

**Εθνικό Μετσόβειο Πολυτεχνείο**  
**Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών**  
Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών – 8ο Εξάμηνο  
1η Άσκηση – Ακ. Έτος 2021-2022

Κυριακόπουλος Γεώργιος – el18153

## Εισαγωγή

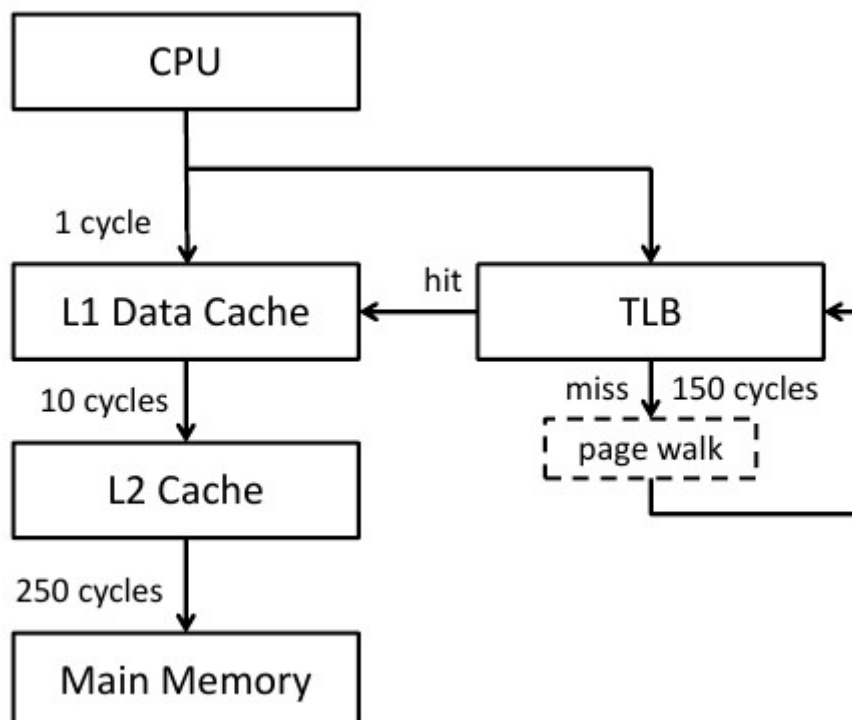
---

Θα ασχοληθούμε με μία προσομοίωση ιεραρχίας μνήμης. Θα χρησιμοποιήσουμε το πρόγραμμα PIN για τη δυναμική εισαγωγή κώδικα, 8 από τα PARSEC benchmarks (blackscholes, bodytack, canneal, fluidanimate, freqmine, raytrace, swaptions, streamcluster) και το εργαλείο PINTOOL για την προσομοίωση του περιβάλλοντος.

Χρησιμοποιώντας ως βασική μετρική επίδοσης το IPC (Instructions Per Cycle), θα μελετηθεί η μεταβολή του, καθώς και της επίδοσης των διαφόρων επιπέδων cache, τις οποίες μεταβάλλουμε, μία σε κάθε υποερώτημα. Για το πρώτο ζητούμενο, θεωρούμε ότι ο κύκλος ρολογιού δε μεταβάλλεται με τις αλλαγές στις παραμέτρους των διαφόρων cache (size, associativity, block size), ενώ στο δεύτερο θεωρούμε πως κάθε διπλασιασμός του associativity ή του size προκαλεί αύξηση του κύκλου ρολογιού κατά 10% και 15% αντίστοιχα.

Γενικότερα, για τις διάφορες καθυστερήσεις των cache έχουμε τα εξής δεδομένα, τα οποία συνοψίζονται και στο παρακάτω σχήμα:

- TLB Hit: 0 cycles (παράλληλα με την L1 cache)
- TLB Miss: 150 cycles
- L1 Hit: 1 cycle
- L2 Hit: 10 cycles
- Main memory access: 250 cycles



## 1ο ζητούμενο

---

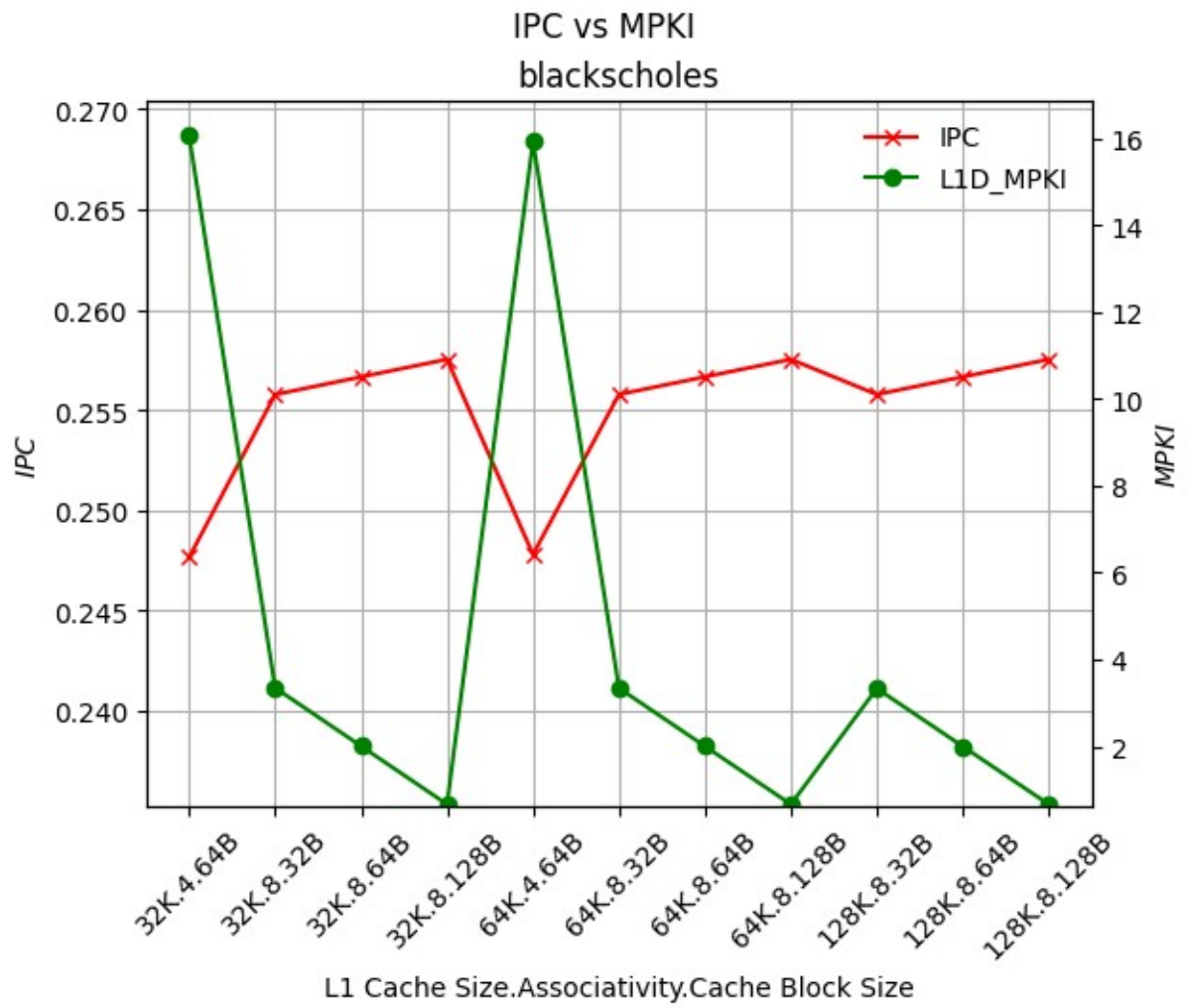
### L1 cache

Για αυτό το πείραμα θεωρούμε πως ο κύκλος ρολογιού, όπως και ο αριθμός εκτελούμενων εντολών παραμένει σταθερός και πως οι μεταβολές στο IPC που παρατηρούνται στα γραφήματα αντιστοιχούν σε διαφορές στην απόδοση. Θα διατηρήσουμε τις παραμέτρους της L2 cache και του TLB σταθερές και ίσες με:

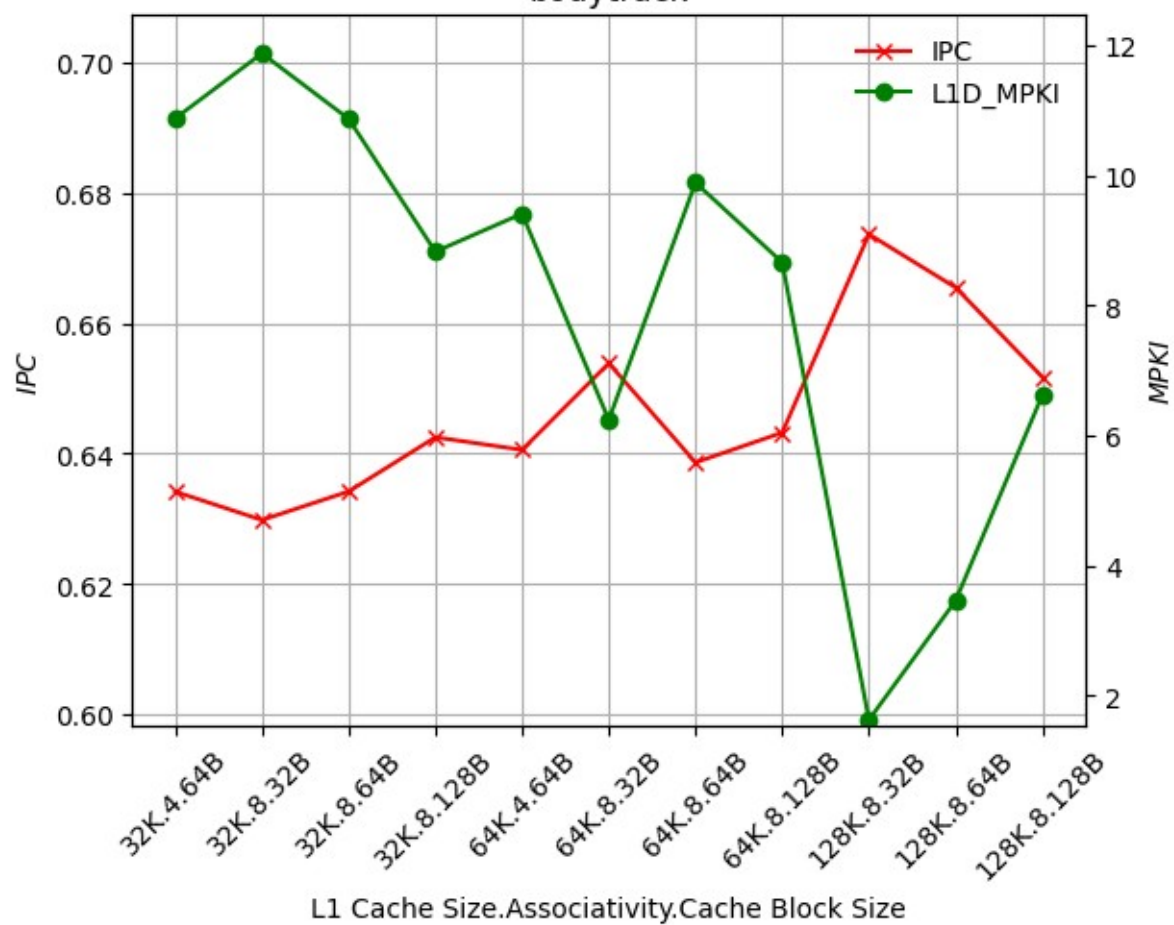
- L2 size = 1024 KB
- L2 associativity = 8
- L2 block size = 128 B
- TLB size = 64 entries
- TLB associativity = 4
- TLB page size = 4096 B

