

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΗΛΕΚΤΡΟΛΟΓΟΙ ΜΗΧΑΝΙΚΟΙ ΚΑΙ ΜΗΧΑΝΙΚΟΙ ΥΠΟΛΟΓΙΣΤΩΝ

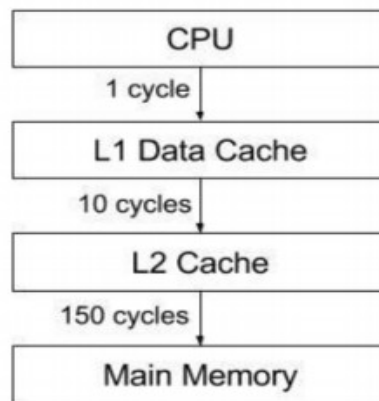
ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΚΟΥΤΣΟΓΙΑΝΝΑΚΟΠΟΥΛΟΣ ΙΩΑΝΝΗΣ
03109061

Εισαγωγή

Η άσκηση αυτή αποτελείται από δύο μέρη. Στο πρώτο μέρος θα μελετήσουμε την επίδραση των βασικότερων παραμέτρων ιεραρχίας κρυφής μνήμης στην απόδοση της εφαρμογής. Στο δεύτερο, για μία συγκεκριμένη παραμετροποίηση της της ιεραρχίας μνήμης ο τρόπος που μεταβάλλονται διάφορες μετρικές απόδοσης στο χρόνο.

Χρησιμοποιήσαμε την παρακάτω ιεραρχία κρυφής μνήμης:



Για τον υπολογισμό της επίδοσης των εφαρμογών που χρησιμοποιούνται στις προσομοιώσεις χρησιμοποιείται ένα απλό μοντέλο όπου ξερούμε ότι κάθε εντολή απαιτεί 1 κύκλο για την εκτέλεσή της (IPC=1). Επιπρόσθετα, οι εντολές που πραγματοποιούν πρόσβαση στη μνήμη (είτε load είτε store) προκαλούν επιπλέον καθυστερήσεις ανάλογα με το πού βρίσκονται τα δεδομένα τους, όπως φαίνεται στο σχήμα. Συνολικά, ο αριθμός των κύκλων υπολογίζεται ως:

$$\text{Cycles} = \text{Inst} + \text{L1_Accesses} * \text{L1_hit_cycles} + \text{L2_Accesses} * \text{L2_hit_cycles} + \text{Mem_Accesses} * \text{Mem_acc_cycles}$$

7.1 Μελέτη επίδρασης παραμέτρων ιεραρχίας μνήμης στην απόδοση της εφαρμογής

7.1.1 L1 cache

Για όλες τις περιπτώσεις που εξετάζονται στο πείραμα αυτό, οι παράμετροι της L2 cache θα διατηρηθούν σταθερές και συγκεκριμένα ίσες με:

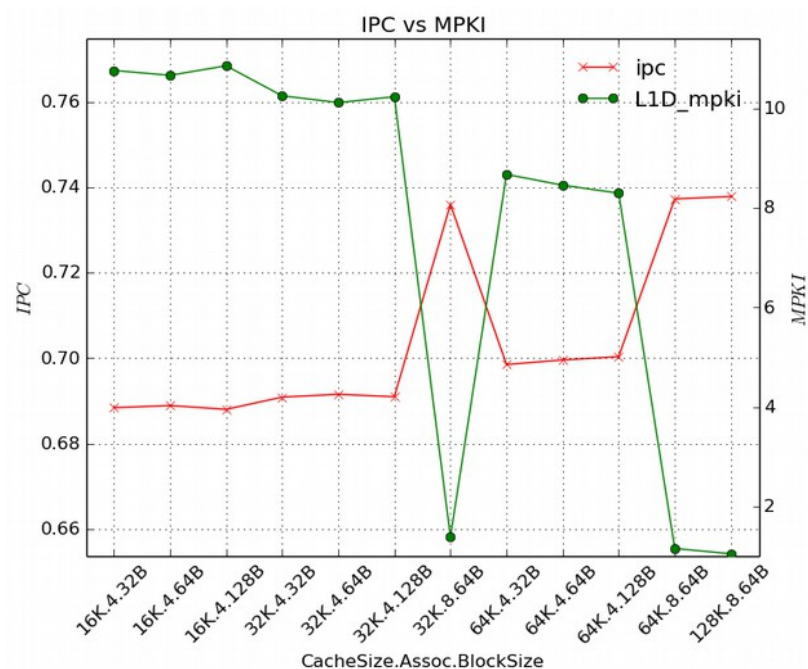
>L2 size = 1024 KB >L2 associativity = 8 >L2 block size = 128B

Εκτελέσαμε τα benchmarks για τις παρακάτω L1 caches:

L1 size	L1 associativity	L1 cache block size
16KB	4	32B, 64B, 128B
32KB	4	32B, 64B, 128B
32KB	8	64B

L1 size	L1 associativity	L1 cache block size
64KB	4	32B, 64B, 128B
64KB	8	64B
128KB	8	64B

1. **blackscholes:** χρησιμοποιείται για οικονομική ανάλυση προσομοιωμένοι με προβλήματα μερικων παραγωγων

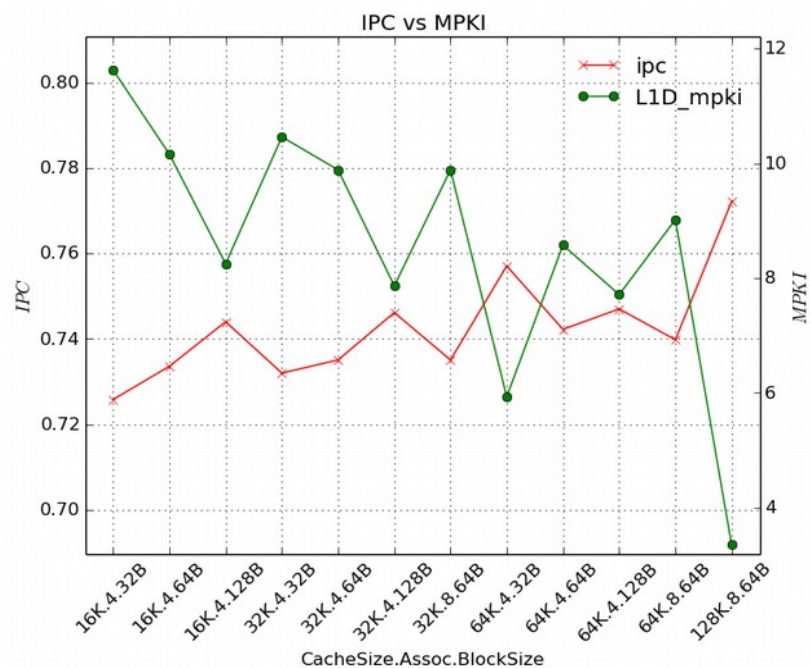


Στο σχήμα παρατηρούμε χαμηλές τιμές σε misses per kilo instructions όταν η cache είναι 8-way associative και το block size είναι 64 B, ανεξαρτήτως του μεγέθους της cache. Επίσης παρατηρούμε μείωση των mpki όσο αυξάνεται το μέγεθος της cache, καθώς όπως είναι φυσικό, όσο αυξάνεται το μέγεθος της cache τόσο

μειώνεται ο αριθμός των mprki. Το mprki για caches μεγέθους κάτω των 64KB είναι αυξημένο, ωστόσο χωρίς αυτό να είναι σε πραγματικά υψηλά επίπεδα - μικρότερο από 5% - επομένως το συγκεκριμένο πρόγραμμα δεν έχει μεγάλες απαιτήσεις σε μέγεθος cache.

Για τις instructions per cycle παρατηρούμε ότι όσο πιο μικρό είναι το mprki τόσο μεγαλύτερο είναι το ipc και αντίστροφα. Αυτό δικαιολογείται από το γεγονός ότι όσο μικρότερος είναι ο αριθμός των misses τόσο λιγότερους κύκλους απαιτεί κάθε εντολή, καθώς δε χρειάζεται να προσπελαστούν κατώτερα επίπεδα μνήμης. Ωστόσο αυτό εξαρτάται από το είδος της εφαρμογής, καθώς σε άλλη περίπτωση π.χ. αύξηση του block size θα μπορούσε να επιφέρει μείωση του mprki (καλύτερη αξιοποίηση του χώρου της cache) αλλά και ταυτόχρονα μείωση του ipc καθώς, μεγαλύτερα blocks ισοδυναμούν με αύξηση του χρόνου προσπέλασης.

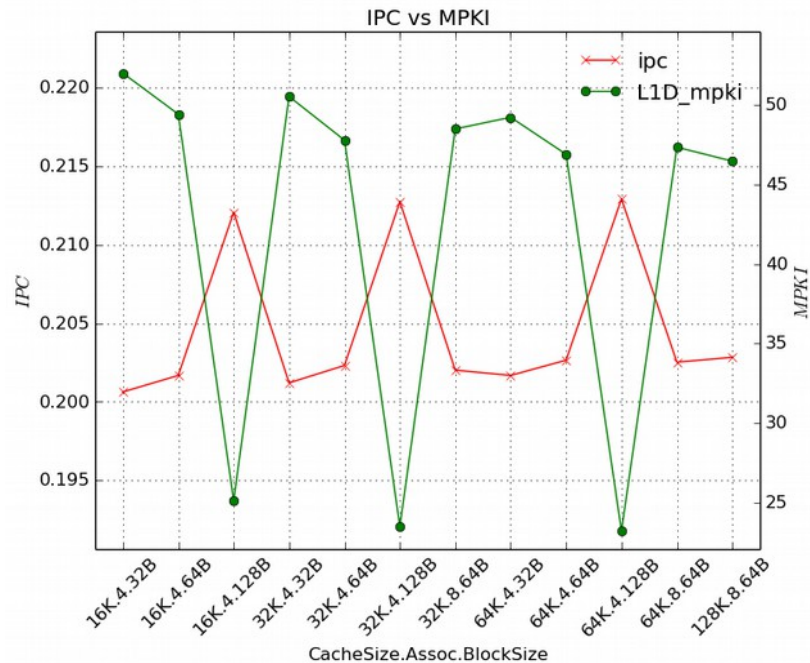
2. **Bodytrack:** χρησιμοποιείται για την περιγραφή της κατάτασης του ανθρώπινου σώματος



Παρατηρούμε ότι αύξηση του μεγέθους του block size επιφέρει μείωση του mprki και ταυτόχρονα αύξηση του ipc. Ωστόσο αυτό παρατηρείται αυστηρά σε συνδυασμούς όπου το cache size είναι κάτω των 64KB. Αλλαγή στην associativity δεν επιφέρει εδώ σημαντικές αλλαγές σε κανένα από τα δύο μελετούμενα μεγέθη, αν και παρατηρείται σε πολύ μικρό ποσοστό αύξηση του mprki και μείωση του ipc. Μακροπρόθεσμα το μέγεθος της cache ελαχιστοποιεί το mprki και αυξάνει το ipc χωρίς όμως να μπορούμε να καταλήξουμε με ασφάλεια στο συμπέρασμα αυτό για το ipc καθώς ανάλογα με το μέγεθος του block μπορεί να αυξηθεί πολύ ο χρόνος αναζήτησης με αποτέλεσμα τη μείωση του ipc.

3. Canneal

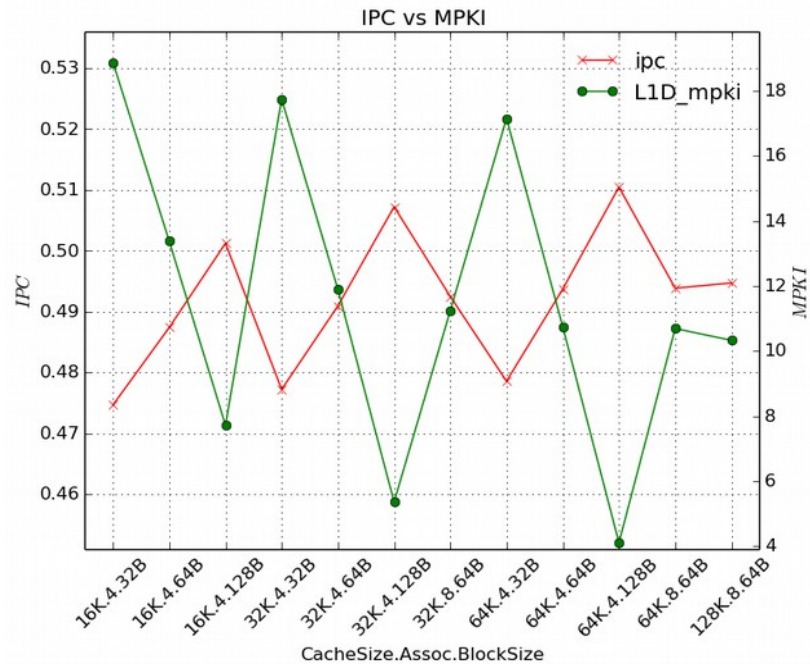
Πρόκειται για ένα πρόγραμμα που χρησιμοποιείται για την ελασιστοποίηση του κόστους κατασκευής και σχεδιασμού μικροκυκλωμάτων.



Παρατηρούμε ότι αύξηση του μεγέθους του block size επιφέρει μείωση του mprki και ταυτόχρονα αύξηση του ipc. Αλλαγή στο associativity από 4 σε 8 και πάλι επιφέρει σε πολύ μικρό ποσοστό αύξηση του mprki και καμία μεταβολή στο ipc. Τέλος αύξηση του μεγέθους της cache επιφέρει μείωση του mprki και με τη σειρά του αύξηση του ipc ωστόσο νωρίς να επιφέρει μεγάλη διαφοροποίηση. Παρατηρούμε ότι αύξηση του μεγέθους του block size επιφέρει ακόμα μεγαλύτερη μείωση του mprki και ταυτόχρονα αύξηση του ipc. Αλλαγή στην associativity από 4 σε 8 και πάλι επιφέρει αμελητέα αύξηση του mprki και καμία μεταβολή στο ipc. Τέλος αύξηση του μεγέθους της cache επιφέρει μείωση του mprki και με τη σειρά του αύξηση του ipc.

