



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών

Εαρινό Εξάμηνο 2023-2024

ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

1η Σειρά Ασκήσεων

Ιωάννης Τσαντήλας
03120883

Περιεχόμενα

Ερώτημα 1	2
L ₁ cache.....	2
Γραφικές Παραστάσεις	3
L ₂ cache.....	6
Γραφικές Παραστάσεις	7
TLB	10
Γραφικές Παραστάσεις	11
Prefetching	14
Γραφικές Παραστάσεις	15
Συμπεράσματα.....	18

Ερώτημα 1

L₁ cache

Για σταθερές παραμέτρους της L₂ cache και του TLB έγινε εκτέλεση των 8 benchmarks για τους διαφορετικούς συνδυασμούς παραμέτρων της L₁ cache (13 συνολικά). Έγιναν οι παρακάτω αλλαγές στο δοσμένο κώδικα **"l1_run.sh"**, για να ενημερωθούν σωστά οι παράμετροι των L₂ cache και του TLB καθώς και οι διάφορες παράμετροι για την L₁ cache.

- L2size = 2048
- L2assoc = 16
- L2bsize = 256
- TLBe = 64
- TLBp = 4096
- TLBa = 4
- L2pref = 0

Ο κώδικας **"run_l1.sh"** εκτελέστηκε με ορισμένα για 8 benchmarks, συγκεκριμένα τα *blackscholes*, *bodytrack*, *canneal*, *fluidanimate*, *freqmine*, *rtview*, *swaptions* και *streamcluster*, που βρίσκονται στον κατάλογο **/advcomparch-ex1-helppcode/run_l1.sh**.

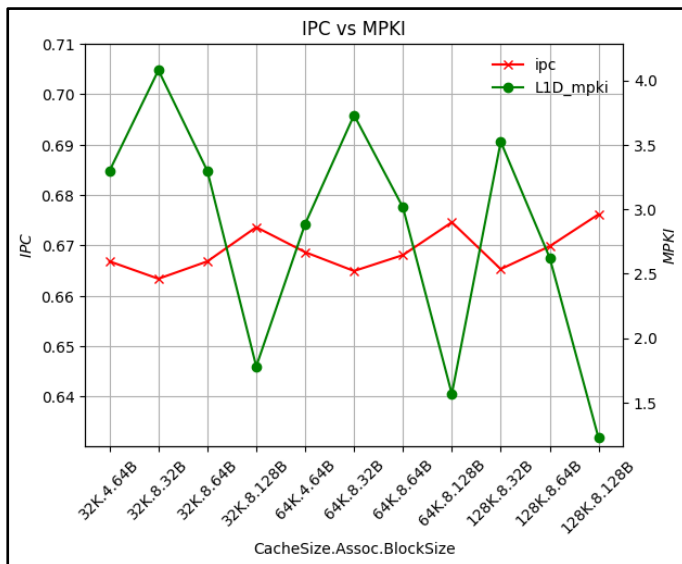
Μετά την εκτέλεση του script **"run_l1.sh"**, δημιουργήθηκαν 13 αρχεία που περιγράφουν την απόδοση κάθε εφαρμογής για τις διάφορες παραμέτρους της ιεραρχίας μνήμης.

Το αρχείο **"plot_l1.sh"** εκτελέστηκε για κάθε benchmark με ορίσματα τα 13 αρχεία που δημιουργήθηκαν, καταλήγοντας στη δημιουργία γραφικών παραστάσεων για κάθε benchmark.

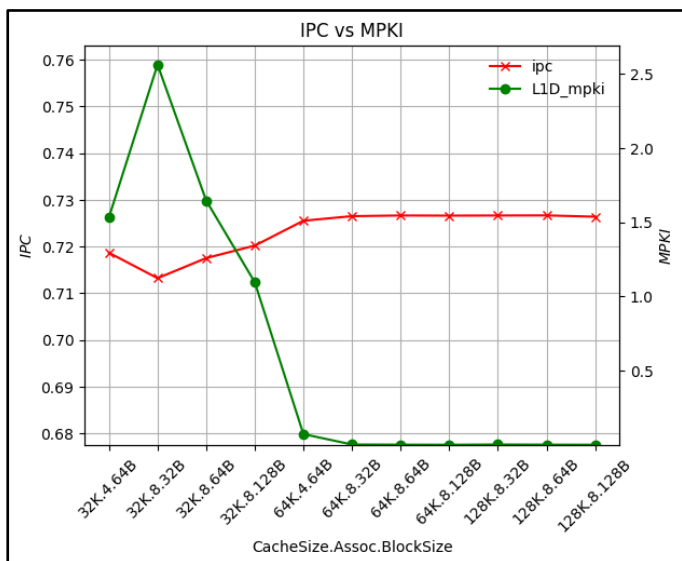
Στις παραστάσεις αυτές παρουσιάζονται οι καμπύλες για το *MPKI* (misses per 1000 instructions) για την L₁ cache και η καμπύλη *IPC* (instructions per cycle).

(Οι παραστάσεις φαίνονται από την επόμενη σελίδα)

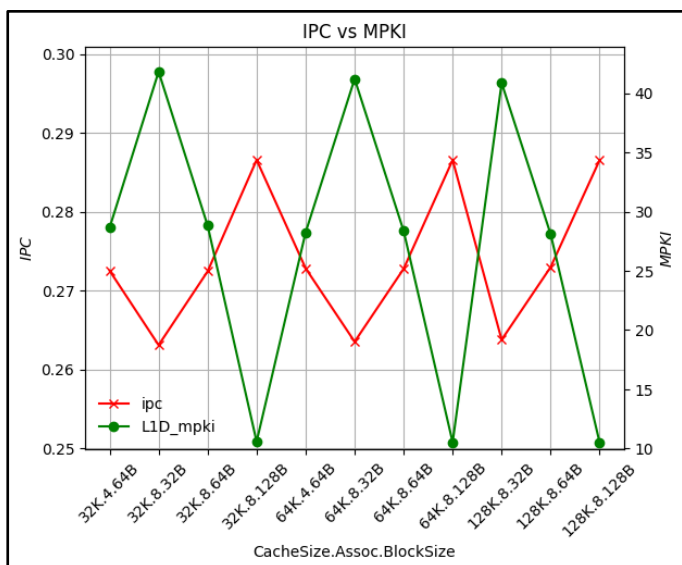
Γραφικές Παραστάσεις



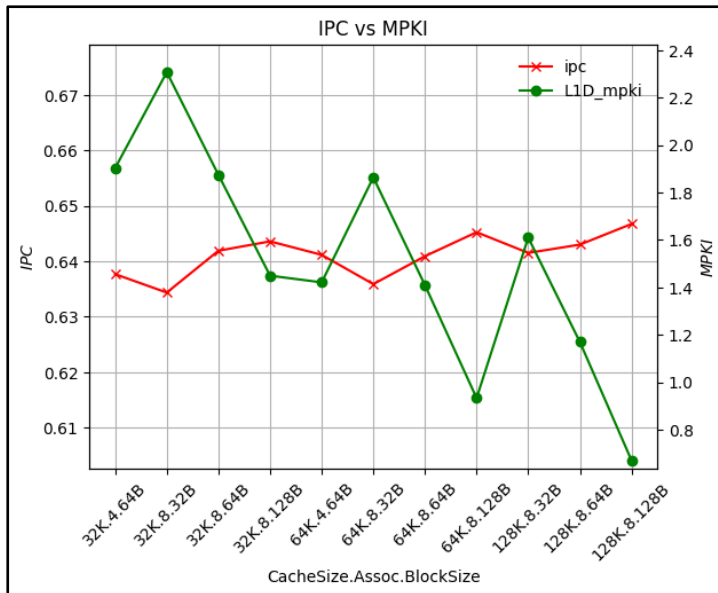
Σχήμα 1: Blacksholes.



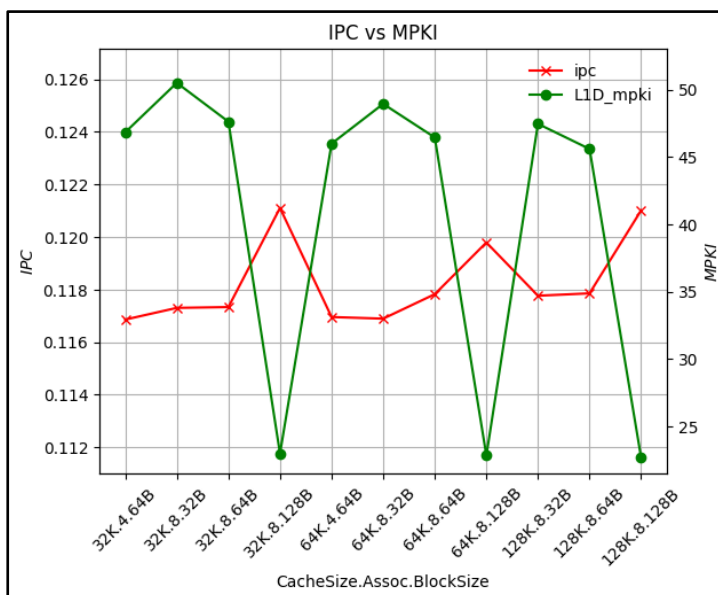
Σχήμα 2: Bodytrack.



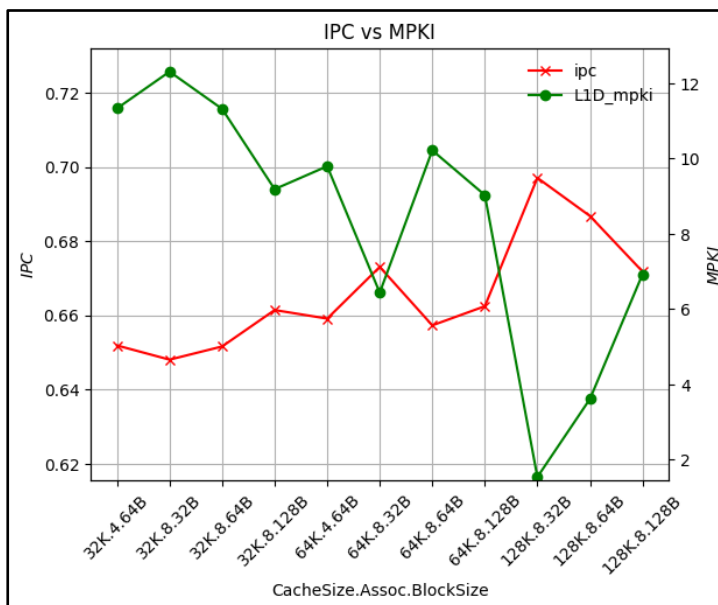
Σχήμα 3: Canneal.



