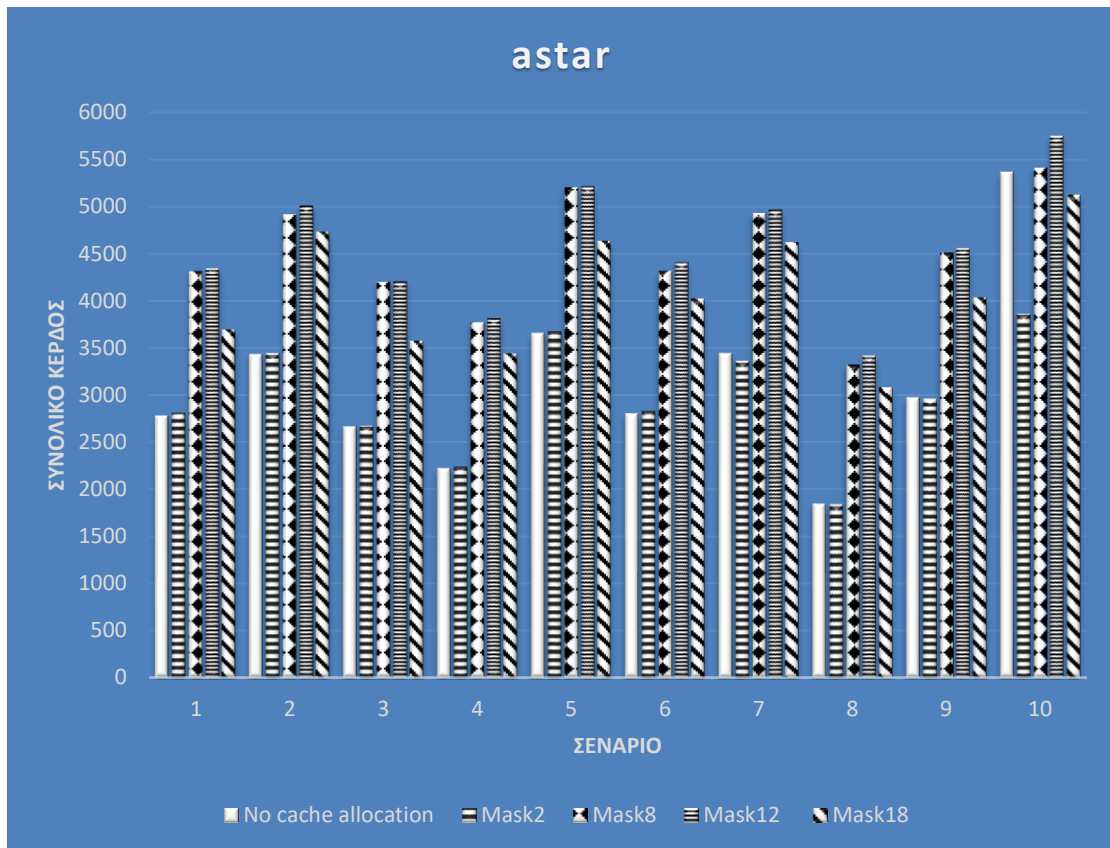


Γράφημα 4-13: Normalized IPC σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC - xalancbmk

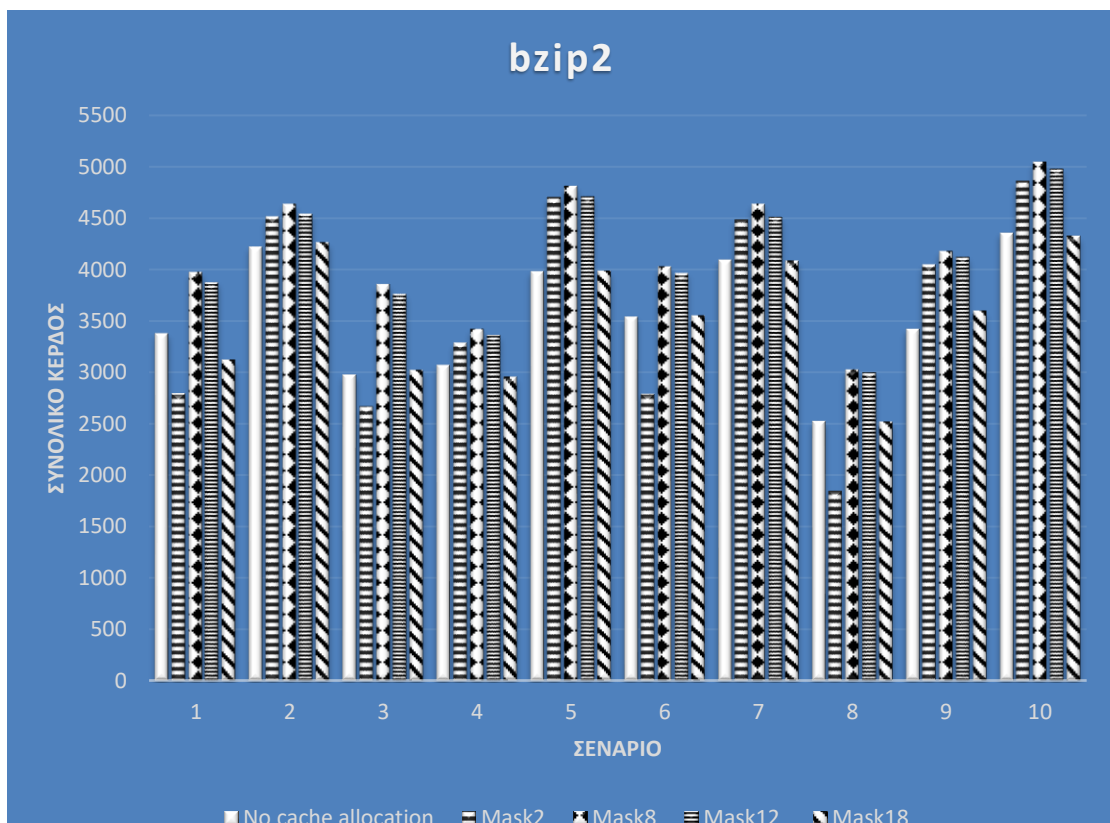
Όπως είναι προφανές, όσο πιο πολλά cache ways κατανέμονται στον πυρήνα όπου τρέχει το gold VM, τόσο πιο πολύ αυξάνεται το IPC του. Έτσι και στις πιο πάνω γραφικές παραστάσεις όσο αυξάνονται τα cache ways, το Normalized IPC κοντεύει όλο και πιο πολύ στη μονάδα. Μερικά benchmarks όμως φαίνεται, σε κάποιες περιπτώσεις παρουσιάζουν ελαφρώς μειωμένο Normalized IPC στη περίπτωση του Mask18. Το φαινόμενο αυτό μπορεί να οφείλεται στο γεγονός πως αυξάνεται ο ανταγωνισμός για τη χρήση του διαύλου δεδομένων από το ένα επίπεδο της cache στο επόμενο. Αξιοσημείωτο είναι και το γεγονός πως στα σενάρια 4 και 10 όλα τα benchmarks συνεχίζουν να αποδίδουν καλύτερα από τις υπόλοιπες περιπτώσεις, ενώ στα σενάρια 3, 7 και 8 το Normalized IPC που παρουσιάζουν τα benchmarks συνεχίζει να είναι χαμηλότερο σε σχέση με τις άλλες περιπτώσεις. Από τα πιο πάνω συμπεραίνουμε πως ένας στατικός τρόπος κατανομής της LLC δεν μπορεί να είναι πάντοτε αποτελεσματικός αφού η συμπεριφορά των εφαρμογών αλλάζει ανάλογα με τις διεργασίες που συνεκτελούνται μαζί τους στο σύστημα.

4.2.4 Συνολικό κέρδος στο σύστημα σε σενάρια παράλληλης εκτέλεσης με στατική κατανομή της LLC

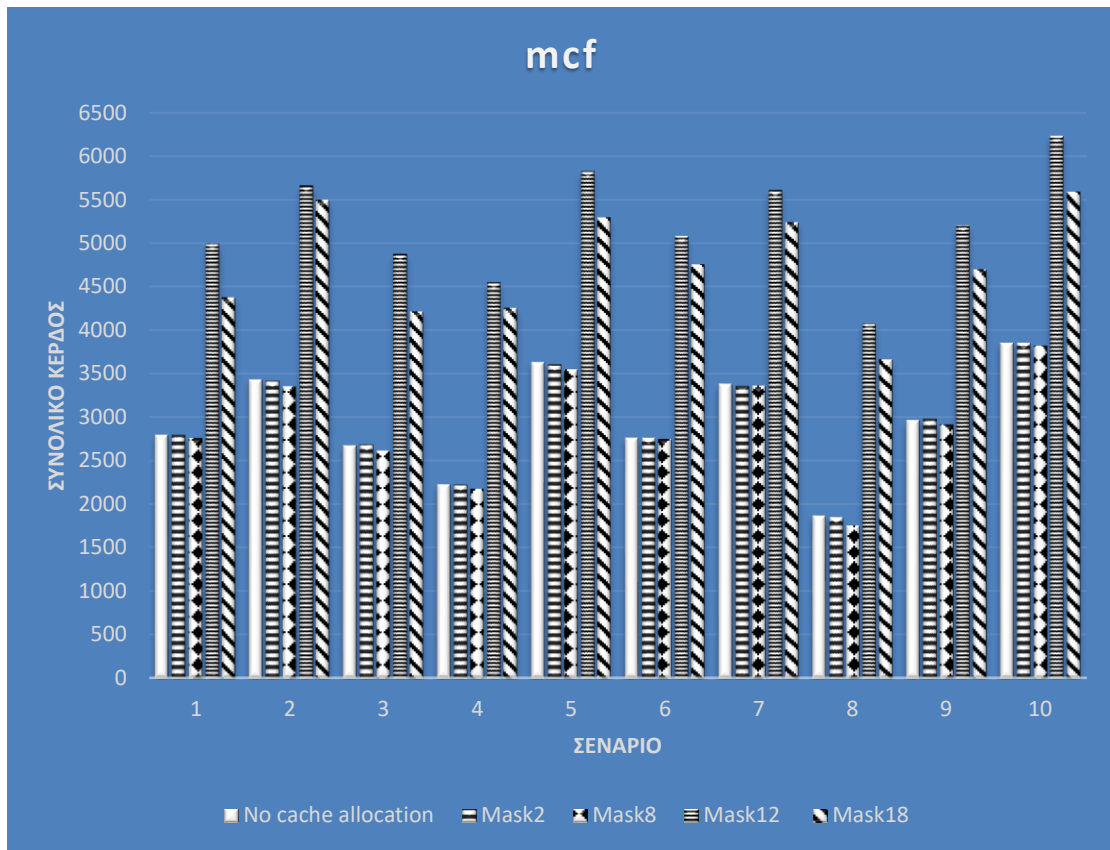
Ενώ η αυξημένη κατανομή των cache ways στο gold VM αυξάνει την απόδοση του, για τα silver VMs συνεπάγεται αυτόματα μεγαλύτερος ανταγωνισμός για πρόσβαση στα λιγότερα cache ways που τους απομένουν. Ο ανταγωνισμός ανάμεσα στα silver VMs είναι δυνατόν να οδηγήσει σε μείωση του κέρδους του συστήματος αφού μπορεί να μην τηρούν το SLO που συμφωνήθηκε για αυτά. Στις γραφικές παραστάσεις που ακολουθούν αναπαρίσταται για κάθε μια από τις περιπτώσεις της προηγούμενης ενότητας το συνολικό κέρδος που επιφέρουν όλα τα VMs, silver και gold. Στην ενότητα 8.2 του παραρτήματος βρίσκονται σε μορφή πίνακα οι πληροφορίες για το συνολικό κέρδος που εισπράττεται στο σύστημα σε κάθε περίπτωση για κάθε benchmark όταν αυτό εκτελείται ως gold. Ο τρόπος υπολογισμού του κέρδους που επιφέρει ένα VM αναλύθηκε στην ενότητα 4.1.2.



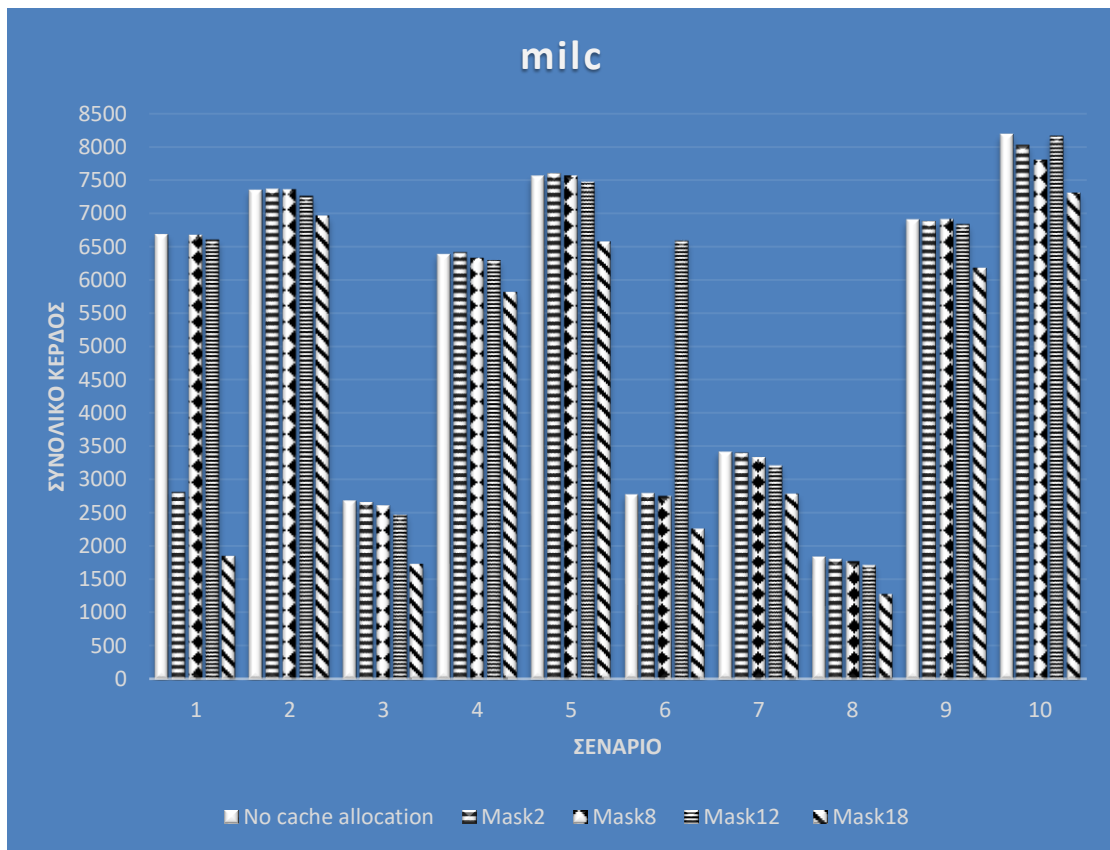
Γράφημα 4-14: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – astar ως gold



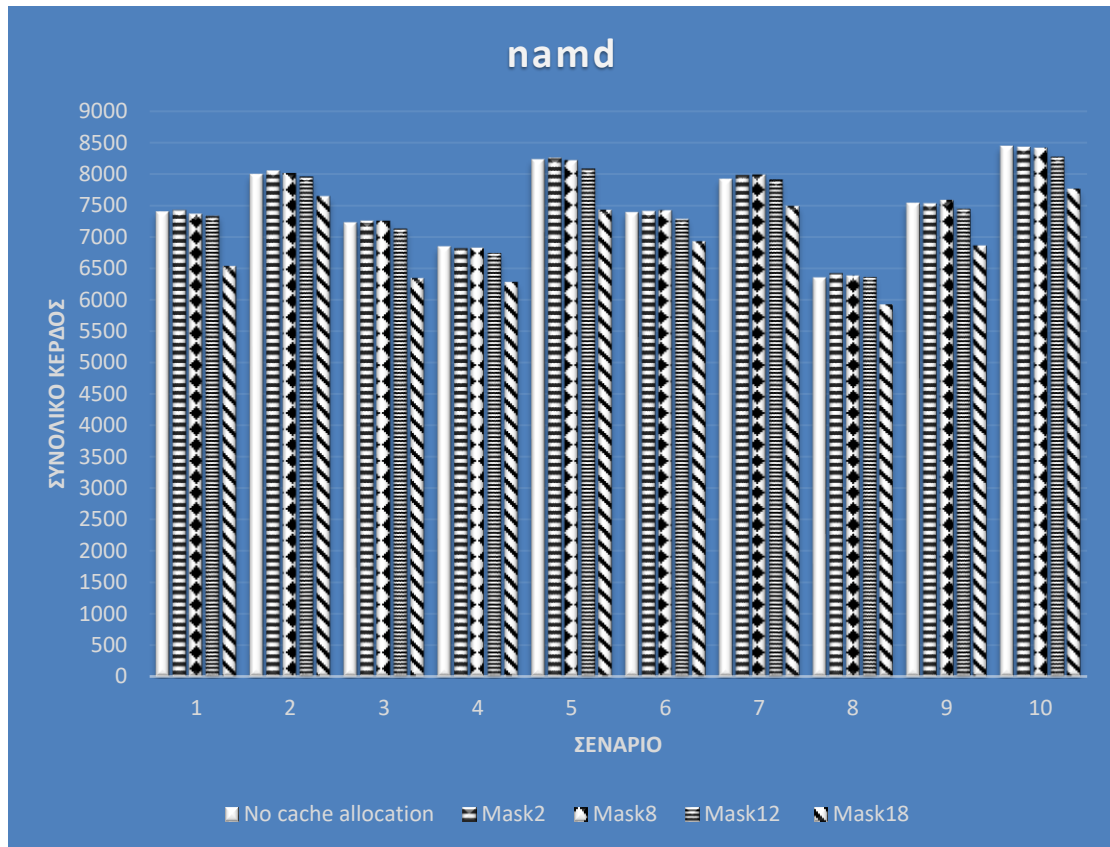
Γράφημα 4-15: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – bzip2 ως gold



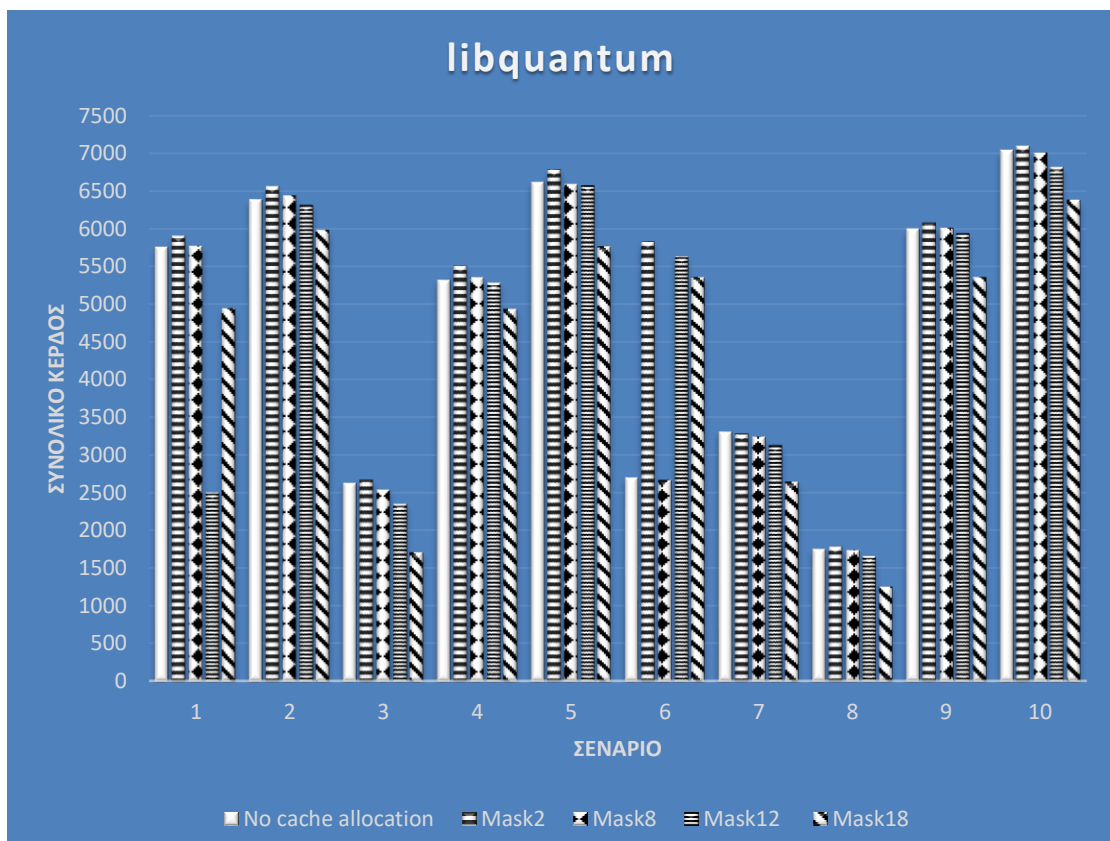
Γράφημα 4-16: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – mcf ως gold