4. Facesim

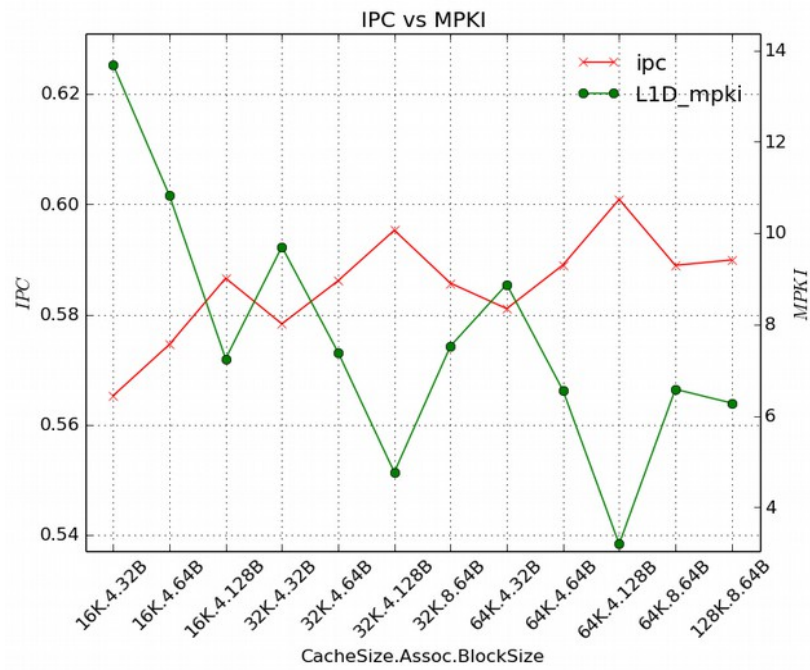
Πρόκειται για ένα πρόγραμμα που χρησιμοποιείται για την προσομοίωση κινήσεων ενός προσώπου



Παρατηρούμε ότι αύξηση του μεγέθους του block size επιφέρει μεγάλη μείωση του mprk και ταυτόχρονα αύξηση του ipc. Αλλαγή στην associativity από 4 σε 8 και πάλι επιφέρει σε πολύ μικρό ποσοστό αύξηση του mprk και καμία μεταβολή του ipc. Τέλος αύξηση του μεγέθους της cache επιφέρει μείωση του mprk και με τη σειρά του αύξηση του ipc ωστόσο χωρίς να επιφέρει μεγάλη διαφοροποίηση, με εξαίρεση την περίπτωση των 128KB, όπου φαίνεται πιο έντονα

5. ferret

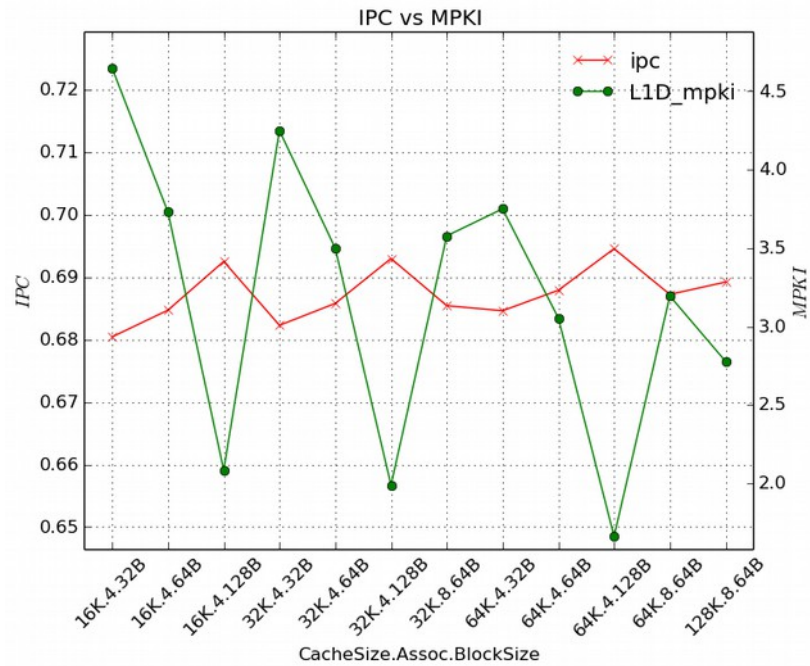
Πρόκειται για ένα πρόγραμμα που χρησιμοποιείται για την αναζήτηση παρόμοιων περιεχομένων.



Παρατηρούμε ότι αύξηση του μεγέθους του block size επιφέρει ακόμα μεγαλύτερη μείωση του mprki και ταυτόχρονα αύξηση του ipc. Αλλαγή στην associativity από 4 σε 8 και πάλι επιφέρει αμελητέα αύξηση του mprki και καμία μεταβολή στο ipc. Τέλος αύξηση του μεγέθους της cache επιφέρει μείωση του mprki και με τη σειρά του αύξηση του ipc.

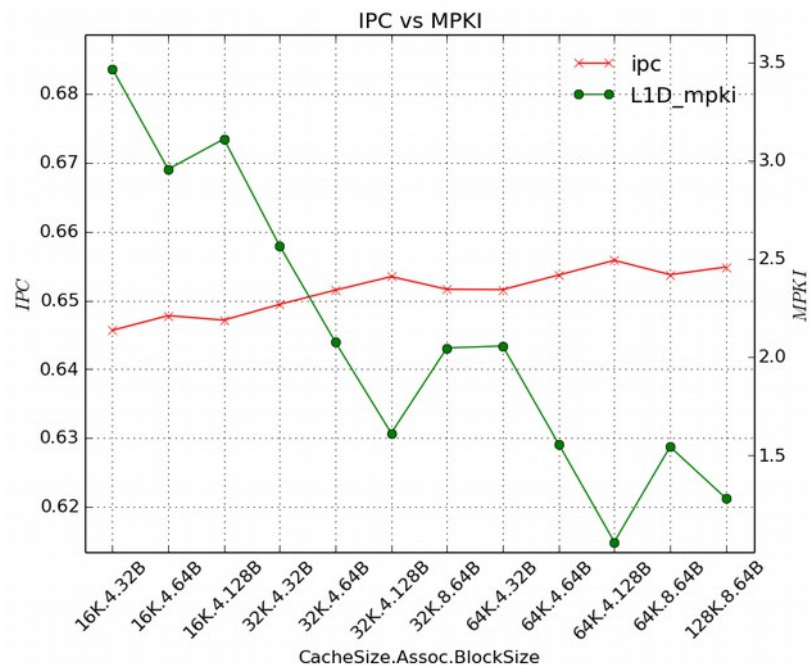
6. Fluidanimate

Πρόκειται για ένα πρόγραμμα που χρησιμοποιείται για τη μελέτη της δυναμικής των υγρών μέσω της μεθόδου SPH.



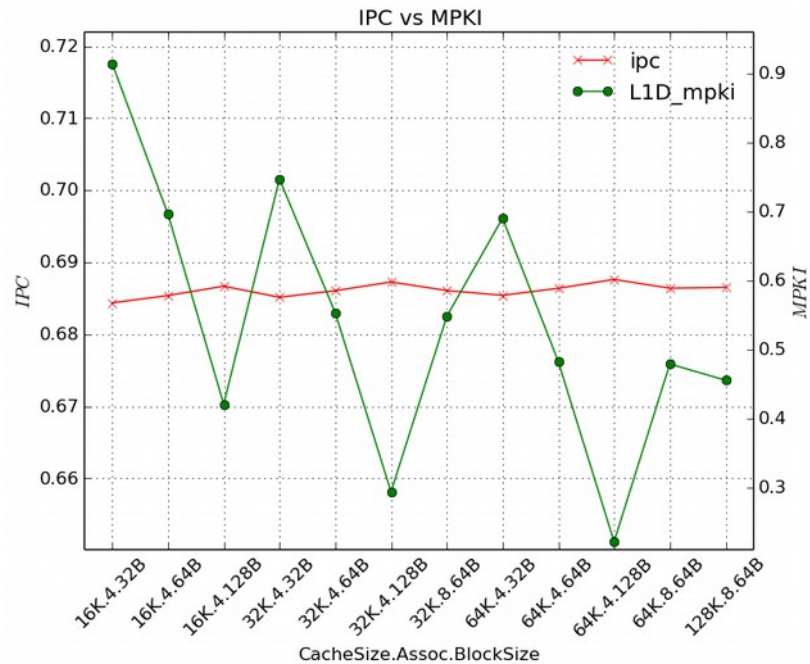
Οι παρατηρήσεις είναι παρόμοιες με αυτές για το ferret, με μόνη διαφορά τον μικρότερο αριθμό mpmi σε όλες τις περιπτώσεις, με αποτέλεσμα να οδηγείται σχεδόν σε εξάλειψη σε πολλές περιπτώσεις και ως αποτέλεσμα τον αυξημένο αριθμό ipc.

7. Freqmine: εξαγωγή συχνά χρησιμοποιούμενων σετ αντικειμένων



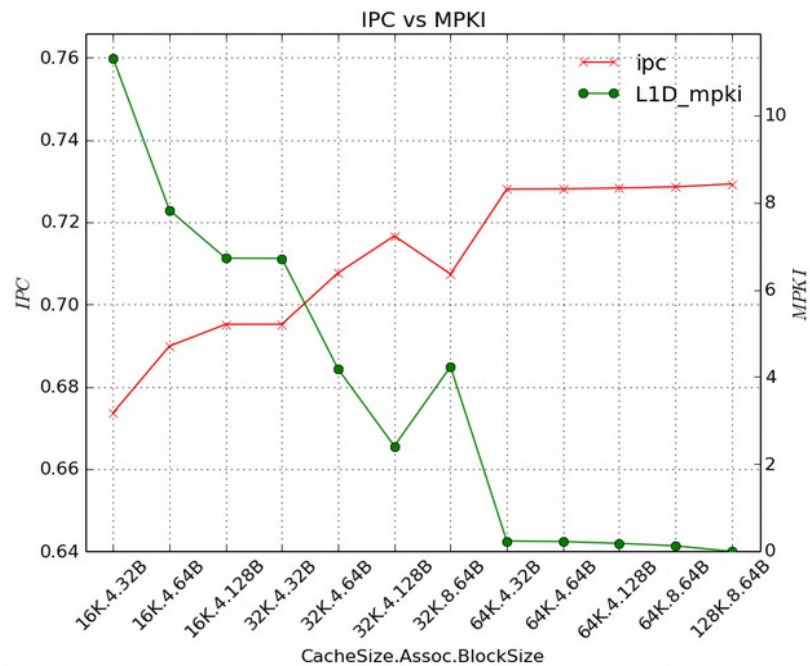
Καταλήγουμε σε παρόμοιες παρατηρήσεις με το fluadamine σχετικά με το mpki και το ipc. Ωστόσο παρότι παρατηρούνται οι μεταβολές σε σχέση με το ipc είναι σαφώς μικρότερη η μεταβολή του

8. **rtview**: Πρόκειται για ένα πρόγραμμα που χρησιμοποιείται για real-time raytracing



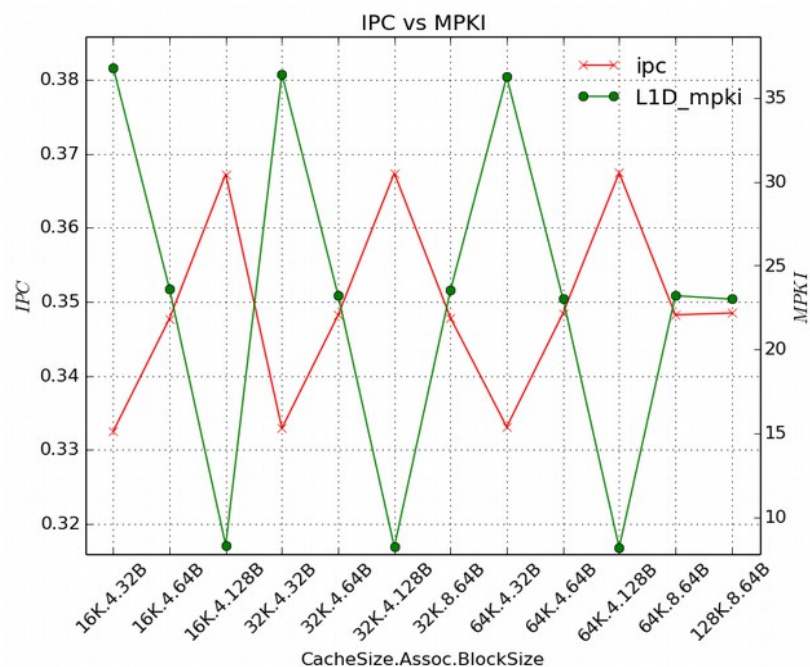
Καταλήγουμε σε παρόμοιες παρατηρήσεις με το fluadamine σχετικά με το mpki. Ωστόσο παρότι παρατηρούνται οι ίδιες μεταβολές σχετικά με το ipc είναι σαφώς μικρότερη η μεταβολή του, με αποτέλεσμα να είναι σχεδόν στατικό.

9. **Swaptions** : Πρόκειται για ένα πρόγραμμα που χρησιμοποιείται για τη διαδικτυακή ομαδοποίηση input streams.



Καταλήξαμε σε παρόμοιες παρατηρήσεις με το ferret σχετικά με τα mпки και ipc, χωρίς ωστόσο να παίζει ρόλο το μέγεθος της cache.

10. **Streamcluster:** Πρόκειται για ένα πρόγραμμα που χρησιμοποιείται για την καταγραφή των swaptions.



Καταλήγουμε σε παρόμοιες παρατηρήσεις με το ferret σχετικά με τα mrci και irc, αλλά καθώς το μέγεθος της cache επιδρά σε μεγάλο βαθμό οδηγούμαστε στο συμπέρασμα πως το πρόγραμμα έχει μικρές απαιτήσεις μνήμης με αποτέλεσμα την εξάλειψη του mrci και την σταθεροποίηση του irc.

7.1.2 L2 cache

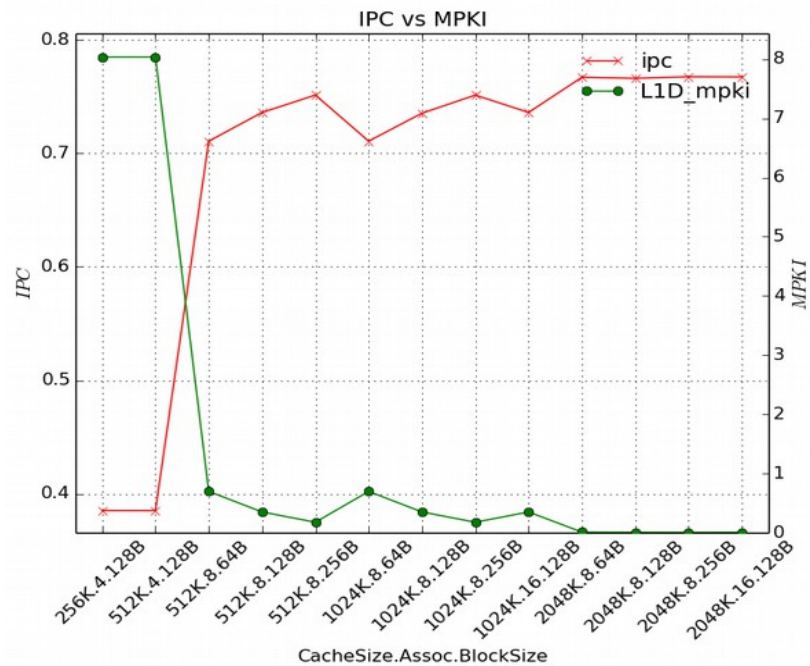
Για όλες τις περιπτώσεις που εξετάζονται στο πείραμα αυτό, οι παράμετροι της L1 cache θα διατηρηθούν σταθερές και συγκεκριμένα ίσες με:

>L size = 32 KB >L1 associativity = 8 >L1 block size = 64B

Εκτελέσαμε τα benchmarks για τις παρακάτω L1 caches

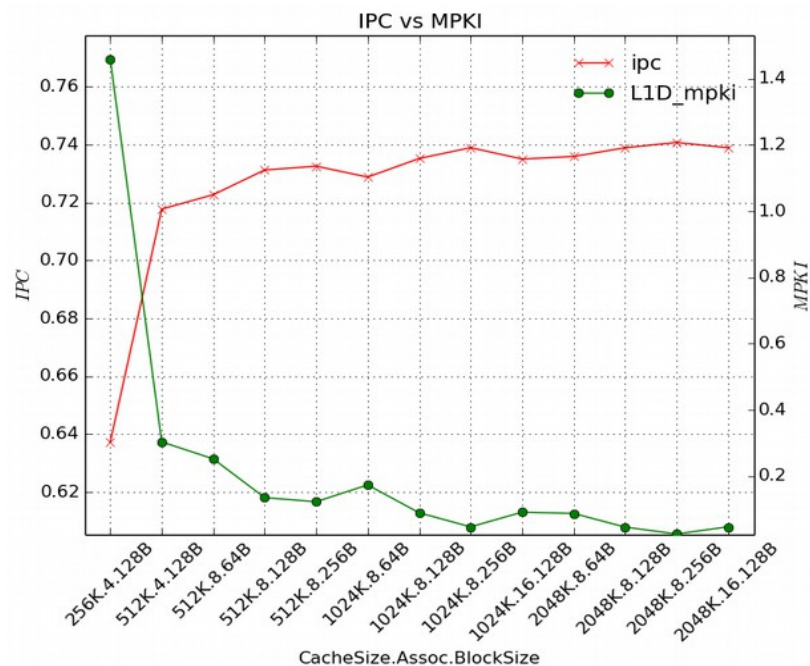
L2 size	L2 associativity	L2 cache block size
256 KB	4	128B
512 KB	4	128B
512 KB	8	64B, 128B, 256B
1024 KB	8	64B, 128B, 256B
1024 KB	16	128B
2048 KB	8	64B, 128B, 256B
2048KB	16	128B

1. blackscholes



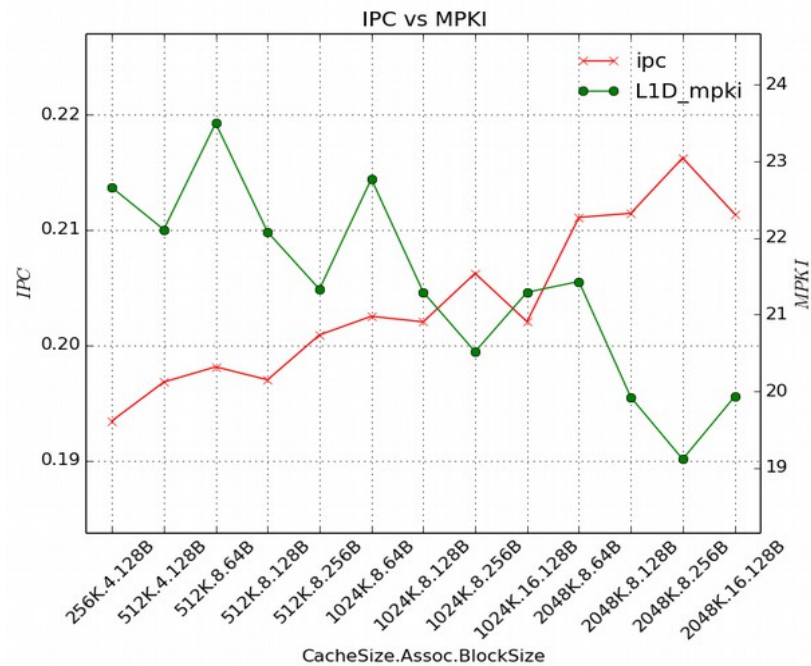
Παρατηρούμε ότι όσο αυξάνεται το block size αυξάνεται το ipc και μειώνονται δραστικά τα misses, ανεξάρτητα από τα υπόλοιπα στοιχεία της cache

2. bodytrack



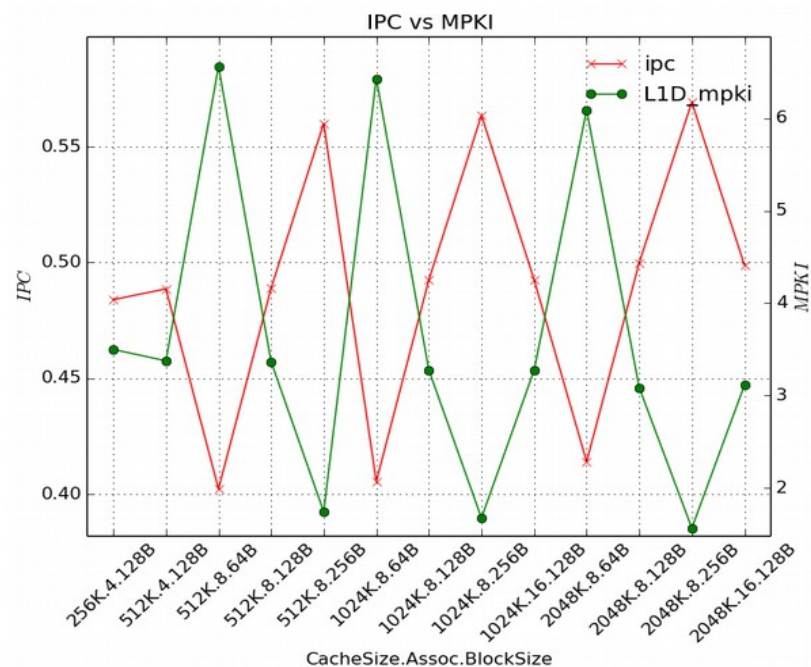
Η εφαρμογή έχει χαμηλές απαιτήσεις σε cache και όπως φαίνεται και στο διάγραμμα είναι κυρίως compulsory.

3. canneal



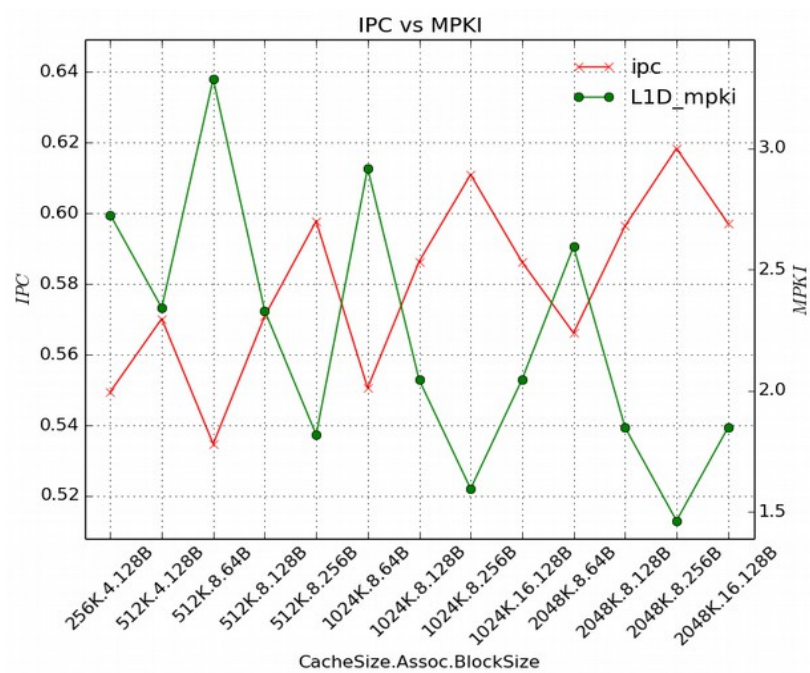
Από το διάγραμμα παρατηρούμε πως η εφαρμογή επηρεάζεται κυρίως από την αλλαγή μεγέθους του block (κυρίως compulsory misses), αν και υπάρχουν και κάποια capacity misses. Φαίνεται επίσης πως δεν υπάρχουν ιδιαίτερα conflict misses

4. facesim



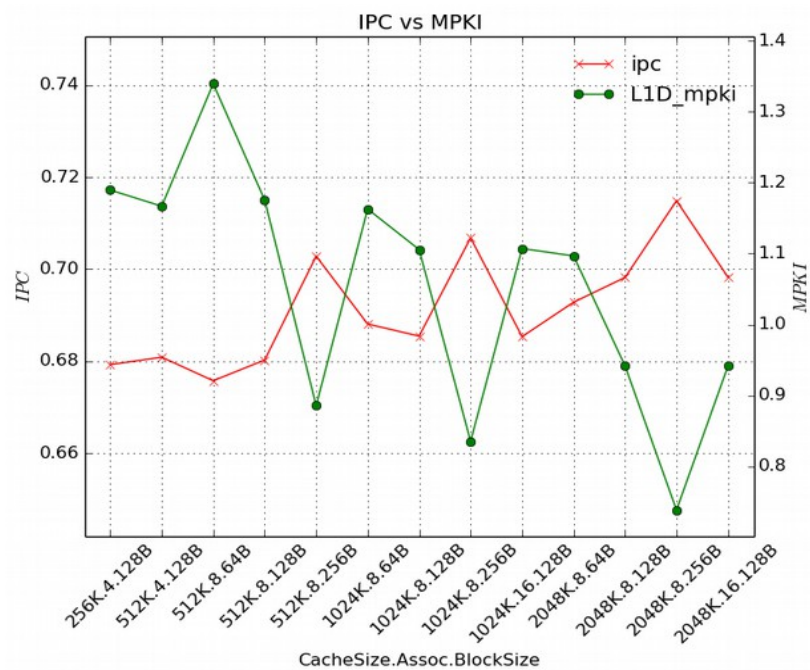
Δεν παίζει ιδιαίτερο ρόλο το μέγεθος της cache, ούτε το associativity. Ωστόσο το μέγεθος του block είναι ιδιαίτερα σημαντικό.

5. ferret



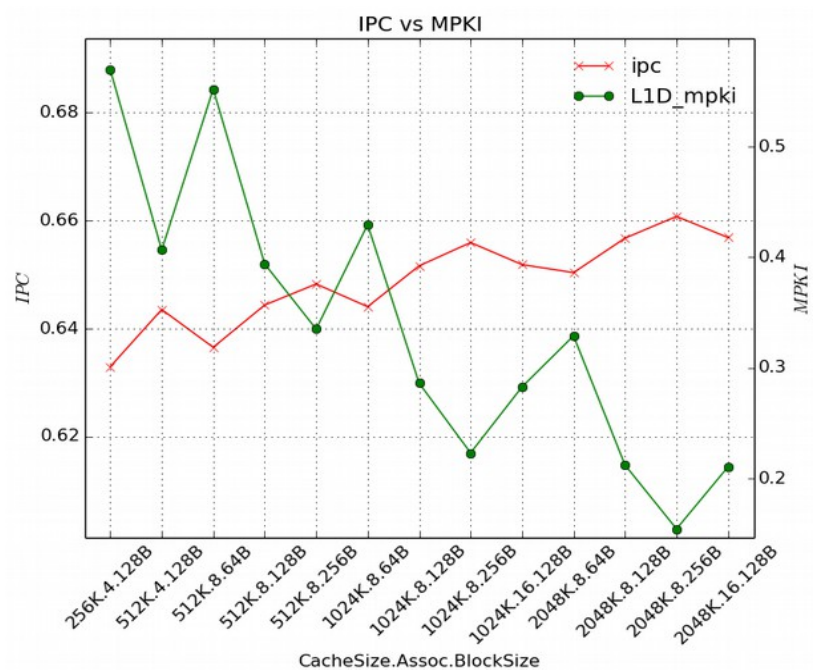
Επιδρούν το μέγεθος της cache και το block size. Και πάλι δεν υπάρχουν conflict misses καθώς το associativity δεν φαίνεται να επηρεάζει τα αποτελέσματα.

6. fluidanimate



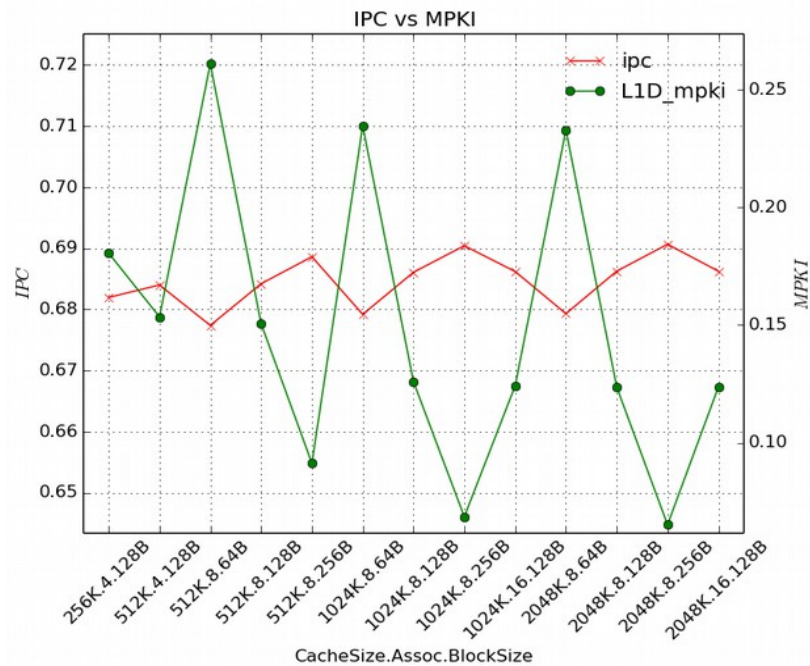
Όμοια με το ferret, επιδρούν το μέγεθος της cache και το block size. Και πάλι δεν υπάρχουν conflict misses καθώς το associativity δεν φαίνεται να επηρεάζει τα αποτελέσματα.

7. freqmine



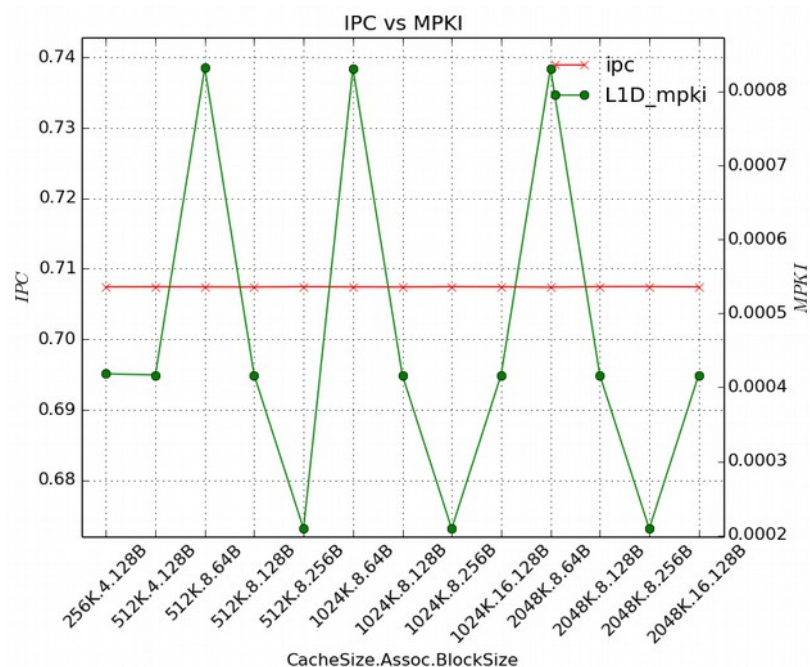
Όμοια με το ferret, επιδρούν το μέγεθος της cache και το block size. Και πάλι δεν υπάρχουν conflict misses καθώς το associativity δεν φαίνεται να επηρεάζει τα αποτελέσματα.