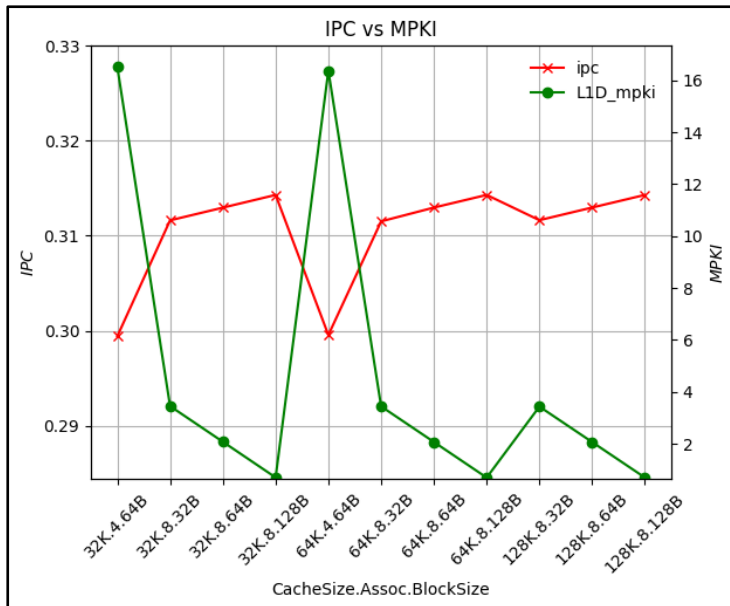
Σχήμα 4: Freqmine.



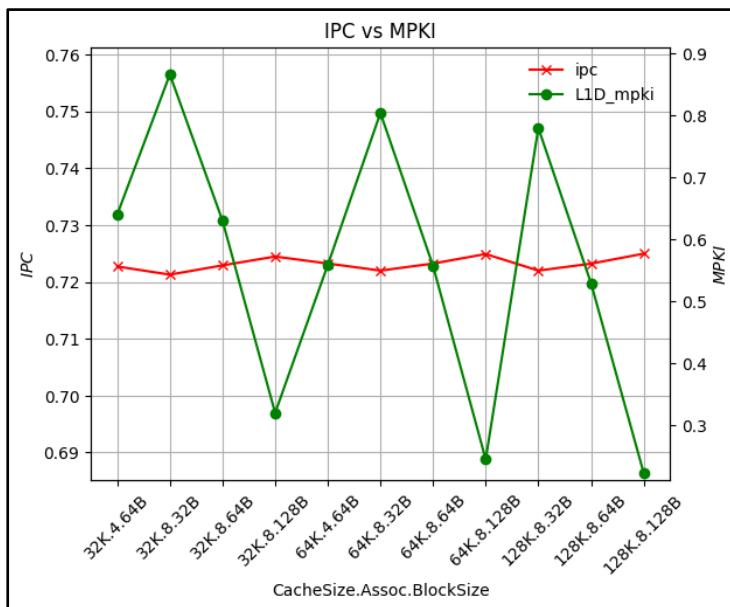
Σχήμα 5: Streamcluster.



Σχήμα 6: Swaptions.



Σχήμα 7: Fluidanimate.



Σχήμα 8: Rtvew.

L₂ cache

Για σταθερές παραμέτρους της L₁ cache και του TLB έγινε εκτέλεση των 8 benchmarks για τους διαφορετικούς συνδυασμούς παραμέτρων της L₂ cache (14 συνολικά). Με βάση το δοσμένο κώδικα για την L₁ cache δημιουργήθηκε ο αντίστοιχος κώδικας “**run_l2.sh**” για την cache L2. Ορίστηκαν σωστά οι παράμετροι για τις τιμές που ζητά η εκφώνηση όπως φαίνεται παρακάτω.

- L1size = 32
- L1assoc = 8
- L1bsize = 64
- TLBe = 64
- TLBp = 4096
- TLBa = 4
- L1pref = 0

Ο κώδικας “**run_l2.sh**” εκτελέστηκε με ορισμένα για 8 benchmarks, συγκεκριμένα τα *blackscholes*, *bodytrack*, *canneal*, *fluidanimate*, *freqmine*, *rtview*, *swaptions* και *streamcluster*, που βρίσκονται στον κατάλογο **/advcomparch-ex1-helpcode/run_l2.sh**.

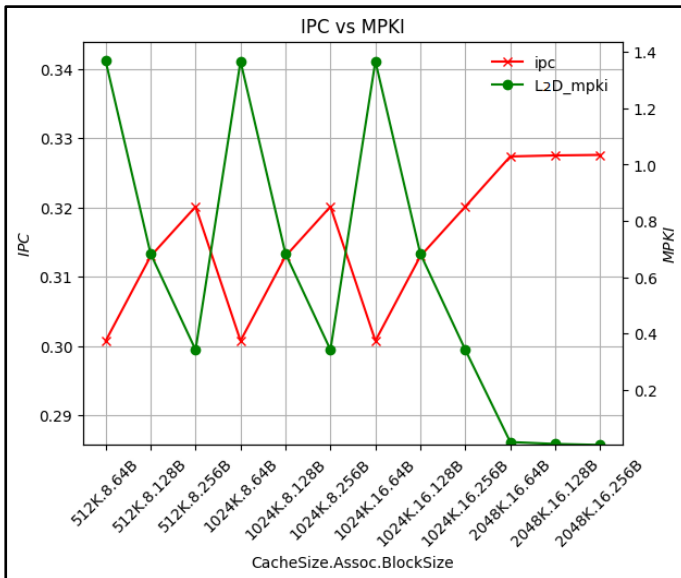
Μετά την εκτέλεση του script “**run_l2.sh**”, δημιουργήθηκαν 14 αρχεία που περιγράφουν την απόδοση κάθε εφαρμογής για τις διάφορες παραμέτρους της ιεραρχίας μνήμης.

Το αρχείο “**plot_l2.sh**” εκτελέστηκε για κάθε benchmark με ορίσματα τα 14 αρχεία που δημιουργήθηκαν, καταλήγοντας στη δημιουργία γραφικών παραστάσεων για κάθε benchmark.

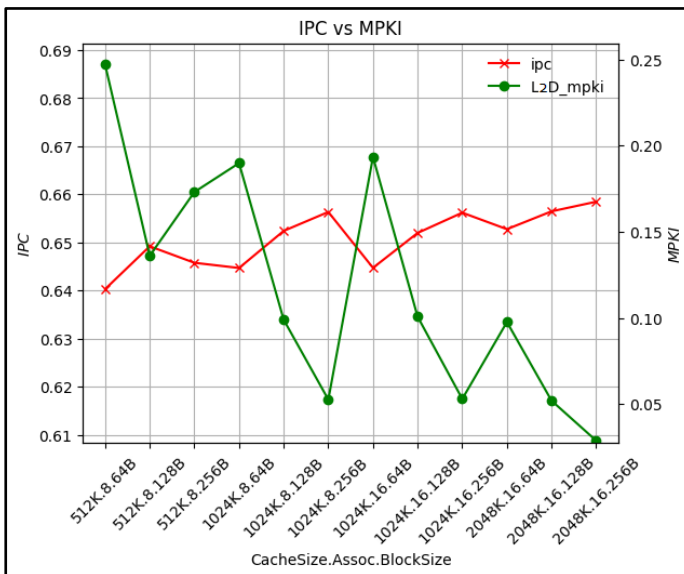
Στις παραστάσεις αυτές παρουσιάζονται οι καμπύλες για το *MPKI* (misses per 1000 instructions) για την L₁ cache και η καμπύλη *IPC* (instructions per cycle).

(Οι παραστάσεις φαίνονται από την επόμενη σελίδα)

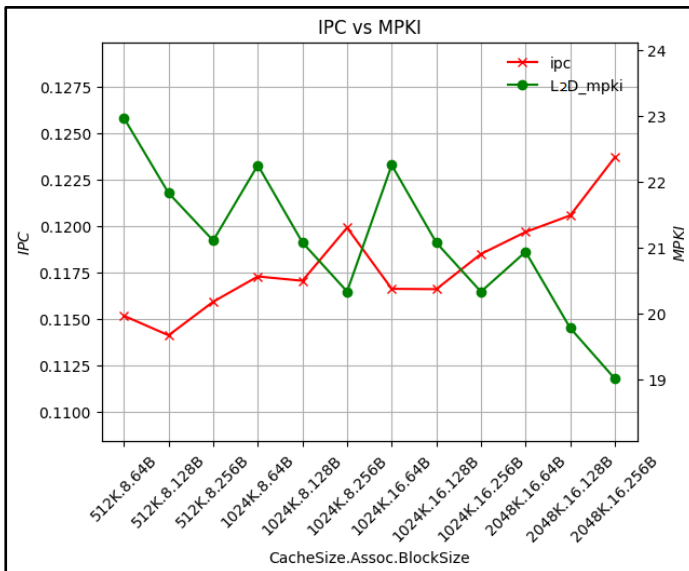
Γραφικές Παραστάσεις



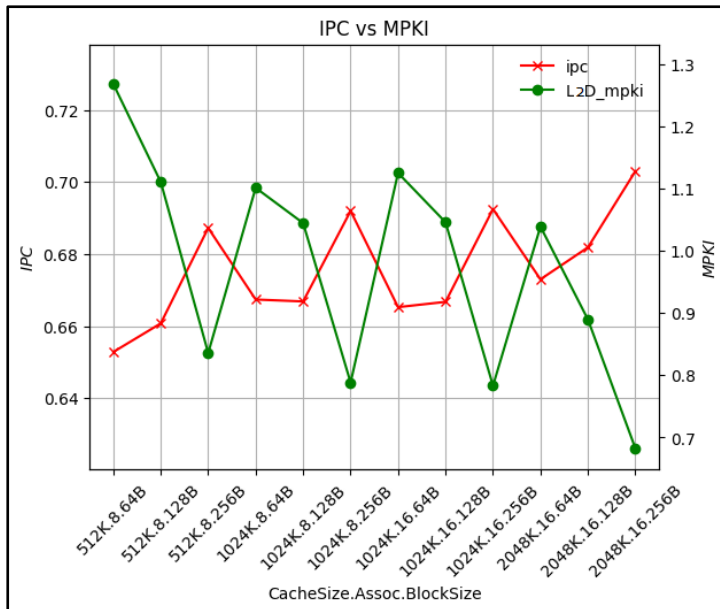
Σχήμα 9: Blacksholes.