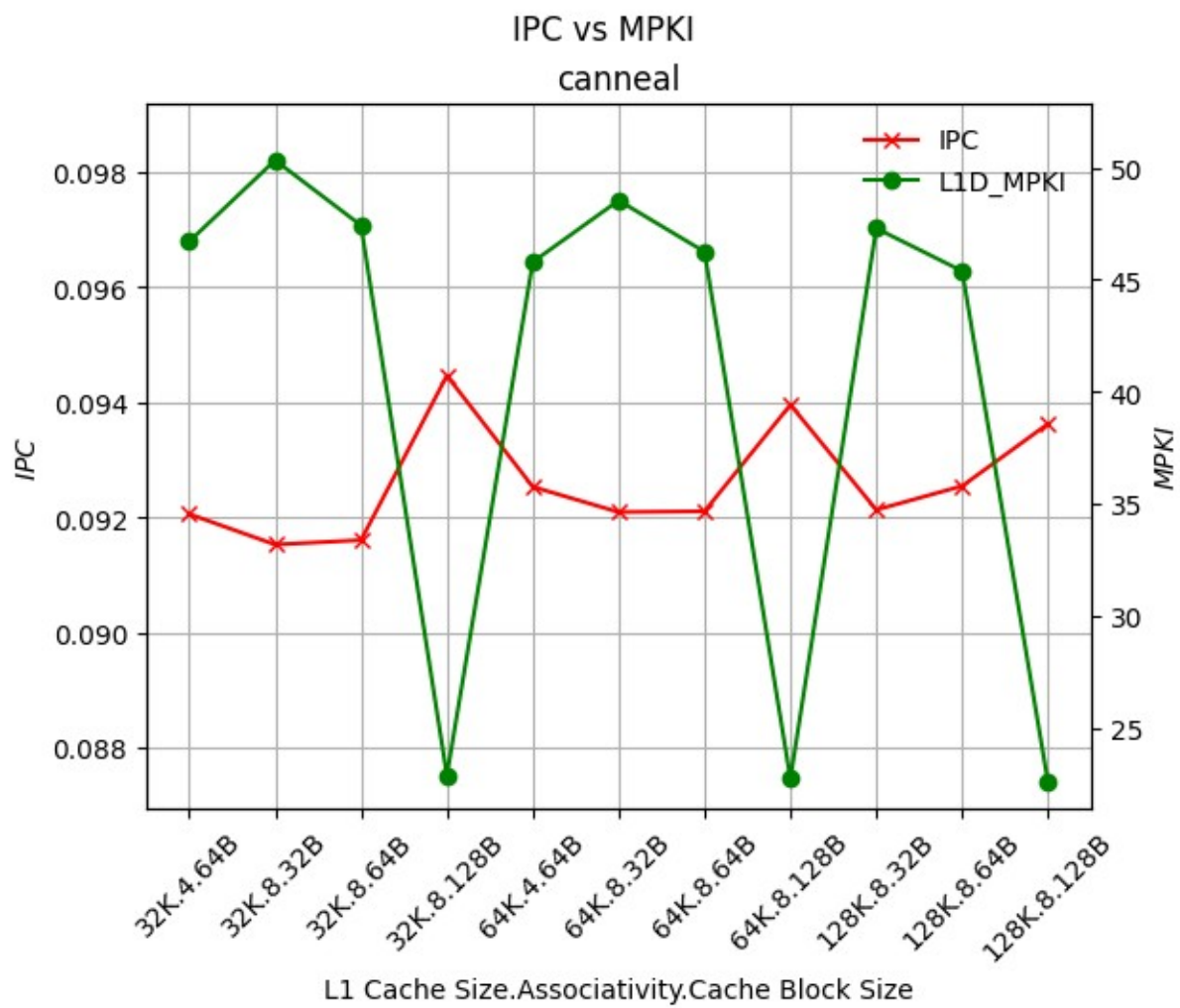
Επίσης, χρησιμοποιήσαμε τις παρακάτω δυνατές τριάδες παραμέτρων για τη L1 cache (με μορφή size.associativity.block\_size): 32.4.64, 32.8.32, 32.8.64, 32.8.128, 64.4.64, 64.8.32, 64.8.64, 64.8.128, 128.8.32, 128.8.64, 128.8.128.

Ακολουθούν τα διαγράμματα που προέκυψαν, ένα για κάθε benchmark.