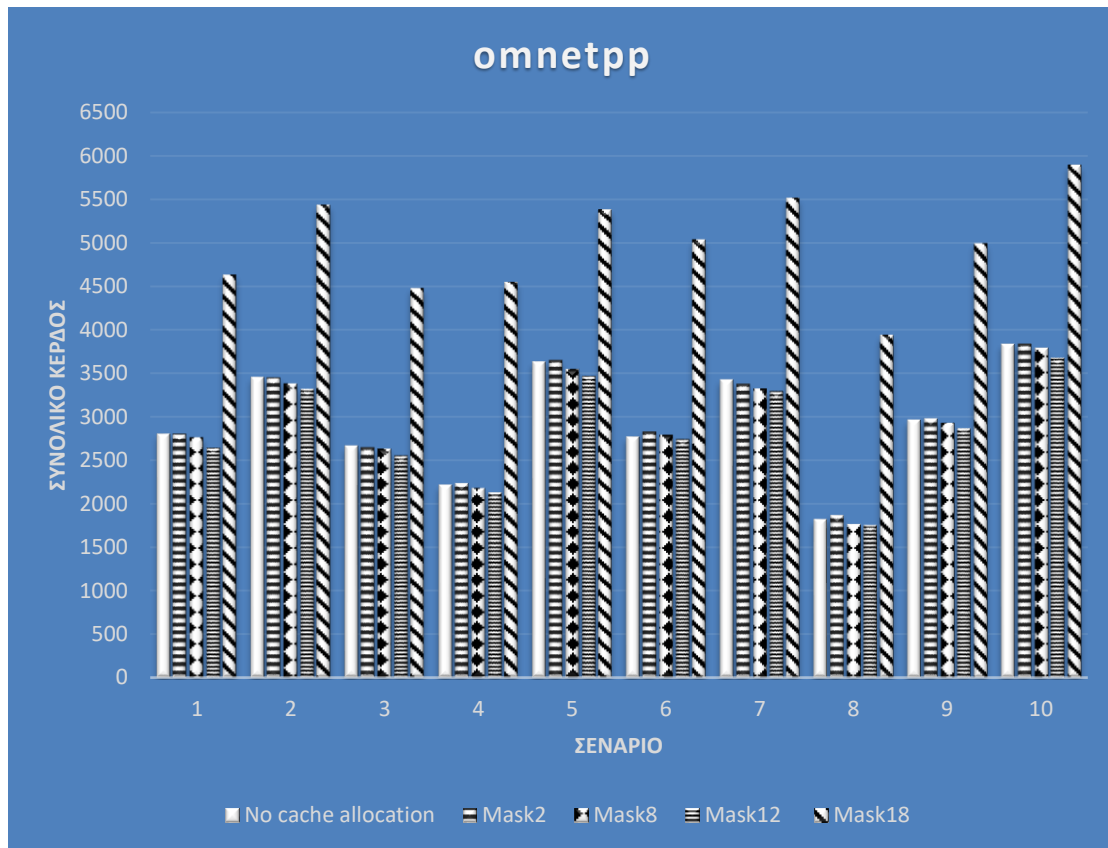
Γράφημα 4-17: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – milc ως gold



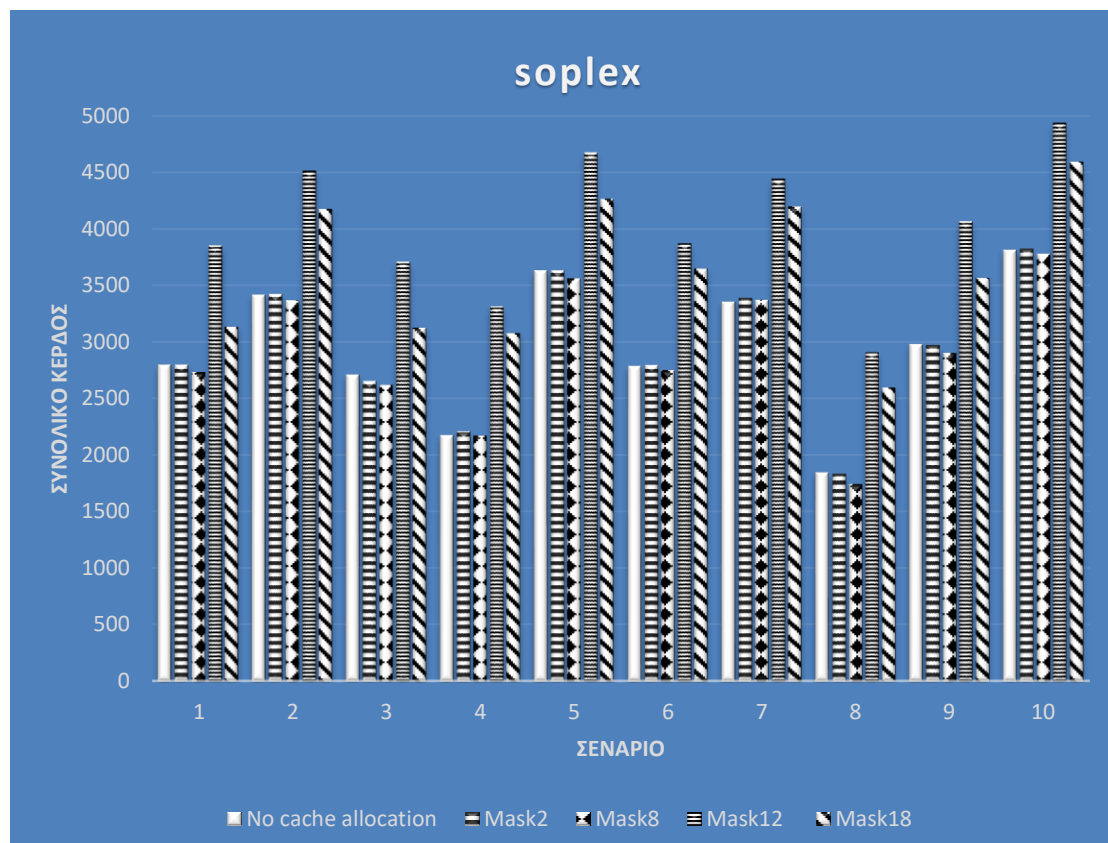
Γράφημα 4-18: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – namd ως gold



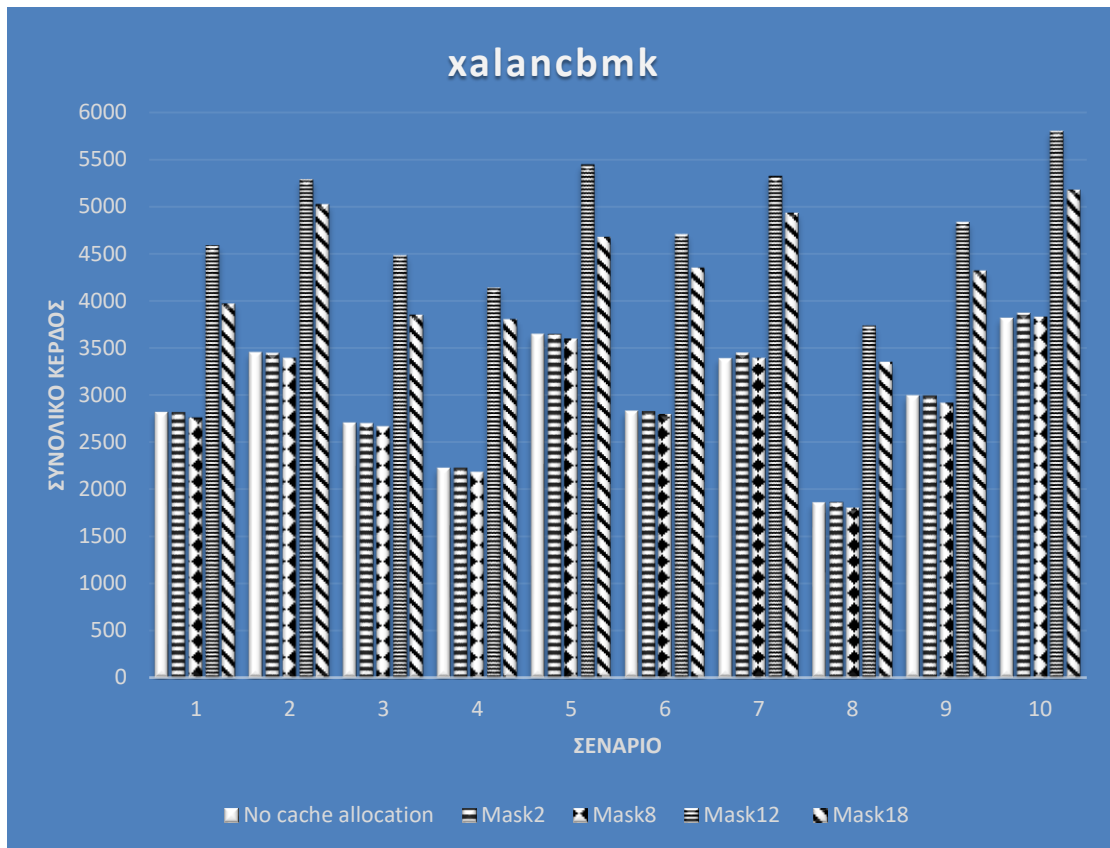
Γράφημα 4-19: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – libquantum ως gold



Γράφημα 4-20: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – omnetpp ως gold



Γράφημα 4-21: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – soplex ως gold



Γράφημα 4-22: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με στατική κατανομή της LLC – xalancbmk ως gold

Εξετάζοντας τα πιο πάνω αποτελέσματα φαίνεται πως μόνο στη περίπτωση του omnetpp το κέρδος στο σύστημα γίνεται μέγιστο στο Mask18, αφού όπως παρατηρείται και στα γραφήματα στην ενότητα 4.2.3, στη περίπτωση του omnetpp το SLO δεν τηρείται στις περιπτώσεις των No cache allocation, Mask2, Mask8 και Mask12. Αξιοσημείωτο είναι ακόμη το γεγονός πως σε όλα τα σενάρια όπου οι libquantum, milc και sorplex παρέχουν μέγιστο κέρδος, το SLO τηρείται οριακά μιας και το Normalized IPC τους (ενότητα 4.2.3) είτε ισούται με 0.84 είτε βρίσκεται οριακά πάνω από αυτή τη τιμή. Παρόλα αυτά τα πλείστα benchmarks φαίνεται να παρέχουν μέγιστο κέρδος στο σύστημα κατά τα Mask8 και Mask12 ενώ λίγες είναι οι περιπτώσεις κατά τις οποίες το κέρδος μεγιστοποιείται στα No cache allocation και Mask2.

Η πτώση του κέρδους στα σενάρια 3, 7 και 8 δικαιολογείται λόγω του μειωμένου Normalized IPC (ενότητα 4.2.3) που εμφανίζουν όλες οι εφαρμογές στα σενάρια αυτά. Ιδικά στις περιπτώσεις των milc και libquantum, κανένα από τα δύο benchmarks δεν καταφέρνει να παρουσιάσει Slowdown μικρότερο του 1.20 σε καμία από τις πέντε διαφορετικές κατανομές της LLC στα τρία αυτά σενάρια. Ως αποτέλεσμα, το συνολικό κέρδος είναι πολύ χαμηλό αφού προέρχεται μόνο από τις silver διεργασίες.

Συγκρίνοντας το μέγιστο συνολικό κέρδος που επιφέρει στο σύστημα το κάθε σενάριο με το Normalized IPC (ενότητα 4.2.3) που εμφανίζει η κάθε εφαρμογή με προτεραιότητα, παρατηρείται πως μέγιστο κέρδος εισπράττεται όταν η εφαρμογή με προτεραιότητα παρουσιάζει Normalized IPC κοντά στο 0.95 (Slowdown ίσο με 1.05). Τέλος, παρατηρείται επίσης το γεγονός πως αν το Normalized IPC του gold benchmark υπερβεί το 0.95 τότε το συνολικό κέρδος είτε μειώνεται, λόγω του ανταγωνισμού μεταξύ των silver benchmarks, είτε παραμένει σταθερό.

4.3 Χαρακτηριστικά Εφαρμογών

Από τις πληροφορίες που εξαγάγαμε στην ενότητα 4.2 σχετικά με την επίδοση των benchmarks και το κέρδος που λαμβάνεται στο σύστημα, σε κάθε σενάριο παράλληλης εκτέλεσης και κατανομής της LLC, μπορούμε τώρα να καταλήξουμε σε κάποια ξεχωριστά χαρακτηριστικά που παρουσιάζει η κάθε εφαρμογή. Εκτός όμως από τα ξεχωριστά χαρακτηριστικά της κάθε εφαρμογής, έχει παρατηρηθεί και μια κοινή συμπεριφορά σε όλα τα benchmarks. Πιο συγκεκριμένα φαίνεται πως όταν το Normalized IPC ξεπερνά το 0.95 το συνολικό κέρδος στο σύστημα είτε μειώνεται, είτε παραμένει σταθερό. Δηλαδή το Slowdown της εφαρμογής δεν είναι αναγκαίο να πέφτει πέραν του 1.05. Έτσι, για να τηρεί η gold εφαρμογή το SLO και για να αυξάνεται το συνολικό κέρδος στο σύστημα, πρέπει το Slowdown της εφαρμογής με προτεραιότητα να κυμαίνεται μεταξύ των 1.05 και 1.20.

Στο πίνακα 4-5 αναγράφονται τα χαρακτηριστικά που έχουν παρατηρηθεί για το κάθε benchmark, όπου:

- $IPC_{isolation}$: το IPC της εφαρμογής σε απομονωμένη εκτέλεση
- CW: αριθμός cache ways που κατανεμήθηκαν στην εφαρμογή, όταν έτρεχε σε gold VM, για τα οποία σε όλα τα σενάρια το συνολικό κέρδος στο σύστημα ήταν μέγιστο.
- Strict: ένα benchmark θα χαρακτηρίζεται ως Strict εάν το Normalized IPC που εμφανίζει στην περίπτωση της κατανομής των CW κυμαίνεται από 0.84 μέχρι 0.85. Δηλαδή το Slowdown κυμαίνεται από 1.18 μέχρι 1.2, τηρώντας έτσι οριακά το SLO.

Benchmark	IPC _{isolation}	CW	Strict
astar	1.01	12	OXI
bzip2	1.49	8	OXI
Bwaves	2.38	8	OXI
gamess	2.73	2	OXI
mcf	0.51	12	NAI
milc	1.17	12	NAI
cactusADM	1.87	8	OXI
leslie3d	2.06	12	OXI
namd	2.23	2	OXI
gobmk	1.24	2	OXI
povray	2.36	2	OXI
calculix	3.07	8	OXI
GemsFDTD	1.72	8	OXI
libquantum	2.05	2	NAI
tonto	2.24	8	OXI
omnetpp	0.85	18	OXI
lbm	1.40	8	OXI
soplex	1.12	12	NAI
gcc	1.44	8	OXI
zeusmp	1.88	8	OXI
perlbench	2.46	8	OXI
gromacs	2.41	8	OXI
dealII	2.33	8	OXI
hmmer	2.60	2	OXI
sjeng	1.60	8	OXI
h264ref	2.86	2	OXI
sphinx3	2.26	12	OXI
xalancbmk	1.86	12	OXI

Πίνακας 4-5: Συμπεριφορά των benchmarks

5

Δυναμικός Μηχανισμός Διατήρησης του SLO με

Κατανομή της LLC

Στο κεφάλαιο 2 παρουσιάστηκαν διάφορες υλοποιήσεις για την διαχείριση της cache. Όπως αναφέρθηκε και στην ενότητα 2.2, οι υλοποιήσεις που συνδυάζουν τεχνικές profiling ή online monitoring με μηχανισμούς κατανομής της LLC τείνουν να αυξάνουν την χρησιμοποίηση της cache αυξάνοντας έτσι και το QoS ολόκληρου του συστήματος. Βασισμένοι σε αυτές τις προτάσεις αναπτύξαμε ένα μηχανισμό ο οποίος με τη χρήση του CMT και του CAT έχει την ικανότητα να επιβλέπει την επίδοση της εφαρμογής με προτεραιότητα κατά τον πραγματικό χρόνο εκτέλεσης και να κατανέμει δίκαια ανάμεσα στις εφαρμογές την LLC, αυξάνοντας ταυτόχρονα το κέρδος του συστήματος. Στις υποενότητες που ακολουθούν περιγράφεται και αναλύεται η δομή, η λειτουργία και η αποτελεσματικότητα του μηχανισμού που προτείνουμε.

5.1 Δομή και Λειτουργία

Ο μηχανισμός αναπτύχθηκε με σκοπό να επιβλέπει την επίδοση της εφαρμογής με προτεραιότητα και στόχο να διατηρεί πάντα το SLO που έχει συμφωνηθεί για τέτοιες εφαρμογές, μεγιστοποιώντας παράλληλα το συνολικό κέρδος του συστήματος. Για να αποφευχθεί η μείωση της επίδοσης της εφαρμογής με προτεραιότητα, λόγω του ανταγωνισμού που εμφανίζεται μεταξύ των συνεκτελούμενων διεργασιών, ο μηχανισμός κατανέμει την LLC μεταξύ των διεργασιών, κατά τον χρόνο εκτέλεσης, ανάλογα με της ανάγκες της gold εφαρμογής. Το σύστημα στο οποίο αναπτύχθηκε ο συγκεκριμένος μηχανισμός είναι το ίδιο με αυτό που αναφέρεται στην ενότητα 4.1.

Πιο αναλυτικά, όταν μια εφαρμογή με προτεραιότητα εισέρχεται στο σύστημα, ο μηχανισμός πρώτα ελέγχει το κατά πόσο η εφαρμογή είναι Strict (Πίνακας 5-5: Γενικά χαρακτηριστικά των benchmarks) και συμπεριφέρεται ανάλογα με το εάν η εφαρμογή ανήκει ή όχι στην κατηγορία αυτή.