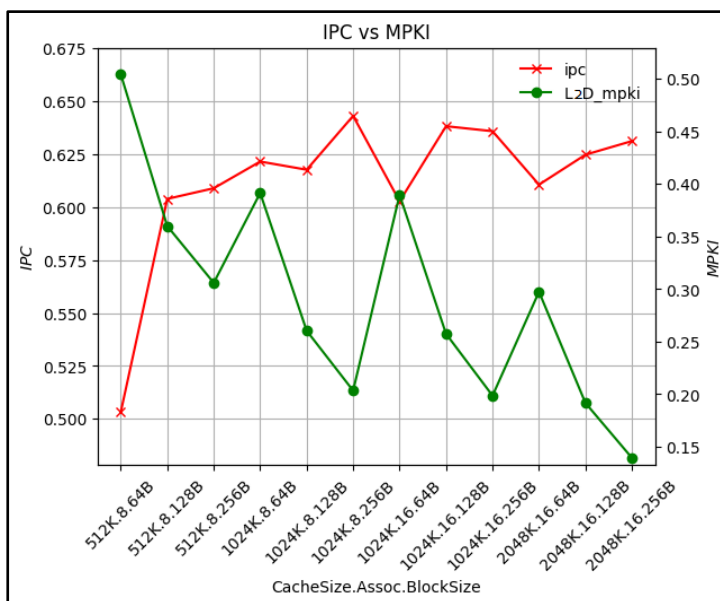
Σχήμα 10: Bodytrack.



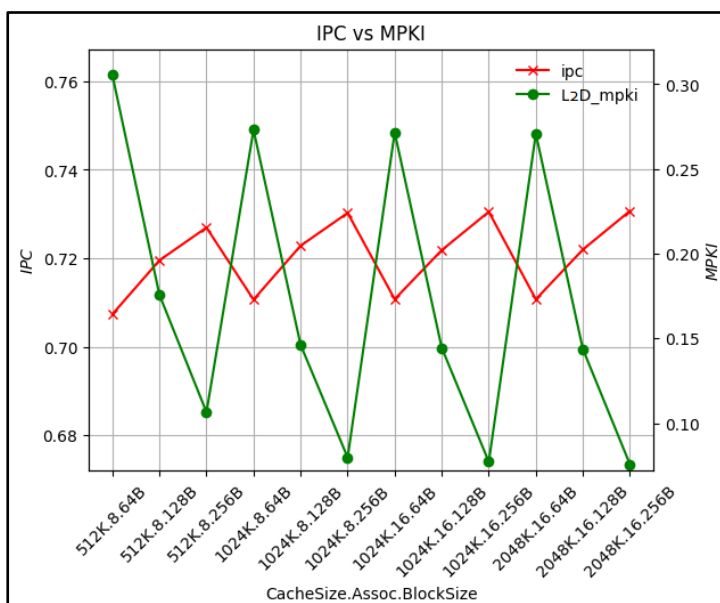
Σχήμα 11: Canneal.



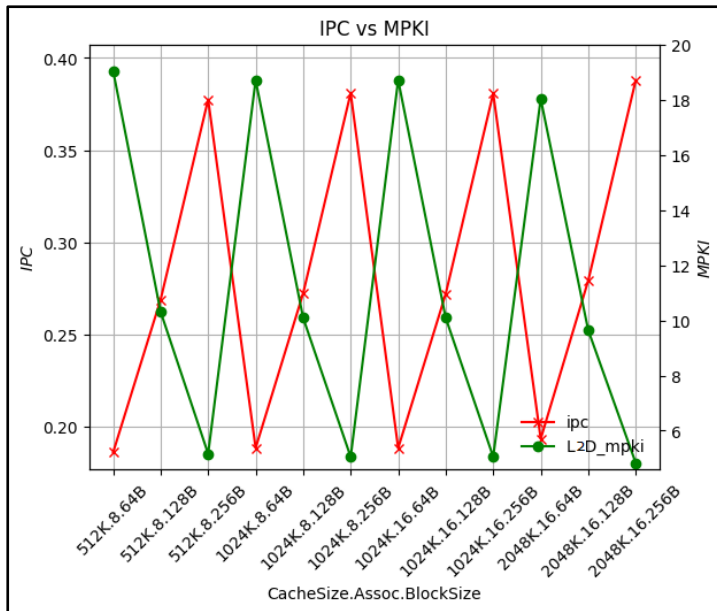
Σχήμα 12: Freqmine.



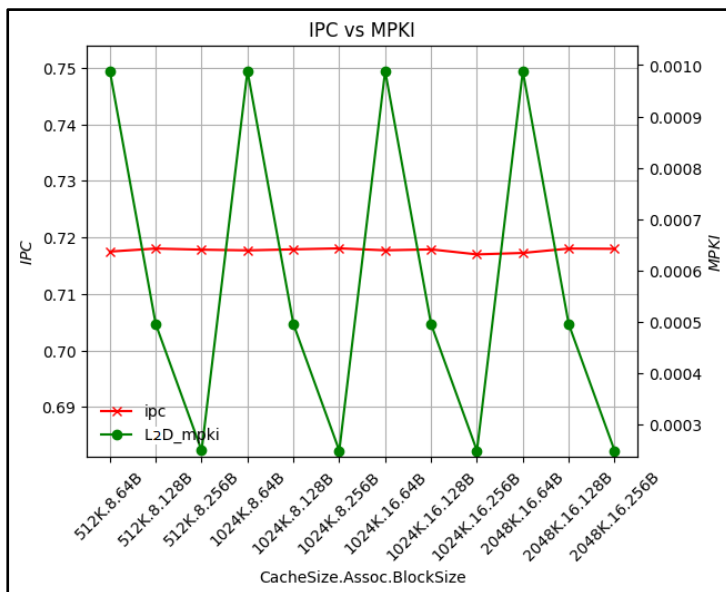
Σχήμα 13: Streamcluster.



Σχήμα 14: Swaptions.



Σχήμα 15: Fluidanimate.



Σχήμα 16: Rtvew.

TLB

Για σταθερές παραμέτρους της L_1 cache και της L_2 έγινε εκτέλεση των 8 benchmarks για τους διαφορετικούς συνδυασμούς παραμέτρων του TLB (11 συνολικά). Όπως και πριν, δημιουργήθηκε ο αντίστοιχος κώδικας “**run_tlb.sh**”. Ορίστηκαν σωστά οι παράμετροι για τις τιμές που ζητά η εκφώνηση όπως φαίνεται παρακάτω.

- $L2size = 2048$
- $L2assoc = 16$
- $L2bsize = 256$
- $L2pref = 0$
- $L1size = 32$
- $L1assoc = 8$
- $L1bsize = 64$
- $L1pref = 0$

Ο κώδικας “**run_tlb.sh**” εκτελέστηκε με ορισμένα για 8 benchmarks, συγκεκριμένα τα *blackscholes*, *bodytrack*, *canneal*, *fluidanimate*, *freqmine*, *rtview*, *swaptions* και *streamcluster*, που βρίσκονται στον κατάλογο **/advcomparch-ex1-helpcode/run_tlb.sh**.

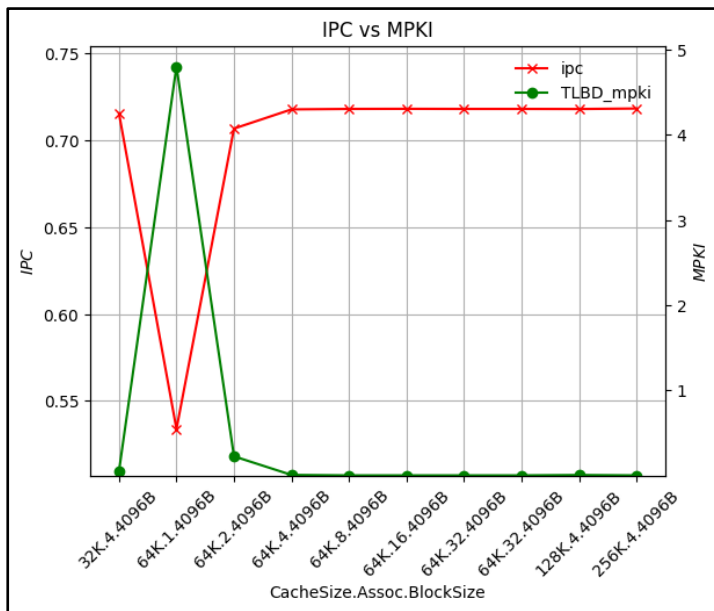
Μετά την εκτέλεση του script “**run_tlb.sh**”, δημιουργήθηκαν 11 αρχεία που περιγράφουν την απόδοση κάθε εφαρμογής για τις διάφορες παραμέτρους της ιεραρχίας μνήμης.

Το αρχείο “**plot_tlb.sh**” εκτελέστηκε για κάθε benchmark με ορίσματα τα 11 αρχεία που δημιουργήθηκαν, καταλήγοντας στη δημιουργία γραφικών παραστάσεων για κάθε benchmark.