8. raytrace



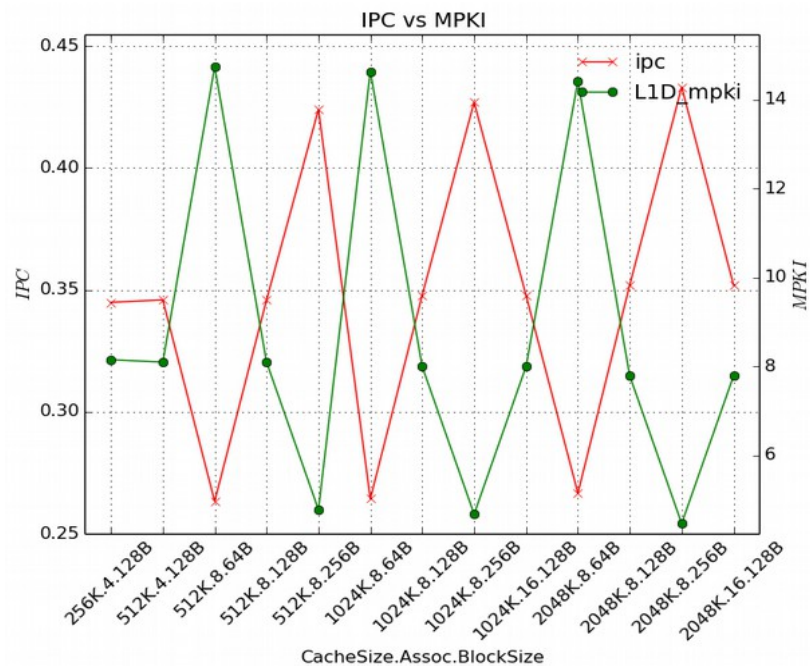
Όμοια με το ferret, επιδρούν το μέγεθος της cache και το block size. Και πάλι δεν υπάρχουν conflict misses καθώς το associativity δεν φαίνεται να επηρεάζει τα αποτελέσματα. Καθώς όμως το mпки είναι ιδιαίτερα χαμηλό, το ipc είναι εξαιρετικά ομαλό και δεν επηρεάζεται ιδιαίτερα.

9. swaptions



Εδώ επιδρά μόνο το μέγεθος του block size. Δεν υπάρχουν capacity ή conflict misses καθώς το associativity και το cache size δεν φαίνεται να επηρεάζουν τα αποτελέσματα.

10. streamcluster



Εδώ επιδρά μόνο το μέγεθος του block size. Δεν υπάρχουν capacity ή conflict misses καθώς το associativity και το cache size δεν φαίνεται να επηρεάζουν τα αποτελέσματα.

Καθώς όμως το mpki είναι σχεδόν μηδενικό, το ipcmp είναι σχεδόν σταθερό.

7.1.3 Write allocation

Σε αυτή την περίπτωση τροποποιήσαμε το κώδικα κατάλληλα για να χρησιμοποιηθεί η πολιτική Write NO Allocation.

➤ L2 size = 1024 KB

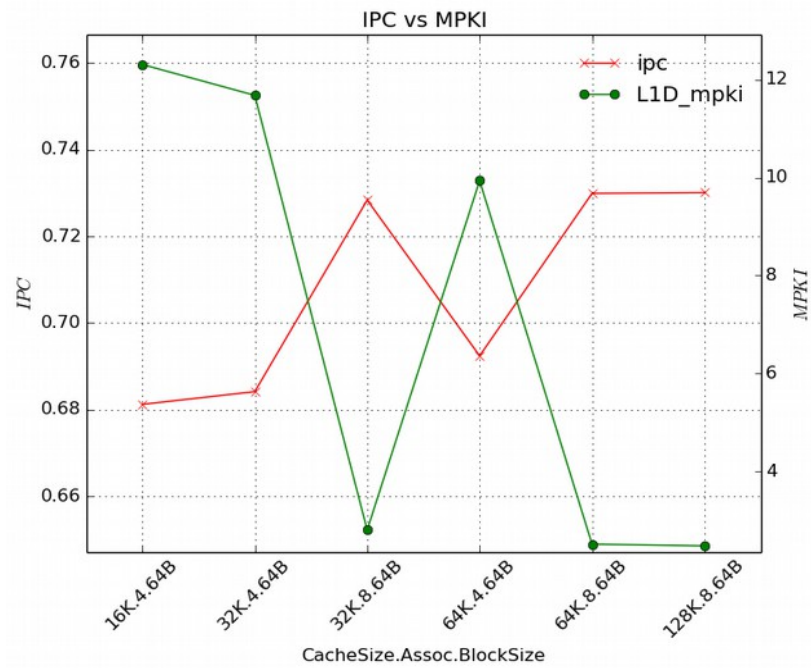
➤ L2 associativity = 8

➤ L2 block size = 128 B

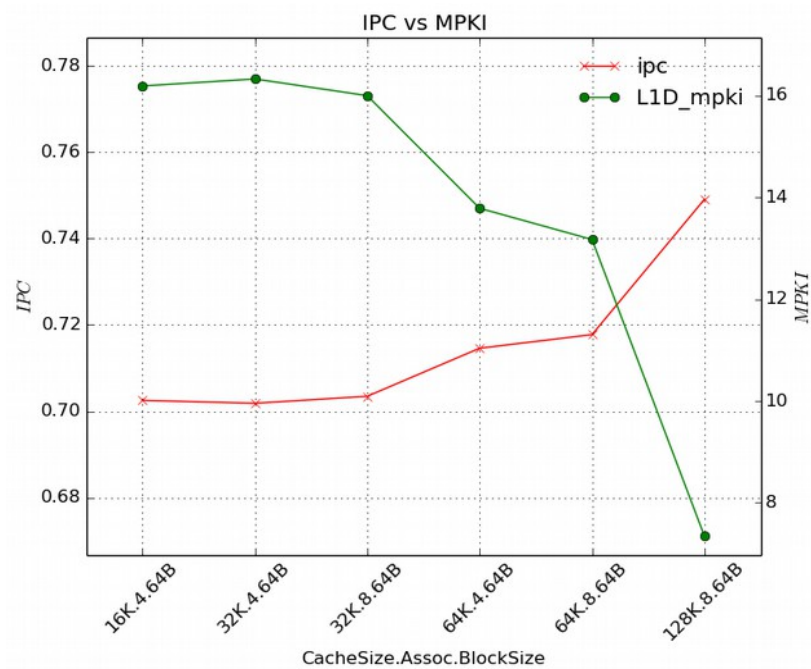
L1 size	L1 associativity	L1 cache block size
16KB	4	64B
32KB	4	64B
32KB	8	64B

