



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Αρχιτεκτονική Υπολογιστών

Ιεραρχία Μνήμης

Χαράλαμπος Παπαδόπουλος
03120199

Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών
2η Άσκηση

1 Εισαγωγή

Στην παρούσα εργασία θα μελετήσουμε διάφορες αρχιτεκτονικές μνήμης χρησιμοποιώντας το εργαλείο προσομοίωσης PIN.

2 L2 Cache

Αρχικά, θα εξετάσουμε την επίδραση που έχουν στην επίδοση τα βασικά χαρακτηριστικά της L2, δηλαδή το μέγεθος, η συσχετιστικότητα (associativity) και το μέγεθος του cache block.

Οι συνδυασμοί που θα δοκιμάσουμε είναι οι εξής:

L2 size (KB)	L2 associativity	L2 cache block size (B)
256	4, 8	64, 128, 256
512	4, 8	
1024	8, 16	
2048	16	

Πίνακας 1: L2 Cache Configuration

Σημειώνεται πως η πολιτική αντικατάστασης, σε όλες τις περιπτώσεις, είναι LRU (Least Recently Used). Για την προσομοίωση χρησιμοποιούμε 7 benchmarks από τη σουίτα `SPEC_CPU2006`.

Για κάθε cache configuration θα χρησιμοποιήσουμε τον γεωμετρικό μέσο όρο των 7 benchmarks προκειμένου να εξάγουμε συμπεράσματα.

Οι μετρικές που θα χρησιμοποιήσουμε είναι οι εξής:

- **Instructions per cycle (IPC):** Μέτρο της απόδοσης ενός επεξεργαστή, που δείχνει τον μέσο αριθμό εντολών που εκτελούνται σε κάθε κύκλο του ρολογιού του επεξεργαστή. Υψηλότερο IPC σημαίνει γενικά καλύτερη απόδοση.

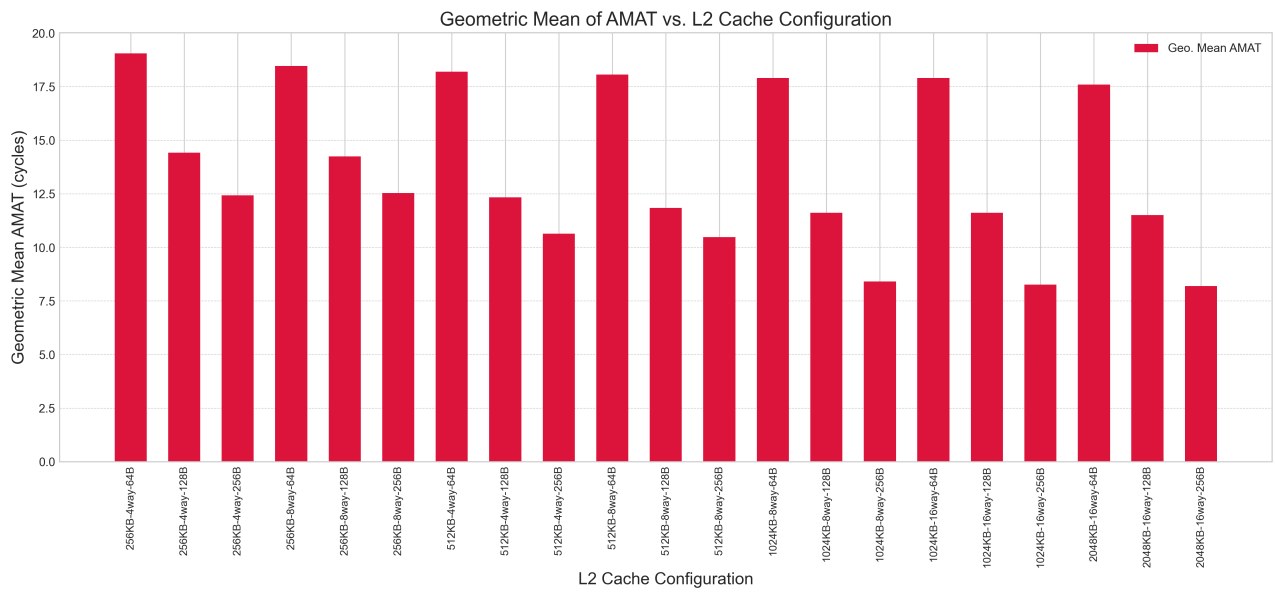
$$\text{Cycles} = \text{Instructions} + (L1_Accesses \times L1_hit_cycles) + (L2_Accesses \times L2_hit_cycles) + (Mem_Accesses \times Mem_acc_cycles)$$

- **Επιτυχία στην κρυφή μνήμη L1:** 1 κύκλος
($L1_hit_cycles = 1$)
- **Επιτυχία στην κρυφή μνήμη L2:** 15 κύκλοι
($L2_hit_cycles = 15$)
- **Πρόσβαση στην κύρια μνήμη:** 250 κύκλοι
($Mem_acc_cycles = 250$)