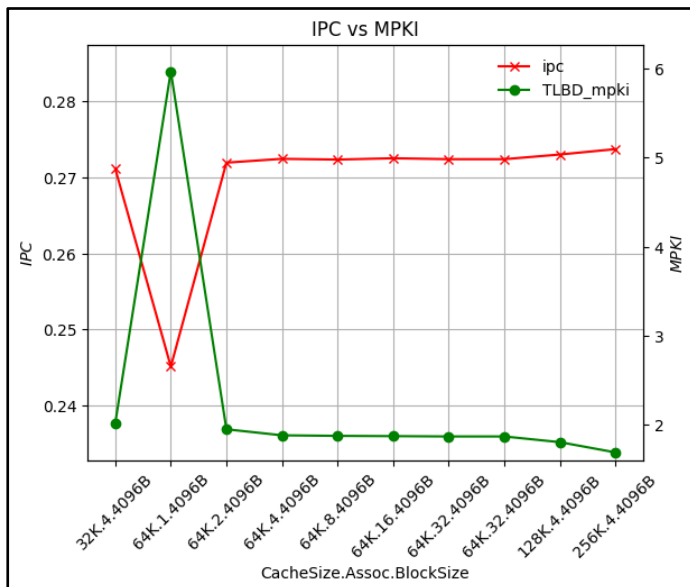
Στις παραστάσεις αυτές παρουσιάζονται οι καμπύλες για το *MPKI* (misses per 1000 instructions) για την L_1 cache και η καμπύλη *IPC* (instructions per cycle).

(Οι παραστάσεις φαίνονται από την επόμενη σελίδα)

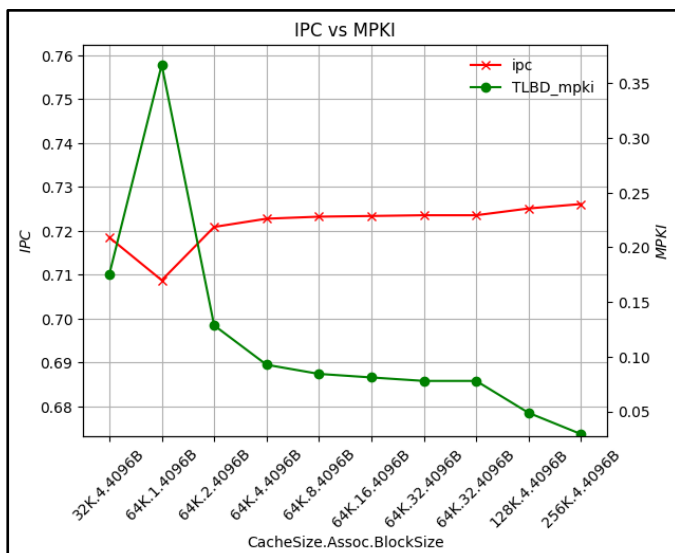
Γραφικές Παραστάσεις



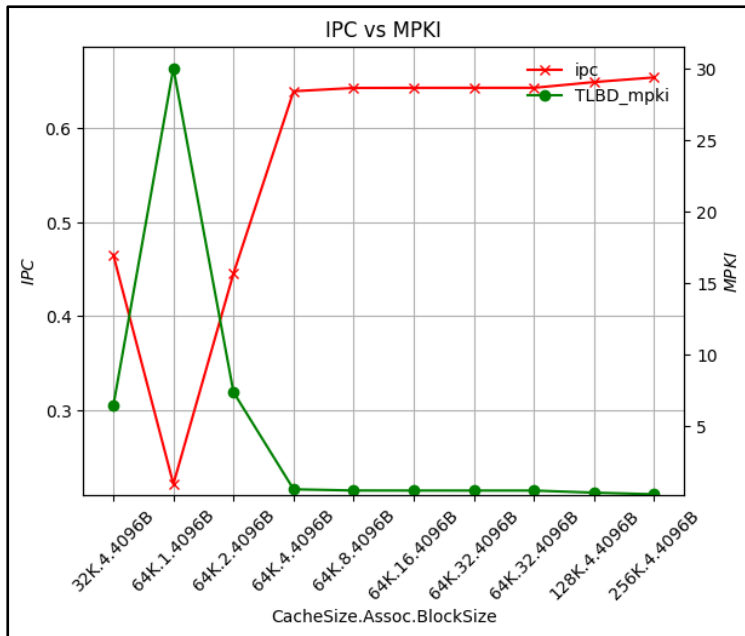
Σχήμα 17: Blacksholes.



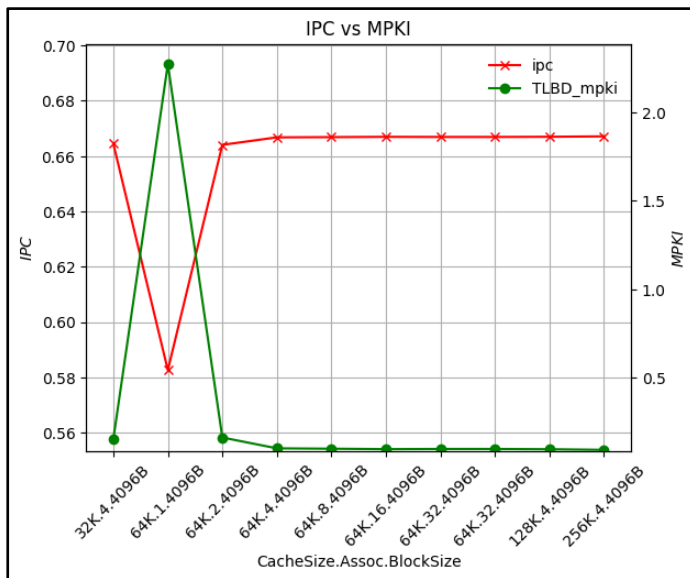
Σχήμα 18: Bodytrack.



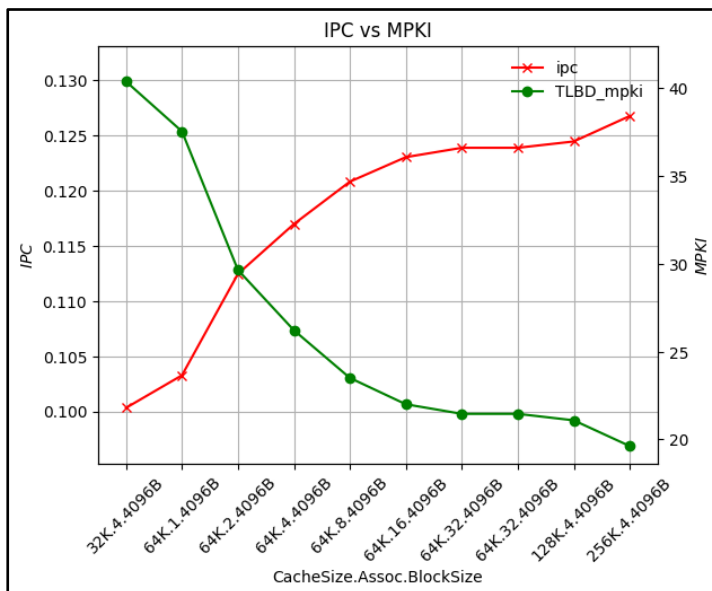
Σχήμα 19: Canneal.



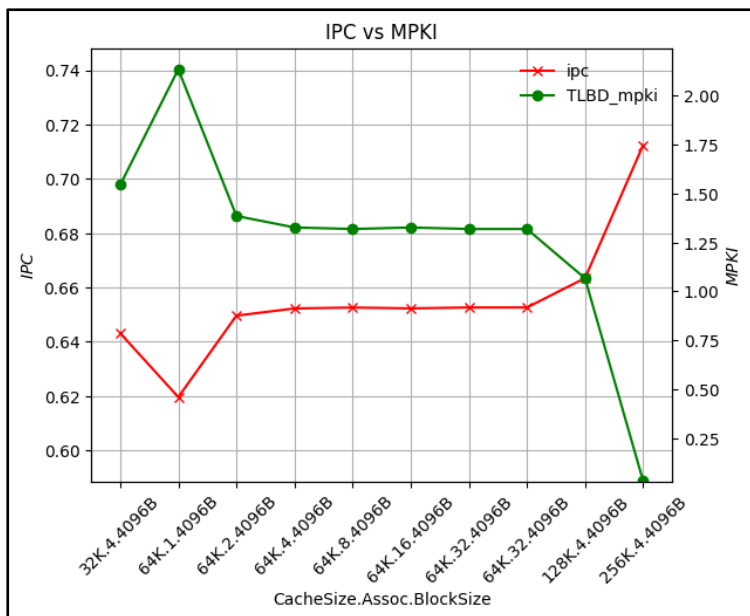
Σχήμα 20: Freqmine.



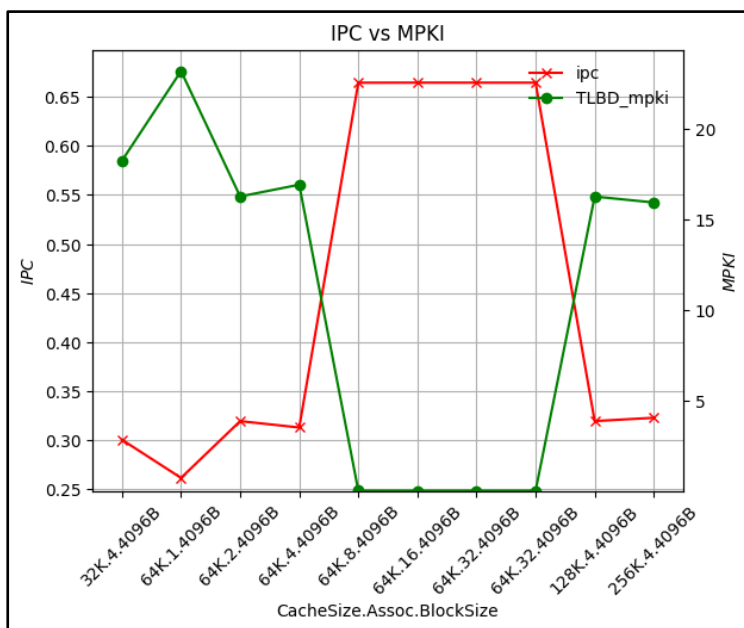
Σχήμα 21: Streamcluster.