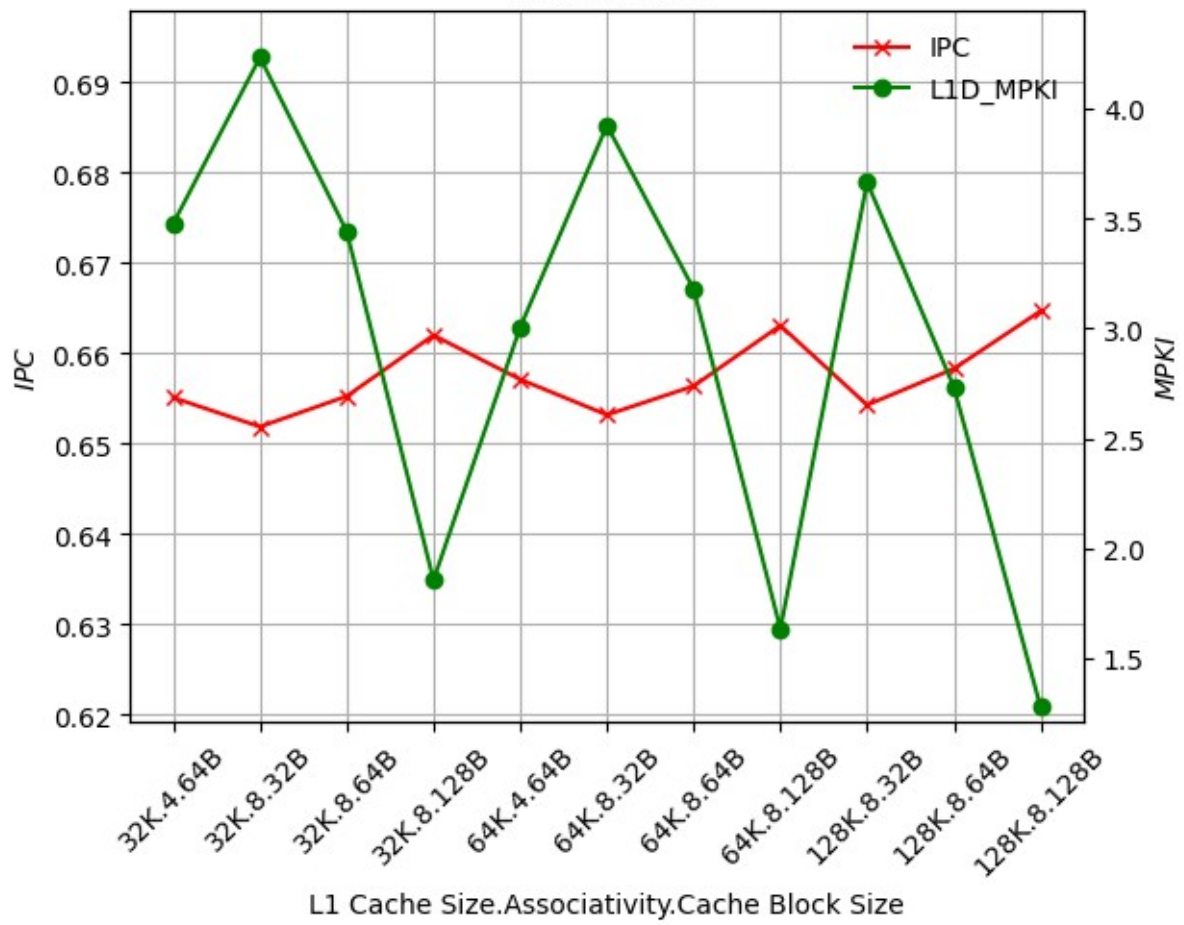


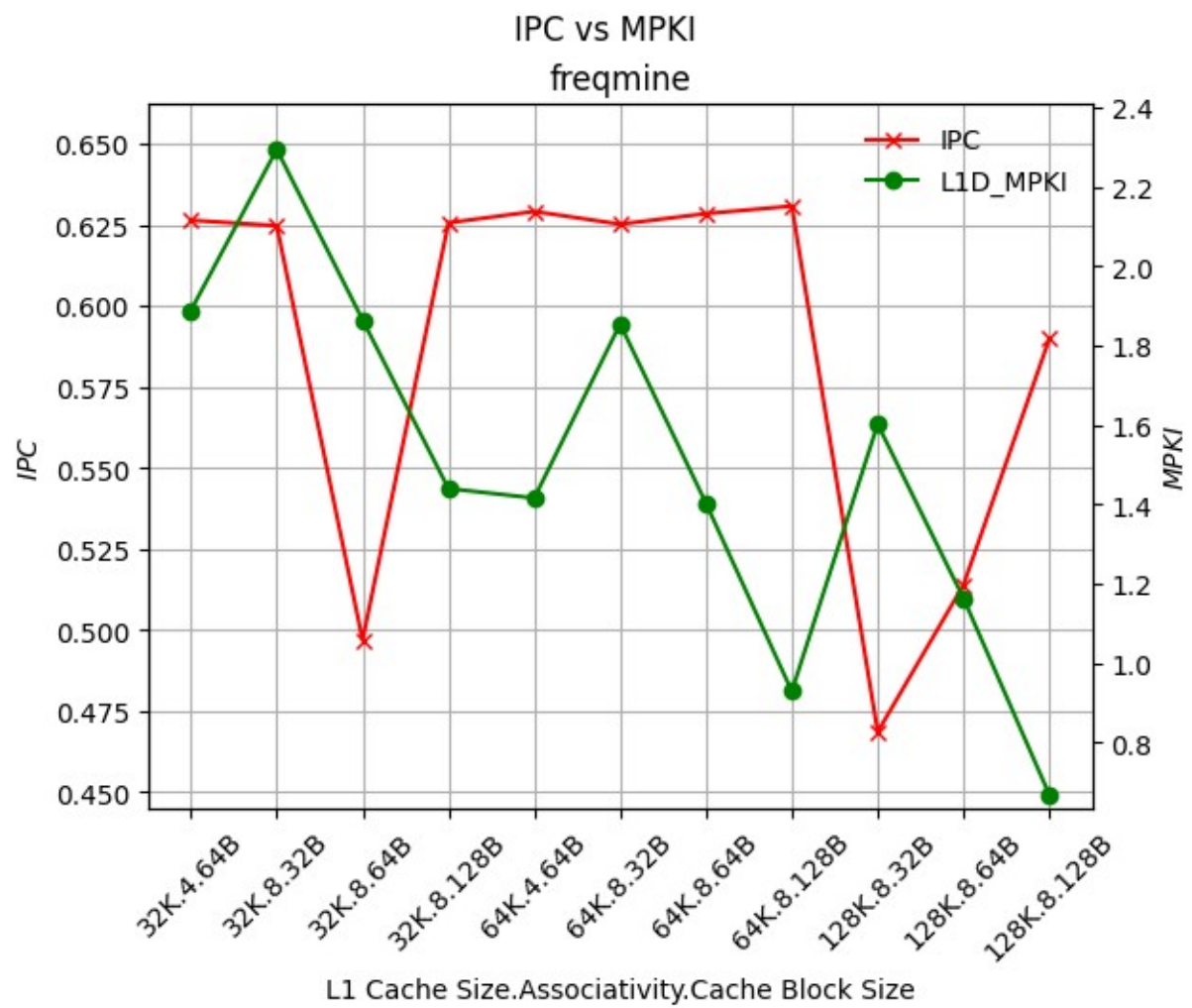
IPC vs MPKI  
bodytrack

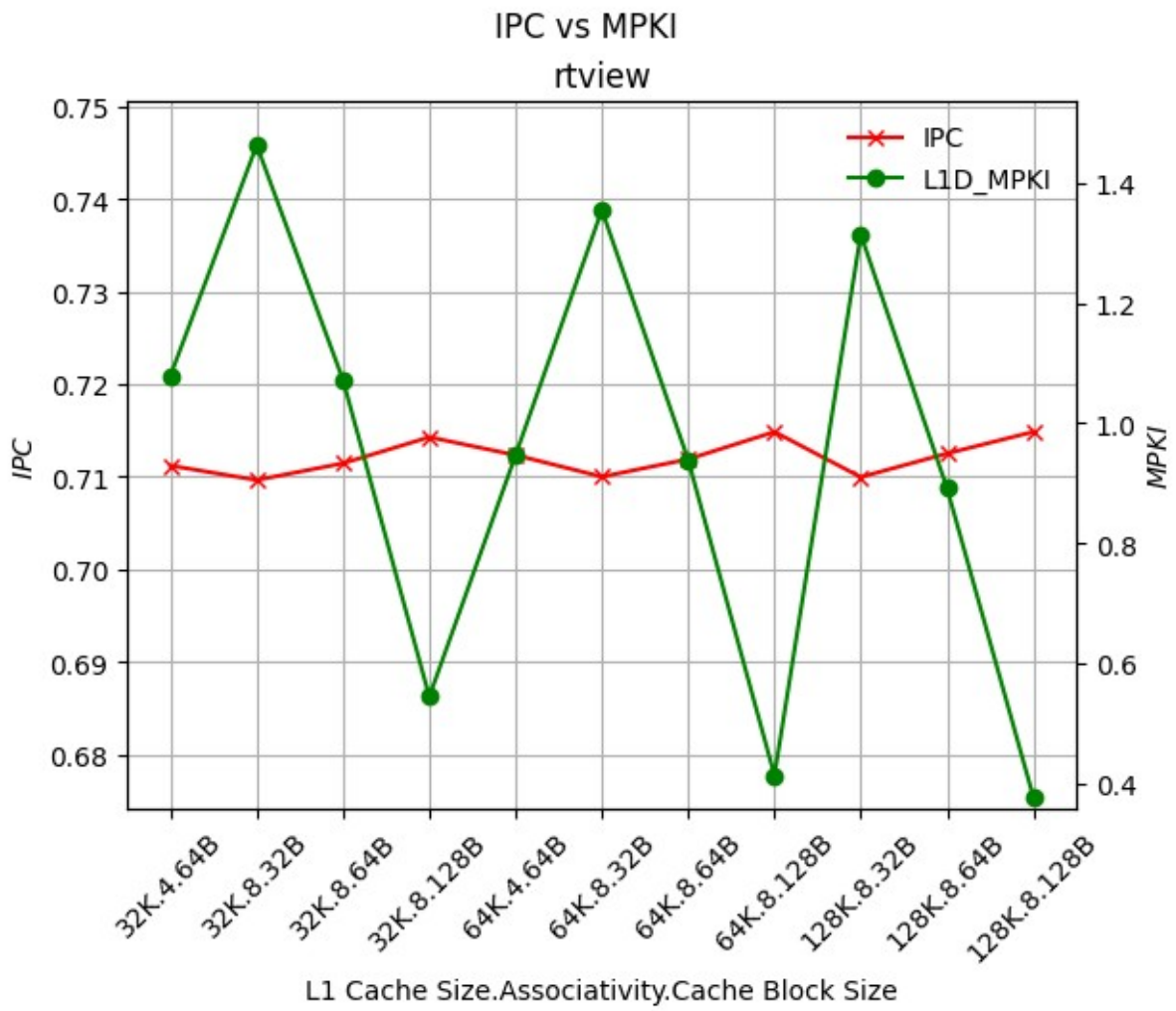




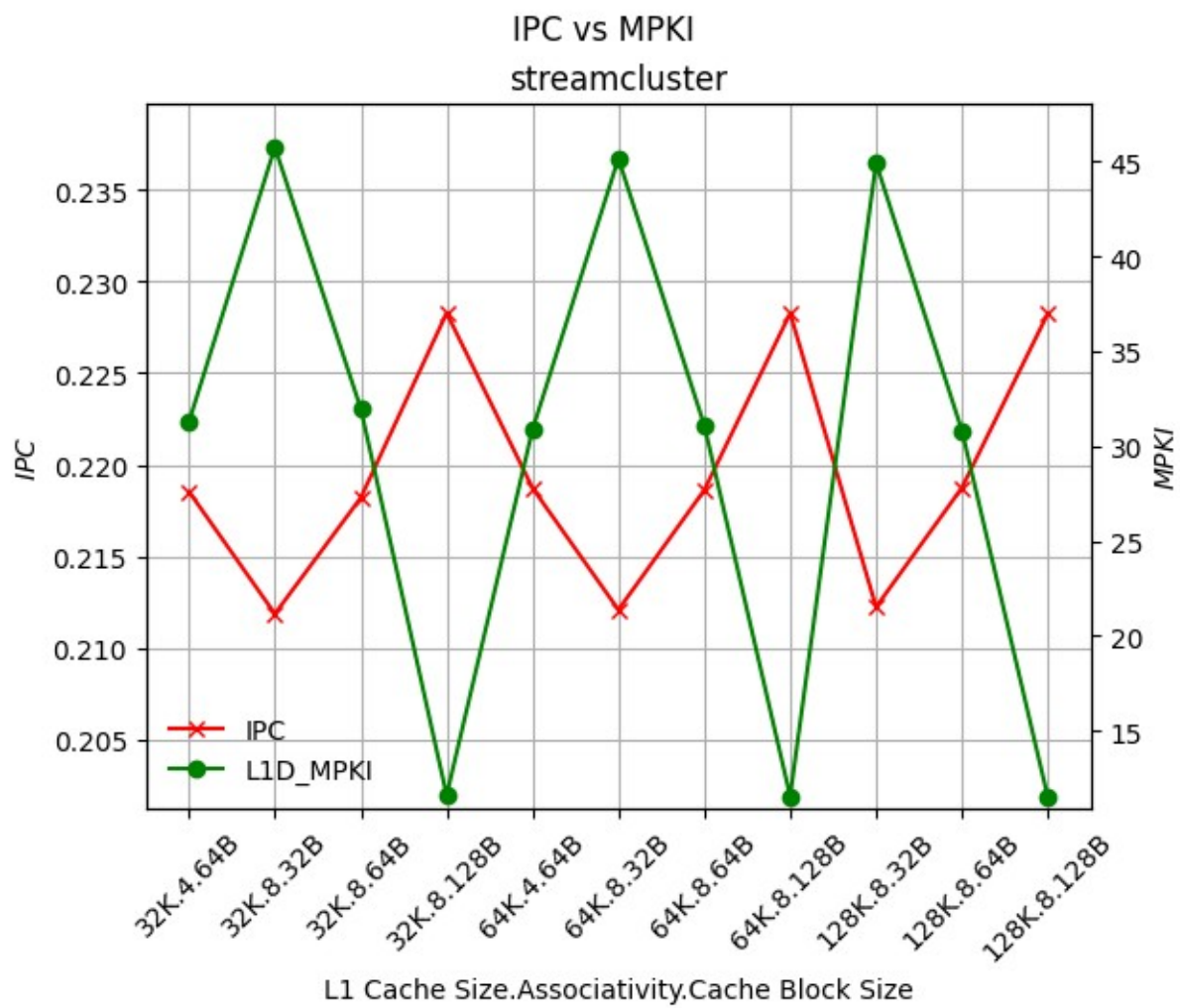
IPC vs MPKI  
fluidanimate



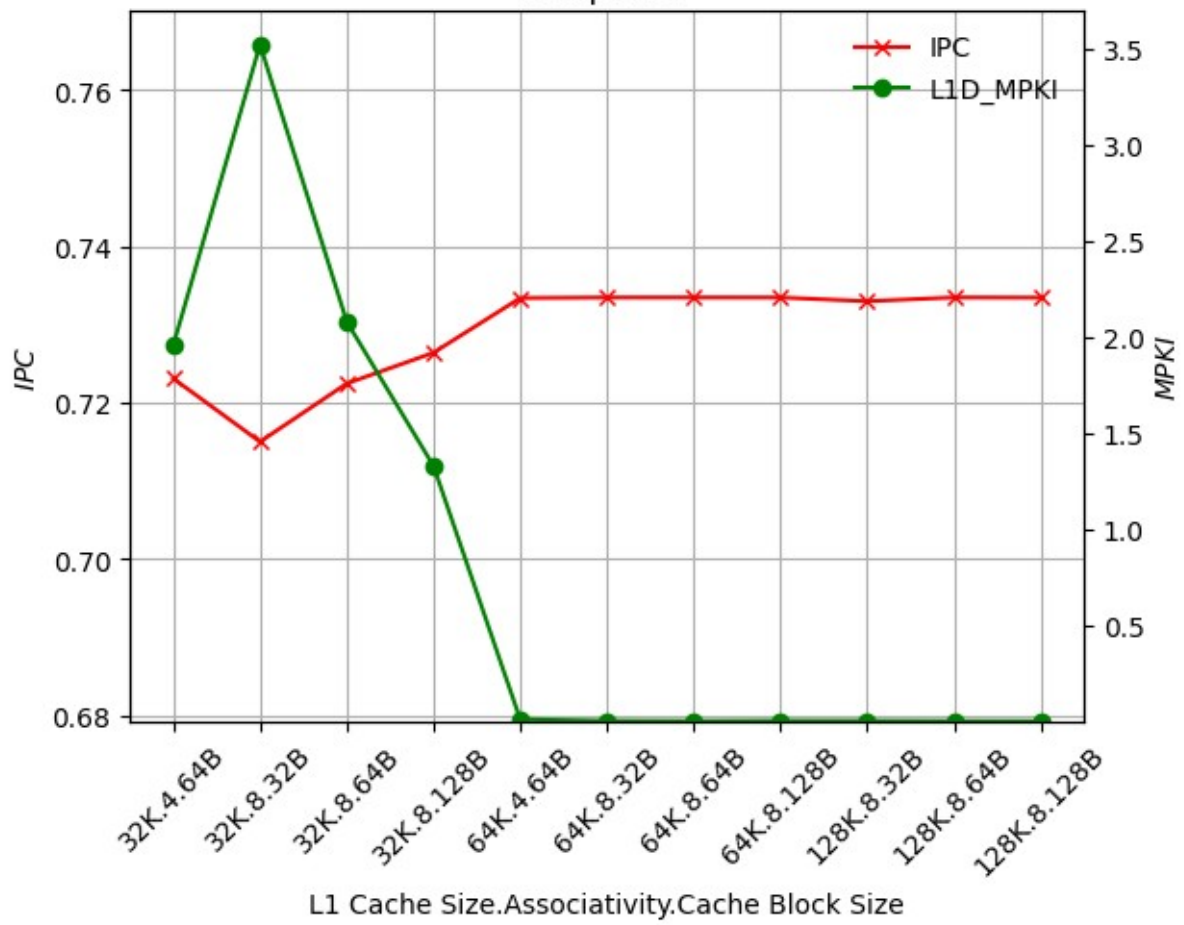








IPC vs MPKI  
swaptions



### Συμπεράσματα για L1 cache

- Φυσικά, ένα γενικό και προφανές συμπέρασμα που ισχύει σε όλα τα γραφήματα, είναι ότι όσο αυξάνονται τα MPKI (Misses Per Kilo Instructions) τόσο μειώνεται το IPC (Instructions Per Cycle), δηλαδή τα συνδέει μια σχέση αντίστροφης αναλογίας, αφού περισσότερα misses συνεπάγονται και περισσότερη καθυστέρηση στα διάφορα επίπεδα των caches, άρα και λιγότερες εκτελεσμένες εντολές ανά κύκλο ρολογιού.
- Στο blackscholes, παρατηρούμε ότι η αύξηση του associativity αυξάνει το IPC. Αυτό μπορεί να οφείλεται σε conflict misses που δεν μπορούν να αποφευχθούν με την αύξηση του μεγέθους, σε αντίθεση με τα capacity misses.
- Στο bodytrack, σε χαμηλά μεγέθη cache (32K) παρατηρείται αύξηση του IPC με την αύξηση του block size, ενώ σε υψηλότερα μεγέθη (64K, 128K) παρατηρείται μείωση του IPC με την αύξηση του block size.
- Στο canneal, fluidanimate και streamcluster φαίνεται να υπάρχει αύξηση του IPC όταν αυξάνεται το block size, κατί που μπορεί να οφείλεται στην επίλυση αρκετών compulsory misses. Στο freqmine, παρατηρείται το ίδιο, αλλά σε μικρότερο βαθμό και με την προσθήκη μίας ιδιόρρυθμης συμπεριφοράς, όπου για ορισμένες αλλαγές (διπλασιασμός associativity από 32.8.32 σε 32.8.64 και διπλασιασμός μεγέθους με υποτετραπλασιασμό του block size από 64.8.128 σε 128.8.32) πέφτει αρκετά το IPC.