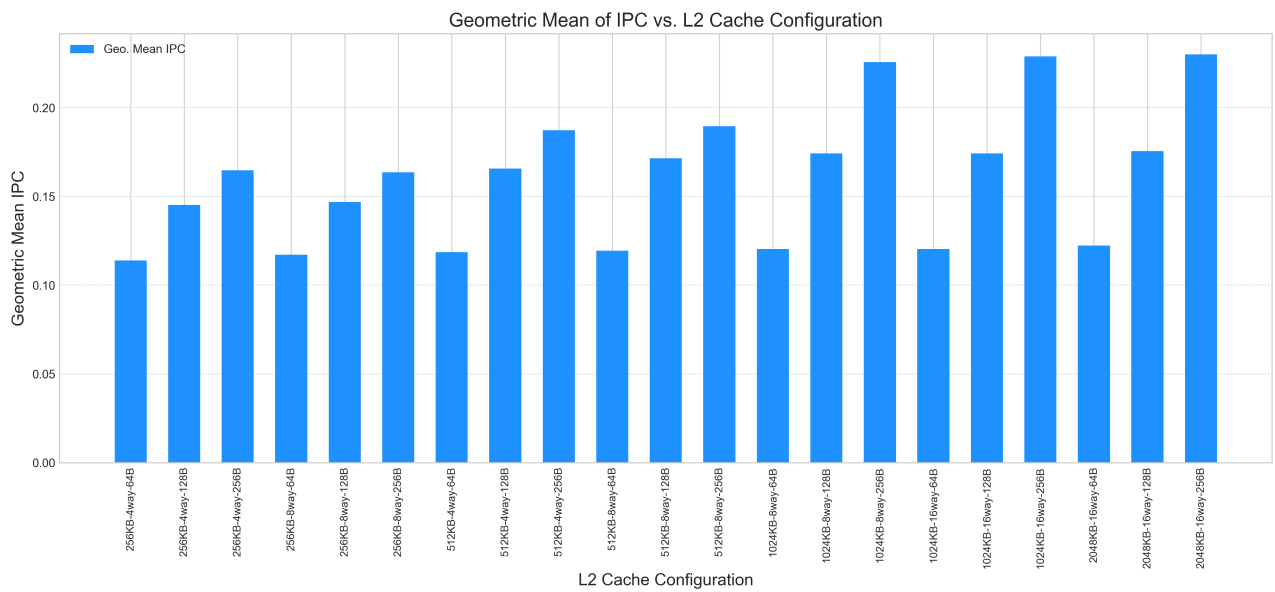
Για την εκτέλεση των προσομοιώσεων τροποποιήσαμε το αρχείο `run_benchmarks.sh`. Συγκεκριμένα, τροποποιήσαμε την παράμετρο *CONFS* = "256_4_64 256_4_128 256_4_256 256_8_64 256_8_128 ..."

προκειμένου να περιλαμβάνει τα configurations που ζητούνται.

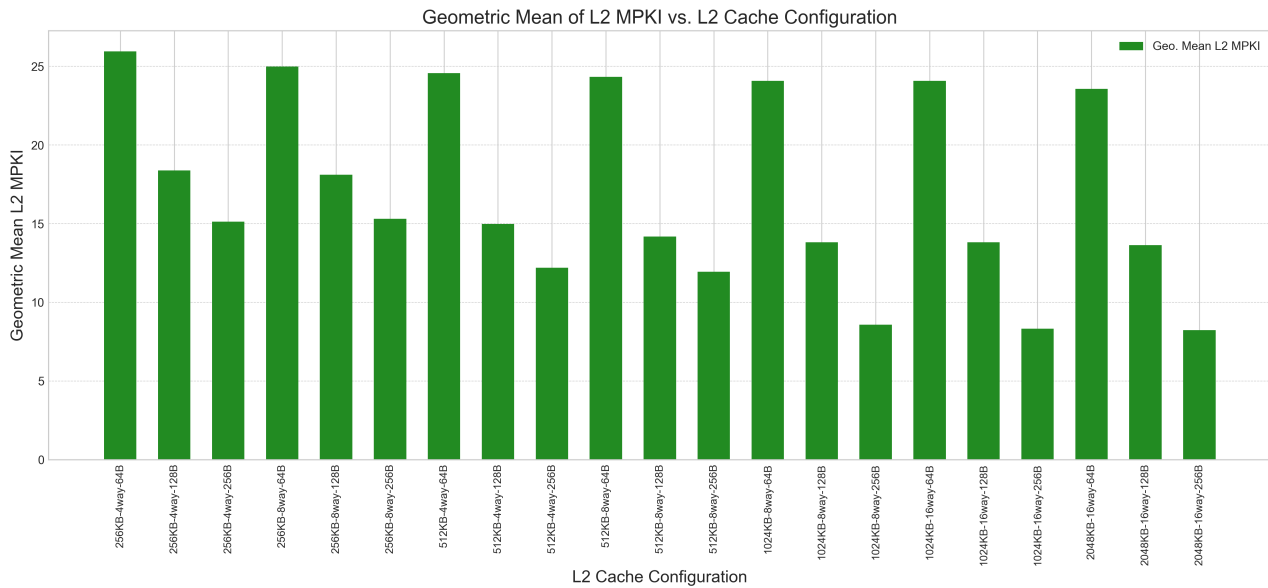
Λαμβάνουμε, λοιπόν, τα εξής αποτελέσματα:



Σχήμα 1: L2 Cache Configuration Results



Σχήμα 2: L2 Cache Configuration Results



Σχήμα 3: L2 Cache Configuration Results

Βάσει των αποτελεσμάτων προσομοίωσης, οι 4 κορυφαίες διαμορφώσεις κρυφής μνήμης L2:

- **Κρυφή μνήμη L2 2048KB, 16 – way associativity, μέγεθος block 256B**

- *IPC*: Υψηλότερο (περίπου 0.225)
- *AMAT*: Χαμηλότερος (περίπου 8.2 κύκλοι)
- *L2 MPKI*: Χαμηλότερο (περίπου 8.0)

Αυτή η διαμόρφωση επιδεικνύει την καλύτερη συνολική απόδοση σε όλες τις βασικές μετρικές.

- **Κρυφή μνήμη L2 1024KB, 8 – way associativity, μέγεθος block 256B**

- *IPC*: Δεύτερο υψηλότερο (περίπου 0.22)
- *AMAT*: Πολύ χαμηλός (περίπου 8.4 κύκλοι)
- *L2 MPKI*: Πολύ χαμηλό (περίπου 8.5)

Αυτή η διαμόρφωση προσφέρει απόδοση πολύ κοντά στην κορυφαία διαμόρφωση με μικρότερο μέγεθος κρυφής μνήμης.

- **Κρυφή μνήμη L2 1024KB, 16 – way associativity, μέγεθος block 256B**

- *IPC*: Τρίτο υψηλότερο (περίπου 0.215)
- *AMAT*: Πολύ χαμηλός (περίπου 8.4 κύκλοι)
- *L2 MPKI*: Πολύ χαμηλό (περίπου 8.5)

- **Κρυφή μνήμη L2 512KB, 8 – way associativity, μέγεθος block 256B**

- *IPC*: Υψηλό (περίπου 0.19)
- *AMAT*: Χαμηλός (περίπου 10.5 κύκλοι)
- *L2 MPKI*: Χαμηλό (περίπου 12.0)

Ακόμη και με μικρότερη χωρητικότητα 512KB, αυτή η διαμόρφωση με 8 – way associativity και μέγεθος block 256B παρέχει ισχυρό *IPC* και καλά χαρακτηριστικά πρόσβασης μνήμης, καθιστώντας την κορυφαία επιλογή μεταξύ των μικρότερων επιλογών κρυφής μνήμης.

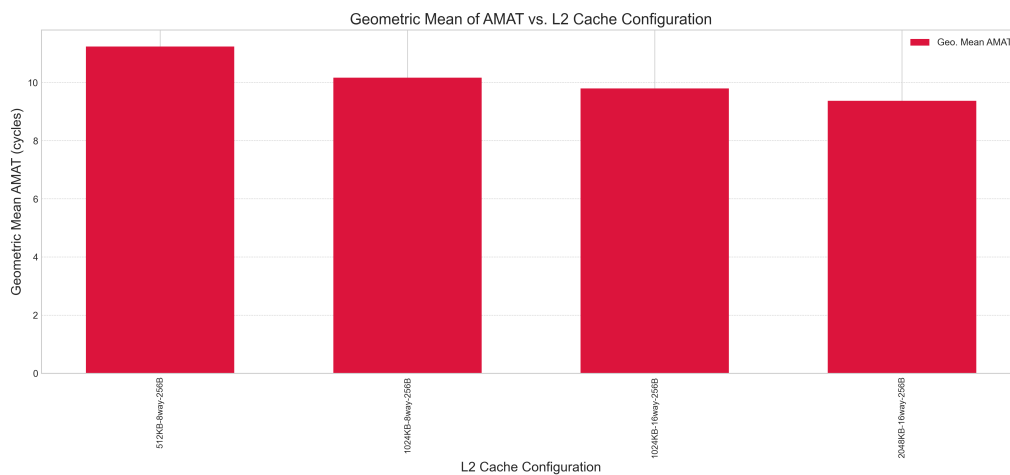
Αυτές οι επιλογές δίνουν προτεραιότητα σε υψηλότερες τιμές *IPC*, οι οποίες υποδεικνύουν καλύτερη συνολική απόδοση του επεξεργαστή υπό την παραδοχή σταθερού κύκλου ρολογιού και αριθμού εντολών. Αυτές οι διαμορφώσεις παρουσιάζουν επίσης ευνοϊκό (χαμηλότερο) Μέσο Χρόνο Πρόσβασης Μνήμης (*AMAT*) (που επηρεάζεται από τις καθυστερήσεις προσπέλασης) και *L2* Αποτυχίες Ανά Χίλιες Εντολές (*MPKI*) ενισχύοντας την αποδοτικότητά τους.