Σχήμα 22: Swaptions.



Σχήμα 23: Fluidanimate.



Σχήμα 24: Rtvew.

Prefetching

Για σταθερές παραμέτρους των L_1 , L_2 και TLB έγινε εκτέλεση των 8 benchmarks για διαφορετικές τιμές της παραμέτρου $L2prf$, δηλαδή για διαφορετικό αριθμό από blocks που για κάθε L_2 miss να φέρνει η cache L_2 . Όπως και πριν, δημιουργήθηκε ένας κώδικας “**run_pref.sh**” και ορίστηκαν σε αυτό σωστά οι παράμετροι για τις τιμές που ζητά η εκφώνηση όπως φαίνεται παρακάτω.

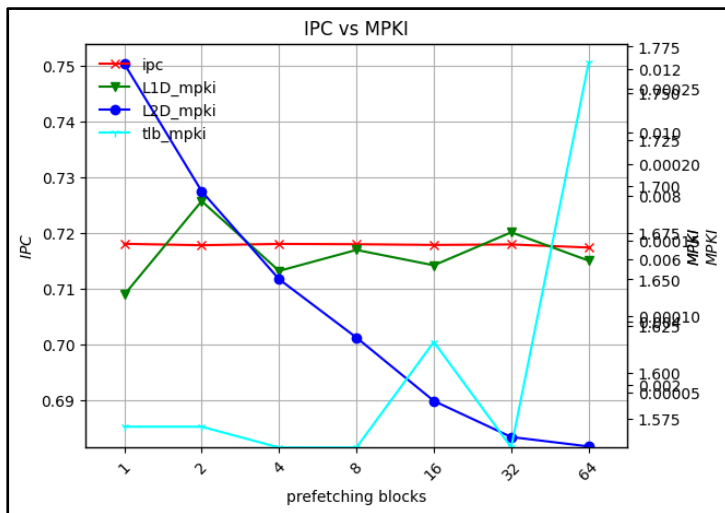
- $L1size = 32$
- $L1assoc = 8$
- $L1bsize = 64$
- $L2size = 2048$
- $L2assoc = 16$
- $L2bsize = 256$
- $TLBe = 64$
- $TLBp = 4096$
- $TLBa = 4$

Ο κώδικας “**run_pref.sh**” εκτελέστηκε με ορίσματα για 8 benchmarks, συγκεκριμένα τα *blackscholes*, *bodytrack*, *canneal*, *fluidanimate*, *fraqmine*, *rtview*, *swaptions*, και *streamcluster*, που βρίσκονται στον κατάλογο **/advcomparch-ex1-helpcode/run_pref.sh**.

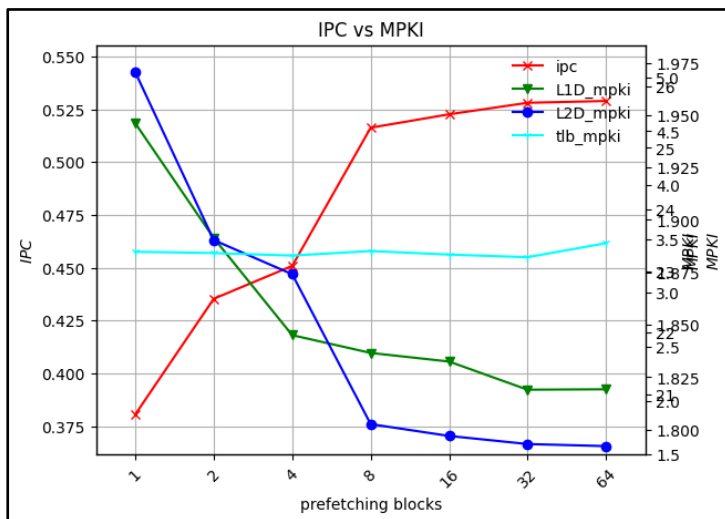
Μετά την εκτέλεση του κώδικα “**run_pref.sh**”, δημιουργήθηκαν 7 αρχεία που περιγράφουν την απόδοση κάθε εφαρμογής για τις διάφορες παραμέτρους της ιεραρχίας μνήμης. Το αρχείο “**plot_pref.sh**” εκτελέστηκε για κάθε benchmark με ορίσματα τα 7 αρχεία που δημιουργήθηκαν, καταλήγοντας στη δημιουργία γραφικών παραστάσεων για κάθε benchmark.

Στις παραστάσεις αυτές παρουσιάζονται οι καμπύλες για το *MPKI* (misses per 1000-instructions) για την $L1$ cache και η καμπύλη *IPC* (instructions per cycle).

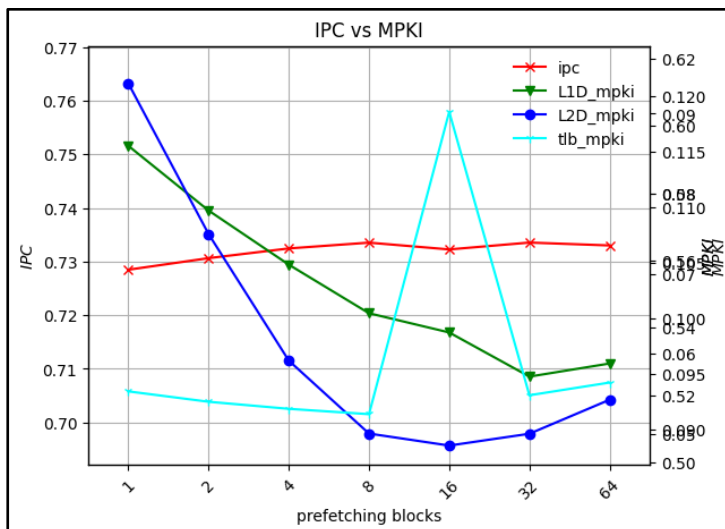
Γραφικές Παραστάσεις



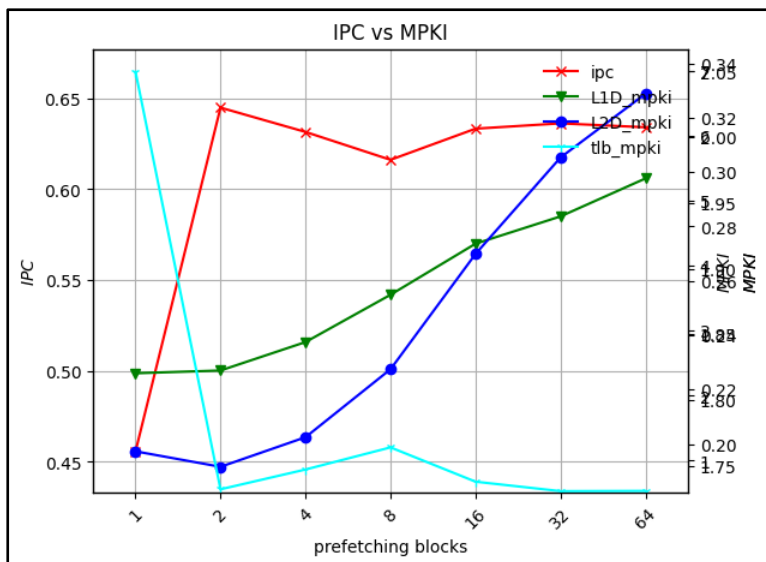
Σχήμα 25: Blacksholes.



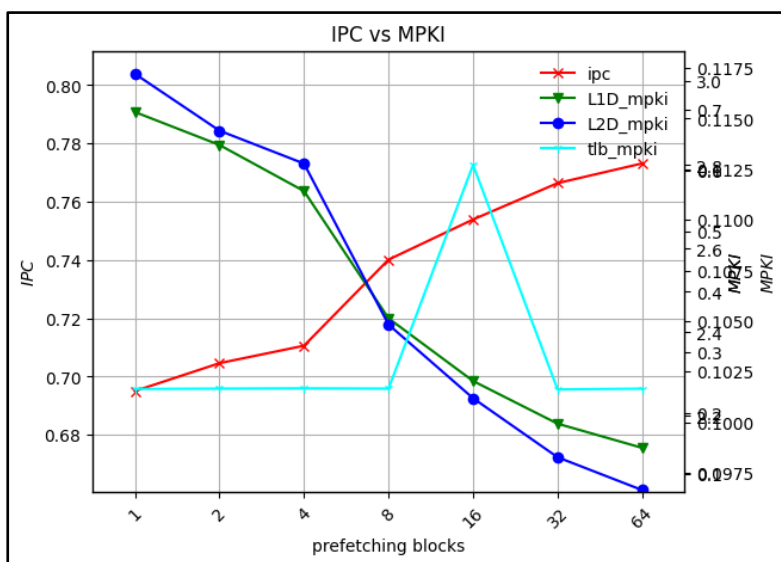
Σχήμα 26: Bodytrack.



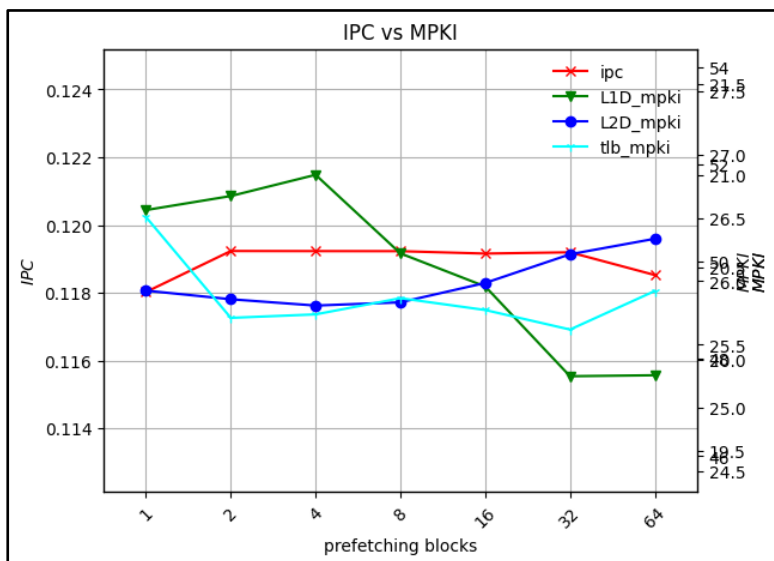
Σχήμα 27: Canneal.