- Στο rtview παρατηρείται μία παρόμοια πορεία με τα παραπάνω, με μικρότερη, όμως, ευαισθησία στις οποιεσδήποτε αλλαγές των παραμέτρων.
- Στο swaptions, παρατηρούμε πως, αρχικά, το IPC αυξάνεται ανάλογα με την αύξηση του block size όπως και στα παραπάνω, ενώ, όταν η cache φτάνει τα 64K, το IPC παραμένει σε μία σταθερή τιμή. Αυτό, πιθανόν να οφείλεται στην επίλυση όλων των capacity misses και στην έλλειψη compulsory και άλλων misses, ώστε να μην επηρεάζεται το MPKI και επομένως και το IPC.
- Γενικότερα, παρατηρούμε ότι πρώτο το block size και δεύτερο το associativity είναι οι πιο σημαντικοί παράμετροι που επηρεάζουν την απόδοση, σχετικά με τις παραμέτρους της L1 cache, ενώ το μέγεθος της επηρεάζει ουσιαστικά ελάχιστα το αποτέλεσμα (κατά κύριο λόγο μόνο στο swaptions benchmark).

## 1ο ζητούμενο

---

### L2 cache

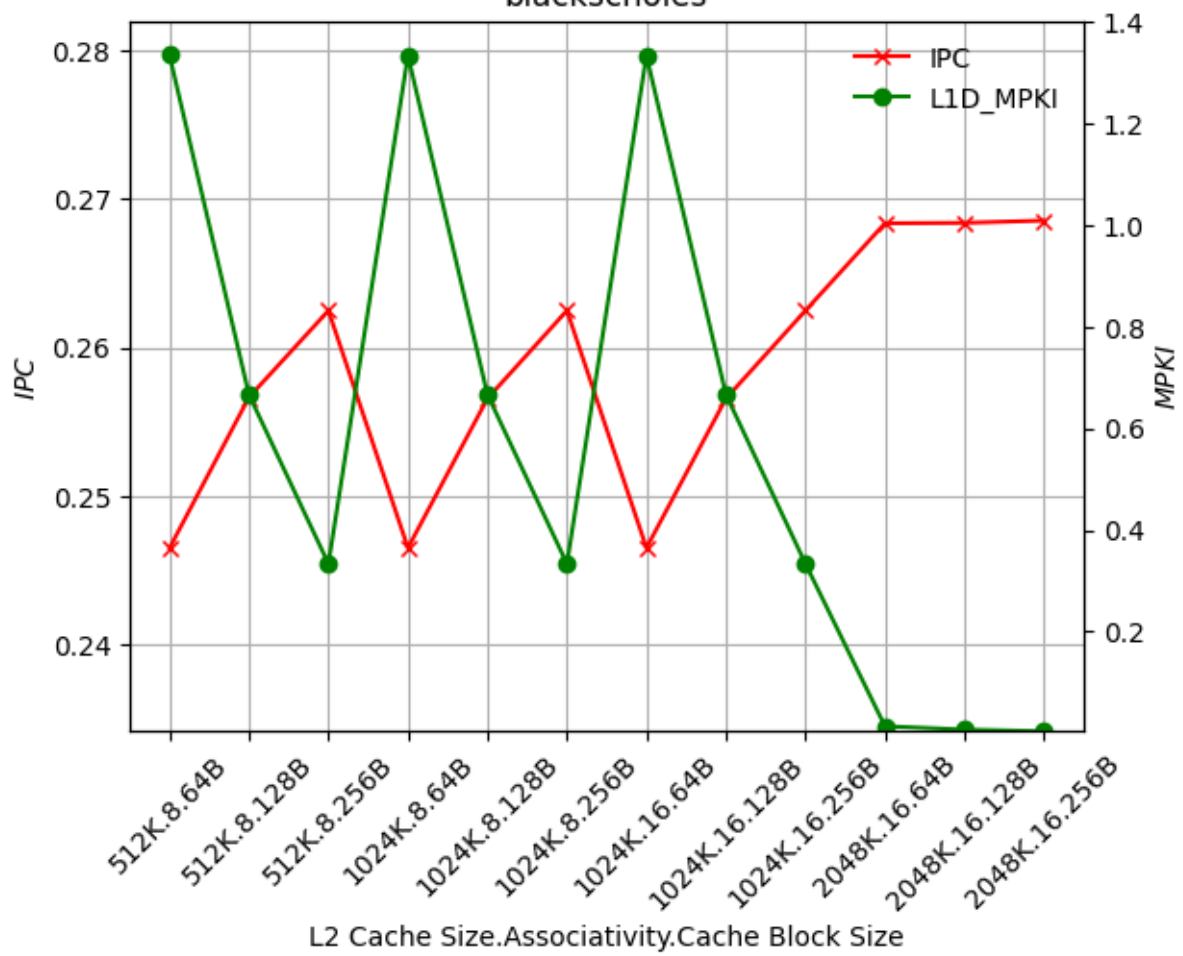
Για αυτό το πείραμα θεωρούμε πως ο κύκλος ρολογιού, όπως και ο αριθμός εκτελούμενων εντολών παραμένει σταθερός και πως οι μεταβολές στο IPC που παρατηρούνται στα γραφήματα αντιστοιχούν σε διαφορές στην απόδοση. Θα διατηρήσουμε τις παραμέτρους της L1 cache και του TLB σταθερές και ίσες με:

- L1 size = 32 KB
- L1 associativity = 8
- L1 block size = 64 B
- TLB size = 64 entries
- TLB associativity = 4
- TLB page size = 4096 B

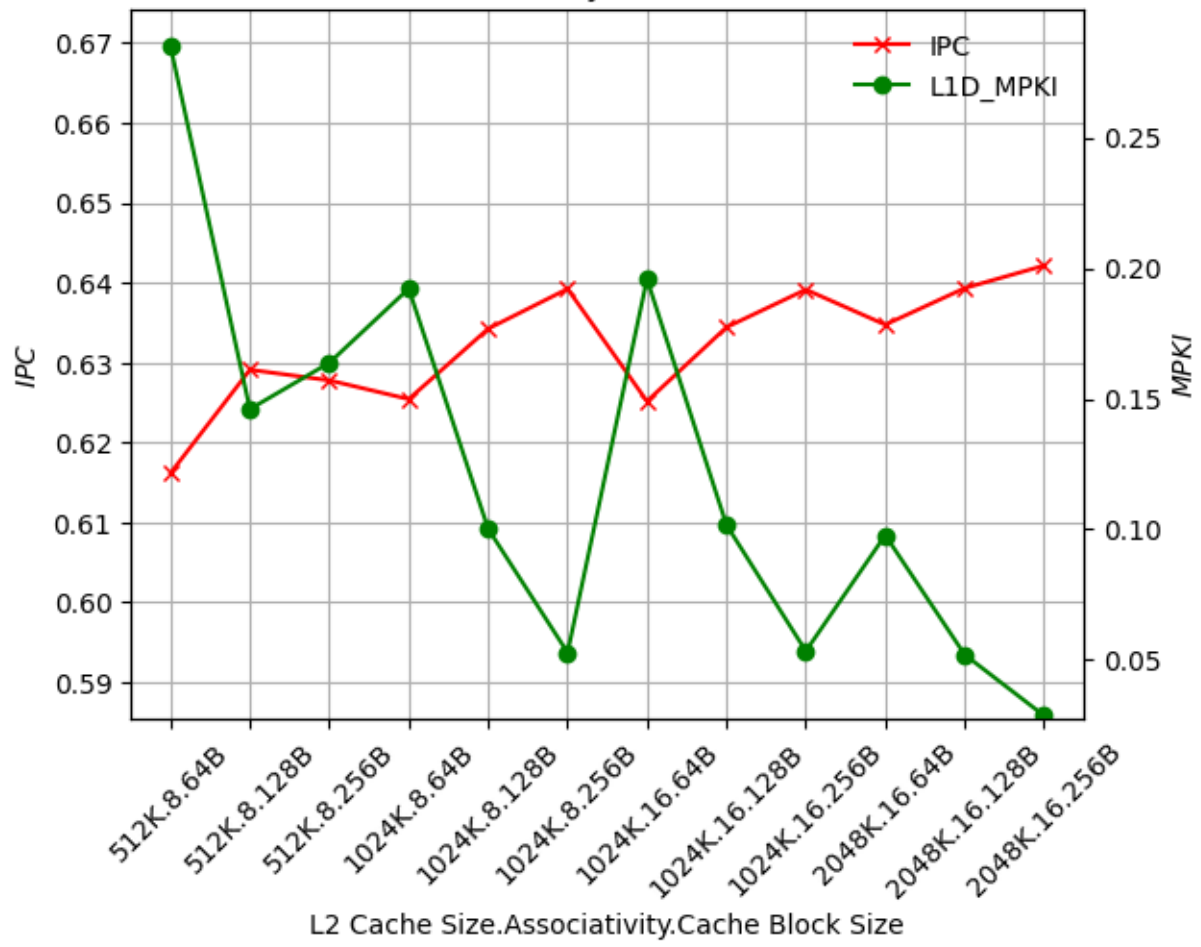
Επίσης, χρησιμοποιήσαμε τις παρακάτω δυνατές τριάδες παραμέτρων για τη L2 cache (με μορφή size.associativity.block\_size): 512.8.64, 512.8.128, 512.8.256, 1024.8.64, 1024.8.128, 1024.8.256, 1024.16.64, 1024.16.128, 1024.16.256, 2048.16.64, 2048.16.128, 2048.16.256.