3 Πολιτικές Αντικατάστασης

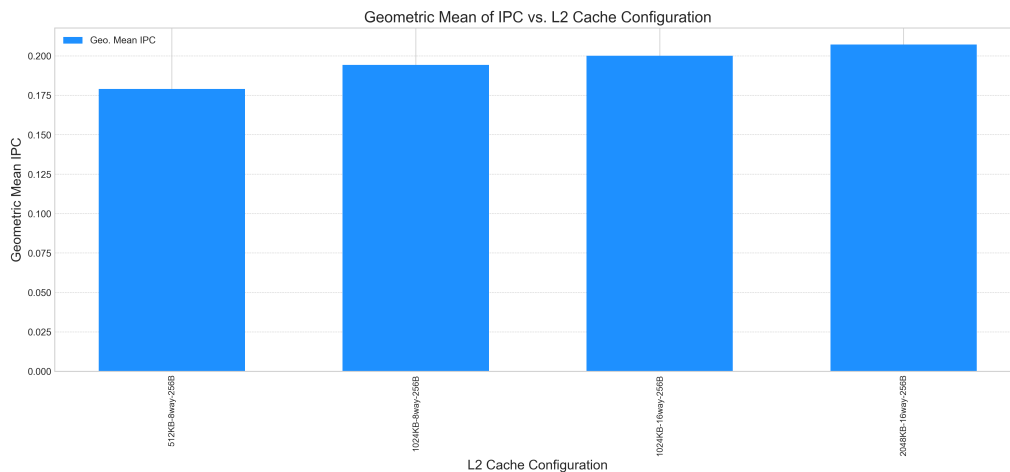
Στη συνέχεια, θα εξετάσουμε την επίδραση που έχουν στην επίδοση οι πολιτικές αντικατάστασης της L2 cache. Θα εξετάσουμε τις εξής πολιτικές αντικατάστασης:

- **Random:** Η πολιτική αυτή επιλέγει τυχαία σε κάθε set το block που θα αντικατασταθεί.
- **Least Frequently Used (LFU):** Η πολιτική Least Frequently Used (LFU) καταγράφει πόσες φορές χρησιμοποιείται κάθε block της cache και όταν χρειαστεί να αντικαταστήσει κάποιο, επιλέγει το block του set που έχει χρησιμοποιηθεί λιγότερο.
- **LRU Insertion Policy (LIP):** Η πολιτική LIP αποτελεί παραλλαγή της LRU πολιτικής, με μοναδική διαφορά το χαρακτηρισμό των blocks που εισάγονται κάθε φορά στην cache. Συγκεκριμένα, κάθε block που εισάγεται στο set εξαιτίας κάποιου miss, χαρακτηρίζεται ως το LRU block του set (αντί για το MRU, όπως συμβαίνει στην κλασική LRU πολιτική).
- **Static Re-reference Prediction (SRRIP):** Η πολιτική SRRIP προσπαθεί να αποτρέψει blocks που θα χρησιμοποιηθούν στο μακρινό μέλλον να καταλάβουν πολύτιμο χώρο στην cache. Για να το πετύχει αυτό, εκτιμά πότε θα χρησιμοποιηθεί ξανά κάθε block (θα έχουμε δηλαδή re-reference) και όταν χρειάζεται να αντικαταστήσει κάποιο, επιλέγει το block του set που έχει προβλεφθεί ότι θα χρησιμοποιηθεί στο πιο μακρινό μέλλον. Κάθε block έχει μήκος n bits, όπου n το associativity της cache.

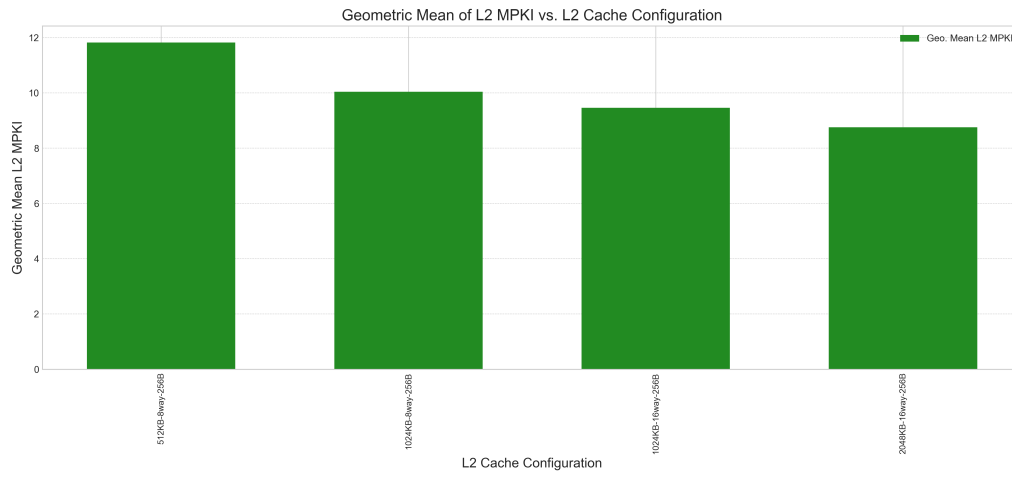
Βρίσκουμε, λοιπόν, τα εξής αποτελέσματα:



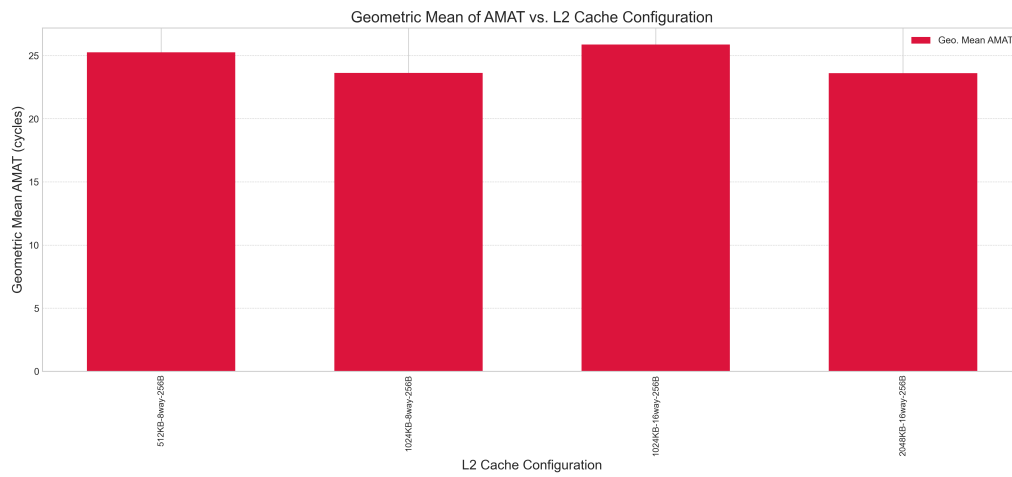
Σχήμα 4: Random Replacement Policy Results



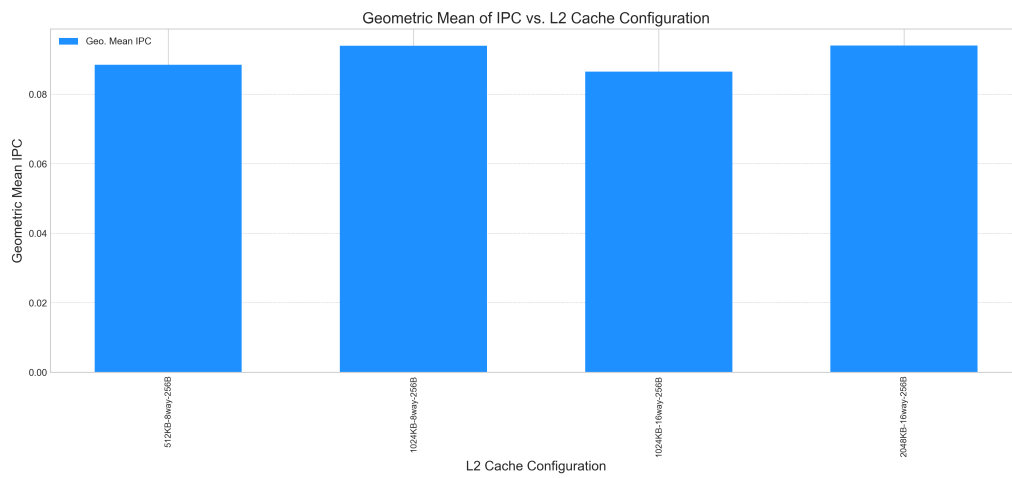
Σχήμα 5: Random Replacement Policy Results