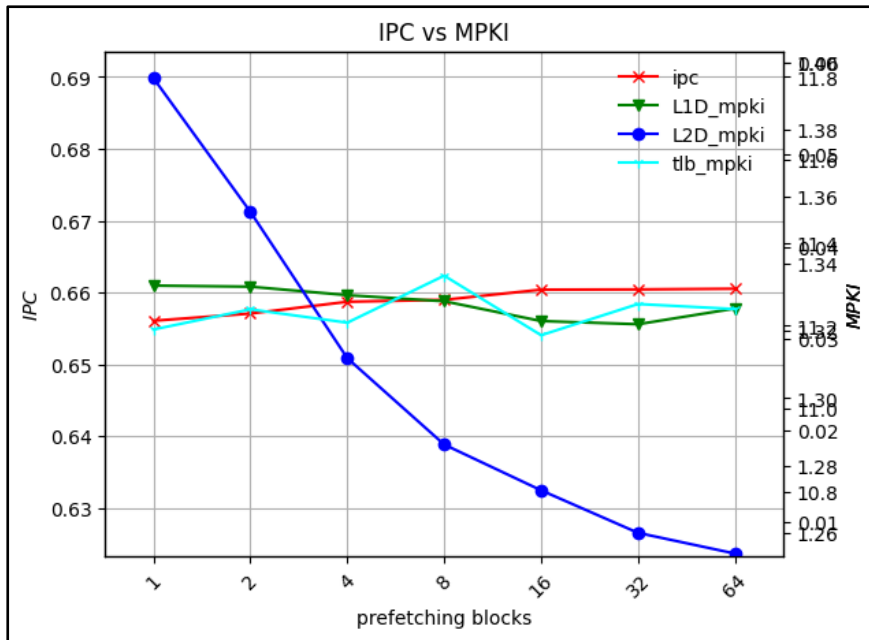
Σχήμα 28: Freqmine.



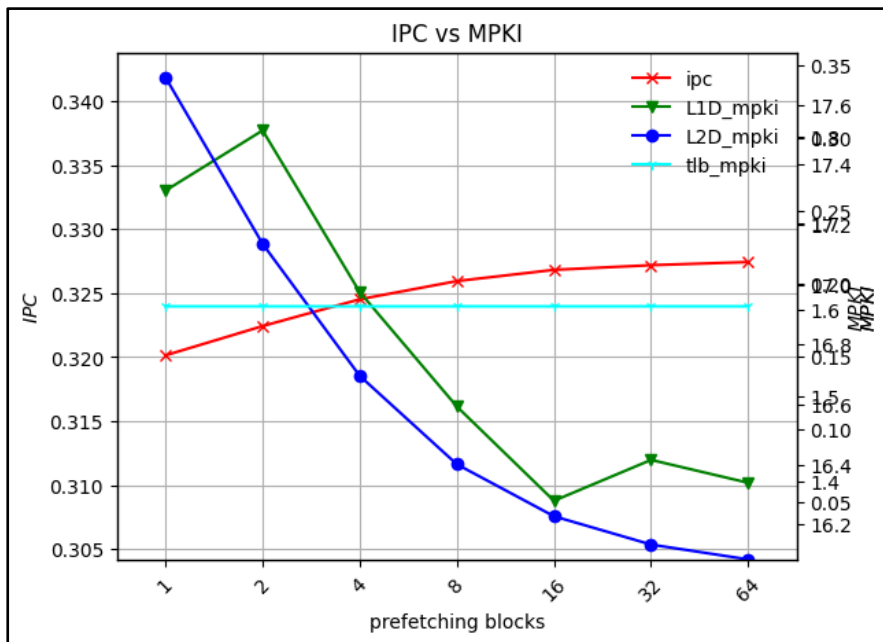
Σχήμα 29: Streamcluster.



Σχήμα 30: Swaptions.



Σχήμα 31: Fluidanimate.



Σχήμα 32: Rtview.

Συμπεράσματα

Με βάση την ανάλυση που αναλύθηκε προηγουμένως, είναι προφανές ότι η αποδοτικότητα της μνήμης, η οποία υποδηλώνεται από τη μείωση των MPKI, βελτιώνεται με την αύξηση πολλών διαστάσεων. Αυτές οι διαστάσεις περιλαμβάνουν το μέγεθος της κρυφής μνήμης, το μέγεθος μπλοκ και τη συσχετιστικότητα μεταξύ των στοιχείων. Τα ευρήματα αυτά υποδηλώνουν μια θετική συσχέτιση μεταξύ αυτών των παραμέτρων και της συνολικής αποδοτικότητας των λειτουργιών μνήμης. Όσον αφορά τις εμπειρικές παρατηρήσεις από διάφορα benchmarks:

- **L1 Cache:** Οι γραφικές αναπαραστάσεις επιβεβαιώνουν σε μεγάλο βαθμό τις θεωρητικές προβλέψεις που υποδηλώνουν βελτιώσεις στην αποδοτικότητα της κρυφής μνήμης L1 καθώς αυξάνεται το μέγεθος του μπλοκ. Η τάση αυτή είναι συνεπής στα περισσότερα benchmarks. Ωστόσο, σημειώνονται εξαιρέσεις στα benchmarks *bodytrack* και *swaptions*, όπου τα δεδομένα παρουσιάζουν ασυνέπειες που δεν ευθυγραμμίζονται με τα αναμενόμενα θεωρητικά αποτελέσματα.
- **L2 Cache:** Ομοίως, τα δεδομένα για την κρυφή μνήμη L2 υποστηρίζουν σε μεγάλο βαθμό τις θεωρητικές προβλέψεις, με σαφείς βελτιώσεις στην αποδοτικότητα καθώς αυξάνεται τόσο το μέγεθος των μπλοκ όσο και η συσχετιστικότητα. Τα αποτελέσματα στα περισσότερα benchmarks ευθυγραμμίζονται με αυτό το μοτίβο. Παρ' όλα αυτά, το benchmark *bodytrack* ξεχωρίζει καθώς εμφανίζει αποκλίσεις από τις προβλεπόμενες βελτιώσεις, γεγονός που υποδηλώνει ασυμφωνία μεταξύ θεωρητικών προβλέψεων και εμπειρικών δεδομένων.
- **TLB:** Η ανάλυση του TLB αντικατοπτρίζει τις θετικές τάσεις που παρατηρήθηκαν σε άλλους τομείς, με βελτιώσεις στην αποδοτικότητα καθώς αυξάνεται η συσχέτιση. Η βελτίωση αυτή παρατηρείται σταθερά σε όλα σχεδόν τα benchmarks. Σημειώνεται επίσης ότι ένας υψηλός βαθμός συσχετιστικότητας του TLB δεν μεταβάλλει σημαντικά τις τιμές MPKI, υποδεικνύοντας ότι δεν υπάρχει ουσιαστική επίπτωση στην απόδοση από αυτή την παράμετρο και μόνο.
- **Prefetching:** Παρατηρούνται βελτιώσεις στις επιδόσεις με την αύξηση της παραμέτρου L2 prefetch. Αυτό υποδηλώνει ότι η προσαρμογή αυτής της παραμέτρου μπορεί να οδηγήσει σε παρατηρήσιμα οφέλη όσον αφορά την αποδοτικότητα και την αποτελεσματικότητα της χρήσης της μνήμης.

Benchmark	MPKI L₁ cache	MPKI L₂ cache
<i>Blacksholes</i>	-	+ -
<i>Bodytrack</i>	-	+ -
<i>Canneal</i>	+	+ -
<i>Fluidanimate</i>	-	-
<i>Freqmine</i>	+	+
<i>Rtview</i>	-	-
<i>Streamcluster</i>	-	-
<i>Swaptions</i>	-	+ -

Από τον παραπάνω πίνακα παρατηρούμε πως για τα περισσότερα benchmarks καθώς αυξάνεται το prefetching στην L₂ cache μειώνεται το MPKI, και έτσι αυξάνεται η επίδοση. Παρατηρούμε ακόμη αυξομειώσεις στην επίδοση της L₁ cache για αρκετά benchmarks. Έτσι,

συμπεραίνουμε πως η αύξηση της παραμέτρου prefetching έχει διαφορετική συμπεριφορά, η οποία εξαρτάται από το benchmark που εκτελείται.

Ερώτημα 2

Σε αυτό το ερώτημα υποθέτουμε ότι η διάρκεια του κύκλου δεν είναι σταθερή. Με την υπόθεση ότι κάθε διπλασιασμός του associativity ή του μεγέθους προκαλεί αύξηση του κύκλου κατά 5% ή 15 % αντίστοιχα εξάγουμε τα παρακάτω αποτελέσματα για κάθε υποερώτημα.

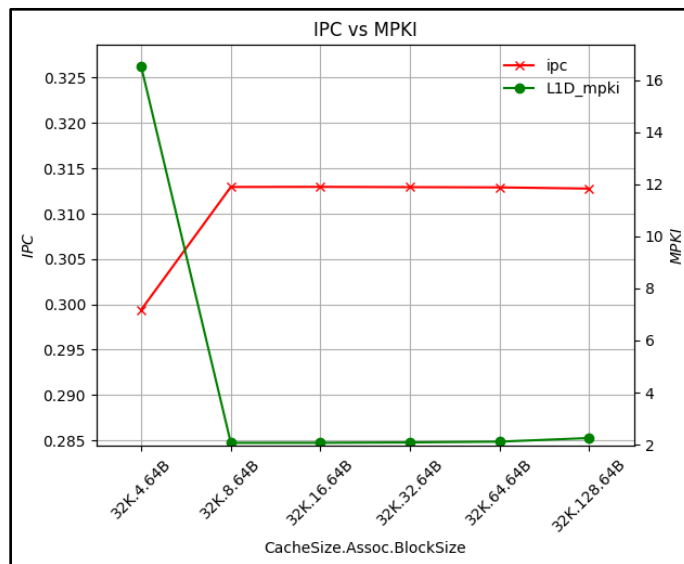
L₁ cache

Για την 1^η προσομοίωση θέτουμε L₁ size = 32 KB, L₁ associativity = 4, L₁ cache block size = 32 B.