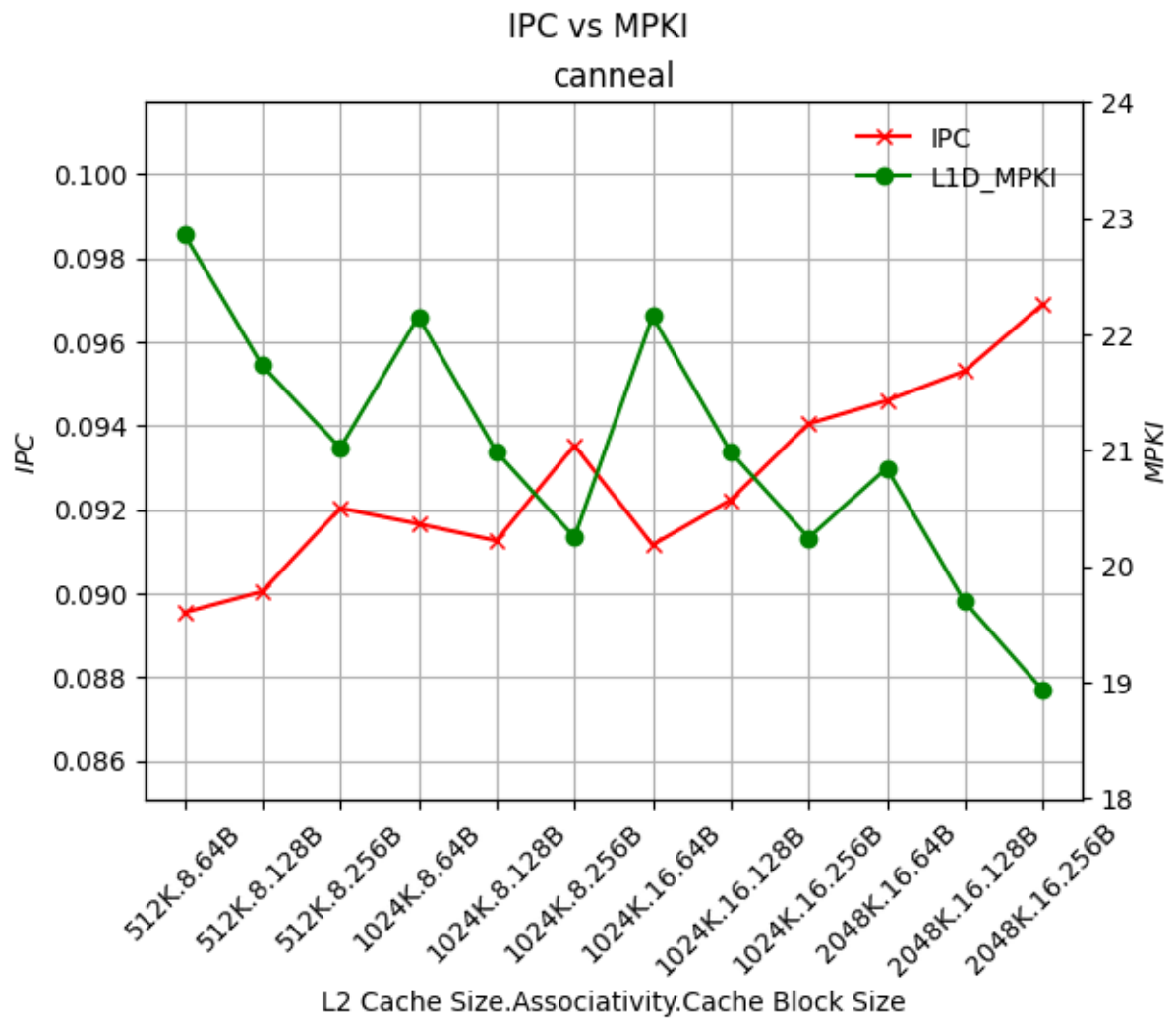
Ακολουθούν τα διαγράμματα που προέκυψαν, ένα για κάθε benchmark.