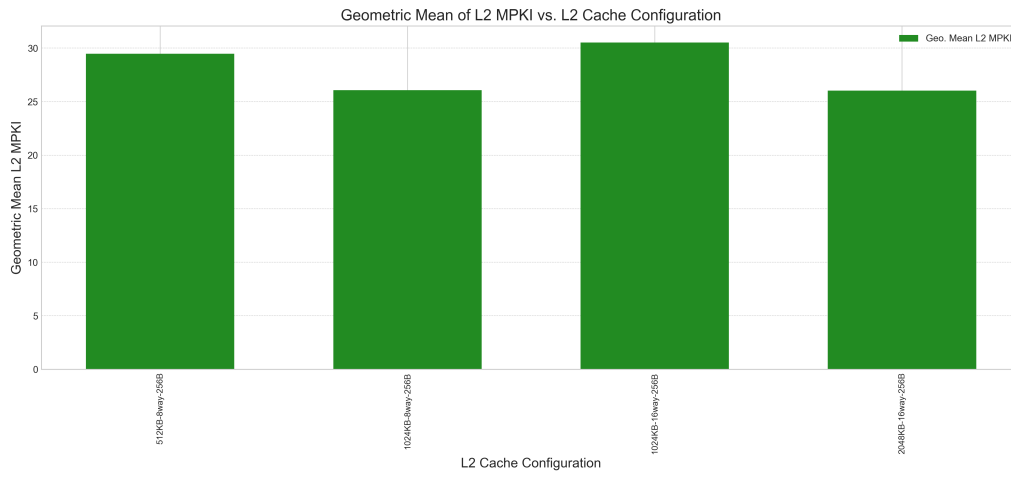
Σχήμα 6: Random Replacement Policy Results