L ₁ size	L ₁ Associativity	L ₁ Cache Block Size	Παράγοντας αύξησης κύκλου
32 KB	4	64 KB	1
32 KB	8	64 KB	1.05
32 KB	16	64 KB	1.05 · 1.05
32 KB	32	64 KB	1.05 · 1.05 · 1.05
32 KB	64	64 KB	1.05 · 1.05 · 1.05 · 1.05
32 KB	128	64 KB	1.05 · 1.05 · 1.05 · 1.05 · 1.05

Σχήμα 33: Διπλασιασμός L1 associativity (5% αύξηση του κύκλου)

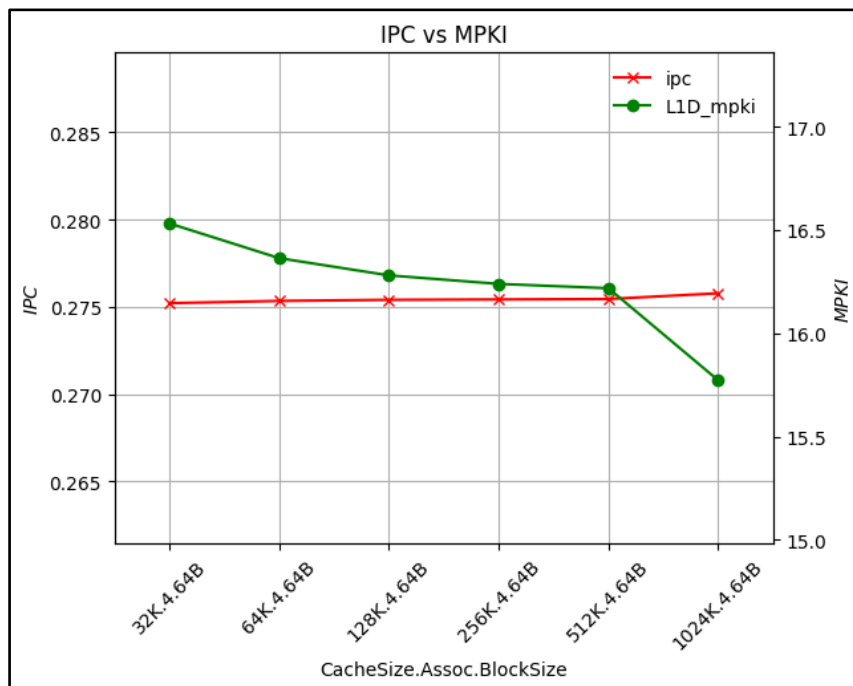
Τρέχουμε τον κώδικα “run2_l1a.sh” και εξάγουμε την παρακάτω γραφική για το benchmark “blackscholes”.



L ₁ size	L ₁ Associativity	L ₁ Cache Block Size	Παράγοντας αύξησης κύκλου
32 KB	4	64 KB	1
64 KB	4	64 KB	1.15
128 KB	4	64 KB	1.15 · 1.15
256 KB	4	64 KB	1.15 · 1.15 · 1.15
512 KB	4	64 KB	1.15 · 1.15 · 1.15 · 1.15
1024 KB	4	64 KB	1.15 · 1.15 · 1.15 · 1.15 · 1.15

Σχήμα 34: Διπλασιασμός L1 associativity (15% αύξηση του κύκλου)

Τρέχουμε τον κώδικα “run2_l1s.sh” και εξάγουμε την παρακάτω γραφική για το benchmark “blackscholes”.



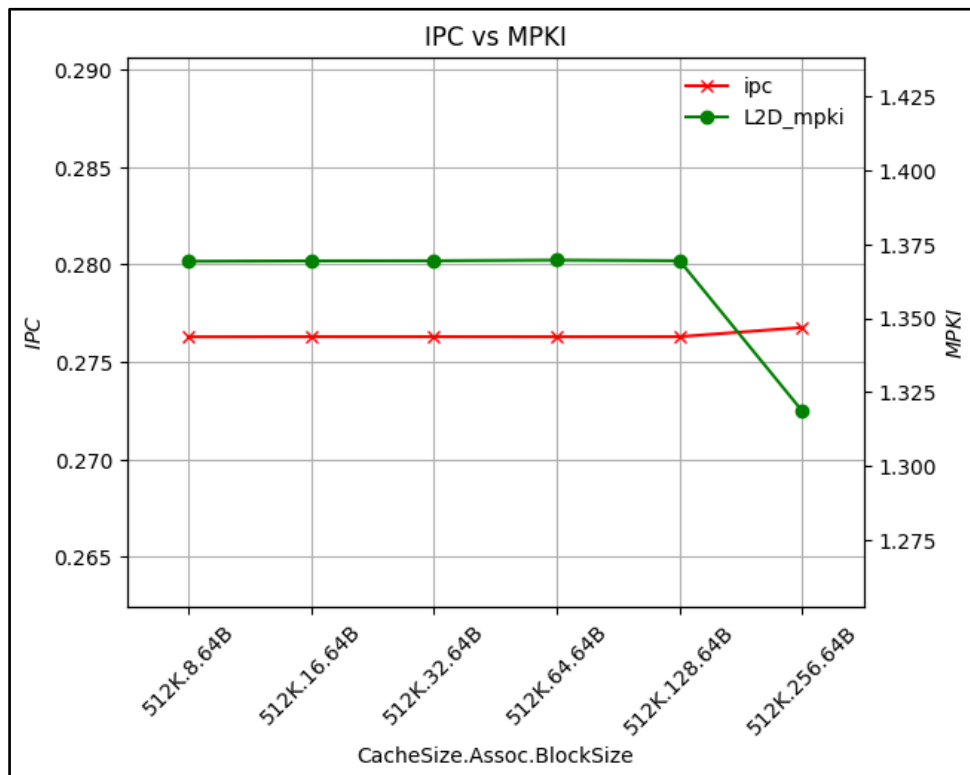
L₂ cache

Για την 1^η προσομοίωση θέτουμε **L₂ size** = 512 KB, **L₂ associativity** = 4, **L₂ cache block size** = 128 B.

L ₂ size	L ₂ Associativity	L ₂ Cache Block Size	Παράγοντας αύξησης κύκλου
512 KB	4	64 KB	1
512 KB	8	64 KB	1.05
512 KB	16	64 KB	1.05 · 1.05
512 KB	32	64 KB	1.05 · 1.05 · 1.05
512 KB	64	64 KB	1.05 · 1.05 · 1.05 · 1.05
512 KB	128	64 KB	1.05 · 1.05 · 1.05 · 1.05 · 1.05

Σχήμα 35: Διπλασιασμός L₂ associativity (5% αύξηση του κύκλου)

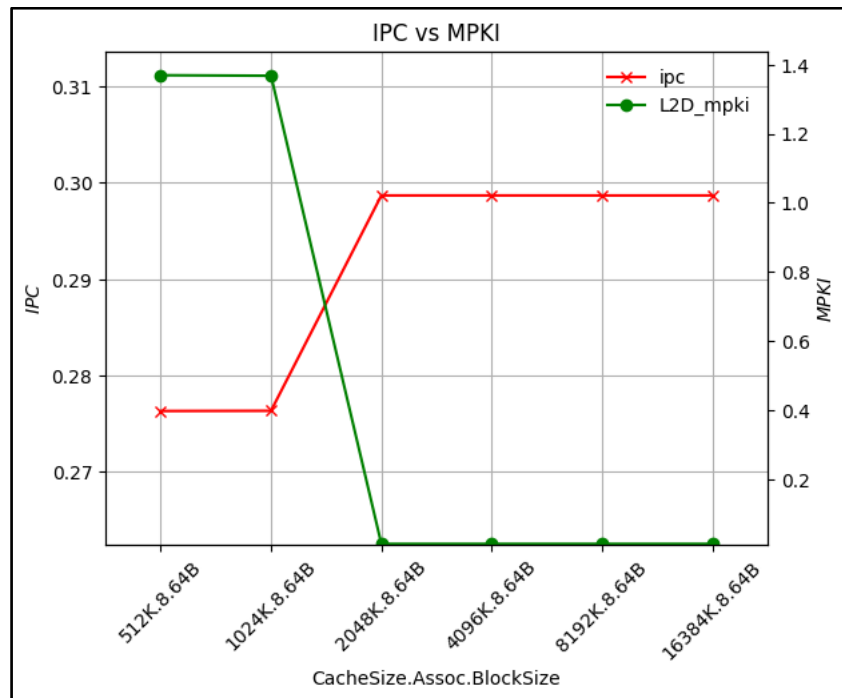
Τρέχουμε τον κώδικα “run2_l2a.sh” και εξάγουμε την παρακάτω γραφική για το benchmark “blackscholes”.



L ₁ size	L ₁ Associativity	L ₁ Cache Block Size	Παράγοντας αύξησης κύκλου
512 KB	8	64 KB	1
1024 KB	8	64 KB	1.15
2048 KB	8	64 KB	1.15 · 1.15
4096 KB	8	64 KB	1.15 · 1.15 · 1.15
8192 KB	8	64 KB	1.15 · 1.15 · 1.15 · 1.15
16384 KB	8	64 KB	1.15 · 1.15 · 1.15 · 1.15 · 1.15

Σχήμα 36: Διπλασιασμός L₂ associativity (15% αύξηση του κύκλου)

Τρέχουμε τον κώδικα “run2_l2s.sh” και εξάγουμε την παρακάτω γραφική για το benchmark “blackscholes”.