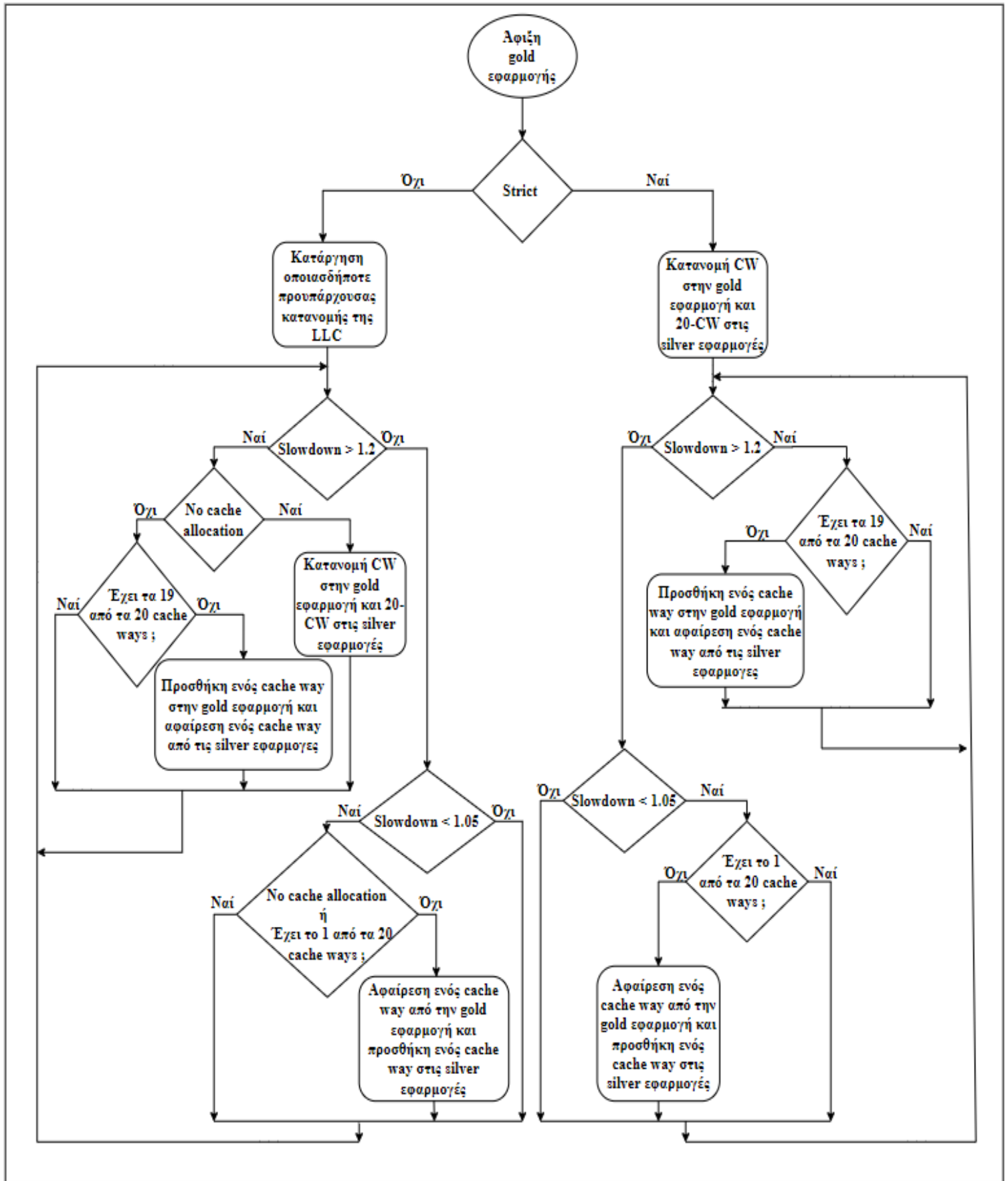
Στη περίπτωση μιας Strict εφαρμογής ο μηχανισμός κατανέμει μέσω του CAT στον πυρήνα όπου χρονοδρομολογείται η gold εφαρμογή, πριν ξεκινήσει την εκτέλεση της, τόσα cache ways όσα και τα CW που αναγράφονται στο πίνακα 4-5 για την εκάστοτε διεργασία. Τα εναπομείναντα 20-CW cache ways κατανέμονται σε όλες τις silver εφαρμογές. Καθ' όλη την εκτέλεση της gold εφαρμογής λαμβάνονται μετρήσεις του IPC της, μέσω του CMT, και ανά καθορισμένο χρονικό διάστημα υπολογίζεται το Slowdown που παρουσιάζει. Όπως έχει αποδειχθεί από τις μετρήσεις που πήραμε στην ενότητα 4.2 προκειμένου να τηρείται το SLO της εφαρμογής με προτεραιότητα και ταυτόχρονα να μεγιστοποιείται το κέρδος στο σύστημα, χωρίς να υποφέρουν οι silver εφαρμογές, το Slowdown της εφαρμογής με προτεραιότητα πρέπει να κυμαίνεται μεταξύ του 1.05 και του 1.2. Εάν ο μηχανισμός ανιχνεύσει Slowdown μεγαλύτερο του 1.20 τότε η gold εφαρμογή έχει ανάγκη από περισσότερα cache ways. Έτσι ο μηχανισμός κατανέμει ένα ακόμη cache way στην εφαρμογή με προτεραιότητα ενώ αποδεσμεύει ένα cache way από τις silver εφαρμογές. Στην αντίθετη περίπτωση που ανιχνευθεί Slowdown μικρότερο από 1.05 τότε σημαίνει πως τα cache ways που έχουν κατανομηθεί στην gold εφαρμογή είναι υπεραρκετά με αποτέλεσμα ο μηχανισμός να αποδεσμεύει ένα cache way από την εφαρμογή με προτεραιότητα και να κατανέμει ένα cache way περισσότερο στις silver εφαρμογές.

Όταν στο σύστημα καταφθάσει μια εφαρμογή με προτεραιότητα η οποία δεν χαρακτηρίζεται ως Strict, ο μηχανισμός καταργεί την οποιαδήποτε προϋπάρχουσα κατανομή της LLC που υπήρχε στο σύστημα δίνοντας έτσι απεριόριστη πρόσβαση στην LLC σε όλες τις διεργασίες. Όπως και στην περίπτωση μιας Strict εφαρμογής, ο μηχανισμός λαμβάνει μετρήσεις για το IPC της gold εφαρμογής. Την πρώτη φορά που θα ανιχνεύσει ο μηχανισμός Slowdown μεγαλύτερο του 1.20 θα κατανέμει στον πυρήνα όπου εκτελείται εφαρμογή με προτεραιότητα τα CW που της αναλογούν (Πίνακας 4-5) και στις silver εφαρμογές τα εναπομείναντα 20-CW. Στην περίπτωση όμως που ο μηχανισμός ανιχνεύσει Slowdown μικρότερο του 1.05 προτού ανιχνευθεί κάποια παραβίαση του SLO, δεν γίνεται καμία αλλαγή στην κατανομή της cache αφού όλες οι διεργασίες έχουν πλήρη πρόσβαση σε αυτή και η gold εφαρμογή δείχνει να

εκτελείται ανεπηρέαστα. Τέλος όταν παρατηρηθεί παραβίαση του SLO ή Slowdown μικρότερο του 1.05, μετά από την πρώτη παραβίαση του SLO και της ανάλογης κατανομής της cache που εφαρμόζεται, ο μηχανισμός συμπεριφέρεται όπως και στη περίπτωση των Strict εφαρμογών. Δηλαδή προσθέτει ή αφαιρεί αντίστοιχα ένα cache way στην εφαρμογή με προτεραιότητα και το αποσπά ή το δίνει αντίστοιχως στις silver εφαρμογές.

Στη γενική και ακραία περίπτωση όπου στη gold εφαρμογή έχουν κατανεμηθεί τα 19 από τα 20 cache ways αλλά το Slowdown συνεχίζει να είναι μεγαλύτερο από 1.20, ή στη περίπτωση που η gold εφαρμογή κατέχει το 1 από τα 20 ways με το Slowdown να είναι μικρότερο από 1.05, ο χρήστης ενημερώνεται πως η gold εφαρμογή κατέχει το μέγιστο ή το ελάχιστο μέρος της cache αντίστοιχα, με αποτέλεσμα ο μηχανισμός να μην μπορεί να εφαρμόσει κάποια άλλη αλλαγή σχετικά με την κατανομή της LLC. Η όλη διαδικασία επιτήρησης της επίδοσης της εφαρμογής με προτεραιότητα συνεχίζει να επαναλαμβάνεται συνεχώς με σκοπό να διατηρεί το Slowdown της gold εφαρμογής μεταξύ των τιμών 1.05 και 1.20.

Στο διάγραμμα ροής (Εικόνα 5-1) παρουσιάζεται ο αλγόριθμος του μηχανισμού από τη στιγμή που εισέρχεται στο σύστημα η εφαρμογή με προτεραιότητα. Τα μπλοκ συνθήκης σχετικά με το εάν η εφαρμογή ανήκει στην κατηγορία Strict, τη τιμή του Slowdown της εφαρμογής και τον αριθμό των cache ways στα οποία έχει πρόσβαση μια εφαρμογή αφορούν πάντα την εφαρμογή με προτεραιότητα.



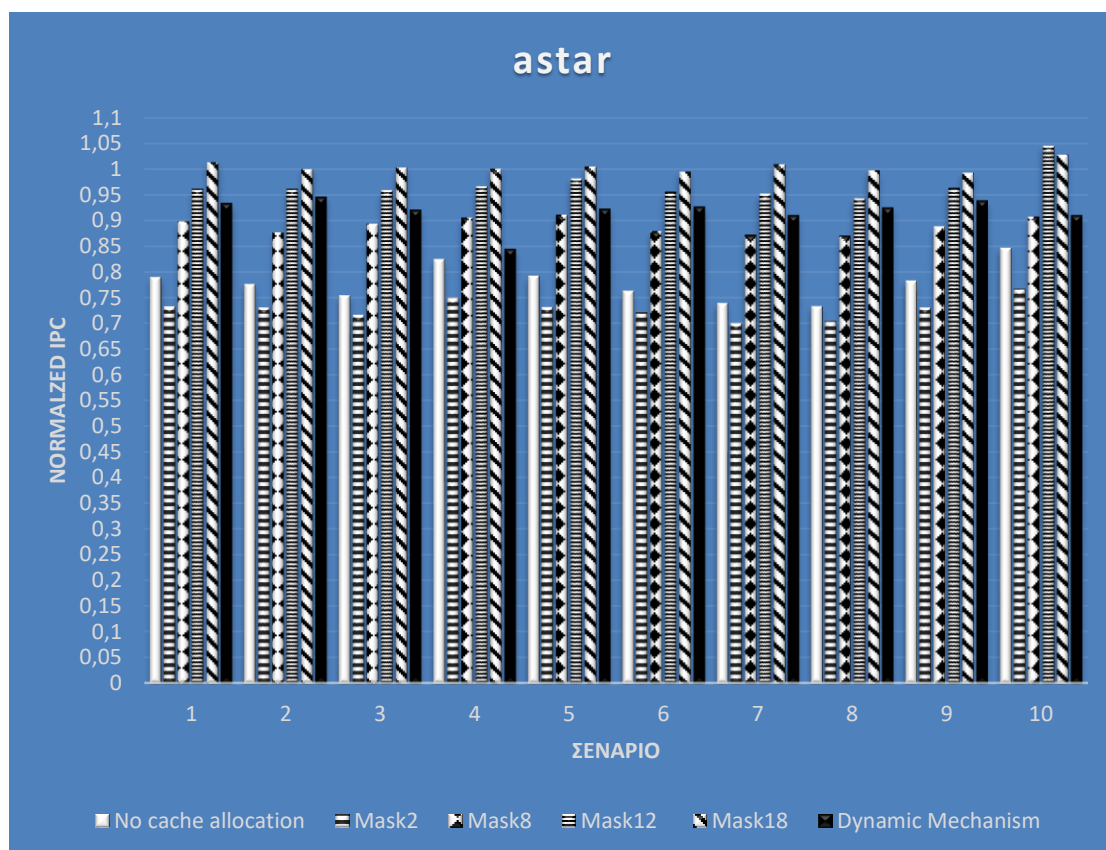
Εικόνα 5-1: Διάγραμμα ροής του μηχανισμού

5.2 Αξιολόγηση Μηχανισμού

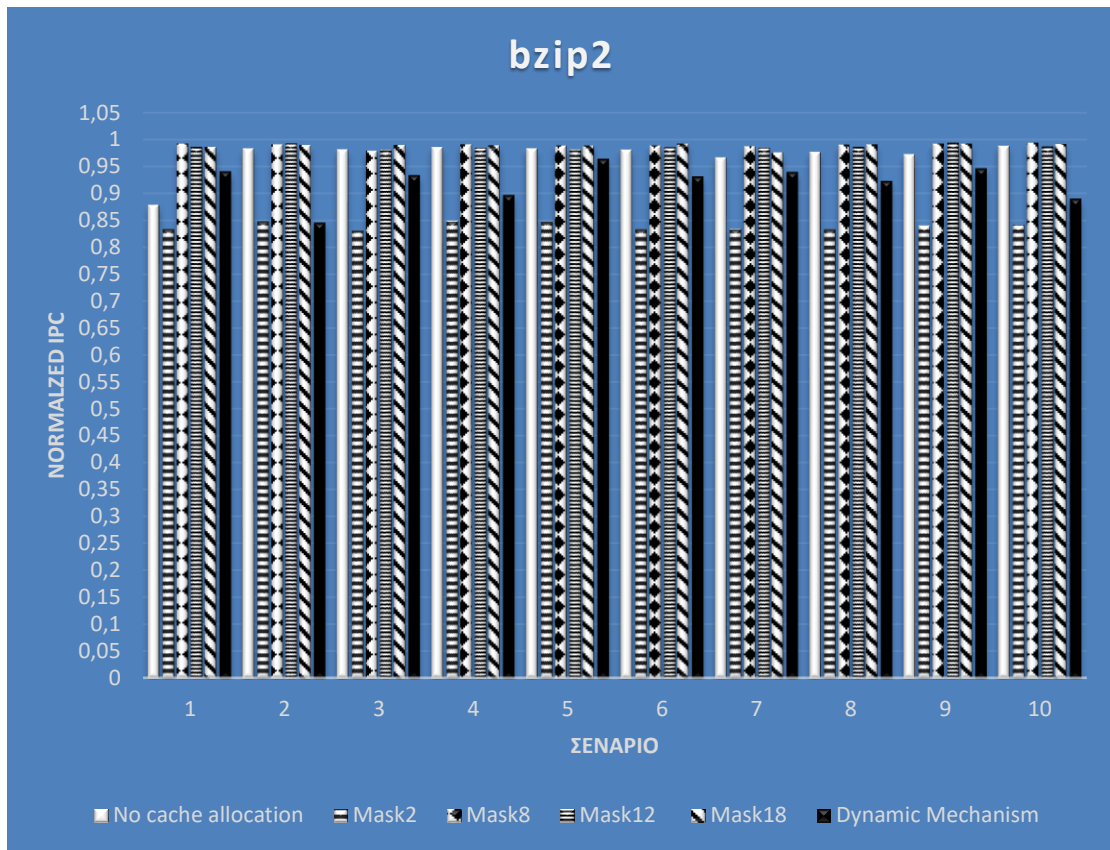
Στην παρούσα ενότητα παρουσιάζονται τα αποτελέσματα που εξαγάγαμε για την επίδοση της εφαρμογής με προτεραιότητα και το συνολικό κέρδος του συστήματος κατά την εφαρμογή του μηχανισμού που αναπτύξαμε.

5.2.1 Επίδοση εφαρμογών

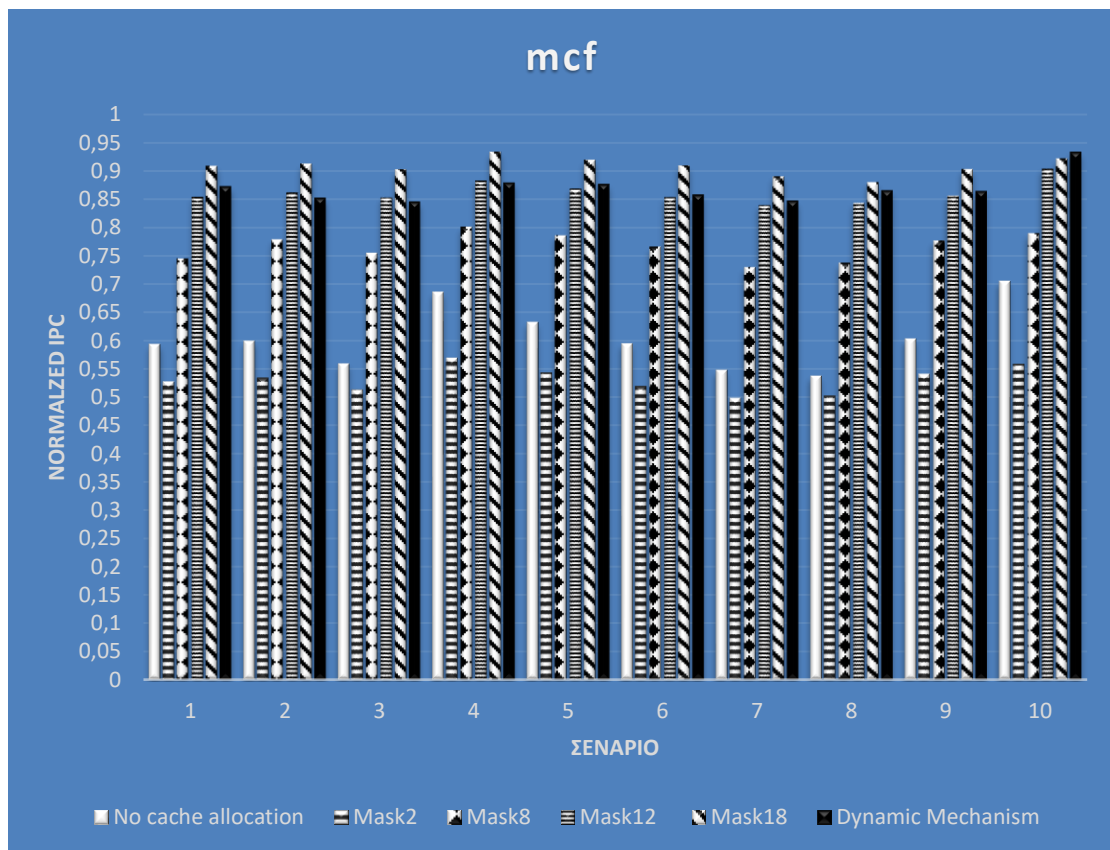
Εφαρμόζοντας τον μηχανισμό στα ίδια benchmarks και στο ίδιο σύστημα που περιεγράφηκαν στο κεφάλαιο 4, λήφθηκαν οι πιο κάτω μετρήσεις για το Normalized IPC της κάθε εφαρμογής. Στις γραφικές παραστάσεις που ακολουθούν αναπαρίστανται και οι τιμές των πειραματικών διαδικασιών της ενότητας 4.2. Οι πληροφορίες για όλα τα benchmarks βρίσκονται στην ενότητα 8.1 του παραρτήματος.