L1 size	L1 associativity	L1 cache block size
64KB	4	64B
64KB	8	64B
128KB	8	64B

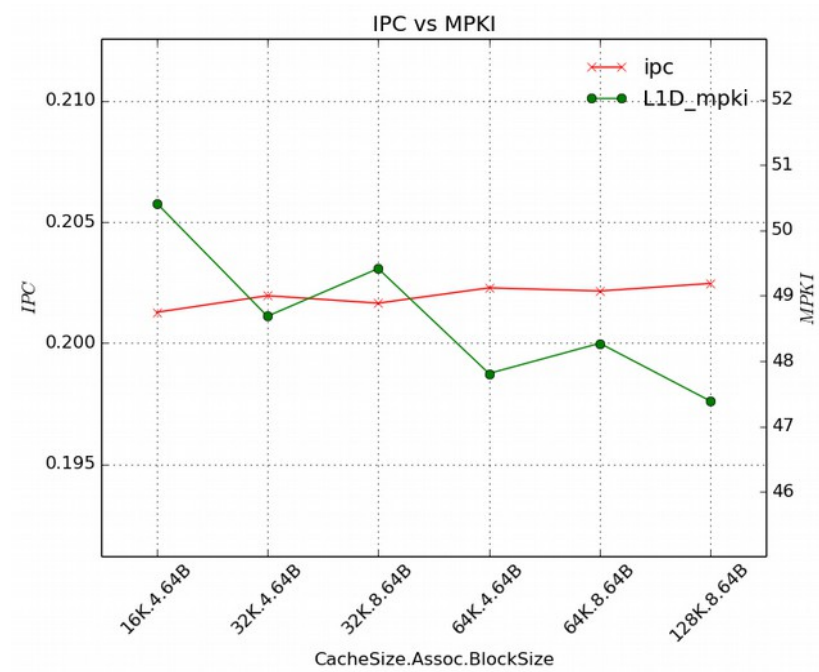
1. blackscholes



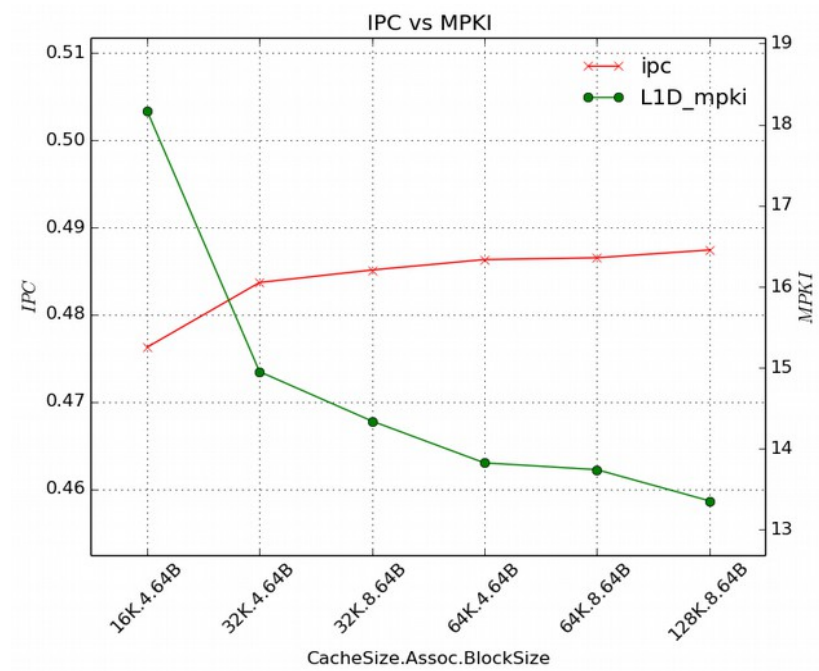
2. bodytrack



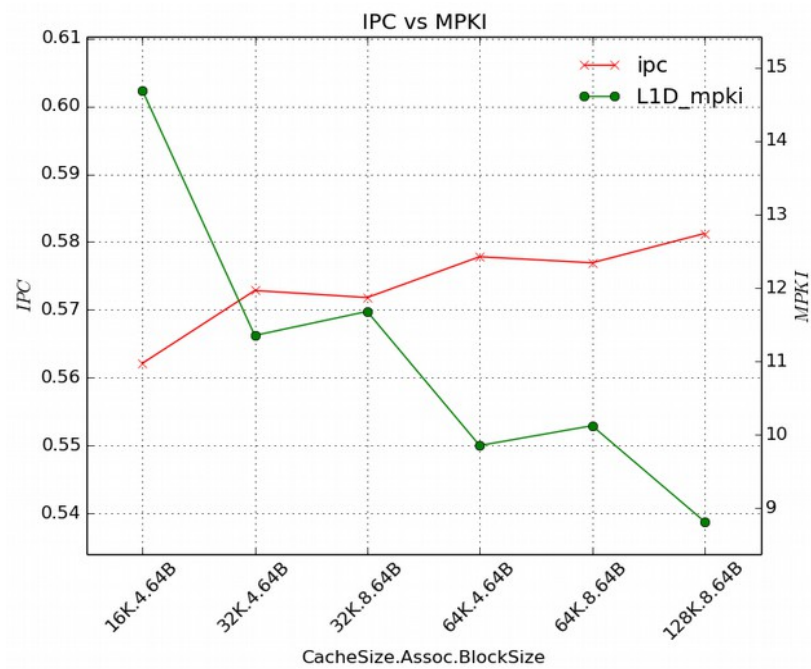
3. canneal



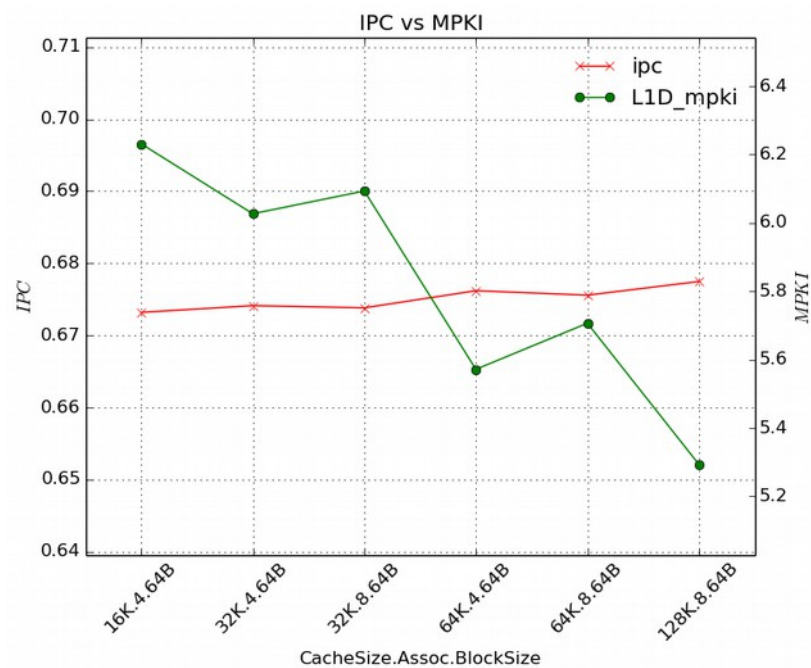
4. facesim



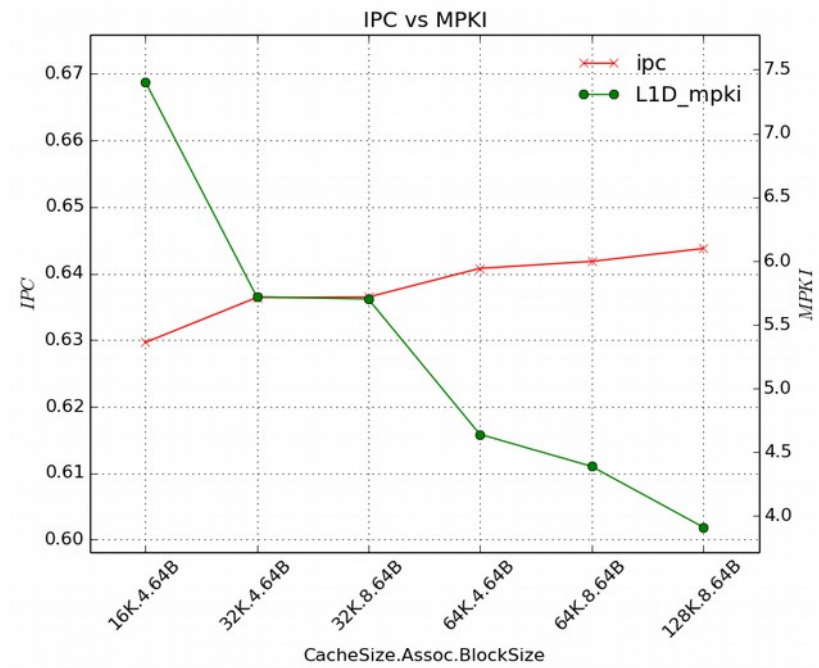
5. ferret



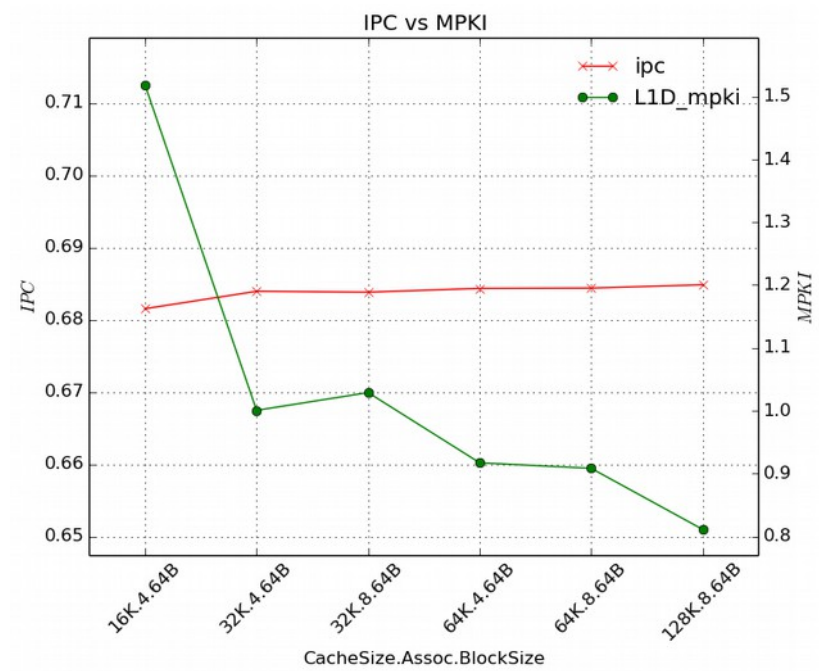
6. fluidanimate



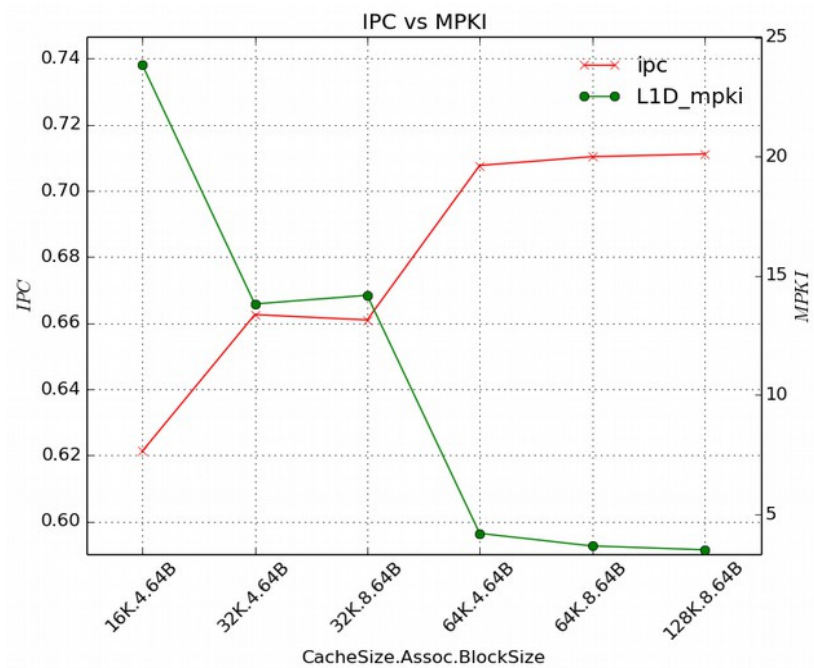
7. freqmine



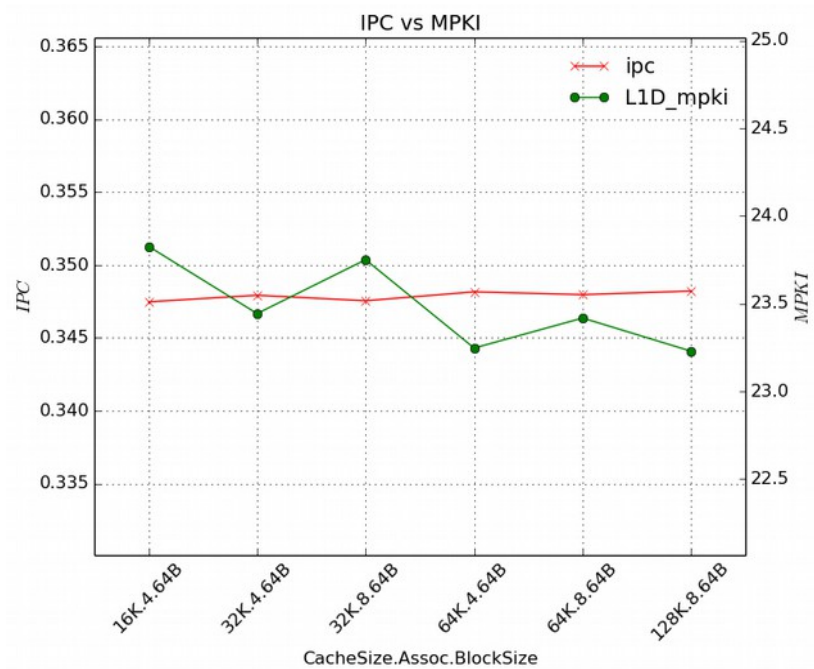
8. raytrace



9. swaptions



10. streamcluster

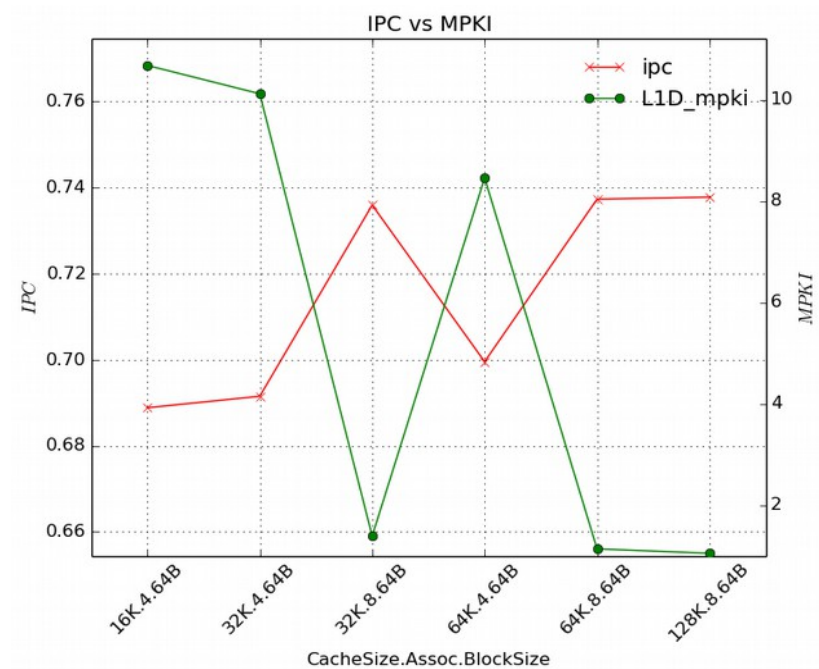


7.1.4 Πολιτική αντικατάστασης

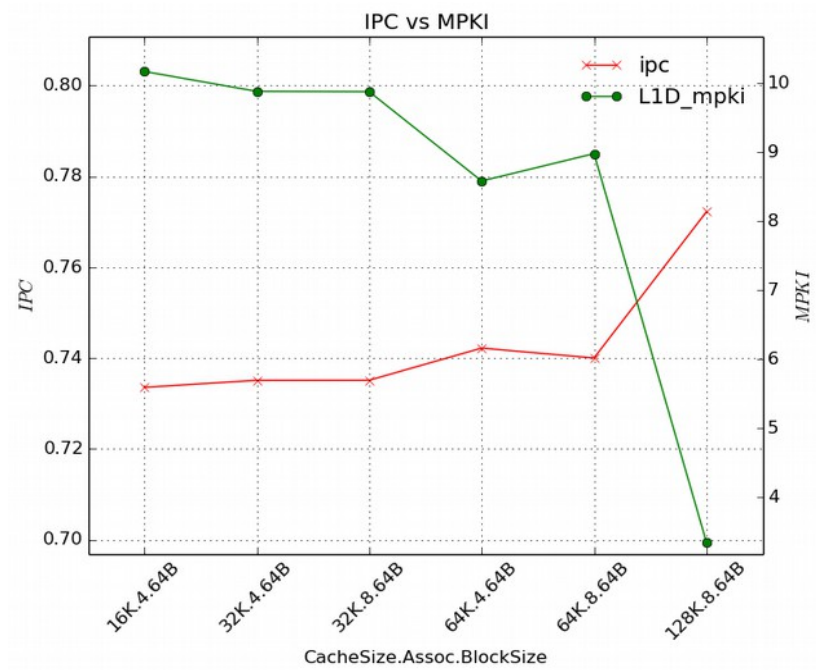
Τροποποιήσαμε κατάλληλα τον κώδικα και τρέξαμε προσοειώσεις με βάση τις παρακάτω πολιτικές αντικατάστασης

- **Random:** Η επιλογή του block που θα αντικατασταθεί γίνεται τυχαία. Έχει αρκετές ομοιότητες με την LRU, αλλά λόγω της τυχειότητας των αντικαταστάσεων δεν μπορούμε να βγάλουμε συμπεράσματα, καθώς δεν υπάρχει σταθερότητα στις μετρήσεις.