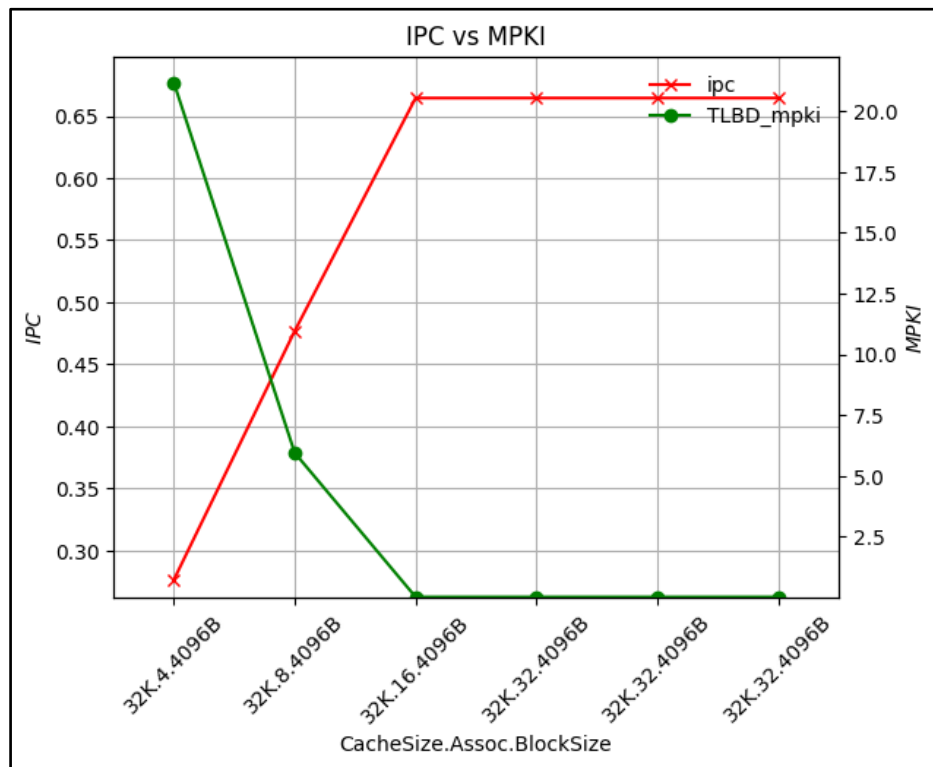
TLB

Για την 1^η προσομοίωση θέτουμε **TLB size = 512 KB**, **TLB associativity = 4**, **TLB cache block size = 128 B**.

TLB size	TLB Associativity	TLB Block Size	Παράγοντας αύξησης κύκλου
32 KB	4	4096 KB	1
32 KB	8	4096 KB	1.05
32 KB	16	4096 KB	1.05 · 1.05
32 KB	32	4096 KB	1.05 · 1.05 · 1.05
32 KB	64	4096 KB	1.05 · 1.05 · 1.05 · 1.05
32 KB	128	4096 KB	1.05 · 1.05 · 1.05 · 1.05 · 1.05

Σχήμα 37: Διπλασιασμός TLB associativity (5% αύξηση του κύκλου)

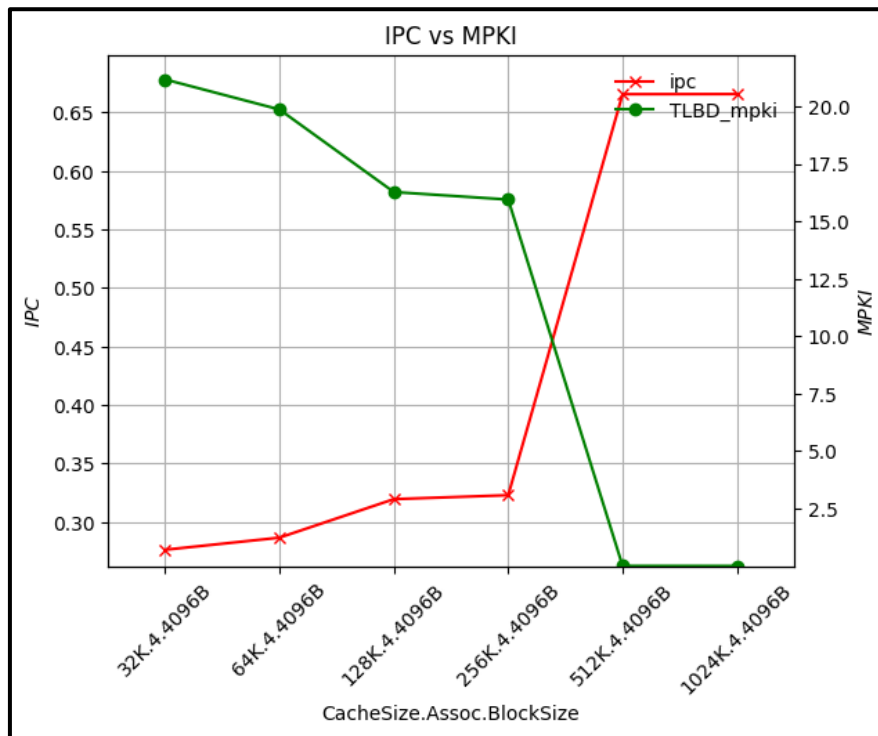
Τρέχουμε τον κώδικα “run2_tlba.sh” και εξάγουμε την παρακάτω γραφική για το benchmark “blackscholes”.



TLB size	TLB Associativity	TLB Block Size	Παράγοντας αύξησης κύκλου
32 KB	8	4096 KB	1
64 KB	8	4096 KB	1.15
128 KB	8	4096 KB	1.15 · 1.15
256 KB	8	4096 KB	1.15 · 1.15 · 1.15
512 KB	8	4096 KB	1.15 · 1.15 · 1.15 · 1.15
1024 KB	8	4096 KB	1.15 · 1.15 · 1.15 · 1.15 · 1.15

Σχήμα 38: Διπλασιασμός TLB associativity (15% αύξηση του κύκλου)

Τρέχουμε τον κώδικα “run2_tlbs.sh” και εξάγουμε την παρακάτω γραφική για το benchmark “blackscholes”.



Συμπεράσματα

Η ανάλυση των γραφημάτων που παρουσιάζονται δείχνει μια αξιοσημείωτη τάση στη συνδεσιμότητα και το μέγεθος των στοιχείων μνήμης. Έχει παρατηρηθεί ότι καθώς διπλασιάζεται η συσχέτιση μεταξύ των στοιχείων, παρατηρείται έντονη μείωση των Misses Per Kilo Instruction (MPKI), η οποία βελτιώνει σημαντικά την απόδοση της κρυφής μνήμης L1, της κρυφής μνήμης L2 και του Translation Lookaside Buffer (TLB). Η βελτίωση αυτή συνεχίζεται μέχρι να φθάσει σε ένα συγκεκριμένο σημείο διπλασιασμού, πέραν του οποίου δεν είναι εμφανή περαιτέρω αύξηση της απόδοσης. Ο διπλασιασμός της συσχέτισης συσχετίζεται επίσης με μια αύξηση 5% στους κύκλους επεξεργασίας. Από αυτά τα δεδομένα μπορεί να συναχθεί ότι υπάρχει άμεση συσχέτιση μεταξύ της αύξησης των κύκλων και της βελτίωσης της απόδοσης αυτών των στοιχείων μνήμης, μέχρι ένα προκαθορισμένο όριο. Μόλις ξεπεραστεί αυτό το όριο, οι πρόσθετες αυξήσεις στον αριθμό των κύκλων δεν συμβάλλουν σε περαιτέρω βελτίωση των επιδόσεων.

Όσον αφορά το μέγεθος, τα δεδομένα καταδεικνύουν ομοίως ότι η αύξηση του μεγέθους των στοιχείων μνήμης κατά το διπλάσιο έχει ευεργετικό αντίκτυπο στη μείωση του MPKI, ενισχύοντας έτσι την απόδοση βασικών στοιχείων μνήμης, όπως η κρυφή μνήμη L1, η κρυφή μνήμη L2 και η TLB. Αυτή η ενίσχυση της απόδοσης διατηρείται μέχρι να επιτευχθεί ένα ορισμένο όριο μεγέθους, μετά το οποίο η απόδοση εξισορροπείται. Αυτή η αύξηση του μεγέθους σχετίζεται με αύξηση του αριθμού των κύκλων κατά 15%, γεγονός που υποδεικνύει ότι οι βελτιώσεις των επιδόσεων συνδέονται με ένα συγκεκριμένο εύρος αύξησης των κύκλων. Πέρα από αυτό το συγκεκριμένο εύρος, οι πρόσθετες αυξήσεις στον αριθμό των κύκλων δεν αποφέρουν περαιτέρω οφέλη στην απόδοση.

Επιπλέον, κατά τη σύγκριση των επιδράσεων του διπλασιασμού της συσχετιστικότητας σε σχέση με το μέγεθος, καθίσταται εμφανές ότι ο διπλασιασμός της συσχετιστικότητας οδηγεί σε ουσιαστικότερη μείωση του MPKI σε σύγκριση με τις επιδράσεις που παρατηρούνται με την αύξηση του μεγέθους. Επιπλέον, η αύξηση του αριθμού των κύκλων είναι πιο σημαντική με την αύξηση του μεγέθους παρά με την αύξηση της συσχετιστικότητας. Ωστόσο, παρά τις διαφορές αυτές, και οι δύο προσαρμογές οδηγούν σε παρόμοια επίπεδα απόδοσης των στοιχείων μνήμης όταν υποβάλλονται σε συνθήκες εκτεταμένου κύκλου. Αυτή η συγκριτική ανάλυση αναδεικνύει μια σαφή σχέση μεταξύ των παραγόντων του διπλασιασμού της συσχετιστικότητας και του μεγέθους στη δυναμική των επιδόσεων των στοιχείων μνήμης.