## IPC vs MPKI blackscholes

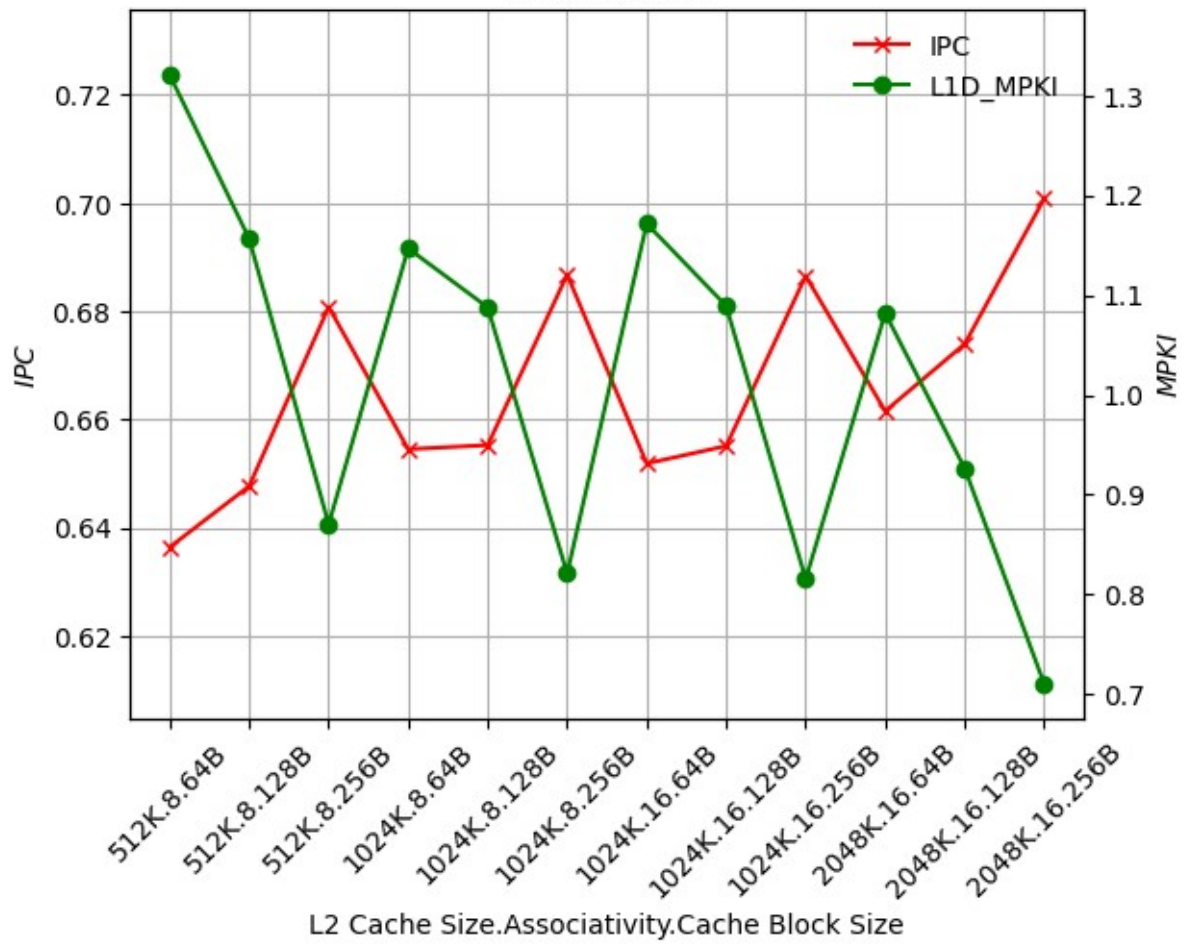


IPC vs MPKI  
bodytrack



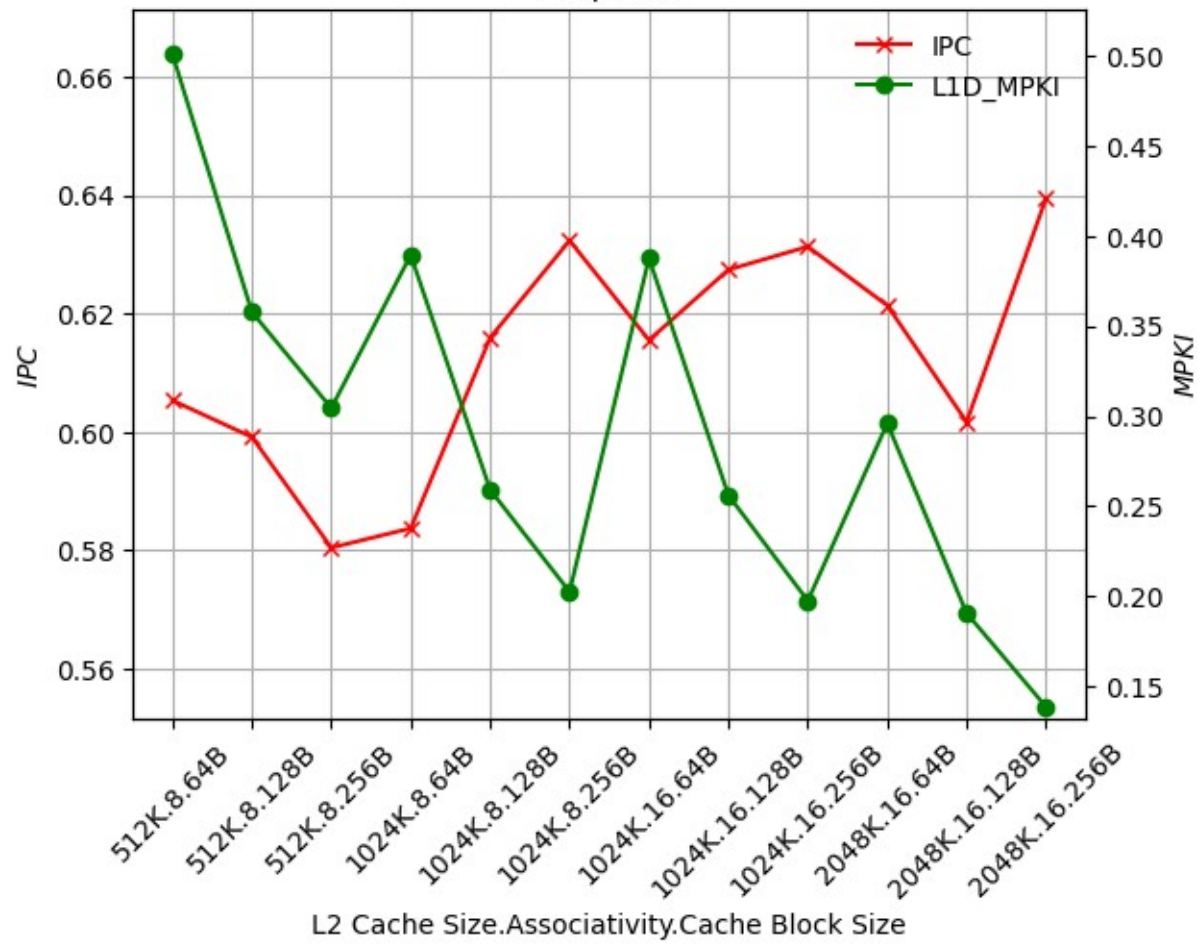


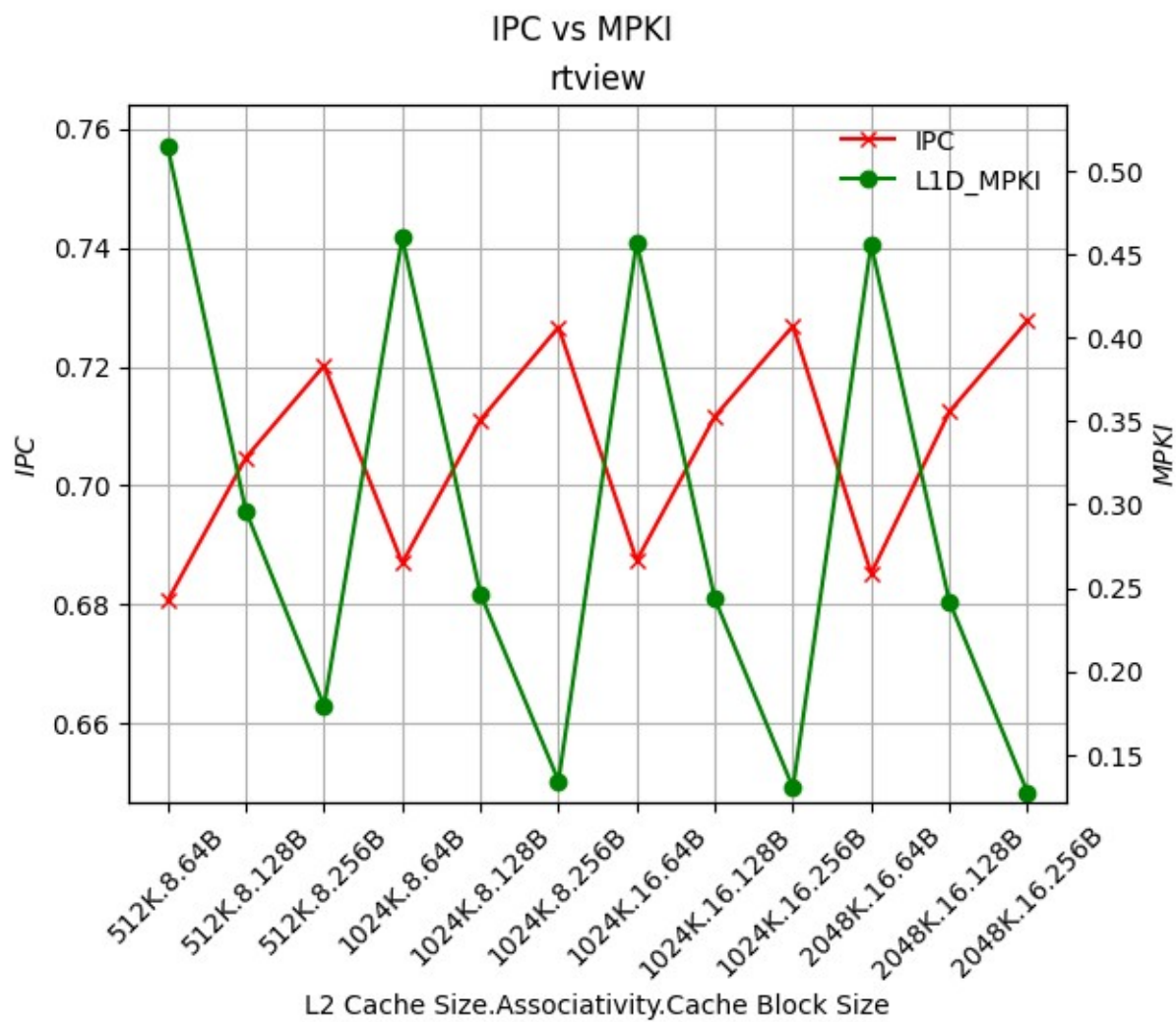
IPC vs MPKI  
fluidanimate

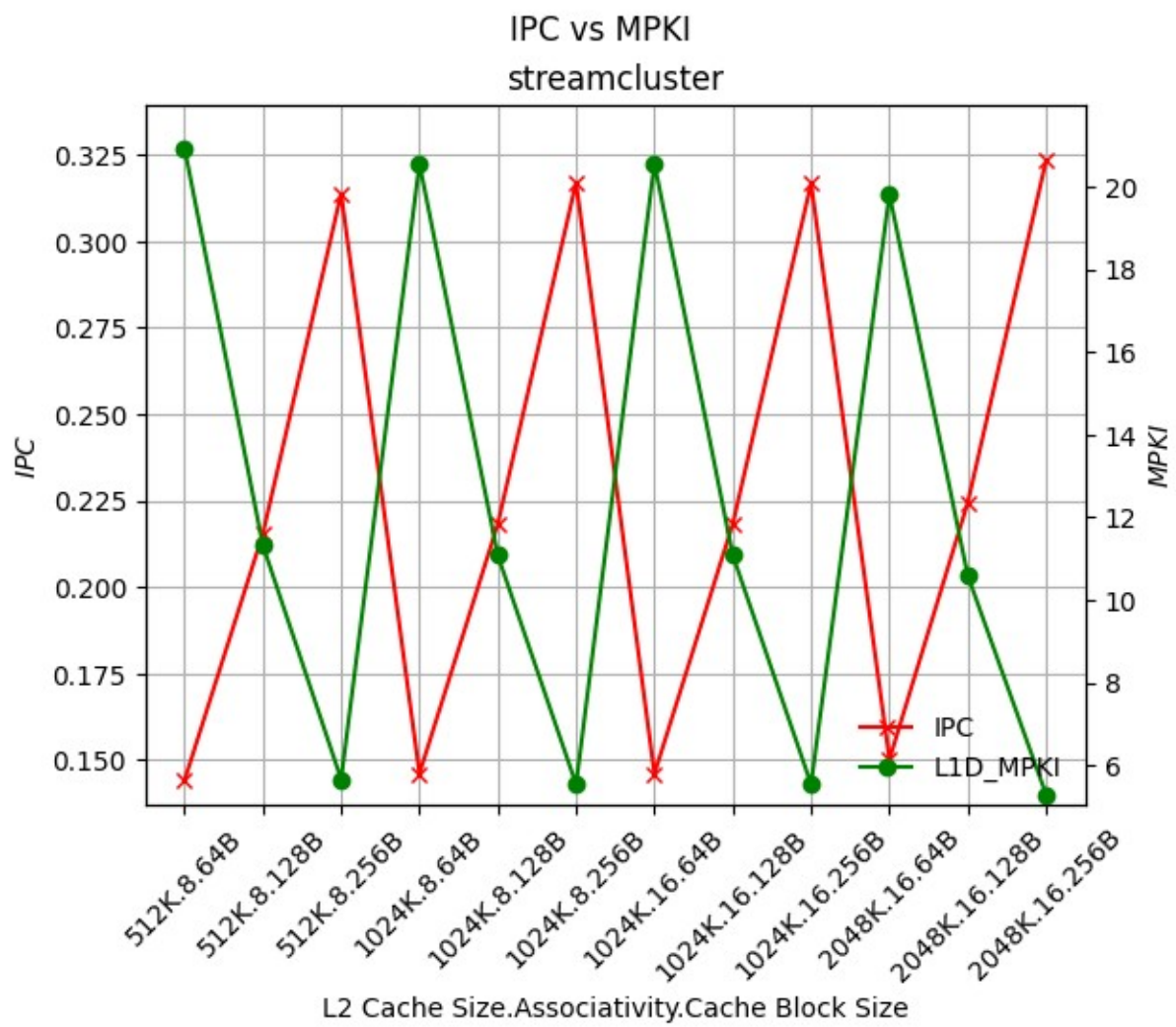


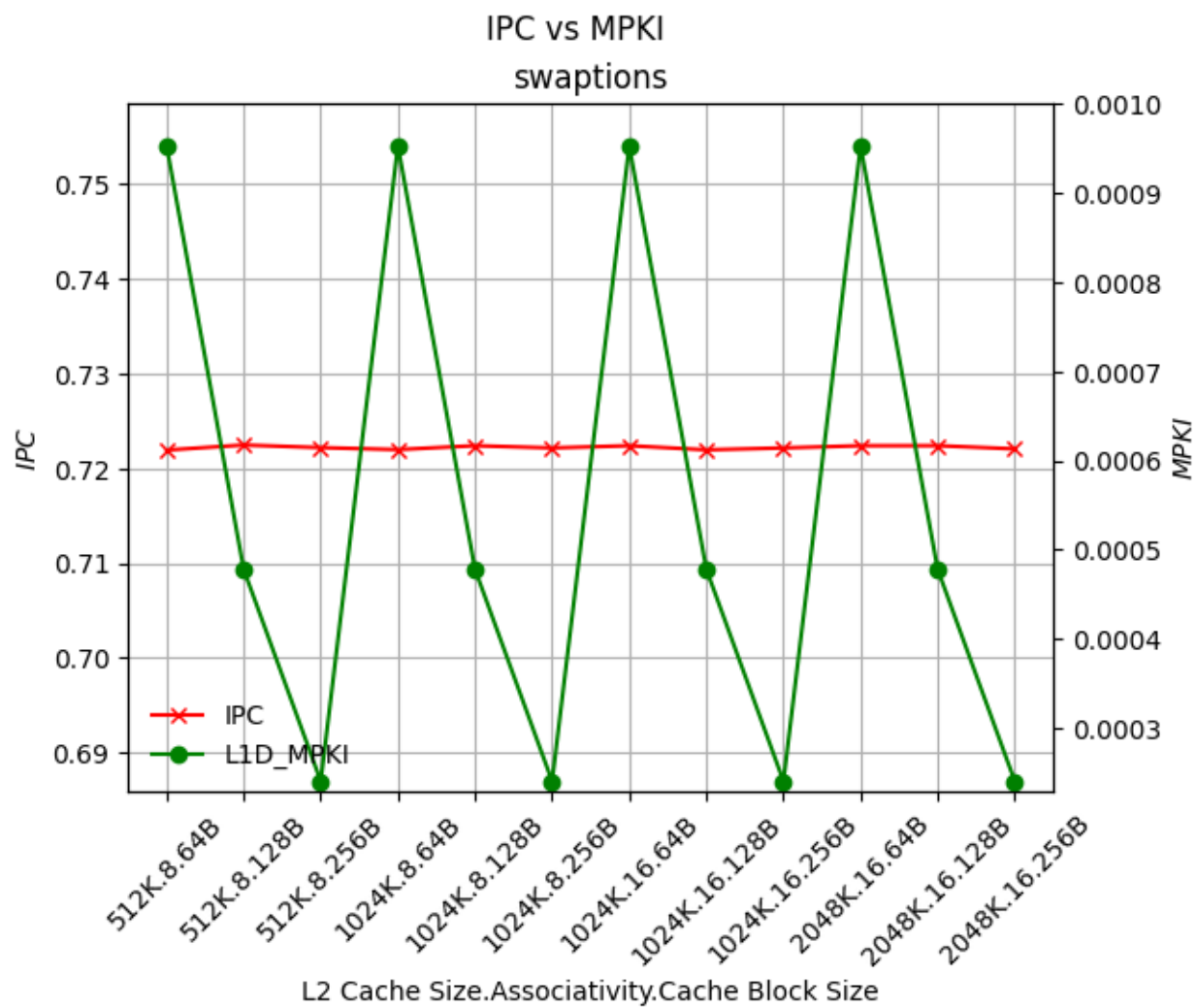


IPC vs MPKI  
freqmine









## 1ο ζητούμενο

---

### Συμπεράσματα για L2 cache

- Στο blackscholes, ενώ το μέγεθος φαίνεται να μην επηρεάζει αρχικά το IPC, όταν φτάσει τα 2048K το IPC μεγιστοποιείται και φαίνεται να παραμένει σταθερό, ανεξάρτητα από περαιτέρω αλλαγές στις παραμέτρους.
- Στο blackscholes, bodytrack, canneal, fluidanimate, rtview και streamcluster παρατηρούμε ότι η αύξηση του block size αυξάνει το IPC, πιθανόν λόγω επίλυσης αρκετών compulsory misses.
- Στο bodytrack, canneal και freqmine φαίνεται πως με αύξηση του μεγέθους της cache αυξάνεται και το IPC, κάτι που λογικά οφείλεται στην επίλυση αρκετών capacity misses.
- Στο freqmine, παρατηρείται αύξηση του IPC με αύξηση του associativity, λόγω επίλυσης conflict misses.
- Στο swaptions, εμφανίζεται πάλι μια σχεδόν σταθερή τιμή για το IPC, ασχέτως από τις διάφορες αλλαγές στις παραμέτρους της cache. Αυτό μπορεί να σημαίνει ότι χρησιμοποιεί συνεχώς τα ίδια δεδομένα, επομένως με ελάχιστα compulsory misses και συνεχώς hit εξυπηρεείται με σχεδόν σταθερό IPC.
- Γενικότερα, παρατηρούμε ότι και πάλι το block size παίζει το μεγαλύτερο παράγοντα για την αύξηση του IPC με το μέγεθος της cache να προσφέρει επίσης ορισμένες βελτιώσεις, όπως και το associativity.