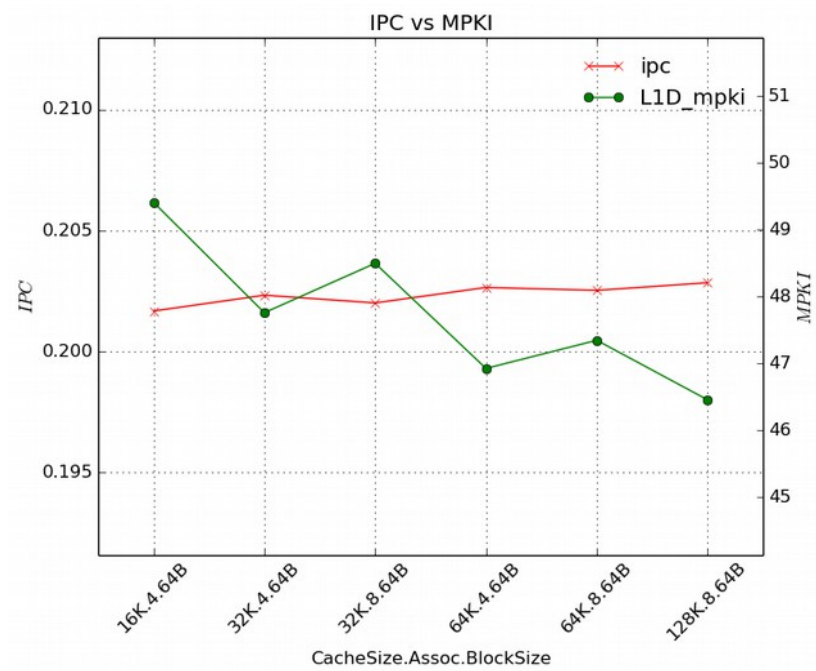
1. blackscholes



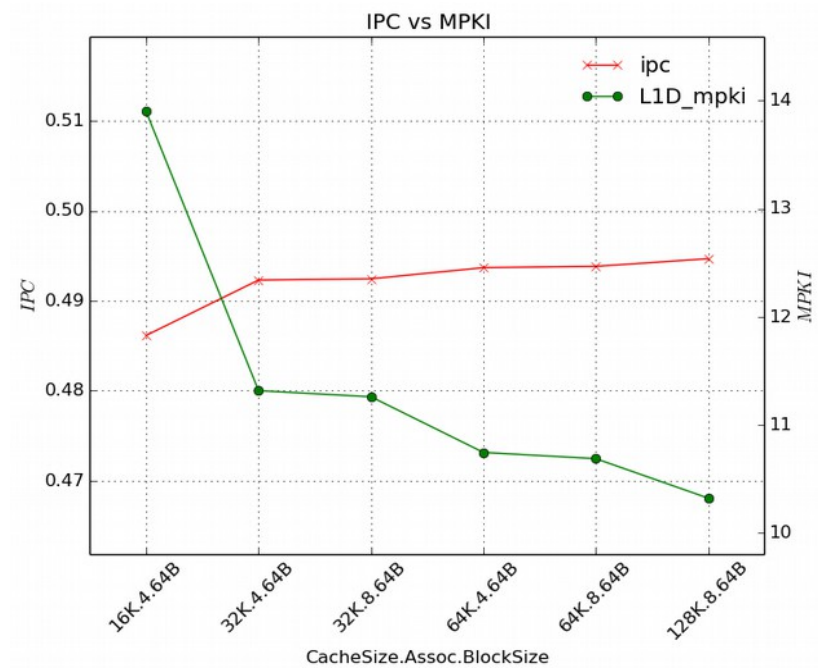
2. bodytrack



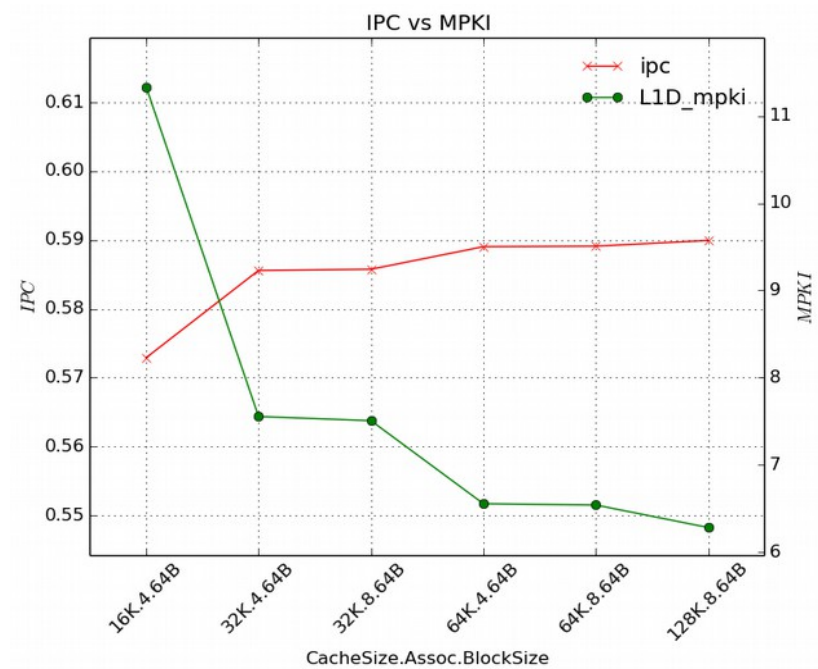
3. canneal



4. facesim

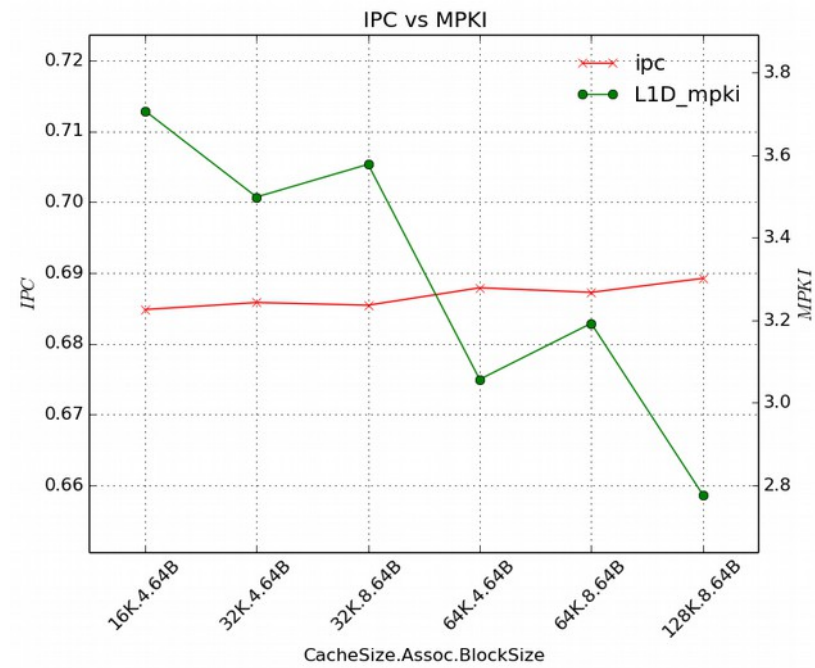


5. ferret

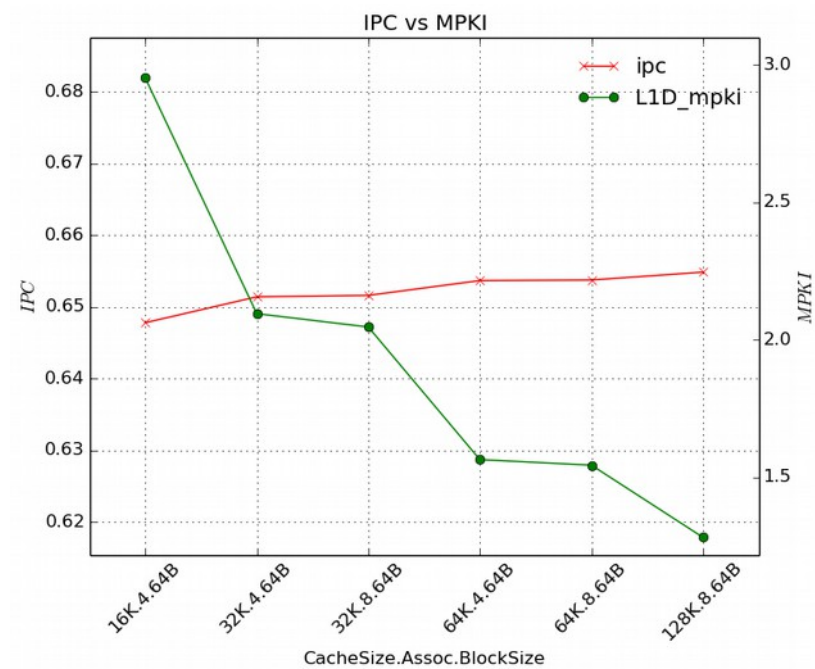


6. fluidanimate

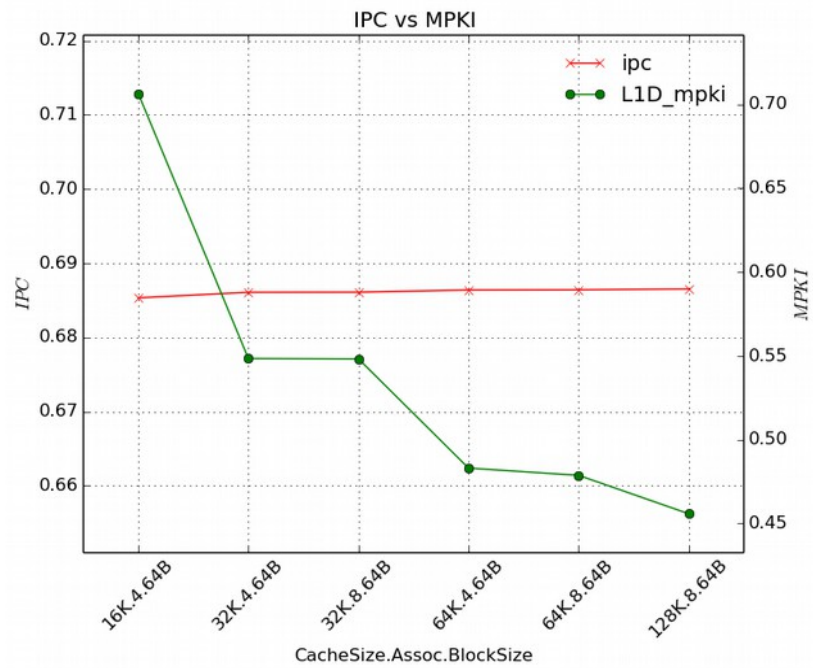
1η Σειρά Ασκήσεων στα Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών



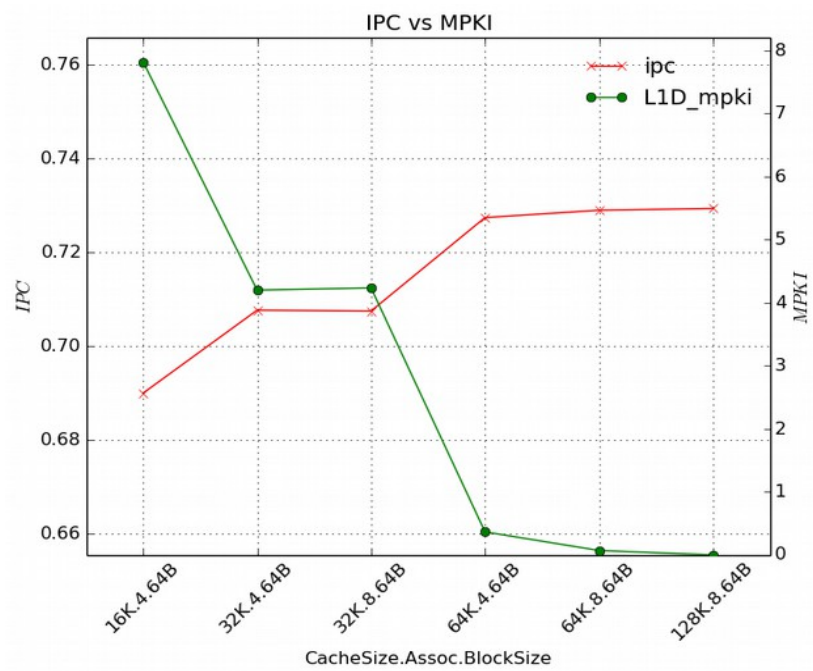
7. freqmine



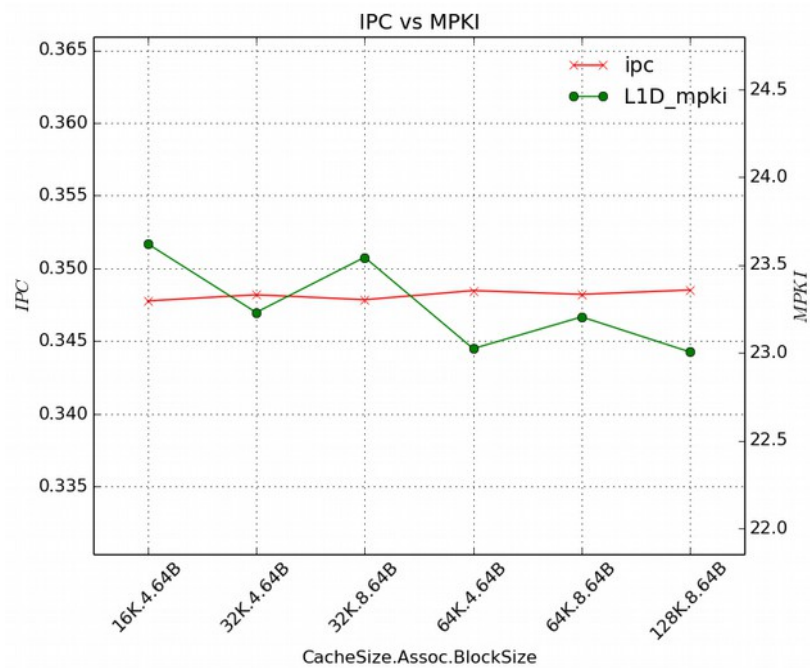
8. raytrace



9. swaptions

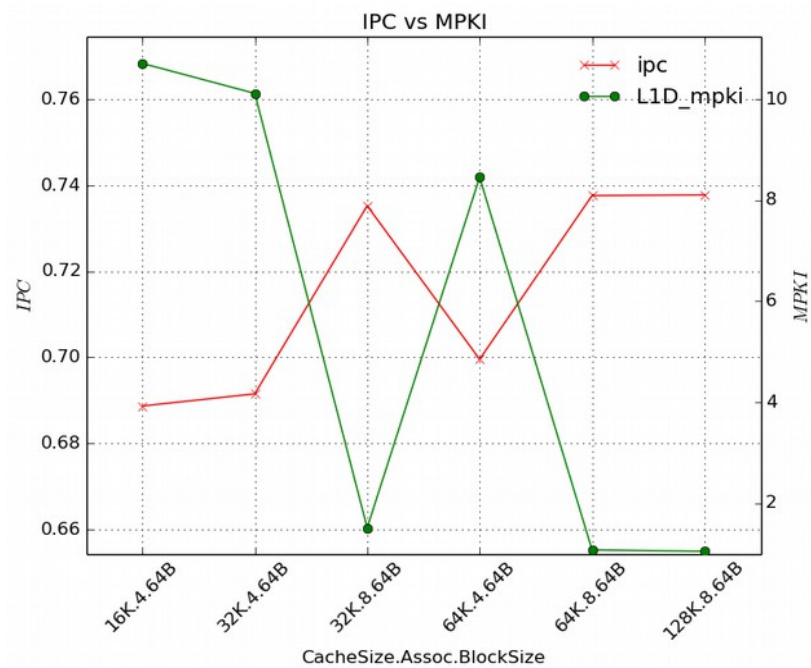


10. streamcluster

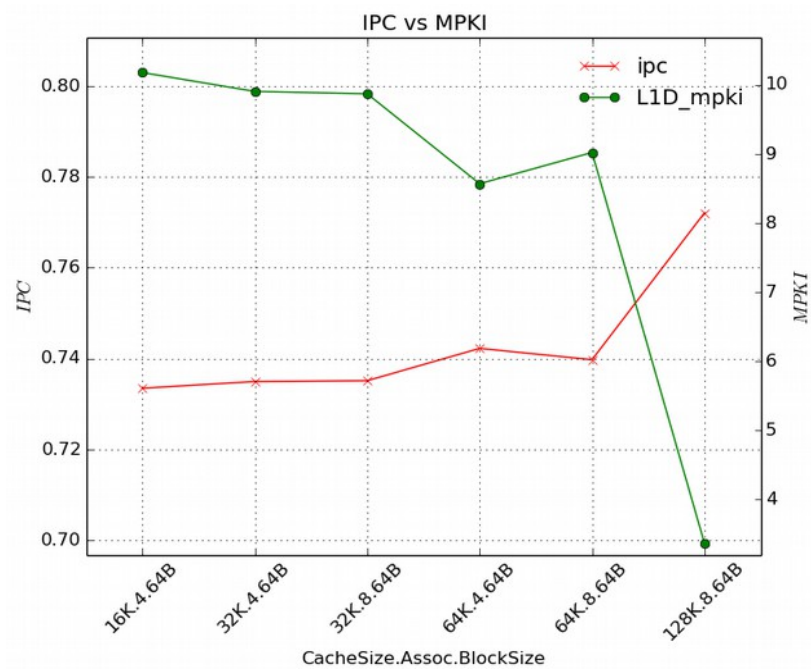


- **Least Frequently Used(LFU):** Η πολιτική LFU μετρά πόσο συχνά χρησιμοποιείται κάθε block και επιλέγει να αντικαταστήσει αυτό που έχει χρησιμοποιηθεί λιγότερο. Μας οδηγεί σε χειρότερα αποτελέσματα σε σχέση με την LRU καθώς και με τη Random. Ειδικότερα:
- Αύξηση του associativity οδηγεί σε αύξηση του mpmi.
- Αύξηση του μεγέθους του Block οδηγεί σε μείωση του mpmi, καθώς μειώνονται τα compulsory misses.
- Αύξηση του μεγέθους της cache κατά κανόνα οδηγεί σε μείωση του mpmi και αύξηση του ipc, καθώς μειώνονται τα capacity misses.