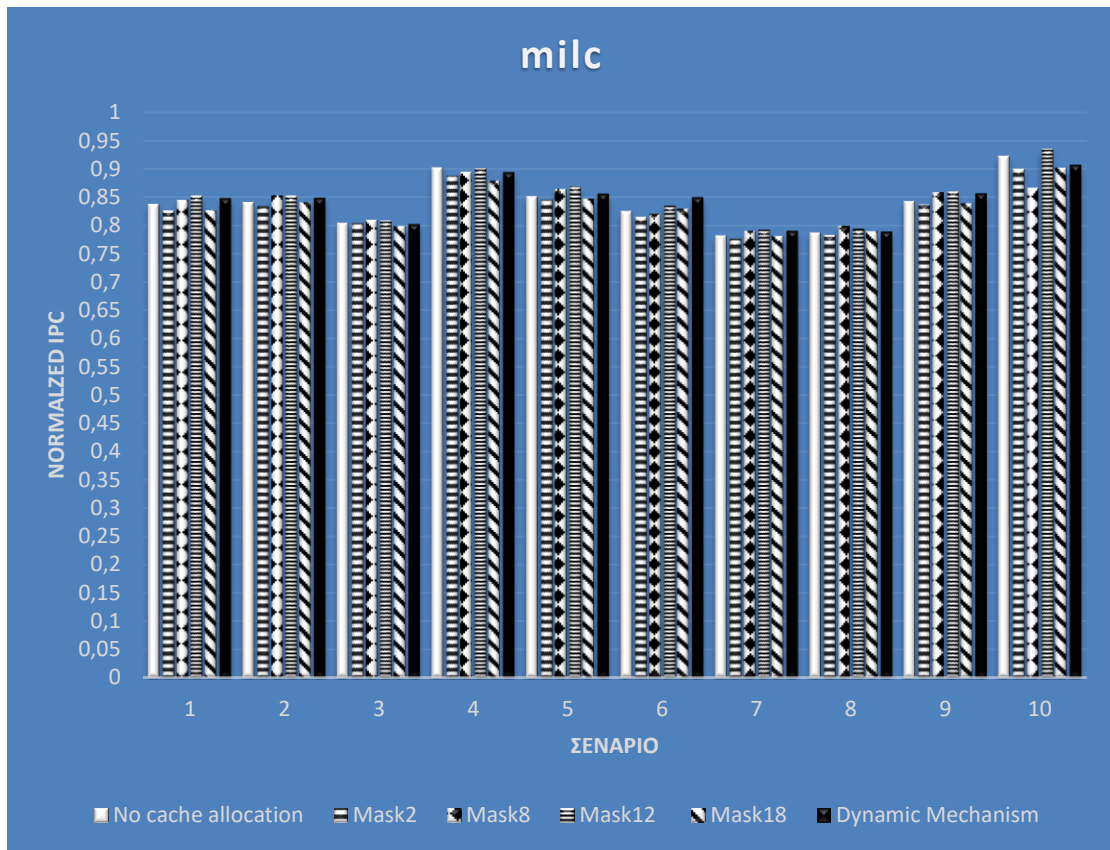
Γράφημα 5-1: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - astar



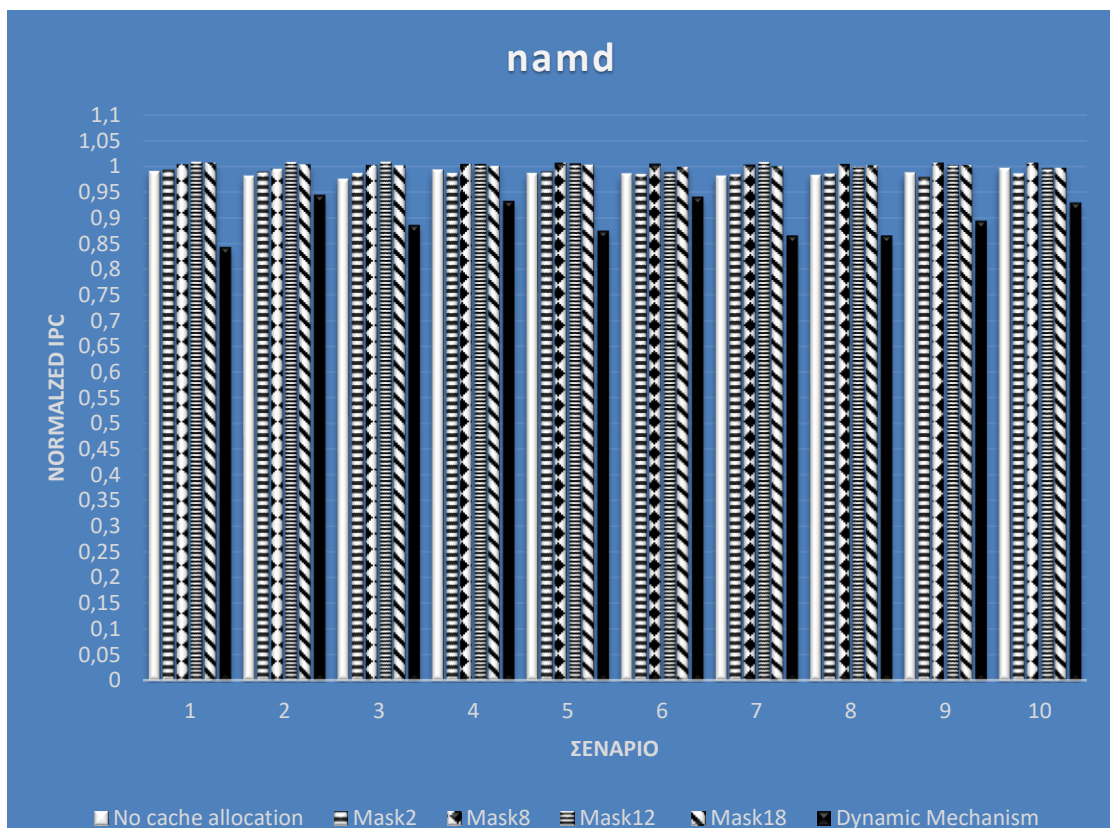
Γράφημα 5-2: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - bzip2



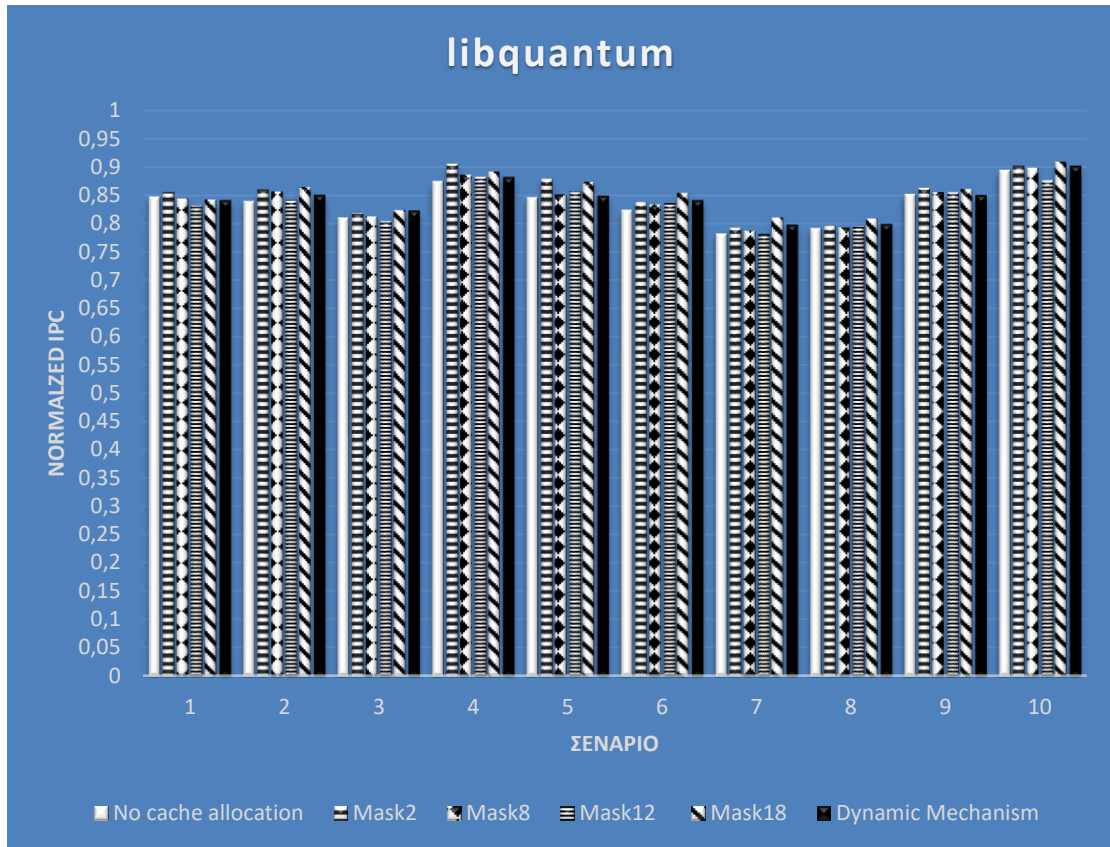
Γράφημα 5-3: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - mcf



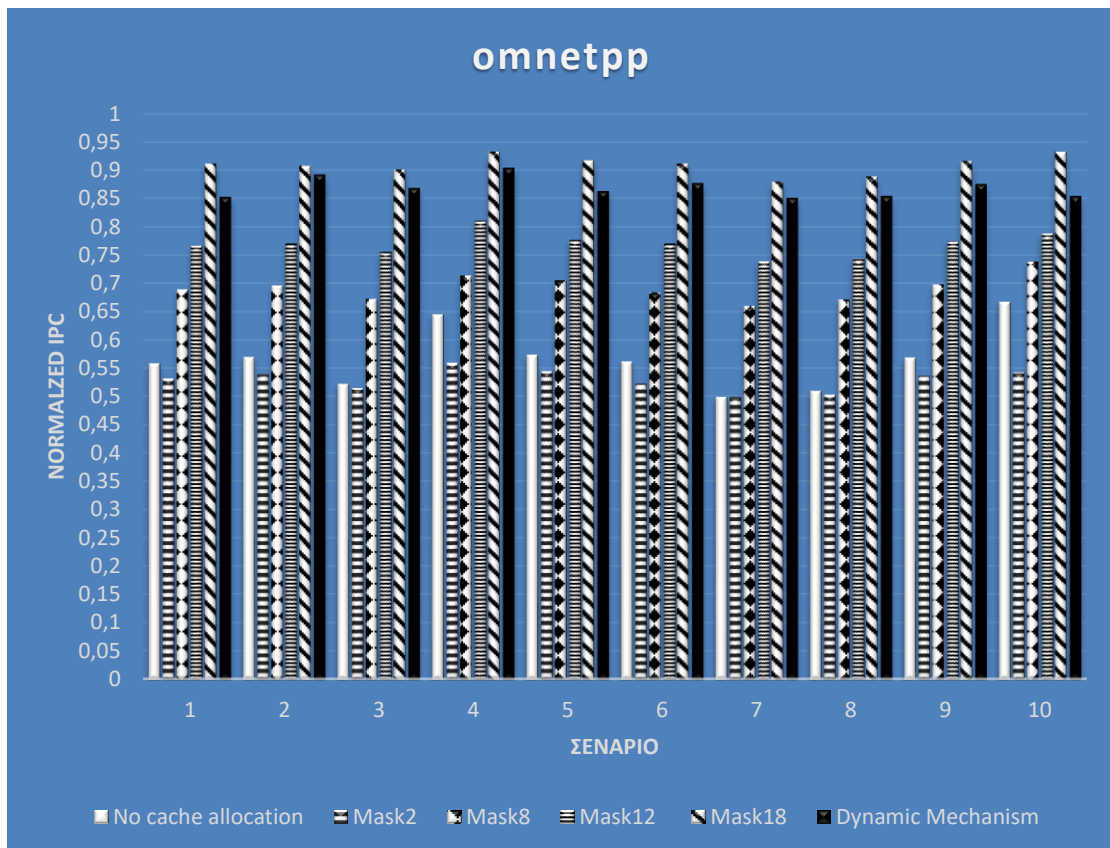
Γράφημα 5-4: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - milc



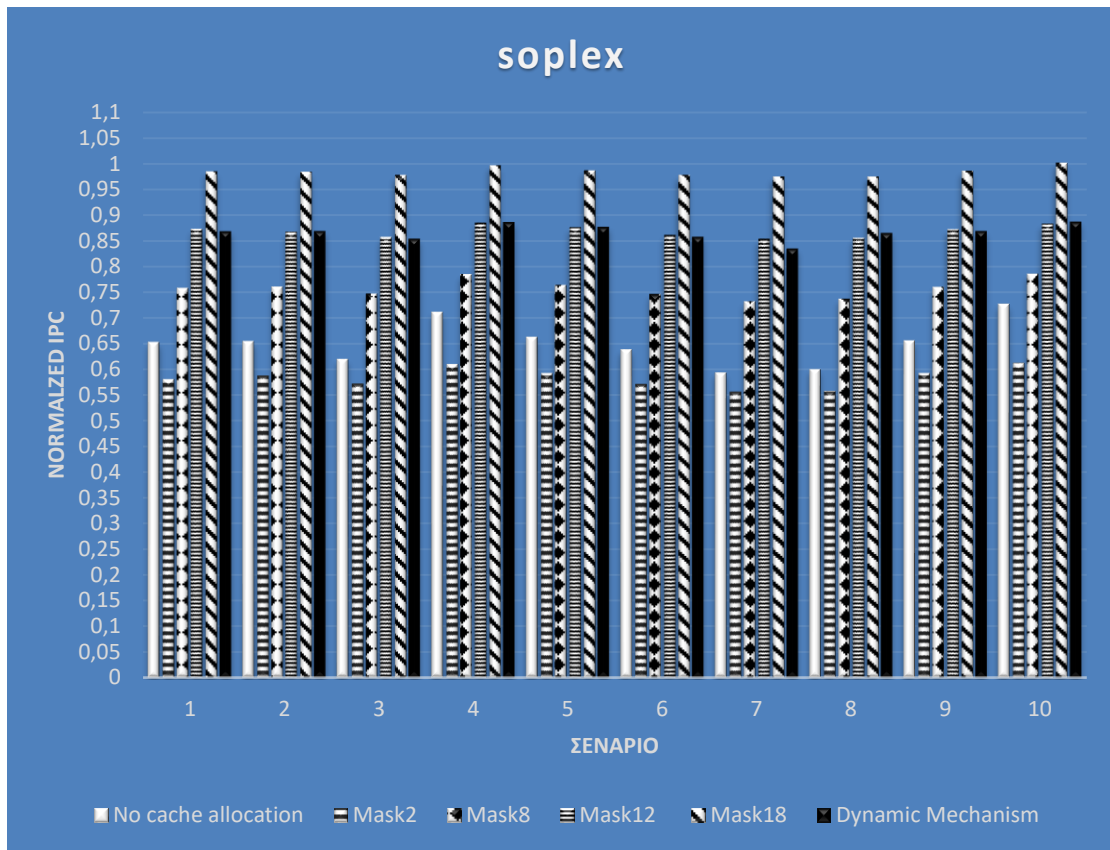
Γράφημα 5-5: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - namd



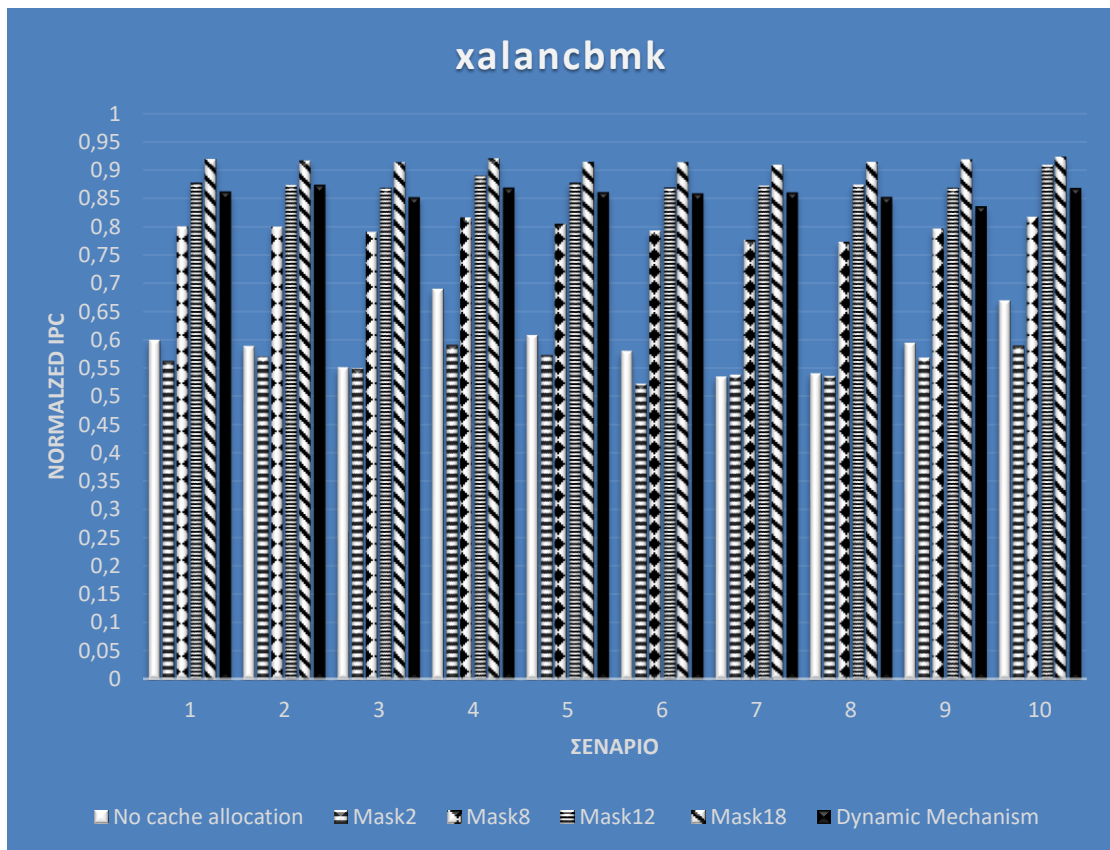
Γράφημα 5-6: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - libquantum



Γράφημα 5-7: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - omnetpp



Γράφημα 5-8: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - soplex

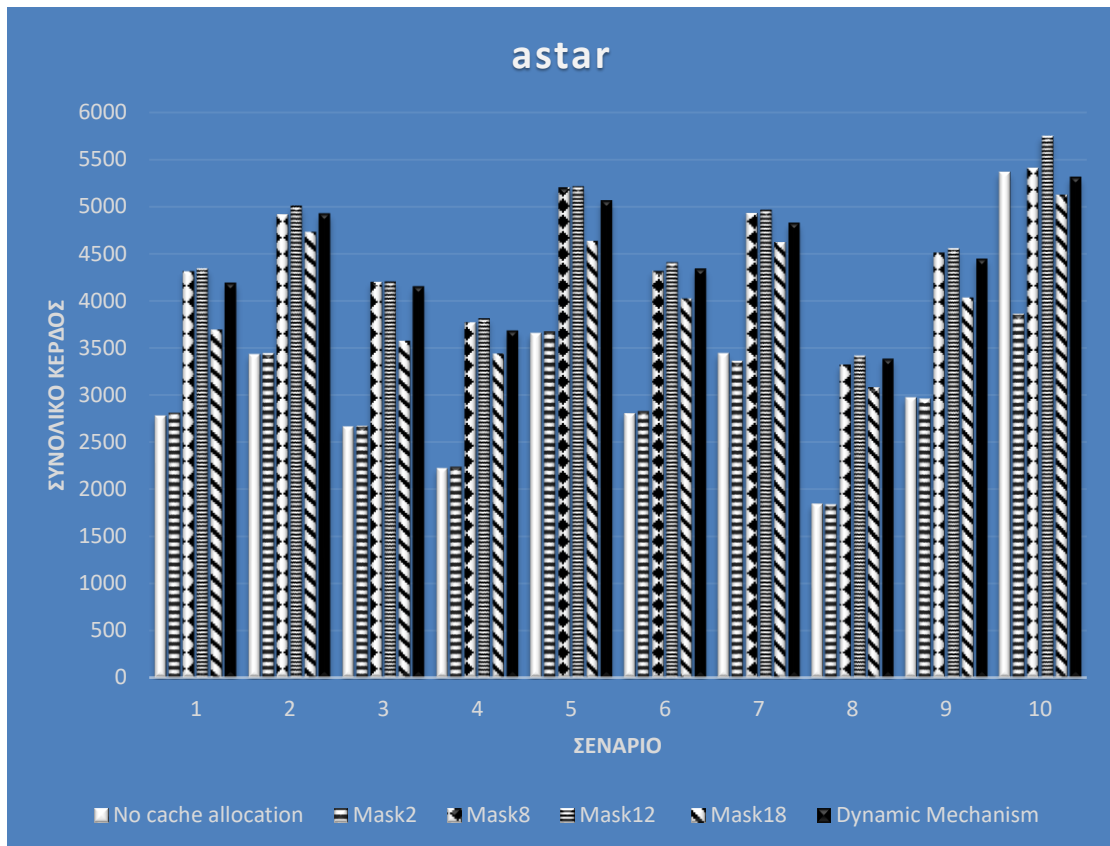


Γράφημα 5-9: Normalized IPC σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού - xalancbmk

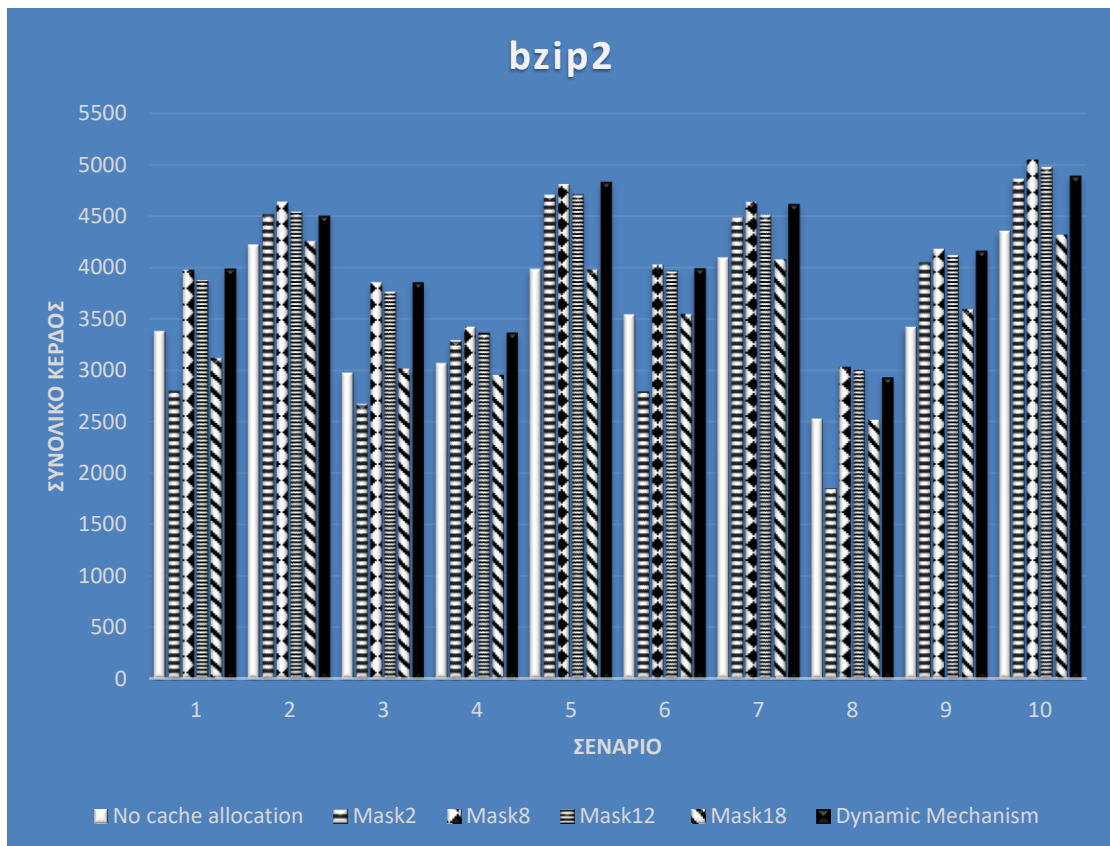
Από τα πιο πάνω γραφήματα φαίνεται πως ο μηχανισμός έχει την ικανότητα να ικανοποιεί τον ένα από τους δύο στόχους που θέσαμε. Δηλαδή είναι σε θέση να ικανοποιεί το SLO των εφαρμογών με προτεραιότητα αφού σε όλες τις περιπτώσεις το Normalized IPC είναι μεγαλύτερο του 0.84 και έτσι το Slowdown παραμένει πάντα μικρότερο από 1.20. Ο λόγος που σε πολλές περιπτώσεις το Normalized IPC της gold διεργασίας, κατά την εφαρμογή του δυναμικού μηχανισμού, είναι μικρότερο από τις άλλες περιπτώσεις κατανομής της LLC οφείλεται στο γεγονός πως ο μηχανισμός, στην περίπτωση άφιξης gold διεργασίας που δεν ανήκει στην κατηγορία Strict, ξεκινά χωρίς καμία κατανομή της LLC στο σύστημα. Έτσι μπορεί να δημιουργηθεί μια μικρή καθυστέρηση μέχρι ο μηχανισμός να φτάσει σε μια κατανομή της LLC που να ικανοποιεί το SLO της εφαρμογής με προτεραιότητα. Ως εκ τούτου το μέσο IPC μειώνεται σε ένα μικρό βαθμό. Παρόλα αυτά, όπως θα δούμε και στην επόμενη ενότητα, το γεγονός αυτό δεν φαίνεται να επηρεάζει το κέρδος που επιφέρουν οι εφαρμογές στο σύστημα.