## 1ο ζητούμενο

---

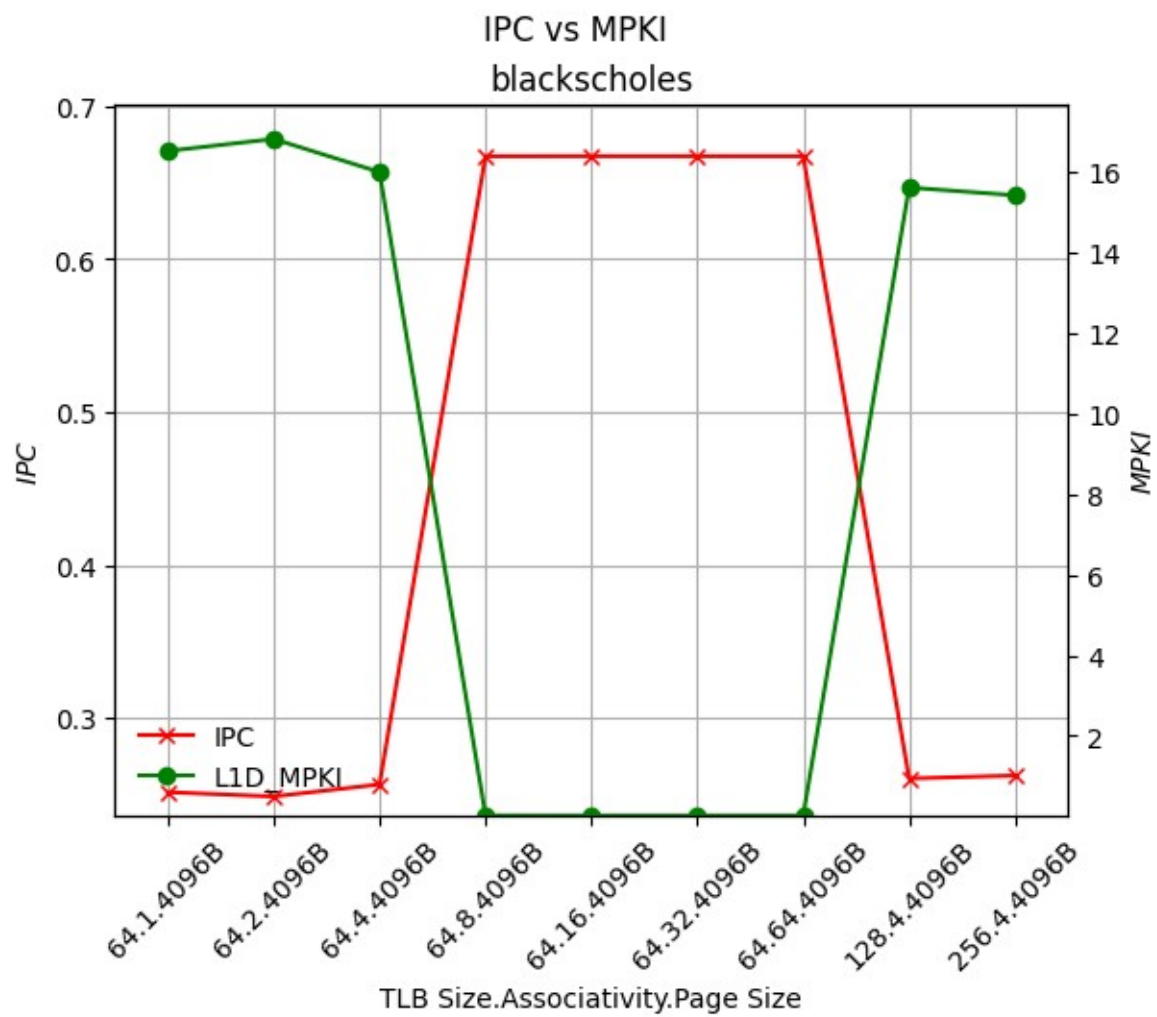
### TLB

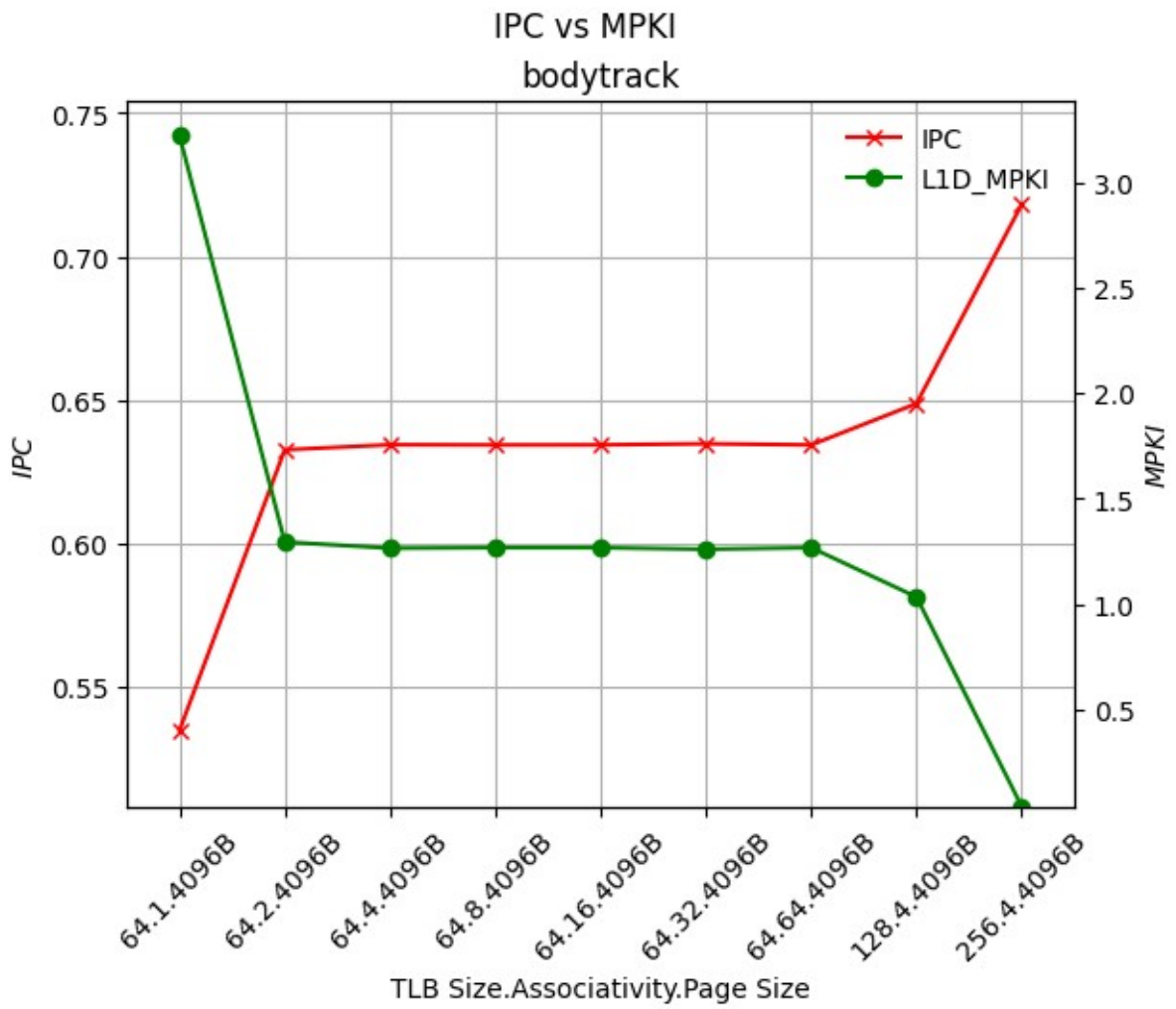
Για αυτό το πείραμα θεωρούμε πως ο κύκλος ρολογιού, όπως και ο αριθμός εκτελούμενων εντολών παραμένει σταθερός και πως οι μεταβολές στο IPC που παρατηρούνται στα γραφήματα αντιστοιχούν σε διαφορές στην απόδοση. Θα διατηρήσουμε τις παραμέτρους της L1 και L2 cache σταθερές και ίσες με:

- L1 size = 32 KB
- L1 associativity = 8
- L1 block size = 64 B
- L2 size = 1024 KB
- L2 associativity = 8
- L2 block size = 128 B

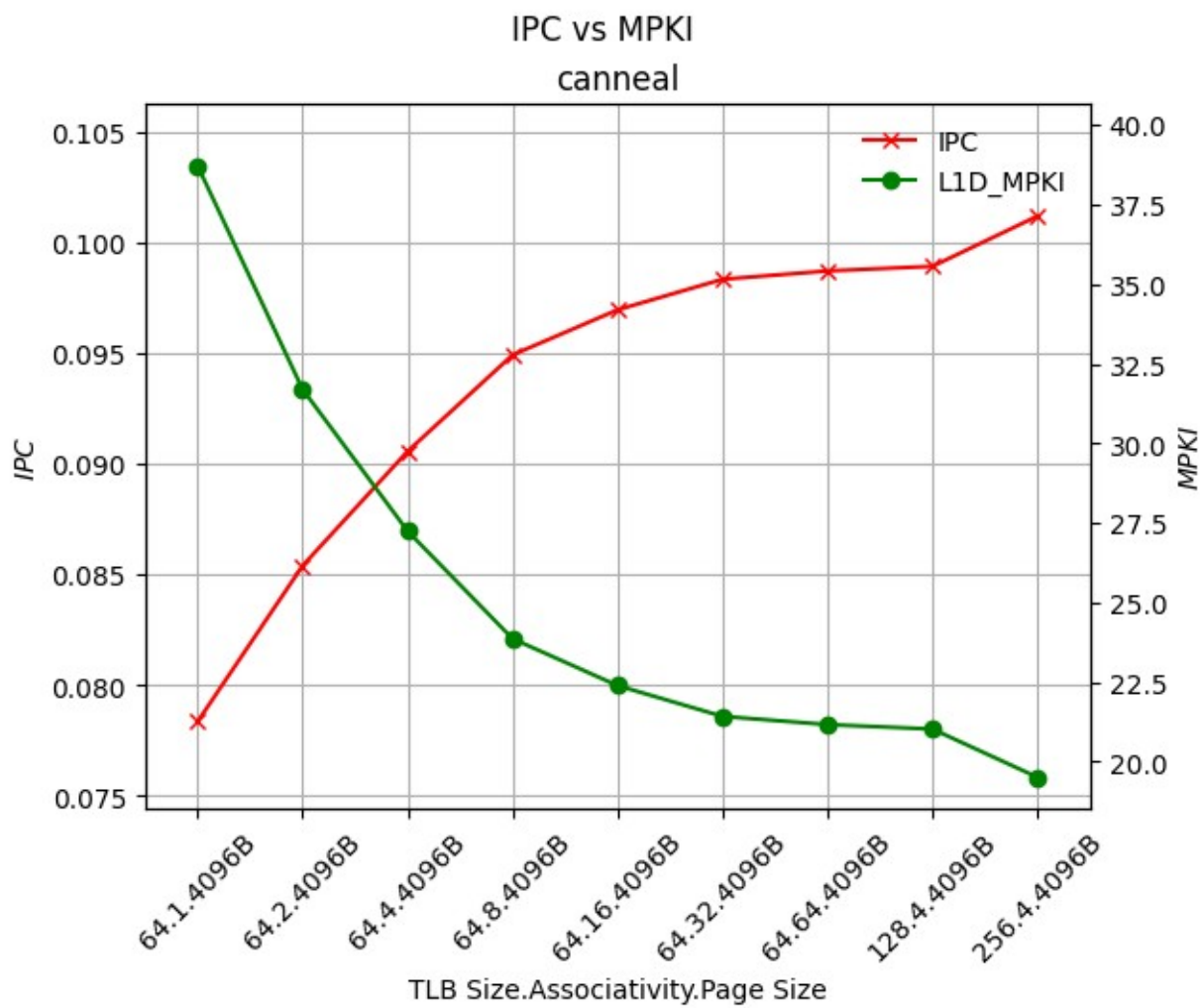
Επίσης, χρησιμοποιήσαμε τις παρακάτω δυνατές τριάδες παραμέτρων για το TLB (με μορφή size.associativity.page\_size): 64.1.4096, 64.2.4096, 64.4.4096, 64.8.4096, 64.16.4096, 64.32.4096, 64.64.4096, 128.4.4096, 256.4.4096.