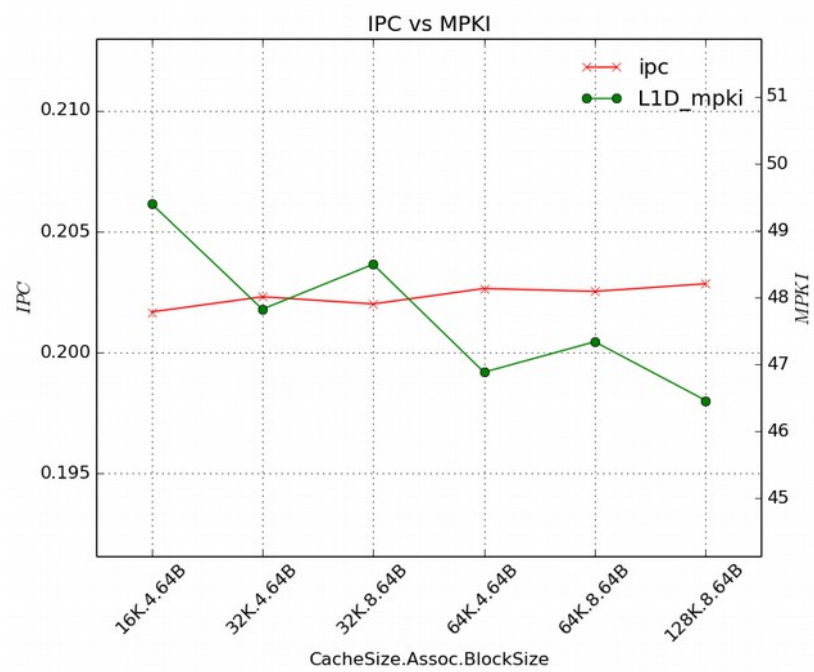
1. blackscholes



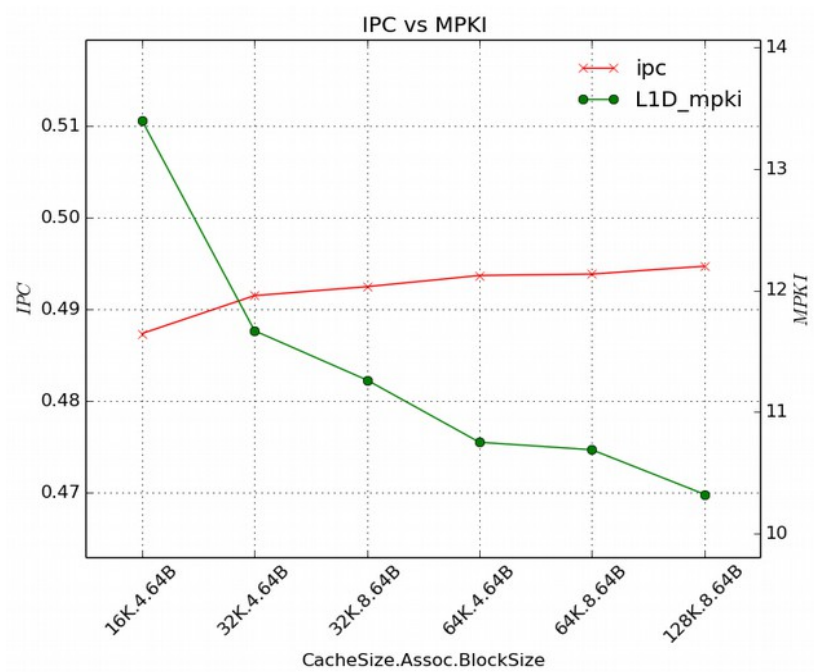
2. bodytrack



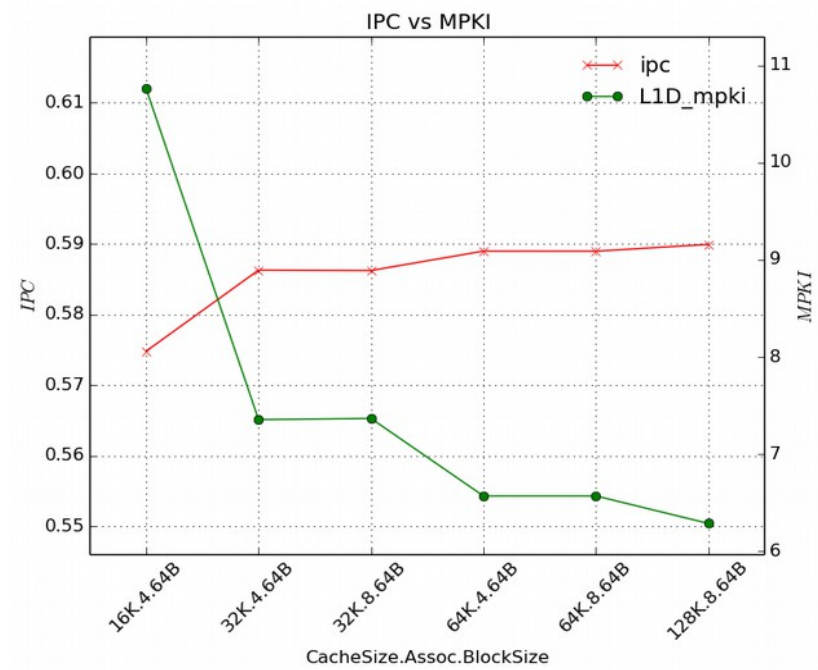
3. canneal



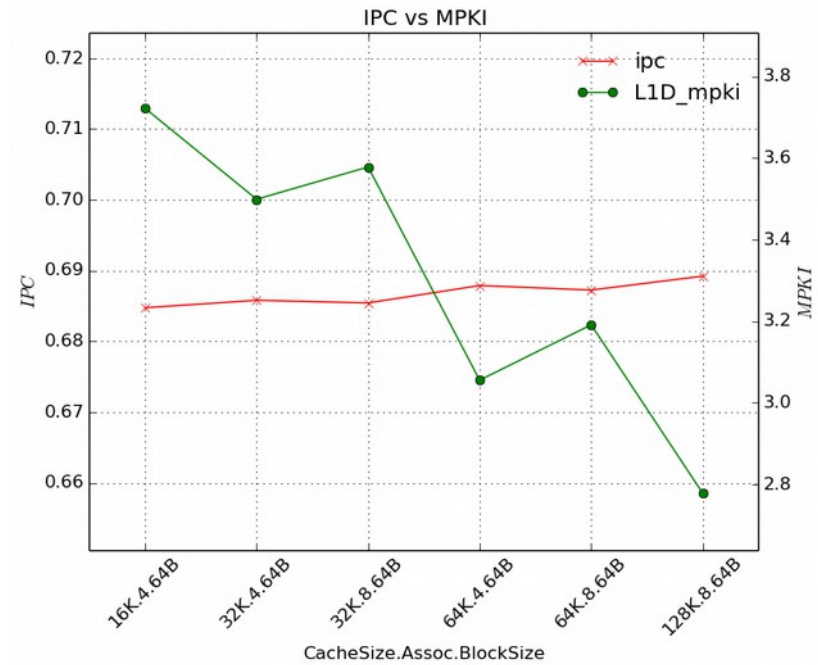
4. facesim



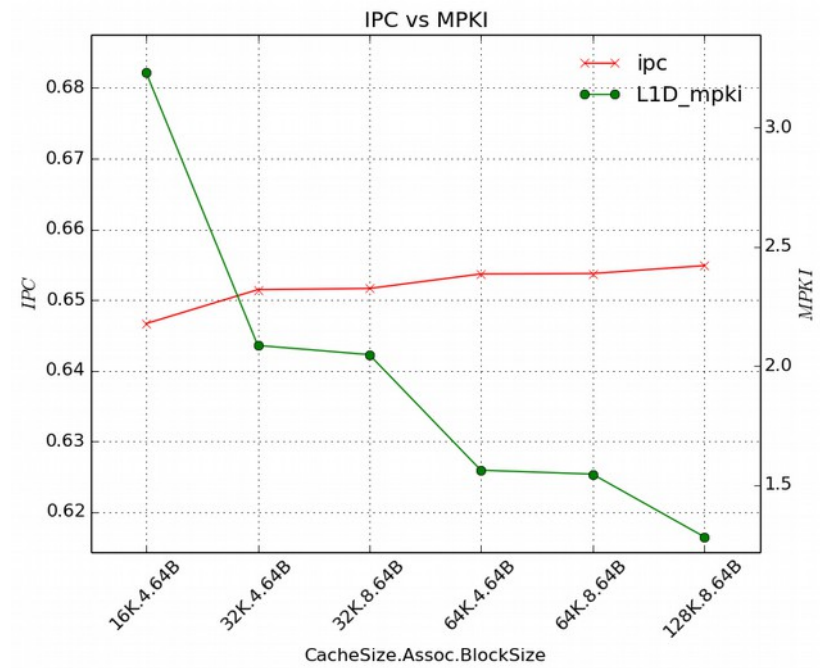
5. ferret



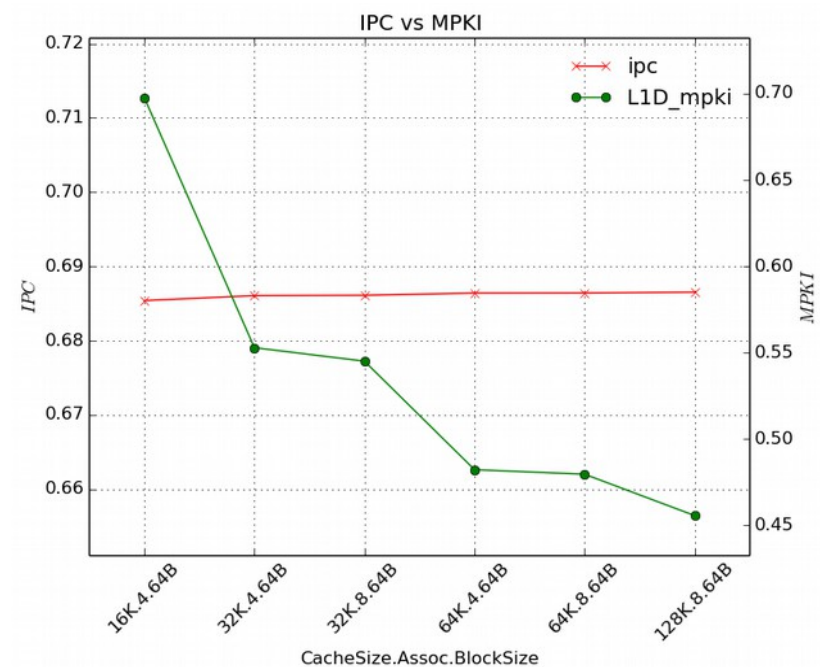
6. fluidanimate



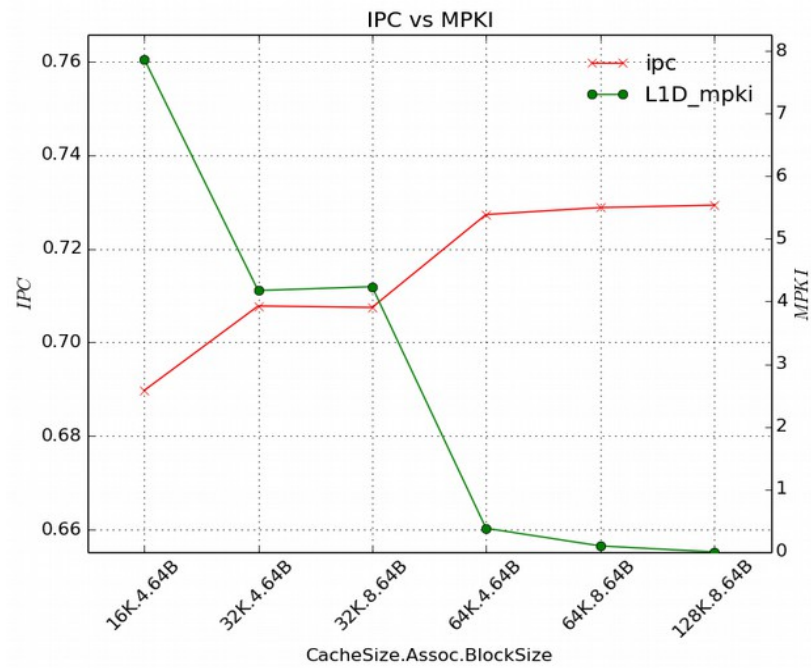
7. freqmine



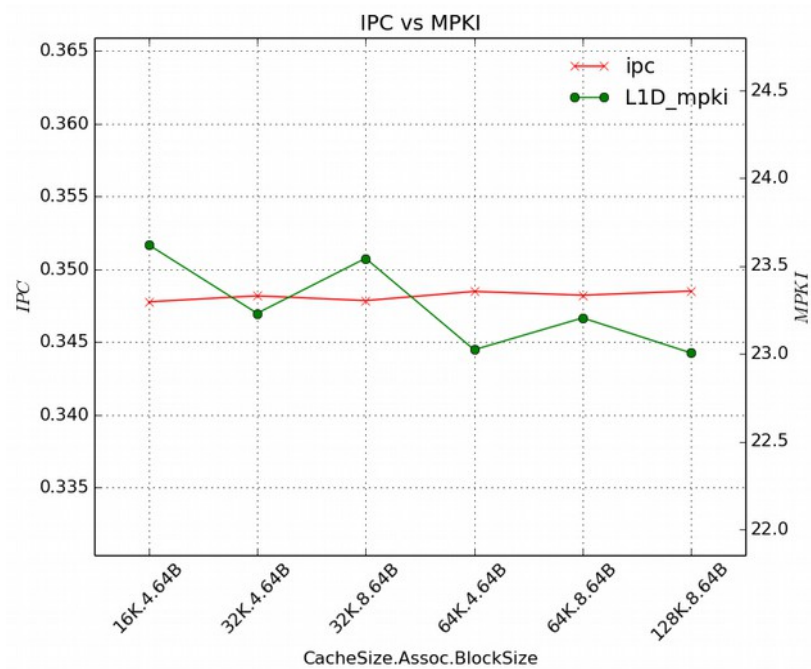
8. raytrace



9. swaptions



10. streamcluster



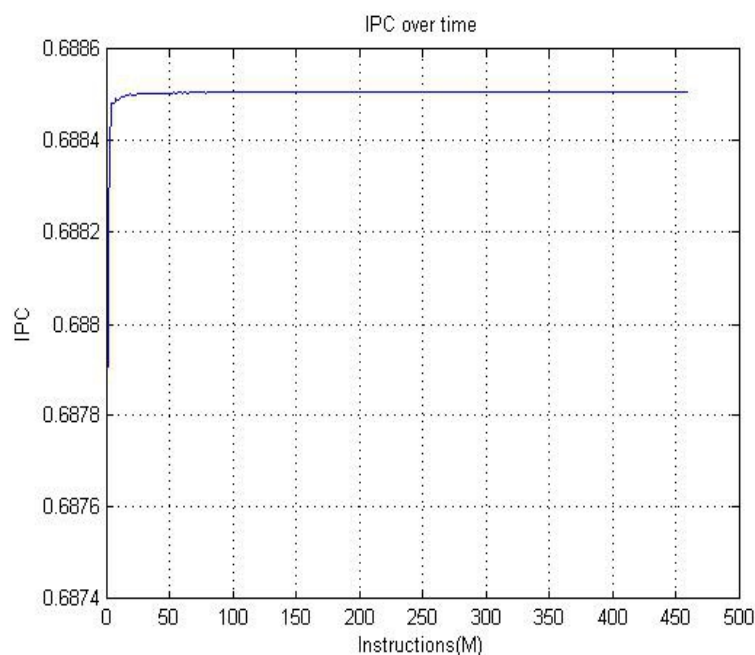
7.2 Μελέτη μεταβολής μετρικών απόδοσης στο χρόνο

Στο δεύτερο μέρος της εργασίας μας ζητήθηκε να εξετηάσουμε για ένα συγκεκριμένο configuration τη συμπεριφορά της επίδοσης σε κάθε benchmark, σε σχέση με το χρόνο. Επιλέξαμε να ρησιμοποιήσουμε ως configuration το παράδειγμα που υπάρχει στην εκφώνηση (L1 32KB, 8-way, block size 64B L2 1024KB, 8-way, block size 128B, write-allocate ιεραρχία με LRU πολιτική αντικατάστασης). Οι αλλαγές στον κώδικα βρίσκονται στο τέλος της αναφοράς. Παρόλο που επιλέγουμε να αποθηκεύουμε κάθε 10M εντολές τα αποτελέσματα, στον άξονα x τιμές εμφανίζονται ανά 500M έτσι ώστε να είναι ευδιάκριτα τα αποτελέσματα. Από τα διαγράμματα καταλήξουμε στο εξής συμπέρασμα:

Τα αποτελέσματα που πήραμε στο πρώτο μέρος μας δίνουν μια γενική εικόνα σχετικά με την προ-σομοίωση. Ανταποκρίνονται κυρίως στο μεσαίο μέρος, καθώς τόσο στην αρχή όσο και στο τέλος παρατηρείται συνήθως αρκετά διαφοροποιημένη συμπεριφορά. Επομένως τα αποτελέσματα του 1ου μέρους δεν ανταποκρίνονται στα πρώτα 10M ή 100M εντολών καθώς πρόκειται για την περιοχή με τις μεγαλύτερες διακυμάνσεις στο ipc.