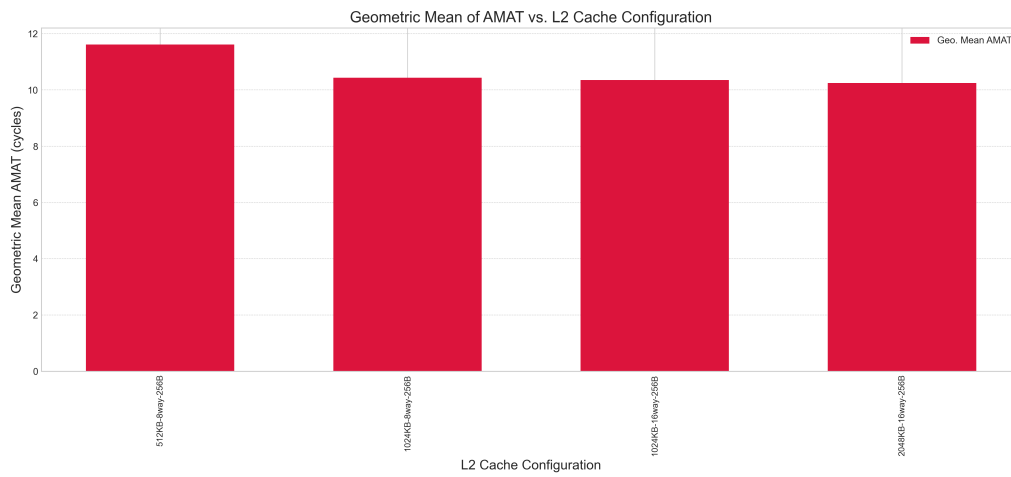
Σχήμα 7: LFU Replacement Policy Results



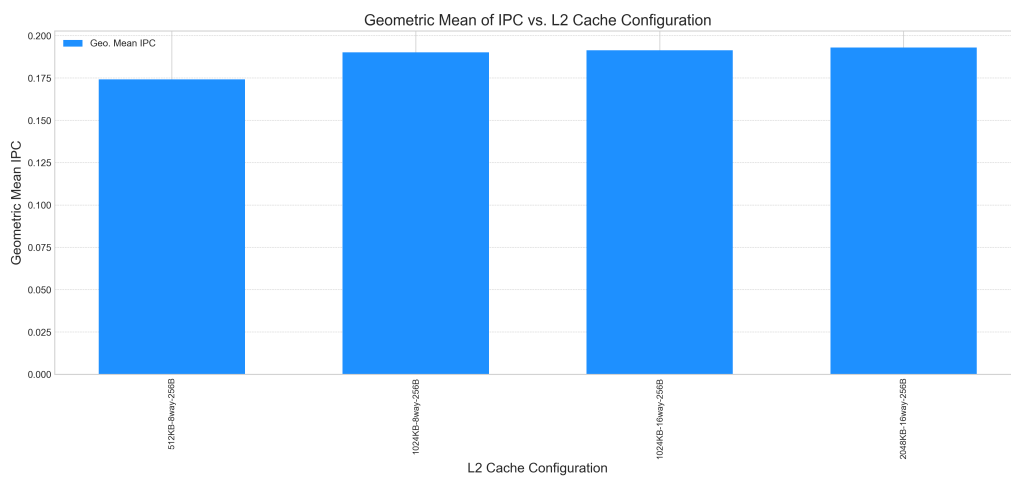
Σχήμα 8: LFU Replacement Policy Results



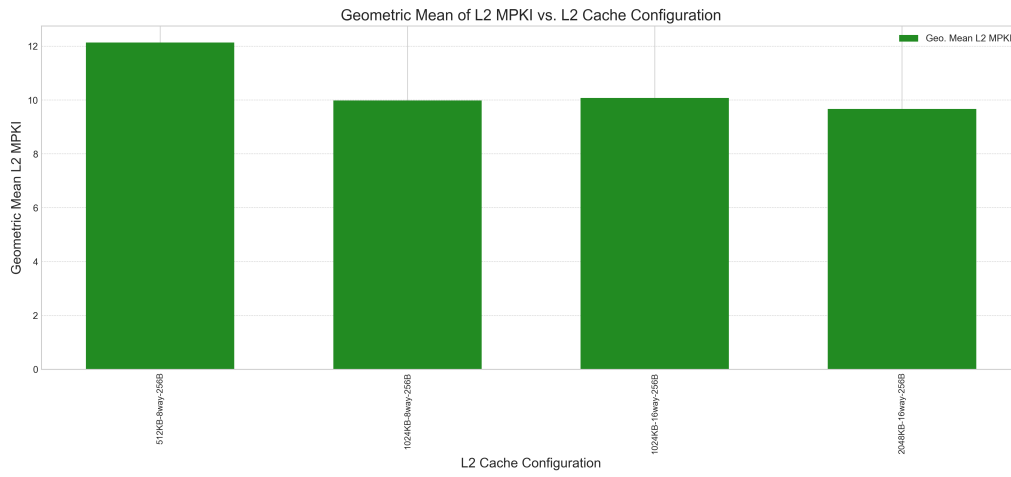
Σχήμα 9: LFU Replacement Policy Results



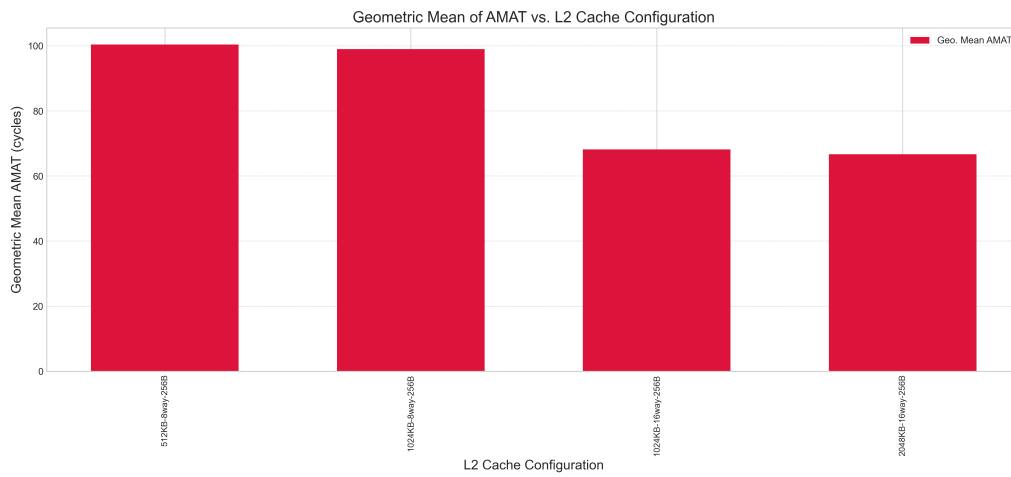
Σχήμα 10: LIP Replacement Policy Results



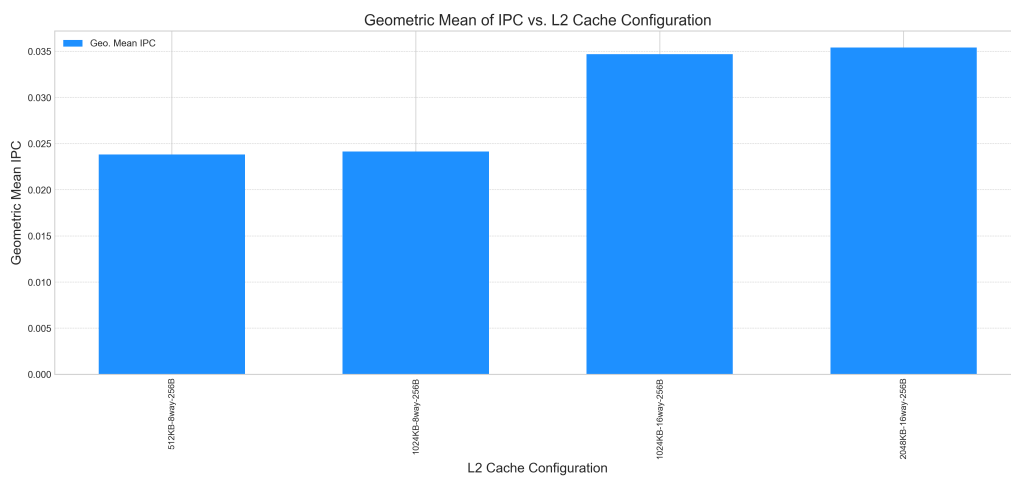
Σχήμα 11: LIP Replacement Policy Results