5.2.2 Συνολικό κέρδος στο σύστημα

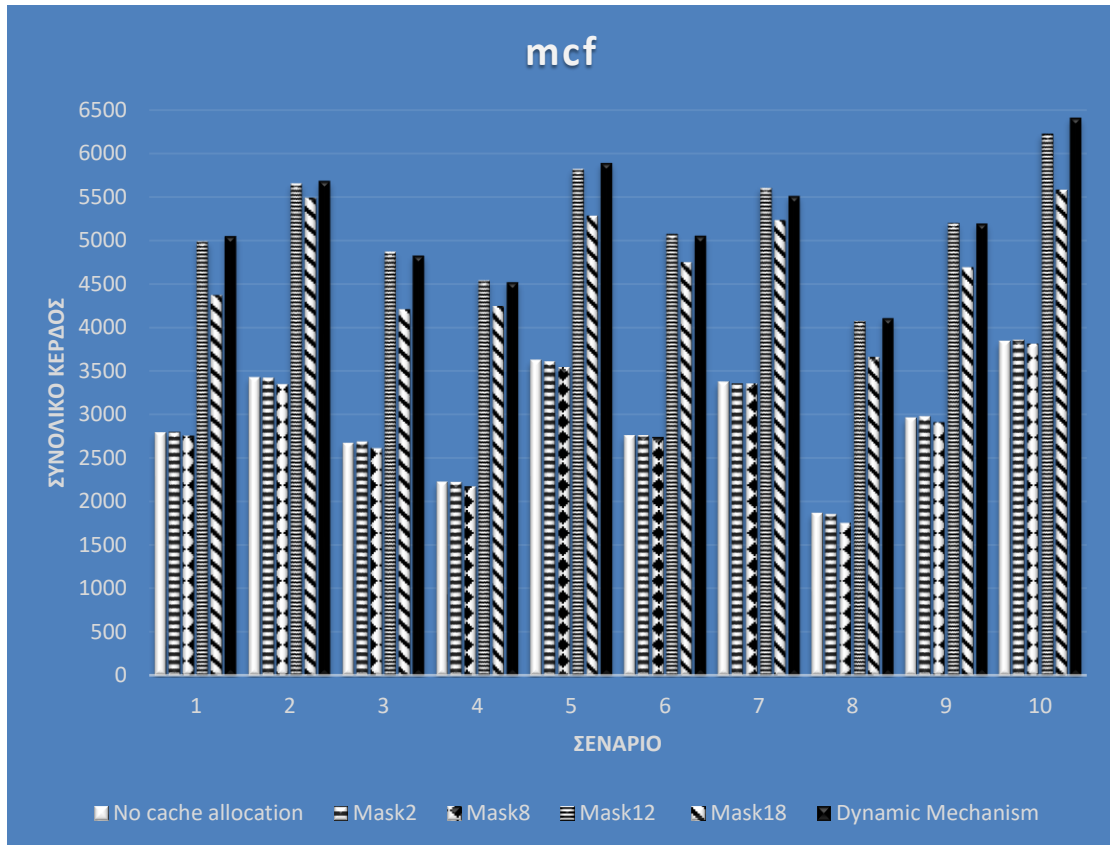
Για να εξετάσουμε το κατά πόσο ο μηχανισμός ικανοποιεί και τον δεύτερο μας στόχο, ο οποίος είναι η αύξηση του κέρδους στο σύστημα, πρέπει να υπολογίσουμε και συνολικό κέρδος που επιφέρουν όλες οι εφαρμογές σε κάθε σενάριο. Στη γραφικές παραστάσεις που ακολουθούν παρουσιάζεται το κέρδος που υπολογίστηκε τόσο κατά την εφαρμογή του μηχανισμού που υλοποιήσαμε, τόσο και των πειραματικών διαδικασιών της ενότητας 4.2 για σύγκριση των αποτελεσμάτων. Οι πληροφορίες για το συνολικό κέρδος που επιφέρουν όλα τα benchmarks βρίσκονται στην ενότητα 8.2 του παραρτήματος.



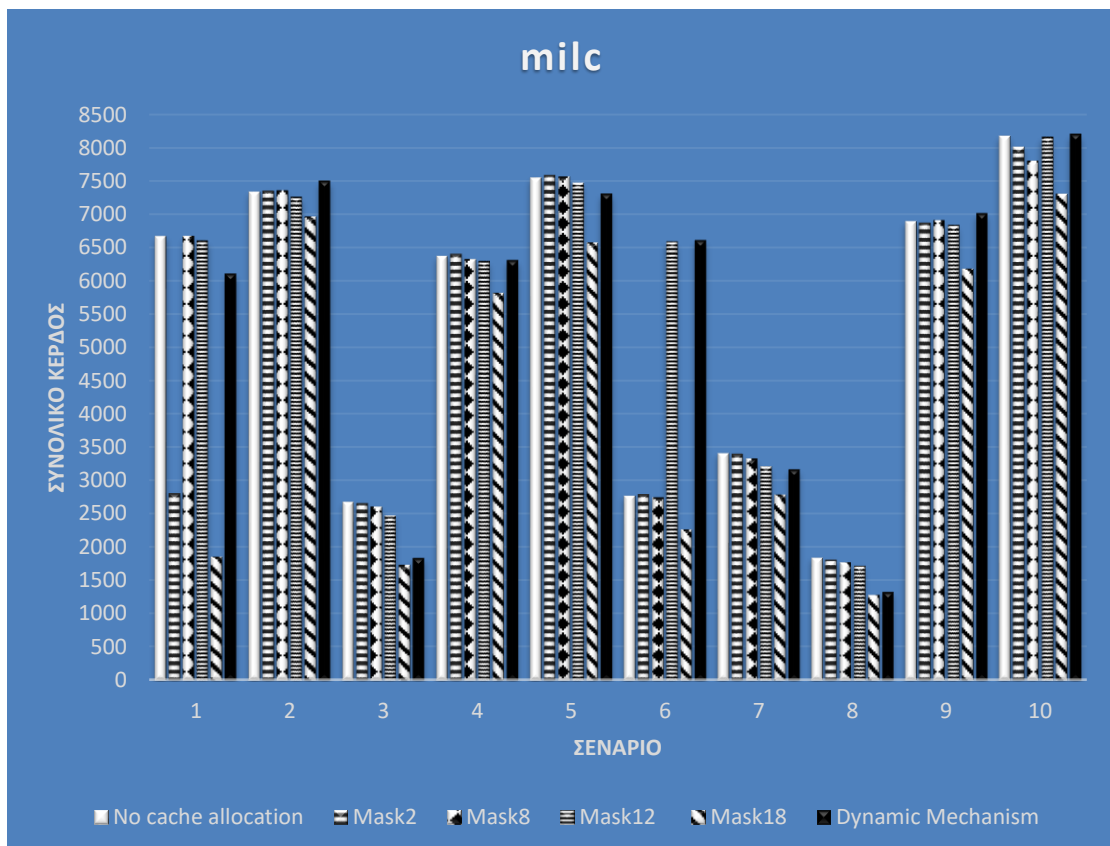
Γράφημα 5-10: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – astar ως gold



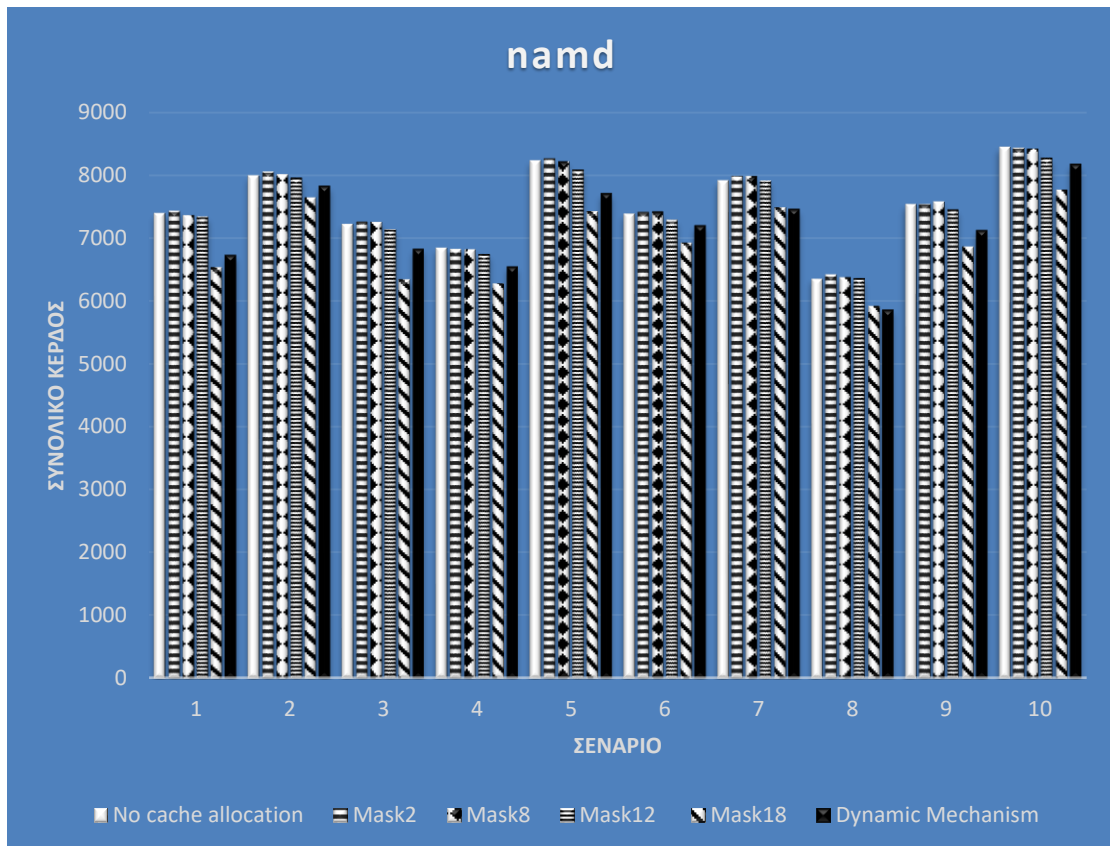
Γράφημα 5-11: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – bzip2 ως gold



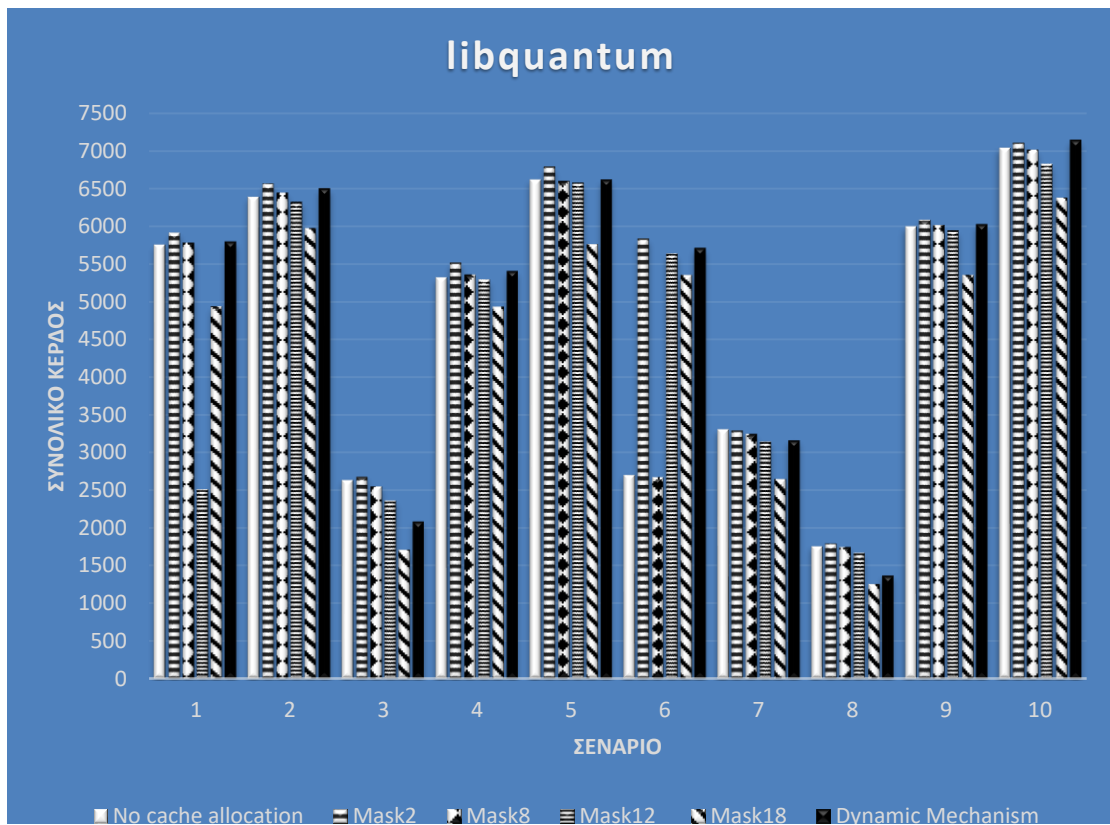
Γράφημα 5-12: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – mcf ως gold



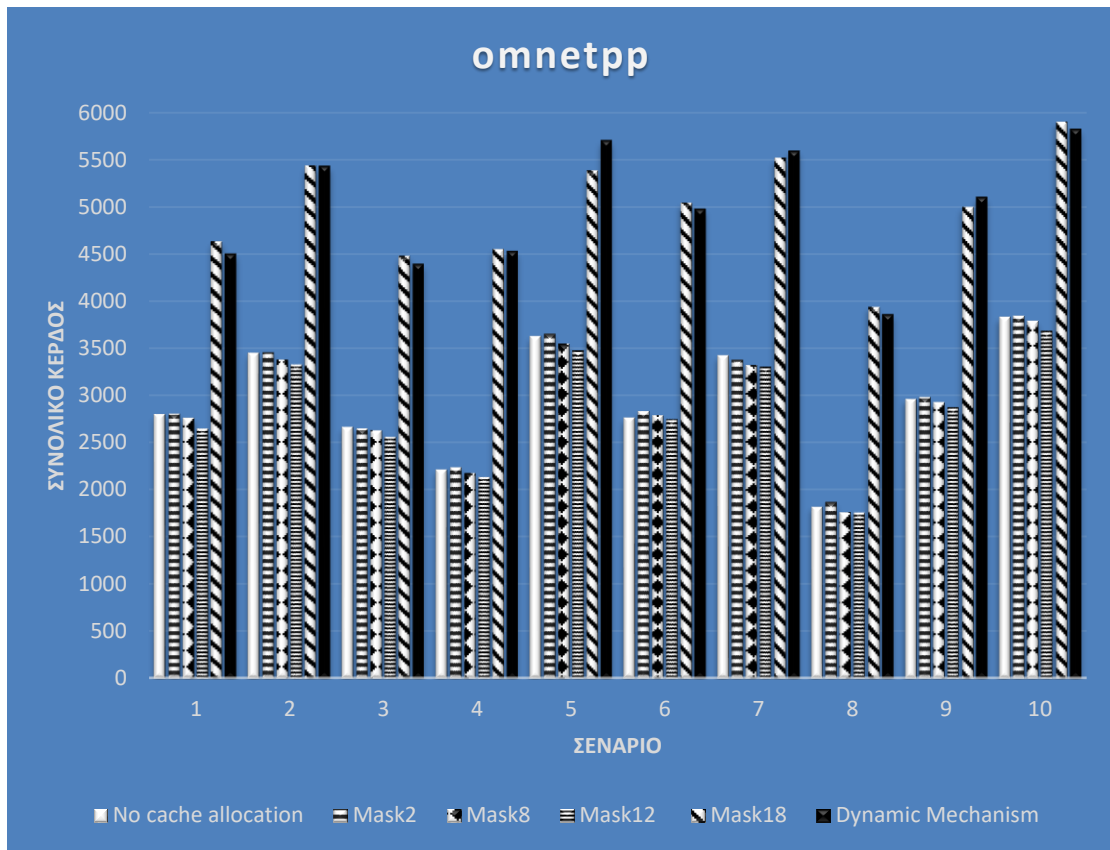
Γράφημα 5-13: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – milc ως gold



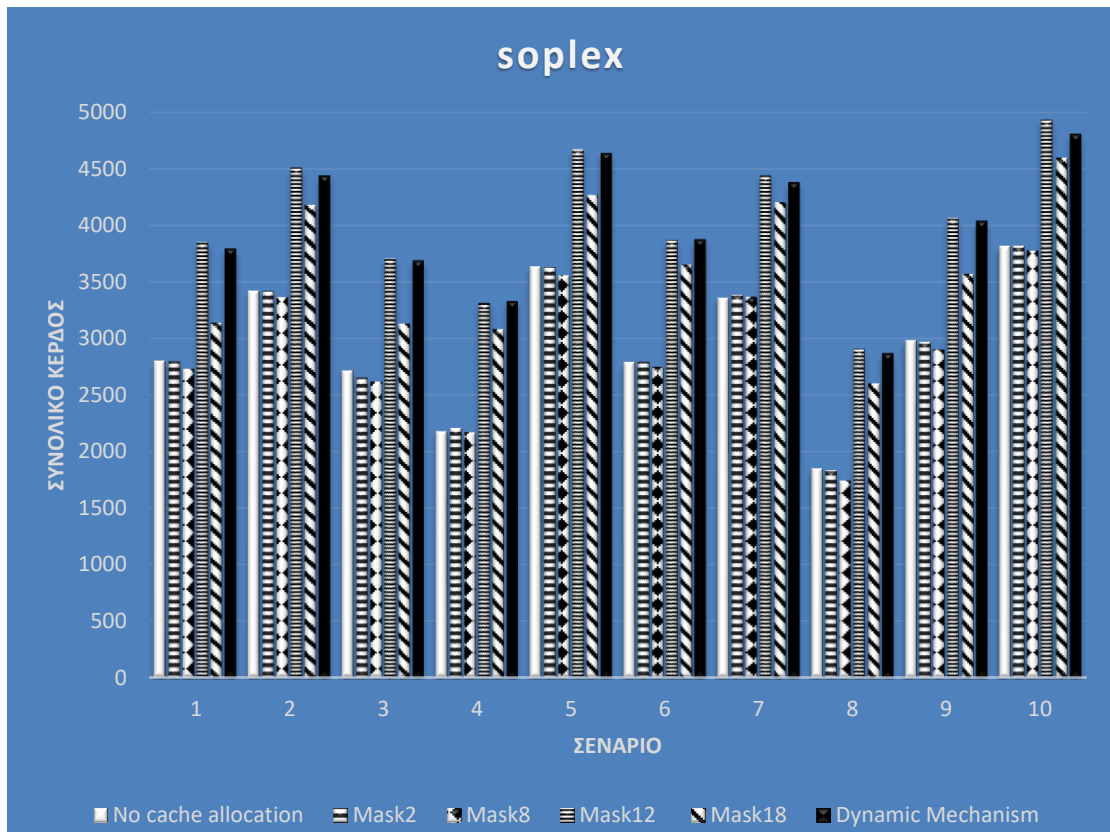
Γράφημα 5-14: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – namd ως gold