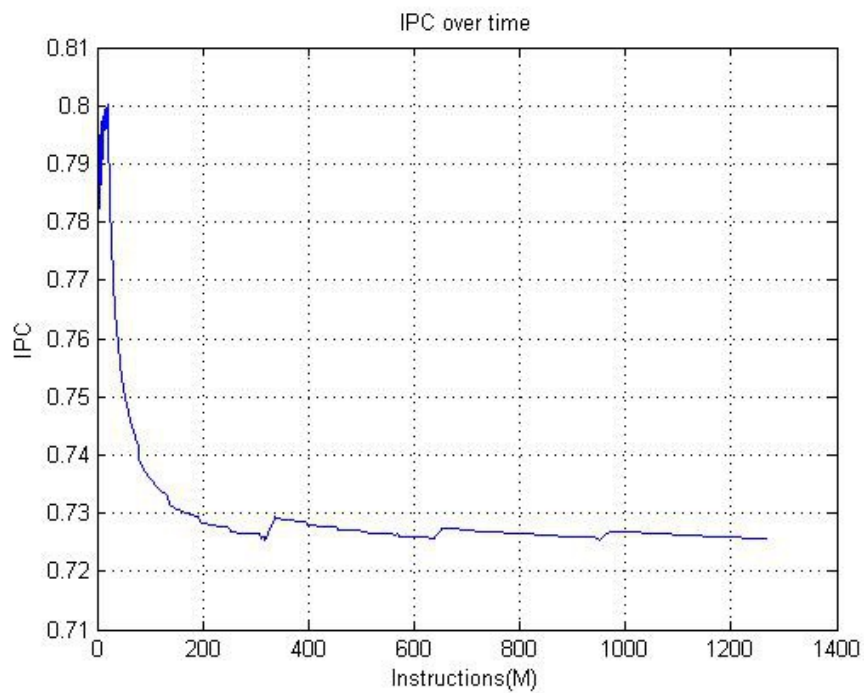
Υπαχουν benchmarks τα οποία παρουσιάζουν μία μορφή περιοδικότητας, πχ το bodytrack. Άλλα εμφανίζουν σχεδόν γραμμικότητα, πχ blackscholes. Επίσης άλλα έχουν μικρές και άλλα μεγάλες διακυμάνσεις, καθώς και διαφορετικό πλήθος εντολών.

1. blackscholes

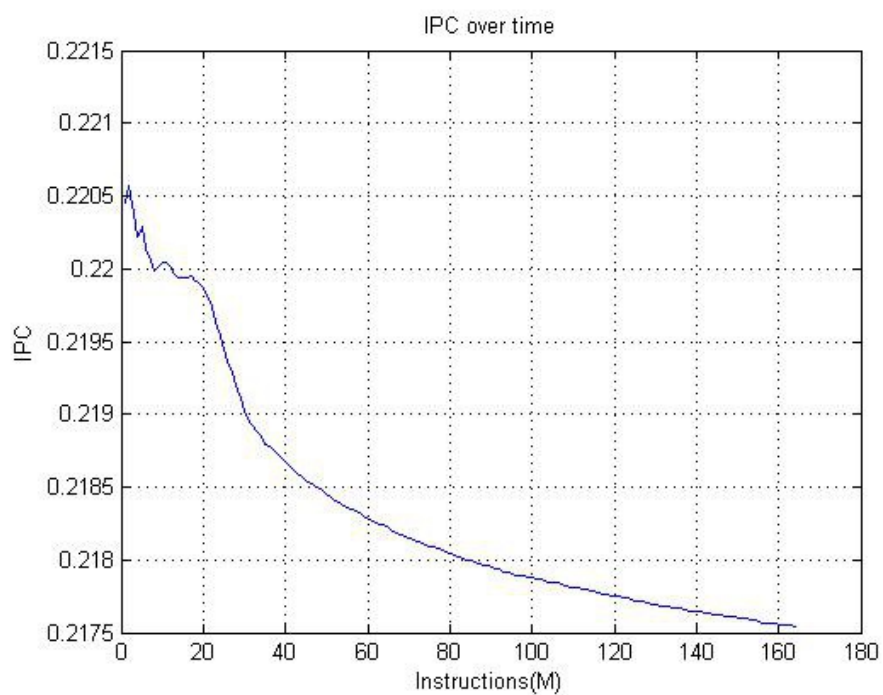


2. bodytrack

1η Σειρά Ασκήσεων στα Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

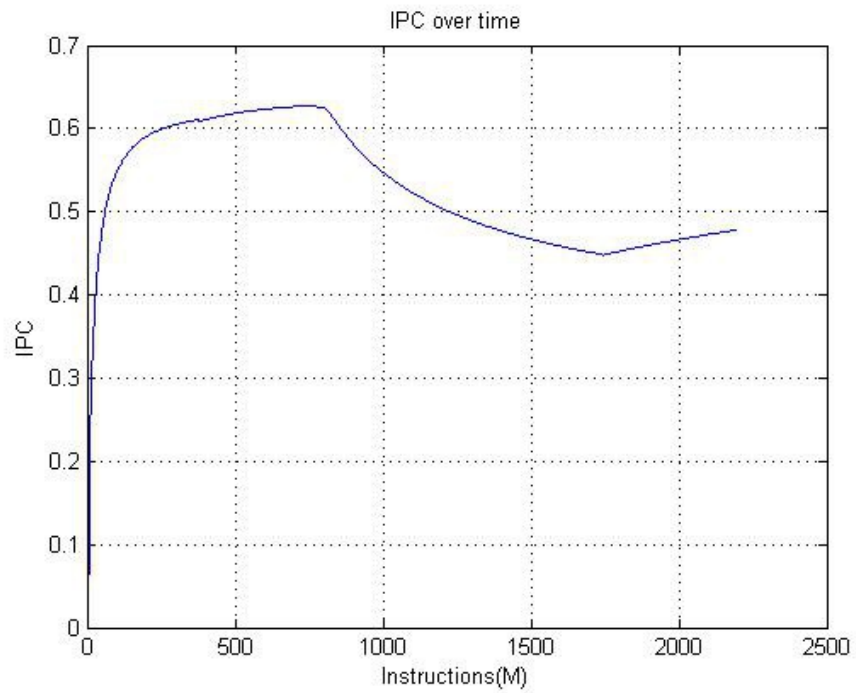


3. canneal

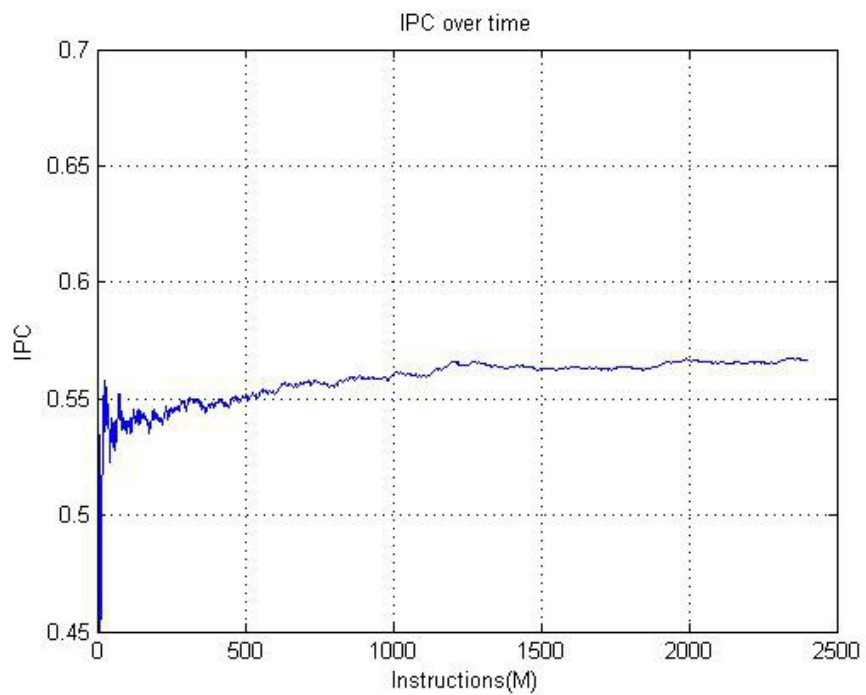


4. facesim

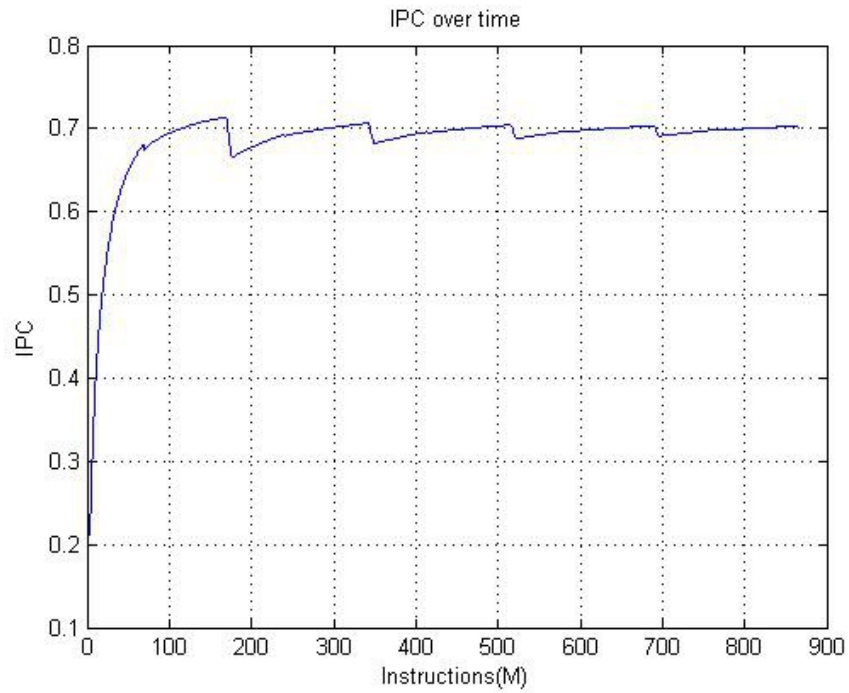
1η Σειρά Ασκήσεων στα Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών



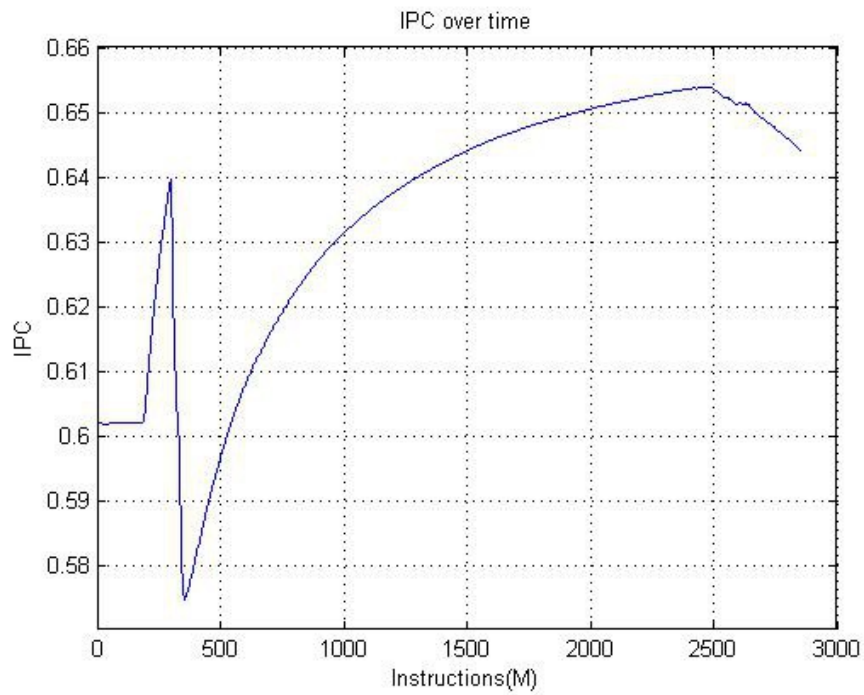
5. ferret



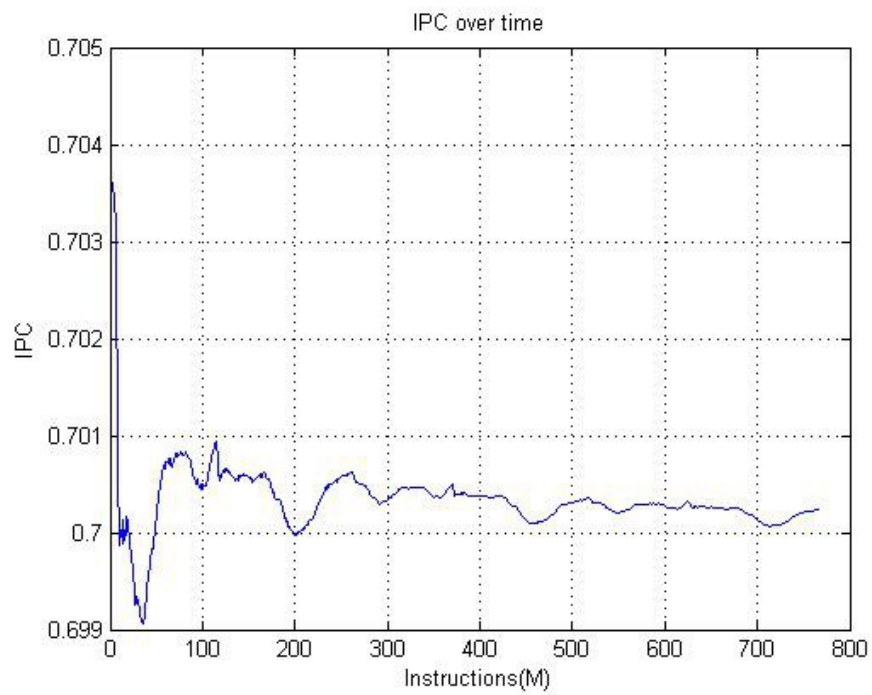
6. fluidanimate



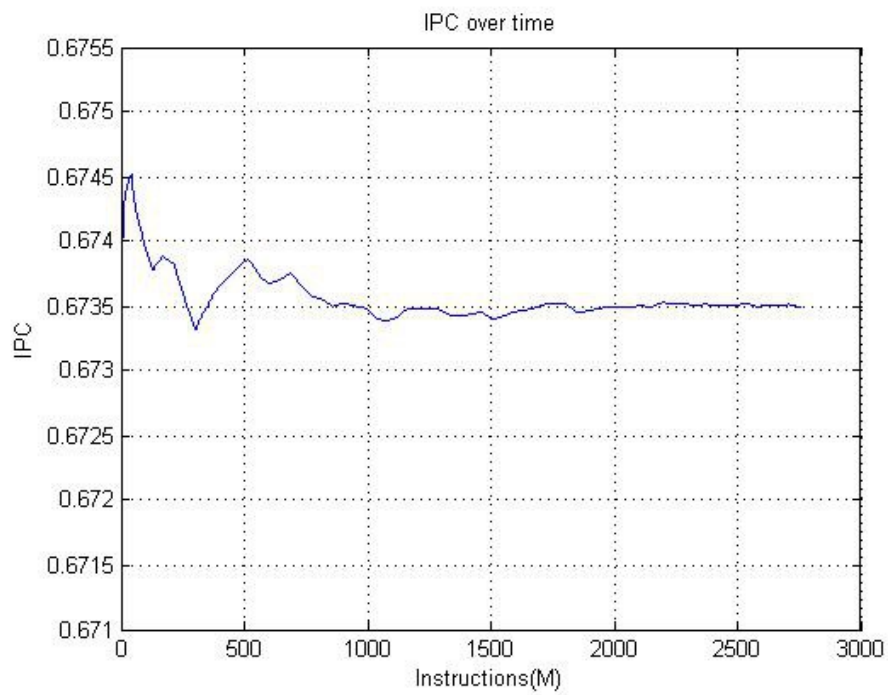
7. freqmine



8. raytrace



9. swaptions



10. streamcluster