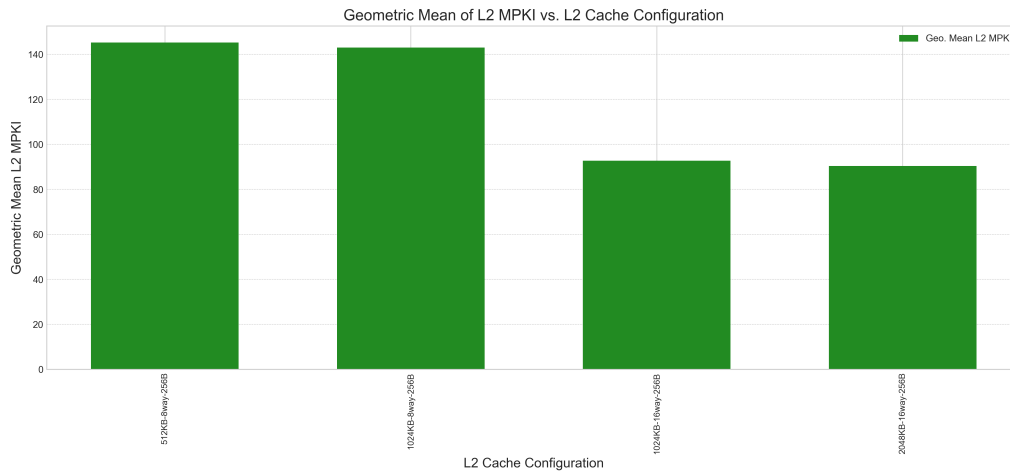
Σχήμα 12: LIP Replacement Policy Results



Σχήμα 13: SRRIP Replacement Policy Results



Σχήμα 14: SRRIP Replacement Policy Results



Σχήμα 15: SRRIP Replacement Policy Results

Συγκρίνοντας τις πέντε πολιτικές αντικατάστασης (LRU, Random, LFU, LIP, SRRIP) βάσει των παραπάνω αποτελεσμάτων:

- Η πολιτική **LRU** επιδεικνύει την καλύτερη συνολική απόδοση για τις εξεταζόμενες διαμορφώσεις L2 cache. Για παράδειγμα, για την διαμόρφωση C4 (2048KB-16way-256B), η LRU πετυχαίνει IPC ~ 0.225 , AMAT ~ 8.2 cycles και L2 MPKI ~ 8.0 . Αυτές οι τιμές είναι ανώτερες από όλες τις άλλες πολιτικές που δοκιμάστηκαν.
- Οι πολιτικές **Random** και **LIP** παρουσιάζουν την καλύτερη απόδοση μεταξύ των νέων πολιτικών που υλοποιήθηκαν, με σχεδόν ταυτόσημες επιδόσεις μεταξύ τους. Για την C4, πετυχαίνουν IPC ~ 0.205 , AMAT ~ 9.2 cycles και L2 MPKI ~ 8.8 . Αν και καλές, υπολείπονται της LRU.
- Η πολιτική **LFU** έχει σημαντικά χαμηλότερη απόδοση από τις LRU, Random και LIP. Οι τιμές IPC είναι πολύ χαμηλές (της τάξης του ~ 0.09), το AMAT είναι υψηλό (της τάξης των $\sim 14 - 15$ κύκλων) και το L2 MPKI επίσης υψηλό (της τάξης των ~ 15).
- Η πολιτική **SRRIP**, όπως υλοποιήθηκε και δοκιμάστηκε βάσει των αποτελεσμάτων της αναφοράς, έδειξε την χειρότερη απόδοση με πολύ μεγάλη διαφορά. Οι τιμές IPC είναι εξαιρετικά χαμηλές (~ 0.03), ενώ οι τιμές AMAT και L2 MPKI είναι πάρα πολύ υψηλές.

Επιλογή Πολιτικής: Με βάση τα αποτελέσματα των προσομοιώσεων, η πολιτική **LRU (Least Recently Used)** θα ήταν η βέλτιστη επιλογή για υλοποίηση στο σύστημα. Παρέχει σταθερά την υψηλότερη απόδοση (υψηλότερο IPC, χαμηλότερο AMAT και L2 MPKI) σε σύγκριση με όλες τις άλλες πολιτικές.

Εάν η LRU δεν ήταν διαθέσιμη για κάποιο λόγο (π.χ., λόγω πολυπλοκότητας υλοποίησης σε ορισμένα σενάρια), τότε η πολιτική Random θα ήταν η επόμενη καλύτερη επιλογή λόγω της καλής της απόδοσης σε σχέση με την απλότητά της. Ωστόσο, εφόσον η LRU αξιολογήθηκε και αποδίδει καλύτερα, παραμένει η προτεινόμενη.

Τέλος, αξίζει να σημειωθεί, πως χρησιμοποιώντας τις πολιτικές LFU και LIP είχαμε πολύ ταχύτερες επιδόσεις. Παρόλα αυτά, είναι δύσκολο να εξαγάγουμε συμπεράσματα για την απόδοσή τους, καθώς ενδέχεται να επηρεάζονται από την αρχιτεκτονική του συστήματος καθώς και άλλους παράγοντες που σχετίζονται με το περιβάλλον εκτέλεσης.