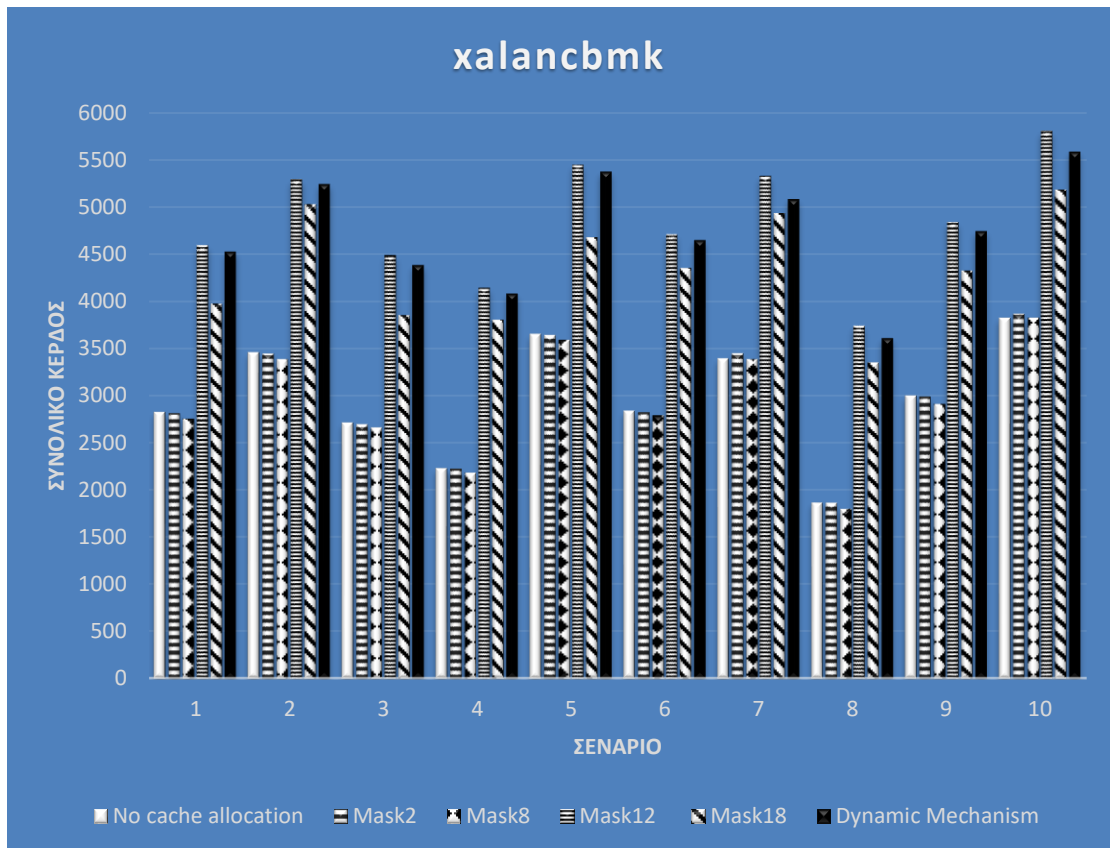
Γράφημα 5-15: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – libquantum ως gold



Γράφημα 5-16: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – omnetpp ως gold



Γράφημα 5-17: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – soplex ως gold



Γράφημα 5-18: Συνολικό κέρδος σε σενάρια συνεκτέλεσης με εφαρμογή του δυναμικού μηχανισμού – xalancbmk ως gold

Από τα αποτελέσματα που εξαγάγαμε και για το κέρδος του συστήματος, φαίνεται πως ο μηχανισμός σε αρκετές περιπτώσεις παράγει περισσότερο συνολικό κέρδος σε σχέση με τη μέγιστη μέτρηση που είχαμε υπολογίσει κατά την μελέτη των εφαρμογών στην ενότητα 4.2.4. Σε άλλες περιπτώσεις αποδίδει κέρδος αρκετά κοντά στα επίπεδα της μέγιστης μέτρησης των προηγούμενων πειραμάτων.

Ιδιαίτερο ενδιαφέρον παρουσιάζεται στη περίπτωση του omnetpp που αποτελεί μια εφαρμογή η οποία δεν ανήκει στην κατηγορία Strict αλλά εμφάνιζε δυσκολία στην διατήρηση του SLO κατά την στατική κατανομή της LLC. Στην εκτέλεση του omnetpp με τη βοήθεια του δυναμικού μηχανισμού, ενώ αρχικά ξεκινά χωρίς καμία μορφή κατανομής της LLC, καταφέρνει να κρατήσει το Slowdown στα επιθυμητά επίπεδα και συνεπώς να αποδώσει πολύ ικανοποιητικό κέρδος.

Όσο αφορά τις εφαρμογές milc και libquantum, δυστυχώς στα σενάρια 3,7, και 8 ούτε ο μηχανισμός που υλοποιήσαμε κατάφερε να διατηρήσει το Slowdown τους σε επίπεδα κάτω του 1.20 με αποτέλεσμα το κέρδος στο σύστημα σε αυτά τα σενάρια να προέρχεται αποκλειστικά από τις silver διεργασίες. Ο μηχανισμός είχε φτάσει να

κατανέμει τα 19 cache ways στην εφαρμογή με προτεραιότητα (milk/libqunatum) και το 1 εναπομένει cache way στις silver εφαρμογές. Ως αποτέλεσμα οι silver εφαρμογές λιμοκτονούσαν ενώ παράλληλα η gold εφαρμογή δεν τηρούσε το SLO αφού δεν υλοποιήθηκε κάποια άλλη τεχνική πρόβλεψης ή επίλυσης τέτοιων καταστάσεων. Το φαινόμενο αυτό μπορεί να οφείλεται στην απαιτητικότητα των silver εφαρμογών και για τους υπόλοιπους πόρους στο σύστημα που είναι κοινοί και προσβάσιμοι από όλες τις διεργασίες.

6

Επίλογος

6.1 Σύνοψη και Συμπεράσματα

Στη παρούσα διπλωματική εργασία είχαμε την ευκαιρία να μελετήσουμε την συμπεριφορά εφαρμογών από τη σουίτα SPEC CPU 2006 σε διάφορα σενάρια συνεκτέλεσης και κατανομής της LLC. Επίσης, καταφέραμε να αναπτύξουμε ένα δυναμικό μηχανισμό διατήρησης της επίδοσης της εφαρμογής με προτεραιότητα στα επίπεδα του SLO μέσω κατανομής της LLC.

Μέσω της χρήσης των CMT και CAT αρχικά αποδείξαμε πως κατά την παράλληλη εκτέλεση η επίδοση των εφαρμογών επηρεάζεται αρνητικά, με την LLC να αποτελεί μια πολύ σημαντική αιτία εμφάνισης ανταγωνισμού μεταξύ των εφαρμογών. Παράλληλα, κατά την μελέτη των εφαρμογών αποδείχθηκε πως η κατανομή των cache ways μεταξύ των πυρήνων μπορεί να αντιμετωπίσει σε μεγάλο βαθμό το φαινόμενο του ανταγωνισμού. Ακόμη, τα αποτελέσματα έδειξαν πως οι εφαρμογές με προτεραιότητα δεν αναζητούν πάντα μεγάλο μέρος της LLC προκειμένου να αποδίδουν στον μέγιστο δυνατό βαθμό. Αντιθέτως κατανέμοντας περισσότερα cache ways από αυτά που χρειάζεται η εκάστοτε εφαρμογή με προτεραιότητα, μπορεί να οδηγήσει σε αύξηση στον ανταγωνισμό μεταξύ όλων των διεργασιών για χρήση του διαύλου δεδομένων αλλά και των silver διεργασιών για τα λιγιστά cache ways που έχουν απομείνει, μειώνοντας έτσι και την επίδοσή τους και το συνολικό κέρδος στο σύστημα. Αφού μελετήσαμε τις εφαρμογές στα διάφορα σενάρια παράλληλης εκτέλεσης και σε διάφορες περιπτώσεις στατικής κατανομής της LLC, καταλήξαμε πως ένας δυναμικός μηχανισμός κατανομής της LLC θα ήταν πολύ πιο αποτελεσματικός. Έτσι, υλοποιήσαμε ένα μηχανισμό που με την βοήθεια του CMT επιβλέπει, κατά τον

πραγματικό χρόνο εκτέλεσης των εφαρμογών, την επίδοση της εφαρμογής με προτεραιότητα και μέσω του CAT κατανέμει τα cache ways ανάλογα με τις ανάγκες της εκάστοτε gold εφαρμογής. Τέλος, αποδείξαμε πως ο μηχανισμός μας σε κάθε σενάριο παράλληλης εκτέλεσης διατηρεί το SLO της εφαρμογής με προτεραιότητα στα επιθυμητά επίπεδα κατανέμοντάς της όσο το δυνατό λιγότερα cache ways και ταυτόχρονα αυξάνει το συνολικό κέρδος στο σύστημα.

6.2 Μελλοντικές Επεκτάσεις

Κλείνοντας, με βάση τα αποτελέσματα της παρούσας διπλωματικής, θα θέλαμε να εισηγηθούμε ορισμένες επεκτάσεις που θα μπορούσαν να υλοποιηθούν μελλοντικά.

Αρχικά ο μηχανισμός, που αναπτύχθηκε, θα μπορούσε να επεκταθεί με σκοπό να επιβλέπει περισσότερες από μια εφαρμογές με προτεραιότητα και να κατανέμει την cache με τέτοιο τρόπο ώστε όλες να τηρούν το SLO.

Επίσης, θα μπορούσαμε να εξετάσουμε την εφαρμογή του μηχανισμού στη περίπτωση μιας πολυνηματικής εφαρμογής με προτεραιότητα, κατανέμοντας διαφορετικά κομμάτια της cache σε κάθε νήμα.

Τέλος, σε κάποιες περιπτώσεις αποδείχθηκε πως ακόμη και η κατανομή των 19 από τα 20 cache ways στην εφαρμογή με προτεραιότητα δεν βελτιώνει αισθητά την επίδοσή της. Αυτό μπορεί να οφείλεται στον ανταγωνισμό μεταξύ των διεργασιών για άλλους πόρους που βρίσκονται στο σύστημα και είναι προσβάσιμοι από όλους τους πυρήνες. Έτσι μια άλλη επέκταση του μηχανισμού θα μπορούσε να αποσκοπεί στην πρόβλεψη αυτών των καταστάσεων κατανέμοντας και τους υπόλοιπους πόρους μεταξύ των εφαρμογών.

7

Βιβλιογραφία

- [1] D. Geer, “Industry trends: Chip makers turn to multicore processors,” *Computer (Long Beach, Calif.)*, vol. 38, no. 5, pp. 11–13, 2005, doi: 10.1109/MC.2005.160.
- [2] R. Hood *et al.*, “Performance impact of resource contention in multicore systems,” *Proc. 2010 IEEE Int. Symp. Parallel Distrib. Process. IPDPS 2010*, no. May 2014, 2010, doi: 10.1109/IPDPS.2010.5470399.
- [3] R. Nathuji, A. Kansal, and A. Ghaffarkhah, “Q-clouds: Managing performance interference effects for QoS-aware clouds,” *EuroSys’10 - Proc. EuroSys 2010 Conf.*, pp. 237–250, 2010, doi: 10.1145/1755913.1755938.
- [4] C. Stoffel, ““Paragon.”,” *Notes Queries*, vol. s9-III, no. 62, p. 168, 1899, doi: 10.1093/nq/s9-III.62.168-a.
- [5] C. Delimitrou and C. Kozyrakis, “Quasar : Resource-Efficient and QoS-Aware Cluster Management.”
- [6] S. Kim, D. Chandra, and Y. Solihin, “Fair cache sharing and partitioning in a chip multiprocessor architecture,” *Parallel Archit. Compil. Tech. - Conf. Proceedings, PACT*, pp. 111–122, 2004, doi: 10.1109/pact.2004.1342546.
- [7] K. Nikas, N. Papadopoulou, D. Giantsidi, V. Karakostas, G. Goumas, and N. Koziris, “DICER: Diligent cache partitioning for efficient workload consolidation,” *ACM Int. Conf. Proceeding Ser.*, 2019, doi: 10.1145/3337821.3337891.
- [8] H. Kasture and D. Sanchez, “Ubik: Efficient cache sharing with strict QoS for latency-critical workloads,” *Int. Conf. Archit. Support Program. Lang. Oper. Syst. - ASPLOS*, pp. 729–742, 2014, doi: 10.1145/2541940.2541944.
- [9] D. Sanchez and C. Kozyrakis, “Vantage: Scalable and efficient fine-grain cache partitioning,” *Proc. - Int. Symp. Comput. Archit.*, pp. 57–68, 2011, doi: 10.1145/2000064.2000073.
- [10] C. Xu, X. Chen, R. P. Dick, and Z. M. Mao, “Cache contention and application performance prediction for multi-core systems,” *ISPASS 2010 - IEEE Int. Symp. Perform. Anal. Syst. Softw.*, pp. 76–86, 2010, doi: 10.1109/ISPASS.2010.5452065.
- [11] S. Chen, C. Delimitrou, and J. F. Martinez, “PARTIES: QoS-Aware Resource Partitioning for Multiple Interactive Services,” *Int. Conf. Archit. Support Program. Lang. Oper. Syst. - ASPLOS*, pp. 107–120, 2019, doi: 10.1145/3297858.3304005.
- [12] N. El-Sayed, A. Mukkara, P. A. Tsai, H. Kasture, X. Ma, and D. Sanchez, “KPart: A Hybrid Cache Partitioning-Sharing Technique for Commodity Multicores,” *Proc. - Int. Symp. High-Performance Comput. Archit.*, vol. 2018-Febru, pp. 104–117, 2018, doi: 10.1109/HPCA.2018.00019.
- [13] L. Subramanian, V. Seshadri, A. Ghosh, S. Khan, and O. Mutlu, “The application slowdown model,” pp. 62–75, 2015, doi: 10.1145/2830772.2830803.
- [14] L. Zhao *et al.*, “Rhythm: Component-distinguishable workload deployment in

- datacenters,” *Proc. 15th Eur. Conf. Comput. Syst. EuroSys 2020*, 2020, doi: 10.1145/3342195.3387534.
- [15] Y. Sfakianakis, C. Kozanitis, C. Kozyrakis, and A. Bilas, “QuMan: Profile-based improvement of cluster utilization,” *ACM Trans. Archit. Code Optim.*, vol. 15, no. 3, 2018, doi: 10.1145/3210560.
- [16] R. Xu *et al.*, “Pythia: Improving datacenter utilization via precise contention prediction for multiple co-located workloads,” *Proc. 19th Int. Middlew. Conf. Middlew. 2018*, pp. 146–160, 2018, doi: 10.1145/3274808.3274820.
- [17] Q. Chen, H. Yang, M. Guo, R. S. Kannan, J. Mars, and L. Tang, “Prophet: Precise QoS prediction on non-preemptive accelerators to improve utilization in warehouse-scale computers,” *ACM SIGPLAN Not.*, vol. 52, no. 4, pp. 17–32, 2017, doi: 10.1145/3037697.3037700.
- [18] D. Novaković, N. Vasić, S. Novaković, D. Kostić, and R. Bianchini, “DeepDive: Transparently identifying and managing performance interference in virtualized environments,” *Proc. 2013 USENIX Annu. Tech. Conf. USENIX ATC 2013*, pp. 219–230, 2019.
- [19] J. Mars, L. Tang, R. Hundt, K. Skadron, and M. Lou Soffa, “Bubble-Up: Increasing utilization in modern warehouse scale computers via sensible co-locations,” *Proc. Annu. Int. Symp. Microarchitecture, MICRO*, pp. 248–259, 2011, doi: 10.1145/2155620.2155650.
- [20] D. Lo, L. Cheng, R. Govindaraju, P. Ranganathan, and C. Kozyrakis, “Improving resource efficiency at scale with Heracles,” *ACM Trans. Comput. Syst.*, vol. 34, no. 2, 2016, doi: 10.1145/2882783.
- [21] X. Zhang, E. Tune, R. Hagmann, R. Jnagal, V. Gokhale, and J. Wilkes, “CPI2: CPU performance isolation for shared compute clusters,” *Proc. 8th ACM Eur. Conf. Comput. Syst. EuroSys 2013*, pp. 379–391, 2013, doi: 10.1145/2465351.2465388.
- [22] Intel, “Intel 64 and IA-32 Architectures Software Developer’s Manual Combined Volumes,” *System*, vol. 3, no. 253665, 2011, doi: 10.1109/MAHC.2010.22.
- [23] Psomadakis, S., Gerangelos, S., Siakavaras, D., Papadakis, I., Vemmou, M., Skalidi, A., Karakostas, V., Nikas, K., Koziris, N., & Goumas, G. (2019). ACTiManager: An end-to-end interference-aware cloud resource manager. *Proceedings of the 20th International Middleware Conference Demos and Posters*.
- [24] A. Fedorova, M. Seltzer, and M. D. Smith, “Improving performance isolation on chip multiprocessors via an operating system scheduler,” *Parallel Archit. Compil. Tech. - Conf. Proceedings, PACT*, pp. 25–36, 2007, doi: 10.1109/PACT.2007.18.

8

Παράρτημα

8.1 Normalized IPC

astar – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.79	0.73	0.90	0.96	1.01	0.93
2	0.78	0.73	0.88	0.96	1.00	0.94
3	0.75	0.71	0.89	0.96	1.00	0.92
4	0.82	0.75	0.90	0.96	1.00	0.84
5	0.79	0.73	0.91	0.98	1.00	0.92
6	0.76	0.72	0.88	0.95	0.99	0.93
7	0.74	0.70	0.87	0.95	1.01	0.91
8	0.73	0.70	0.87	0.94	1.00	0.92
9	0.78	0.73	0.89	0.96	0.99	0.94
10	0.85	0.77	0.91	1.04	1.03	0.91

Πίνακας 8-1: Normalized IPC – astar

bzip2 – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.88	0.83	0.99	0.98	0.98	0.94
2	0.98	0.85	0.99	0.99	0.99	0.84
3	0.98	0.83	0.98	0.98	0.99	0.93
4	0.98	0.85	0.99	0.98	0.99	0.90
5	0.98	0.85	0.99	0.98	0.99	0.96
6	0.98	0.83	0.99	0.98	0.99	0.93
7	0.97	0.83	0.99	0.98	0.97	0.94
8	0.98	0.83	0.99	0.98	0.99	0.92
9	0.97	0.84	0.99	0.99	0.99	0.94
10	0.99	0.84	0.99	0.99	0.99	0.89

Πίνακας 8-2: Normalized IPC - bzip2

bwaves – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.95	0.95	0.97	0.97	0.98	0.97
2	0.96	0.95	0.97	0.98	0.98	0.95
3	0.95	0.94	0.97	0.97	0.97	0.95
4	0.98	0.96	0.99	0.99	0.99	0.91
5	0.96	0.95	0.98	0.99	0.98	0.97
6	0.95	0.95	0.97	0.98	0.98	0.95
7	0.95	0.93	0.96	0.96	0.96	0.96
8	0.94	0.93	0.96	0.96	0.97	0.96
9	0.95	0.94	0.98	0.98	0.99	0.96
10	0.96	0.96	0.99	0.99	0.99	0.98

Πίνακας 8-3: Normalized IPC - bwaves

gamess – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.97	0.98	1.01	1.01	1.01	0.86
2	0.99	0.98	1.01	1.01	1.01	0.92
3	0.98	0.98	1.01	1.01	1.01	0.88
4	0.99	0.97	1.01	1.01	1.00	0.91
5	0.98	0.99	1.01	1.01	1.00	0.86
6	0.98	0.99	1.01	1.01	1.00	0.88
7	0.96	0.94	1.01	1.01	1.01	0.86
8	0.96	0.98	1.01	1.01	1.01	0.90
9	0.99	0.97	1.01	1.00	1.00	0.87
10	1.00	0.99	1.01	1.01	1.01	0.85

Πίνακας 8-4: Normalized IPC - gamess

mcf – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.59	0.53	0.74	0.85	0.91	0.87
2	0.60	0.53	0.78	0.86	0.91	0.85
3	0.56	0.51	0.75	0.85	0.90	0.84
4	0.69	0.57	0.80	0.88	0.93	0.88
5	0.63	0.54	0.78	0.87	0.92	0.88
6	0.59	0.52	0.76	0.85	0.91	0.86
7	0.55	0.50	0.73	0.84	0.89	0.84
8	0.54	0.50	0.74	0.84	0.88	0.86
9	0.60	0.54	0.78	0.86	0.90	0.86

Πίνακας 8-5: Normalized IPC - mcf

milc – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.84	0.83	0.84	0.85	0.83	0.84
2	0.84	0.83	0.85	0.85	0.84	0.85
3	0.80	0.80	0.81	0.81	0.80	0.80
4	0.90	0.89	0.89	0.90	0.88	0.89
5	0.85	0.85	0.86	0.87	0.85	0.85
6	0.83	0.82	0.82	0.83	0.83	0.84
7	0.78	0.78	0.79	0.79	0.78	0.79
8	0.79	0.78	0.80	0.79	0.79	0.79
9	0.84	0.84	0.86	0.86	0.84	0.86

Πίνακας 8-6: Normalized IPC - milc

cactusADM – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.75	0.74	0.96	0.99	0.98	0.91
2	0.75	0.75	0.97	0.99	0.98	0.91
3	0.73	0.73	0.95	0.98	0.97	0.94
4	0.79	0.77	0.98	1.00	1.00	0.87
5	0.79	0.75	0.96	0.99	0.99	0.89
6	0.74	0.74	0.97	0.98	0.99	0.96
7	0.71	0.72	0.96	0.97	0.97	0.93
8	0.71	0.73	0.95	0.97	0.97	0.90
9	0.75	0.75	0.98	0.99	0.98	0.95

Πίνακας 8-7: Normalized IPC - cactusADM

leslie3d – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.89	0.83	0.93	0.95	0.97	0.94
2	0.90	0.83	0.93	0.95	0.98	0.91
3	0.87	0.82	0.92	0.94	0.97	0.88
4	0.94	0.86	0.95	0.97	0.99	0.94
5	0.91	0.84	0.93	0.96	0.98	0.92
6	0.89	0.82	0.92	0.95	0.98	0.90
7	0.84	0.80	0.90	0.93	0.96	0.85
8	0.85	0.81	0.91	0.93	0.96	0.86
9	0.90	0.84	0.93	0.96	0.98	0.91

Πίνακας 8-8: Normalized IPC - leslie3d

namd – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.99	0.99	1.00	1.01	1.01	0.84
2	0.98	0.99	0.99	1.01	1.00	0.94
3	0.97	0.99	1.00	1.01	1.00	0.89
4	0.99	0.99	1.00	1.00	1.00	0.93
5	0.99	0.99	1.01	1.00	1.00	0.87
6	0.99	0.98	1.00	0.99	1.00	0.94
7	0.98	0.98	1.00	1.01	1.00	0.87
8	0.98	0.98	1.00	1.00	1.00	0.86
9	0.99	0.98	1.01	1.00	1.00	0.89

Πίνακας 8-9: Normalized IPC - namd

gobmk – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.91	0.95	0.99	0.99	1.00	0.95
2	0.92	0.95	0.99	1.00	1.01	0.92
3	0.89	0.94	0.99	1.00	1.00	0.90
4	0.94	0.95	0.99	0.99	1.01	0.94
5	0.92	0.95	0.99	0.99	1.01	0.92
6	0.91	0.94	0.99	0.99	1.00	0.91
7	0.89	0.94	0.99	0.99	1.00	0.89
8	0.88	0.94	0.99	0.99	1.00	0.89
9	0.92	0.95	0.99	1.00	1.00	0.92

Πίνακας 8-10: Normalized IPC - gobmk

povray – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.95	0.97	1.00	0.99	1.00	0.85
2	0.96	0.96	1.00	0.99	1.00	0.85
3	0.95	0.97	1.00	1.00	1.00	0.92
4	0.98	0.97	1.00	0.99	0.99	0.86
5	0.96	0.98	1.00	0.99	1.00	0.88
6	0.95	0.95	1.00	1.00	1.00	0.86
7	0.94	0.97	0.99	1.00	1.00	0.85
8	0.94	0.95	1.00	1.00	1.00	0.92
9	0.94	0.95	1.00	1.00	0.99	0.84

Πίνακας 8-11: Normalized IPC - povray

calculix – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.97	1.00	1.02	1.02	1.02	0.98
2	0.98	0.99	1.02	1.02	1.02	0.94
3	0.97	0.99	1.02	1.02	1.02	0.94
4	0.99	1.00	1.02	1.02	1.02	0.96
5	0.98	1.00	1.01	1.03	1.02	0.96
6	0.97	0.99	1.02	1.02	1.01	0.94
7	0.97	0.99	1.01	1.02	1.02	0.97
8	0.96	0.99	1.02	1.02	1.02	0.95
9	0.98	0.99	1.02	1.02	1.03	0.95

Πίνακας 8-12: Normalized IPC - calculix

GemsFDTD – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.93	0.86	0.93	0.94	0.93	0.93
2	0.94	0.87	0.94	0.94	0.93	0.93
3	0.91	0.86	0.92	0.92	0.93	0.92
4	0.97	0.91	0.97	0.98	0.97	0.95
5	0.95	0.89	0.95	0.95	0.95	0.94
6	0.93	0.86	0.93	0.93	0.94	0.92
7	0.89	0.82	0.89	0.89	0.89	0.89
8	0.90	0.84	0.91	0.91	0.90	0.90
9	0.94	0.87	0.93	0.93	0.93	0.94

Πίνακας 8-13: Normalized IPC - GemsFDTD

libquantum – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.85	0.85	0.84	0.83	0.84	0.84
2	0.84	0.86	0.86	0.84	0.86	0.85
3	0.81	0.82	0.81	0.80	0.82	0.82
4	0.87	0.90	0.88	0.88	0.89	0.88
5	0.85	0.88	0.85	0.85	0.87	0.85
6	0.82	0.84	0.83	0.83	0.85	0.84
7	0.78	0.79	0.79	0.78	0.81	0.80
8	0.79	0.80	0.79	0.79	0.81	0.80
9	0.85	0.86	0.85	0.85	0.86	0.85

Πίνακας 8-14: Normalized IPC - libquantum

tonto – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.96	0.96	1.01	1.01	1.02	0.99
2	0.96	0.96	1.01	1.01	1.01	0.85
3	0.94	0.96	1.00	1.02	1.01	0.90
4	0.98	0.96	1.01	1.01	1.01	0.87
5	0.96	0.96	1.01	1.01	1.02	0.98
6	0.95	0.95	1.01	1.01	1.01	0.84
7	0.93	0.95	1.01	1.01	1.01	0.96
8	0.95	0.96	1.01	1.01	1.02	0.95
9	0.96	0.97	1.01	1.00	1.02	0.88

Πίνακας 8-15: Normalized IPC - tonto

omnetpp – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.56	0.53	0.69	0.77	0.91	0.85
2	0.57	0.54	0.70	0.77	0.91	0.89
3	0.52	0.51	0.67	0.76	0.90	0.87
4	0.65	0.56	0.71	0.81	0.93	0.90
5	0.57	0.54	0.71	0.78	0.92	0.86
6	0.56	0.52	0.68	0.77	0.91	0.88
7	0.50	0.50	0.66	0.74	0.88	0.85
8	0.51	0.50	0.67	0.74	0.89	0.85
9	0.57	0.53	0.70	0.77	0.91	0.87

Πίνακας 8-16: Normalized IPC - omnetpp

Ibm – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.85	0.74	0.95	0.96	0.94	0.91
2	0.86	0.75	0.95	0.96	0.95	0.88
3	0.80	0.72	0.93	0.94	0.93	0.88
4	0.95	0.77	0.98	0.98	0.97	0.95
5	0.88	0.76	0.96	0.97	0.96	0.90
6	0.85	0.74	0.94	0.95	0.94	0.85
7	0.76	0.71	0.91	0.91	0.92	0.88
8	0.77	0.71	0.92	0.92	0.92	0.90
9	0.87	0.75	0.95	0.96	0.95	0.89

Πίνακας 8-17: Normalized IPC - Ibm

soplex – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.65	0.58	0.76	0.87	0.98	0.87
2	0.65	0.59	0.76	0.87	0.98	0.87
3	0.62	0.57	0.75	0.86	0.98	0.85
4	0.71	0.61	0.78	0.88	1.00	0.89
5	0.66	0.59	0.76	0.87	0.99	0.88
6	0.64	0.57	0.75	0.86	0.98	0.86
7	0.59	0.56	0.73	0.85	0.97	0.83
8	0.60	0.56	0.74	0.85	0.97	0.86
9	0.66	0.59	0.76	0.87	0.98	0.87

Πίνακας 8-18: Normalized IPC - soplex

gcc – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.95	0.63	0.85	0.95	1.02	0.86
2	0.95	0.63	0.85	0.95	1.02	0.88
3	0.93	0.62	0.83	0.94	1.01	0.87
4	0.97	0.65	0.88	0.96	1.02	0.90
5	0.95	0.64	0.84	0.95	1.01	0.86
6	0.94	0.62	0.84	0.94	1.00	0.87
7	0.93	0.60	0.83	0.93	1.01	0.86
8	0.94	0.60	0.82	0.93	1.00	0.86
9	0.95	0.63	0.84	0.95	1.01	0.86

Πίνακας 8-19: Normalized IPC - gcc

zeusmp – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.93	0.92	0.96	0.98	0.98	0.94
2	0.93	0.92	0.97	0.98	0.97	0.93
3	0.91	0.90	0.95	0.96	0.96	0.91
4	0.96	0.94	0.99	1.00	1.00	0.97
5	0.94	0.93	0.98	0.98	0.99	0.94
6	0.92	0.91	0.95	0.97	0.97	0.92
7	0.89	0.89	0.94	0.95	0.95	0.89
8	0.89	0.89	0.93	0.94	0.95	0.90
9	0.93	0.91	0.97	0.97	0.97	0.93

Πίνακας 8-20: Normalized IPC - zeusmp

perlbench – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.83	0.84	0.97	1.00	1.00	0.94
2	0.84	0.84	0.97	1.00	1.00	0.94
3	0.82	0.83	0.97	1.00	1.00	0.96
4	0.86	0.85	0.98	1.00	1.00	0.94
5	0.84	0.84	0.97	0.99	1.00	0.94
6	0.83	0.83	0.97	1.00	0.99	0.93
7	0.81	0.82	0.96	0.99	0.99	0.96
8	0.81	0.82	0.97	1.00	0.98	0.95
9	0.83	0.82	0.97	0.99	1.00	0.97

Πίνακας 8-21: Normalized IPC - perlbench

gromacs – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.96	0.94	0.99	0.99	0.99	0.86
2	0.96	0.95	0.99	0.99	0.99	0.96
3	0.95	0.94	0.98	0.99	0.99	0.95
4	0.97	0.95	0.99	0.99	0.98	0.97
5	0.97	0.94	0.99	1.00	0.98	0.96
6	0.96	0.95	0.99	0.99	0.98	0.96
7	0.95	0.94	1.00	0.99	0.99	0.94
8	0.95	0.94	0.98	0.99	0.99	0.94
9	0.96	0.94	0.99	0.98	0.98	0.95

Πίνακας 8-22: Normalized IPC - gromacs

dealII – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.96	0.91	0.98	0.99	0.99	0.95
2	0.96	0.92	0.98	0.98	0.99	0.95
3	0.95	0.91	0.97	0.98	0.98	0.93
4	0.98	0.94	0.99	0.99	0.99	0.95
5	0.96	0.92	0.99	0.99	0.99	0.94
6	0.95	0.92	0.99	0.99	0.99	0.93
7	0.94	0.91	0.98	0.98	0.98	0.93
8	0.94	0.91	0.98	0.98	0.98	0.93
9	0.96	0.92	0.99	0.99	0.98	0.93

Πίνακας 8-23: Normalized IPC - dealII

hmmmer – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	1.00	0.98	1.01	1.00	1.00	0.86
2	1.00	0.97	1.01	1.01	1.01	0.93
3	0.99	0.98	1.00	1.01	1.00	0.92
4	0.99	0.98	1.00	1.01	1.01	0.92
5	1.00	0.98	1.00	1.01	1.01	0.86
6	1.00	0.98	1.01	1.01	1.01	0.93
7	1.00	0.97	1.00	1.01	1.01	0.87
8	1.01	0.96	1.01	1.01	1.01	0.91
9	1.01	0.97	1.01	1.01	1.01	0.90

Πίνακας 8-24: Normalized IPC - hmmmer

sjeng – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.91	0.96	0.98	0.99	1.00	0.97
2	0.90	0.95	0.98	0.98	0.99	0.83
3	0.87	0.95	0.97	0.98	0.98	0.92
4	0.93	0.95	0.98	0.98	0.99	0.94
5	0.92	0.96	0.98	0.99	0.99	0.96
6	0.89	0.95	0.97	0.98	0.98	0.93
7	0.86	0.95	0.97	0.97	0.97	0.91
8	0.88	0.95	0.98	0.97	0.98	0.90
9	0.91	0.94	0.99	0.99	0.99	0.94

Πίνακας 8-25: Normalized IPC - sjeng

h264ref – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.93	0.93	1.00	1.01	1.01	0.88
2	0.93	0.93	1.01	1.00	1.01	0.93
3	0.92	0.94	1.01	1.01	1.00	0.89
4	0.96	0.94	1.00	1.01	1.01	0.84
5	0.95	0.95	1.01	1.00	1.01	0.89
6	0.93	0.94	1.00	1.00	1.01	0.87
7	0.88	0.94	1.01	1.00	1.01	0.88
8	0.92	0.94	1.01	1.00	1.01	0.91
9	0.93	0.94	1.00	1.00	1.01	0.91

Πίνακας 8-26: Normalized IPC - h264ref

sphinx3 – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.85	0.71	0.97	1.00	0.99	0.96
2	0.83	0.70	0.97	1.00	0.99	0.94
3	0.77	0.69	0.98	1.00	0.99	0.97
4	0.92	0.72	0.97	1.00	0.99	0.93
5	0.88	0.72	0.98	0.99	0.98	0.95
6	0.80	0.70	0.97	1.00	0.99	0.94
7	0.74	0.67	0.97	1.00	0.99	0.92
8	0.73	0.69	0.97	1.00	0.98	0.91
9	0.83	0.71	0.98	1.00	0.99	0.97

Πίνακας 8-27: Normalized IPC - sphinx3

xalancbmk – Normalized IPC						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	0.60	0.56	0.80	0.88	0.92	0.86
2	0.59	0.57	0.80	0.87	0.92	0.87
3	0.55	0.55	0.79	0.87	0.91	0.85
4	0.69	0.59	0.81	0.89	0.92	0.87
5	0.61	0.57	0.80	0.88	0.91	0.86
6	0.58	0.52	0.79	0.87	0.91	0.86
7	0.53	0.54	0.78	0.87	0.91	0.86
8	0.54	0.53	0.77	0.87	0.91	0.85
9	0.59	0.57	0.79	0.87	0.92	0.83

Πίνακας 8-28: Normalized IPC - xalancbmk

8.2 Συνολικό Κέρδος Συστήματος

astar – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	2778.36	2809.80	4311.17	4339.74	3690.81	4186.56
2	3430.54	3443.68	4918.59	5004.54	4727.49	4922.00
3	2663.46	2670.58	4200.38	4201.13	3572.43	4149.25
4	2222.29	2235.04	3772.64	3810.13	3435.72	3681.48
5	3656.18	3673.93	5202.91	5208.62	4633.33	5059.11
6	2803.82	2828.33	4314.89	4403.93	4019.88	4337.85
7	3441.27	3361.82	4932.47	4961.54	4619.69	4820.65
8	1846.05	1837.93	3324.47	3416.74	3078.65	3380.91
9	2974.96	2959.65	4509.69	4553.85	4030.36	4441.96
10	5365.58	3859.26	5408.64	5747.64	5122.04	5309.52

Πίνακας 8-29: Συνολικό Κέρδος – astar ως gold

bzip2 – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	3381.58	2795.88	3970.74	3875.79	3116.46	3985.47
2	4222.78	4513.58	4632.95	4543.12	4254.53	4498.73
3	2977.57	2668.06	3853.17	3764.06	3015.02	3853.52
4	3070.47	3287.68	3418.79	3365.76	2950.85	3361.91
5	3985.92	4702.28	4805.99	4710.68	3977.46	4830.99
6	3543.59	2789.01	4024.85	3966.55	3543.70	3989.79
7	4097.94	4483.17	4631.68	4512.11	4077.43	4613.90
8	2527.41	1845.81	3026.18	3000.25	2515.46	2926.48
9	3423.56	4045.99	4175.28	4121.35	3590.90	4161.70
10	4356.76	4858.79	5041.17	4978.85	4316.30	4888.54

Πίνακας 8-30: Συνολικό Κέρδος – bzip2 ως gold

bwaves – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	6715.67	6770.04	6821.20	6697.24	5950.30	6830.23
2	7431.58	7410.77	7441.00	7392.22	7061.60	7382.05
3	6671.73	6625.70	6654.21	6431.38	5801.70	6648.85
4	6283.04	6271.03	6295.01	6223.38	5918.86	6019.00
5	7618.86	7613.06	7606.26	7586.29	6732.23	7680.48
6	6735.76	6764.28	6811.43	6718.56	6369.54	6764.59
7	7378.32	7299.15	7354.99	7177.22	6775.43	7382.95
8	5742.98	5713.46	5806.36	5750.40	5367.37	5808.05
9	6909.60	6938.37	7006.98	6959.55	6315.87	6963.05
10	7774.92	7899.14	7954.54	7887.44	7405.96	7942.03

Πίνακας 8-31: Συνολικό Κέρδος – bwaves ως gold

gamess – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	3427.87	3439.90	3390.68	3235.48	2478.28	3338.91
2	4068.66	4054.77	4017.85	3936.07	3411.29	4015.26
3	3295.89	3310.10	3280.95	3160.82	2408.01	3268.40
4	2866.50	2852.31	2805.82	2752.04	2348.82	2806.69
5	4284.86	4269.78	4199.59	4117.27	3360.14	4204.66
6	3430.08	3409.33	3399.59	3363.74	2920.93	3394.71
7	4016.99	3975.16	3953.48	3862.73	3486.24	3989.11
8	2460.10	2464.87	2425.89	2333.31	1918.06	2408.39
9	3602.11	3595.92	3553.18	3480.90	2969.83	3523.80
10	4475.82	4423.14	4442.62	4360.05	3886.15	4410.20

Πίνακας 8-32: Συνολικό Κέρδος – games ως gold

mcf – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	2791.63	2793.23	2753.81	4989.18	4369.66	5051.56
2	3423.79	3417.45	3347.29	5654.11	5489.73	5683.42
3	2667.78	2682.49	2608.52	4870.47	4206.96	4824.91
4	2224.36	2221.14	2171.07	4541.19	4246.56	4521.54
5	3622.85	3604.04	3541.22	5819.10	5284.17	5890.39
6	2757.04	2759.11	2740.93	5075.05	4749.82	5053.83
7	3374.18	3351.62	3354.71	5602.07	5230.74	5513.02
8	1866.00	1851.48	1751.74	4072.36	3659.27	4108.28
9	2960.28	2976.01	2907.50	5196.95	4691.39	5194.33
10	3841.10	3849.24	3809.50	6224.23	5581.04	6410.21

Πίνακας 8-33: Συνολικό Κέρδος – mcf ως gold

mile – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	6679.41	2814.76	6668.64	6600.88	1854.21	6113.08
2	7344.99	7358.29	7353.31	7257.29	6961.24	7499.96
3	2685.58	2664.35	2609.63	2475.39	1734.16	1838.17
4	6380.38	6409.30	6322.90	6292.38	5815.78	6313.81
5	7554.65	7595.16	7559.27	7470.03	6573.85	7311.39
6	2778.58	2796.82	2748.47	6588.74	2265.29	6612.45
7	3419.02	3406.25	3334.39	3214.58	2789.69	3171.71
8	1843.76	1811.62	1771.91	1720.77	1282.31	1331.95
9	6898.82	6877.18	6905.92	6833.77	6179.02	7017.74
10	8182.35	8019.27	7797.70	8155.88	7305.83	8203.19

Πίνακας 8-34: Συνολικό Κέρδος – mile ως gold

cactusADM – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	2818.19	2821.16	9130.11	9315.12	8463.50	8811.74
2	3467.01	3449.34	9953.65	10010.66	9662.27	9575.03
3	2680.32	2663.77	9100.48	9139.10	8383.82	8863.61
4	2221.15	2245.14	8814.87	8869.97	8444.61	8109.03
5	3662.69	3634.17	10084.39	10159.82	9313.50	9518.89
6	2830.04	2833.37	9404.65	9379.88	8941.63	9220.68
7	3449.98	3406.79	9960.87	9769.09	9333.53	9587.96
8	1819.88	1814.55	8253.18	8338.95	7845.95	7858.16
9	2991.20	2984.24	9606.67	9512.76	8907.38	9329.61
10	3823.35	3844.37	10939.73	10440.68	9510.22	10288.68

Πίνακας 8-35: Συνολικό Κέρδος – cactusADM ως gold

leslie3d – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	5754.30	2808.83	5872.50	5783.64	5084.46	5684.99
2	6411.96	3560.54	6481.11	6479.82	6215.38	6403.16
3	5603.57	2670.76	5663.16	5667.07	4960.83	5577.88
4	5332.68	5107.47	5326.58	5346.99	5016.79	5283.91
5	6659.57	6439.34	6662.32	6648.93	5920.99	6590.10
6	5687.62	2731.74	5807.25	5830.60	5498.30	5665.44
7	6214.63	3325.38	6247.99	6378.68	6018.32	6132.98
8	4624.23	1801.44	4746.10	4826.68	4465.77	4649.09
9	5957.31	5777.82	6013.39	6050.59	5505.42	5927.45
10	6971.72	6709.83	6958.72	6932.34	6407.00	6872.85

Πίνακας 8-36: Συνολικό Κέρδος – leslie3d ως gold

namd – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	7399.48	7422.94	7362.77	7335.14	6530.89	6726.14
2	7996.66	8047.32	8013.75	7955.75	7639.78	7819.56
3	7225.09	7252.67	7254.20	7130.97	6336.47	6822.77
4	6845.83	6815.45	6822.25	6741.49	6274.31	6544.24
5	8234.02	8254.17	8217.80	8083.37	7421.14	7705.30
6	7387.96	7404.53	7422.93	7284.09	6921.41	7195.53
7	7920.34	7976.76	7986.32	7909.72	7481.30	7452.85
8	6352.78	6416.40	6380.65	6360.64	5914.98	5858.81
9	7540.41	7530.11	7584.54	7448.93	6856.56	7118.28
10	8447.83	8423.89	8415.90	8271.73	7759.24	8167.76

Πίνακας 8-37: Συνολικό Κέρδος – namd ως gold

gobmk – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	3573.37	3618.20	3549.90	3409.34	2703.89	3600.46
2	4203.22	4253.67	4218.42	4116.93	3847.95	4240.94
3	3474.75	3494.15	3461.02	3334.14	2568.89	3467.38
4	3013.02	3025.03	2985.41	2940.16	2542.45	3039.83
5	4445.39	4451.29	4414.22	4327.76	3499.07	4428.79
6	3588.63	3577.81	3574.77	3560.74	3090.45	3614.65
7	4112.16	4225.76	4191.68	4117.06	3647.65	4188.93
8	2612.39	2664.44	2586.33	2570.13	2092.67	2610.92
9	3758.94	3777.35	3771.47	3657.52	3064.02	3744.18
10	4663.85	4651.28	4629.25	4601.04	4072.81	4634.52

Πίνακας 8-38: Συνολικό Κέρδος – gobmk ως gold

ronray – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	4607.04	4614.50	4571.40	4467.15	3694.01	4403.32
2	5232.01	5252.82	5249.46	5120.02	4678.40	5048.82
3	4468.23	4463.31	4432.46	4375.96	3579.39	4413.65
4	4044.64	4021.16	4026.28	3955.85	3553.46	3846.30
5	5420.60	5455.05	5416.44	5329.90	4716.89	5271.25
6	4622.46	4554.39	4599.10	4559.22	4126.62	4411.34
7	5164.61	5232.14	5198.26	5123.73	4596.99	5023.01
8	3605.81	3614.65	3647.03	3580.24	3146.32	3563.13
9	4761.77	4758.61	4785.68	4697.70	4074.85	4549.56
10	5586.14	5635.09	5615.41	5566.43	4902.33	5507.48

Πίνακας 8-39: Συνολικό Κέρδος – ronray ως gold

calculix – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	11753.28	11988.92	12113.28	12077.29	11269.24	11758.58
2	12442.81	12537.77	12761.44	12683.17	12384.72	12048.63
3	11602.08	11789.51	12032.95	11895.08	11082.01	11316.03
4	11375.11	11406.95	11550.09	11489.23	11060.87	10980.66
5	12731.65	12798.86	12901.67	12870.40	11997.45	12391.07
6	11789.42	11936.79	12128.57	12094.11	11576.97	11443.84
7	12393.08	12558.27	12696.57	12672.08	12190.59	12297.82
8	10695.91	10957.19	11161.69	11093.83	10641.56	10491.83
9	12008.46	12132.90	12285.47	12214.60	11631.96	11739.65
10	13016.49	12989.39	13218.54	12979.71	12529.91	12869.25

Πίνακας 8-40: Συνολικό Κέρδος – calculix ως gold

GemsFDTD – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	6422.35	6168.44	6370.99	6213.42	5490.25	6366.76
2	7056.72	6853.82	7028.16	6953.43	6651.51	7033.80
3	6229.70	6023.44	6166.70	6082.76	5329.49	6181.70
4	5983.60	5799.83	5946.93	5954.72	5633.54	5930.82
5	7311.57	7108.30	7289.98	7142.76	6305.50	7263.64
6	6335.25	6114.72	6356.86	6329.51	5909.90	6367.50
7	6832.49	3382.48	6732.91	6622.99	6240.27	6771.16
8	5329.15	5121.11	5265.41	5319.95	4761.25	5292.67
9	6633.29	6321.92	6581.72	6420.30	5778.66	6599.52
10	7778.73	7516.06	7566.92	7334.01	6991.67	7515.80

Πίνακας 8-41: Συνολικό Κέρδος – GemsFDTD ως gold

libquantum – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	5759.46	5905.15	5769.54	2505.00	4934.86	5795.28
2	6389.91	6557.26	6434.27	6314.31	5971.65	6501.08
3	2634.39	2675.38	2541.93	2356.81	1706.29	2082.49
4	5323.21	5511.06	5350.64	5284.66	4928.22	5407.00
5	6619.64	6783.27	6589.94	6571.35	5756.47	6615.51
6	2704.96	5828.82	2666.99	5632.15	5349.08	5710.58
7	3311.51	3288.47	3240.09	3135.47	2643.93	3161.78
8	1756.82	1791.17	1738.81	1663.30	1252.42	1363.94
9	6001.72	6078.75	6005.76	5936.89	5348.52	6027.97
10	7042.26	7095.38	7001.29	6819.26	6372.81	7143.29

Πίνακας 8-42: Συνολικό Κέρδος – libquantum ως gold

tonto – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	8910.27	8996.83	9157.34	9064.09	8377.37	9121.63
2	9558.72	9554.94	9783.37	9757.98	9311.83	8877.26
3	8724.60	8766.44	8993.27	8957.08	8261.92	8445.54
4	8475.15	8389.73	8606.39	8511.12	8119.56	7757.23
5	9797.90	9789.68	9996.90	9914.98	9288.71	9926.18
6	8871.61	8913.63	9177.60	9157.68	8706.57	8175.71
7	9371.01	9474.43	9750.18	9709.69	9206.20	9485.28
8	7887.84	7937.05	8183.78	8154.51	7755.95	7922.66
9	9136.75	9136.51	9355.32	9170.82	8686.59	8581.26
10	10135.63	10043.23	10199.03	10339.64	9592.77	10114.58

Πίνακας 8-43: Συνολικό Κέρδος – tonto ως gold

omnetpp – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	2797.89	2802.40	2754.05	2639.88	4629.62	4500.97
2	3449.21	3450.84	3372.07	3316.11	5432.24	5431.63
3	2662.74	2647.85	2624.40	2550.98	4477.79	4393.74
4	2211.62	2233.80	2172.44	2127.42	4548.32	4528.62
5	3626.05	3649.21	3539.44	3465.63	5381.95	5705.66
6	2763.64	2828.94	2781.79	2738.93	5037.87	4979.37
7	3421.54	3374.56	3313.22	3291.06	5514.73	5593.79
8	1819.04	1868.17	1757.82	1752.86	3938.59	3858.19
9	2960.20	2978.50	2920.59	2862.95	4993.59	5104.81
10	3829.80	3837.82	3780.75	3673.81	5895.77	5824.79

Πίνακας 8-44: Συνολικό Κέρδος – omnetpp ως gold

Ibm – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	6248.88	6993.63	2659.84	6130.25	7260.49	6289.95
2	2787.02	3383.56	2683.28	2182.85	3586.69	2719.14
3	6682.73	7314.66	6410.65	6214.17	7536.55	6651.23
4	6577.03	7234.85	6412.84	6147.28	7485.89	6623.66
5	5769.40	6747.31	5645.21	5738.29	6587.45	6183.82
6	6418.76	6997.54	6238.17	6120.10	7277.99	6233.98
7	6248.88	6993.63	2659.84	6130.25	7260.49	6289.95
8	2787.02	3383.56	2683.28	2182.85	3586.69	2719.14
9	6682.73	7314.66	6410.65	6214.17	7536.55	6651.23
10	6577.03	7234.85	6412.84	6147.28	7485.89	6623.66

Πίνακας 8-45: Συνολικό Κέρδος – Ibm ως gold

soplex – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	2799.14	3414.55	2710.11	2173.65	3630.54	2787.77
2	2794.98	3417.86	2651.30	2207.12	3627.95	2792.59
3	2729.49	3364.41	2615.78	2167.99	3558.36	2747.83
4	3849.10	4504.92	3704.93	3309.47	4667.10	3865.45
5	3133.91	4175.66	3127.60	3077.63	4262.13	3646.42
6	3789.11	4435.32	3683.84	3325.89	4631.85	3872.52
7	2799.14	3414.55	2710.11	2173.65	3630.54	2787.77
8	2794.98	3417.86	2651.30	2207.12	3627.95	2792.59
9	2729.49	3364.41	2615.78	2167.99	3558.36	2747.83
10	3849.10	4504.92	3704.93	3309.47	4667.10	3865.45

Πίνακας 8-46: Συνολικό Κέρδος – soplex ως gold

gcc – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	2869.56	2802.11	2935.89	2878.10	2100.48	2875.65
2	3517.10	3431.96	3572.63	3523.64	3053.34	3528.95
3	2751.38	2686.89	2833.04	2726.02	1969.62	2751.99
4	2347.77	2240.69	2390.19	2322.15	1941.63	2371.22
5	3706.25	3618.77	3747.58	3679.62	2859.94	3719.51
6	2907.35	2820.62	2916.02	2956.74	2531.02	2910.35
7	3473.44	3367.95	3395.40	3436.54	3049.05	3457.98
8	1992.27	1840.15	1787.66	1986.71	1499.89	1941.68
9	3084.38	2956.16	3142.08	3080.53	2470.51	3089.05
10	3962.00	3850.04	4013.85	4002.13	3305.07	3873.78

Πίνακας 8-47: Συνολικό Κέρδος – gcc ως gold

zeusmp – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	6911.42	6877.66	7011.56	6909.80	6209.50	6890.52
2	7551.37	7489.92	7644.10	7593.51	7092.71	7504.24
3	6678.12	6684.23	6814.98	6734.21	5982.54	6626.72
4	6476.43	6398.78	6529.78	6506.41	6094.18	6470.39
5	7812.11	7773.15	7870.62	7775.17	6999.52	7700.22
6	6845.22	6798.71	6970.12	6993.20	6541.30	6810.03
7	7358.22	7367.17	7463.41	7390.20	6964.51	7247.18
8	5793.14	5783.41	5847.45	5837.02	5498.49	5774.55
9	7102.57	7007.38	7204.86	7127.49	6506.98	6994.43
10	8121.39	8012.69	8179.26	7970.16	7649.59	7970.45

Πίνακας 8-48: Συνολικό Κέρδος – zeusmp ως gold

perlbench – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	2811.87	4458.17	4580.29	4608.75	3825.73	4596.90
2	5114.06	5095.00	5287.55	5266.90	4925.16	5252.18
3	2678.83	2714.22	4487.31	4441.79	3715.62	4532.28
4	3920.55	3898.00	4093.17	4064.29	3623.99	4055.14
5	5295.49	5304.67	5493.43	5411.48	4709.12	5441.57
6	2833.79	2826.18	4686.09	4671.48	4252.54	4637.49
7	3456.24	3411.26	5254.13	5199.86	4820.67	5225.48
8	1846.48	1825.03	3674.33	3707.15	3161.95	3615.61
9	2985.31	2993.85	4796.17	4784.39	4169.24	4829.93
10	5496.21	5503.57	5697.51	5680.47	5063.81	5455.50

Πίνακας 8-49: Συνολικό Κέρδος – perlbench ως gold

gromacs – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	6632.02	6538.89	6568.97	6557.72	5782.26	5969.17
2	7253.84	7177.59	7280.75	7195.91	6874.86	7175.87
3	6437.42	6394.21	6437.84	6417.96	5664.93	6433.13
4	6083.20	5968.81	6081.14	6027.88	5595.54	6019.08
5	7434.66	7379.15	7444.85	7379.81	6752.05	7365.55
6	6599.96	6550.19	6661.49	6633.60	6166.26	6516.96
7	7157.36	7176.99	7229.37	7214.52	6634.32	7003.45
8	5586.76	5576.67	5646.33	5666.69	5150.19	5561.33
9	6775.41	6687.75	6822.06	6705.09	6220.43	6689.48
10	7676.18	7581.86	7699.06	7558.32	7189.19	7514.41

Πίνακας 8-50: Συνολικό Κέρδος – gromacs ως gold

dealII – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	6434.17	6242.25	6428.17	6372.95	5593.77	6357.00
2	7046.06	6931.00	7076.43	6981.25	6721.46	7005.25
3	6233.61	6101.79	6285.75	6204.74	5432.79	6159.31
4	5895.47	5775.29	5907.49	5848.21	5415.50	5805.23
5	7249.81	7131.75	7242.57	7156.81	6438.07	7191.08
6	6400.45	6272.67	6469.58	6449.44	6003.14	6325.84
7	6922.26	6868.10	7000.87	6982.14	6505.69	6911.48
8	5384.23	5317.97	5433.27	5428.23	4943.40	5349.18
9	6575.10	6455.04	6624.07	6576.06	6009.21	6496.41
10	7484.42	7378.88	7512.83	7486.37	6797.92	7457.41

Πίνακας 8-50: Συνολικό Κέρδος – dealII ως gold

hmmcr – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	3409.38	4335.82	4301.91	4220.11	3378.50	4168.10
2	4366.55	4937.38	4923.10	4864.95	4569.92	4888.50
3	3113.71	4196.23	4173.36	4050.99	3303.74	4110.26
4	3247.59	3739.70	3731.72	3678.84	3262.82	3650.36
5	4422.21	5158.44	5105.99	4997.06	4447.36	4993.73
6	3873.63	4323.86	4309.63	4313.22	3874.98	4270.73
7	4453.24	4931.29	4898.41	4866.17	4393.72	4782.74
8	2868.74	3363.03	3336.63	3278.64	2877.12	3246.30
9	3792.24	4480.78	4485.48	4418.81	3789.56	4382.77
10	4815.11	5321.95	5371.48	5310.02	4612.02	5121.07

Πίνακας 8-51: Συνολικό Κέρδος – hmmcr ως gold

sjeng – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	8624.06	8915.73	9006.68	8907.92	8193.05	8984.82
2	9164.44	9480.91	9568.06	9512.53	9039.23	8722.38
3	8228.10	8756.77	8749.58	8752.68	7964.11	8507.48
4	8161.60	8268.19	8367.36	8331.34	7914.38	8135.79
5	9611.26	9765.78	9824.44	9738.35	8915.61	9725.41
6	8484.47	8852.78	8930.17	8864.62	8499.02	8669.09
7	8865.32	9467.94	9521.22	9417.15	9024.49	9177.59
8	7419.69	7819.36	7957.45	7857.32	7491.09	7509.46
9	8796.07	9002.16	9178.86	9076.50	8447.17	8901.37
10	10010.82	9938.42	10063.52	10044.36	9331.41	10029.39

Πίνακας 8-53: Συνολικό Κέρδος – sjeng ως gold

h264ref – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	3512.48	3583.16	3569.65	3452.89	2641.89	3511.26
2	4208.06	4183.16	4187.75	4096.52	3636.47	4199.26
3	3410.05	3414.68	3423.20	3310.89	2542.18	3414.66
4	3017.44	2985.75	2972.32	2912.41	2543.46	2914.55
5	4443.16	4402.32	4351.59	4294.57	3435.98	4360.56
6	3594.53	3582.07	3566.88	3503.69	3114.11	3530.16
7	4241.69	4187.40	4160.67	4069.64	3645.74	4139.87
8	2865.35	2593.93	2604.87	2551.12	2056.07	2576.73
9	3716.46	3746.49	3714.10	3670.35	3051.12	3707.02
10	4604.01	4603.32	4608.42	4514.07	3797.25	4528.83

Πίνακας 8-54: Συνολικό Κέρδος – h264ref ως gold

sphinx3 – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	8093.10	2843.50	8869.51	8919.52	8134.66	8713.05
2	3425.01	3454.92	9472.19	9607.02	9246.97	9271.44
3	2676.87	2752.27	8774.16	8810.76	8055.45	8640.73
4	7981.33	2238.41	8300.49	8435.42	8023.07	8023.34
5	9122.38	3666.11	9708.36	9649.00	9051.27	9391.21
6	2779.65	2844.70	8836.52	8995.52	8541.41	8650.17
7	3409.06	3450.97	9412.49	9569.27	9176.35	9085.94
8	1793.74	1865.43	7869.65	8056.74	7493.13	7516.65
9	2966.04	2991.81	9066.56	9223.55	8597.75	8959.47
10	9741.45	3911.91	9907.61	10213.67	9590.40	9719.67

Πίνακας 8-55: Συνολικό Κέρδος – sphinx3 ως gold

xalancbmk – Συνολικό Κέρδος						
Σενάριο	No Cache Allocation	Mask2	Mask8	Mask12	Mask18	Dynamic Mechanism
1	2817.66	3451.43	2706.48	2225.27	3646.50	2831.42
2	2810.08	3439.82	2694.89	2222.49	3639.30	2819.27
3	2751.62	3383.36	2659.39	2179.01	3587.07	2789.91
4	4588.57	5288.39	4483.21	4138.71	5442.44	4706.22
5	3966.55	5020.75	3845.31	3797.83	4672.49	4346.29
6	4525.69	5242.89	4381.20	4080.85	5374.10	4648.71
7	2817.66	3451.43	2706.48	2225.27	3646.50	2831.42
8	2810.08	3439.82	2694.89	2222.49	3639.30	2819.27
9	2751.62	3383.36	2659.39	2179.01	3587.07	2789.91
10	4588.57	5288.39	4483.21	4138.71	5442.44	4706.22

Πίνακας 8-55: Συνολικό Κέρδος – xalancbmk ως gold