Ακολουθούν τα διαγράμματα που προέκυψαν, ένα για κάθε benchmark.

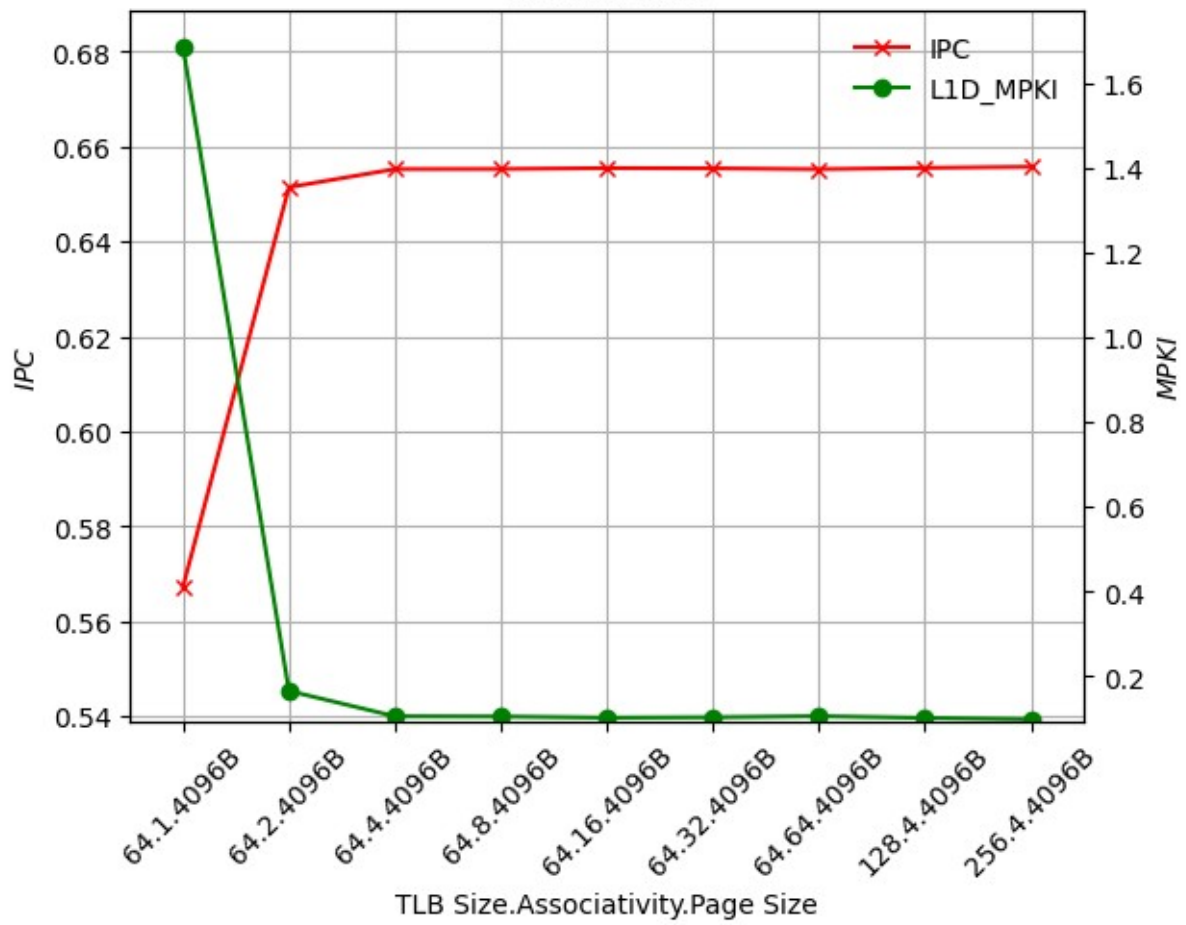


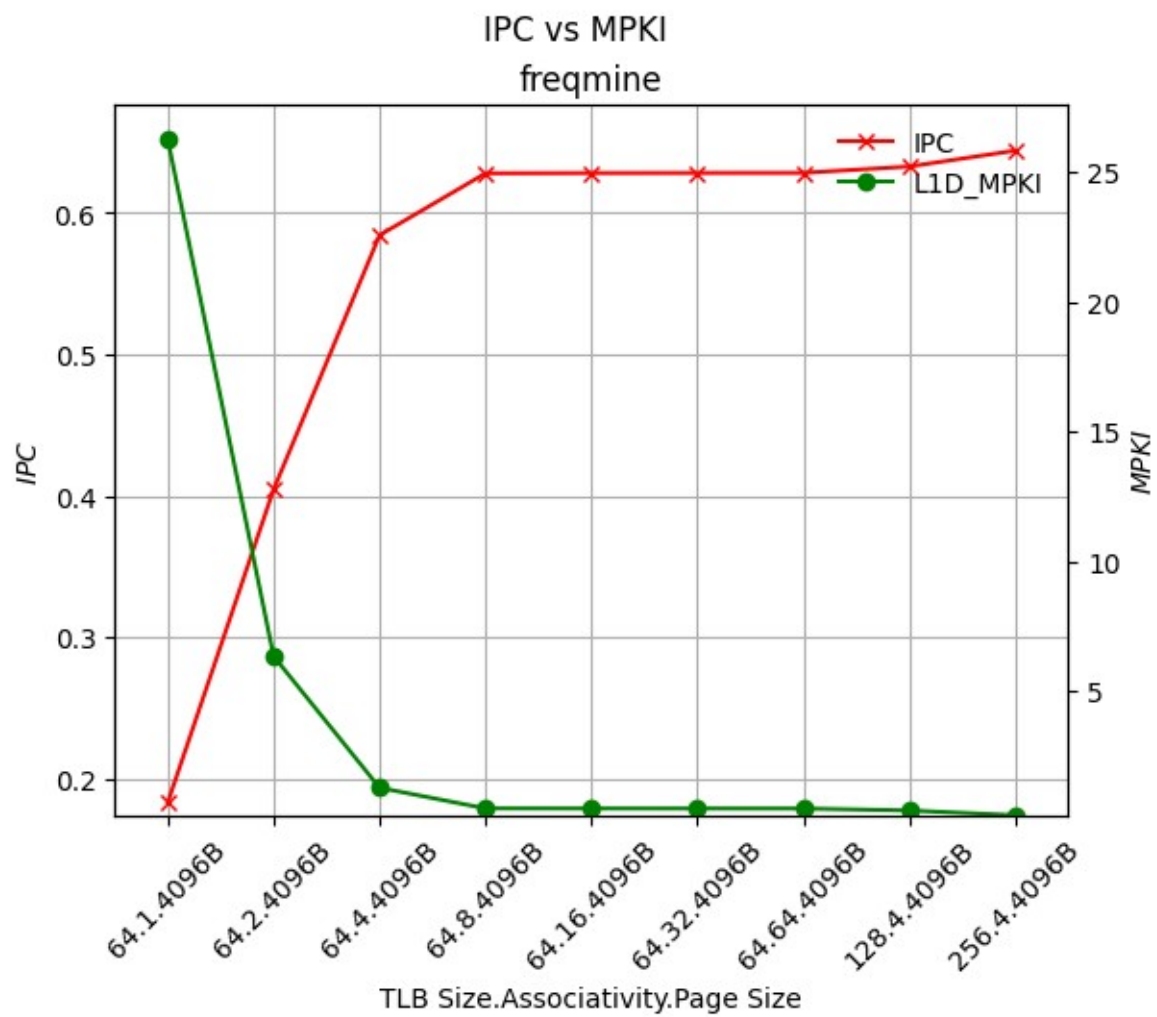






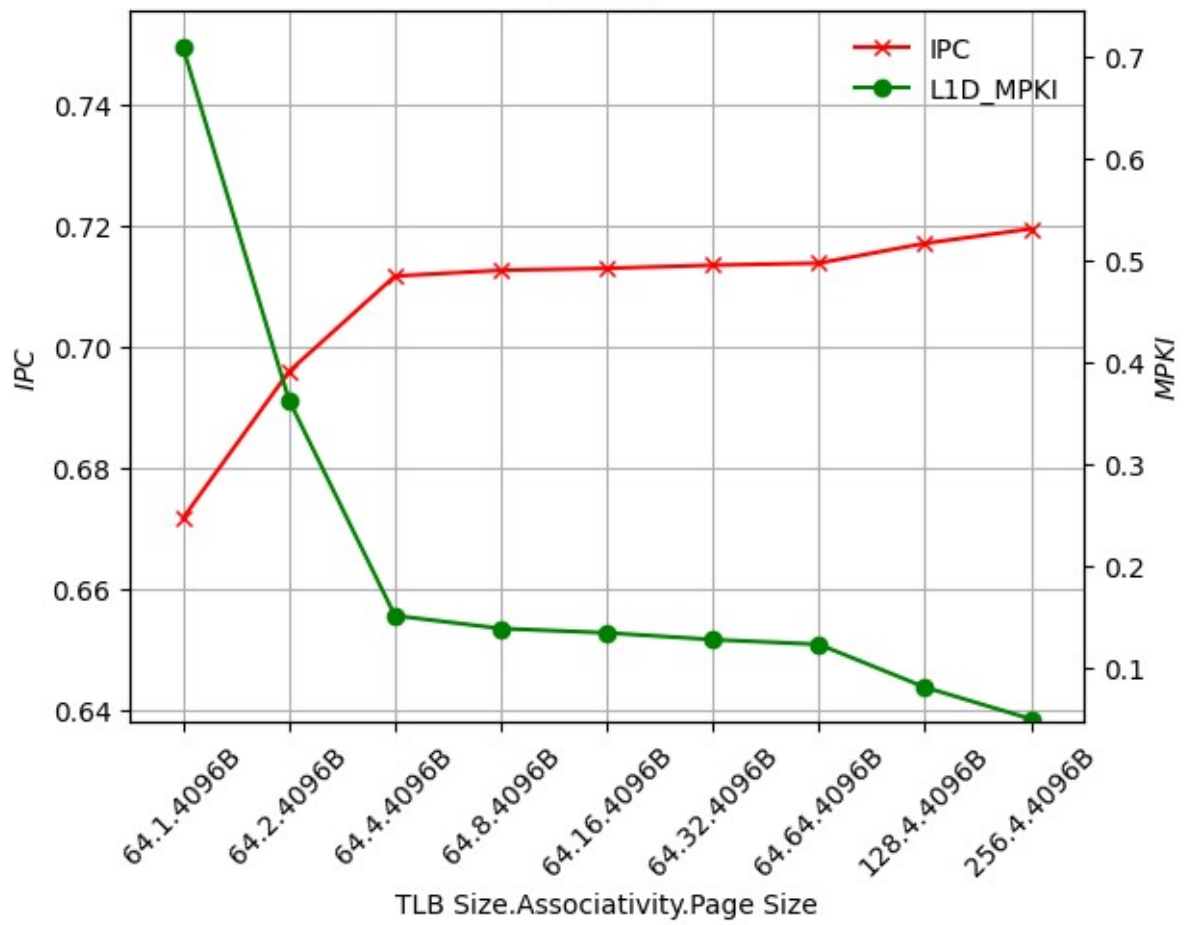
IPC vs MPKI  
fluidanimate

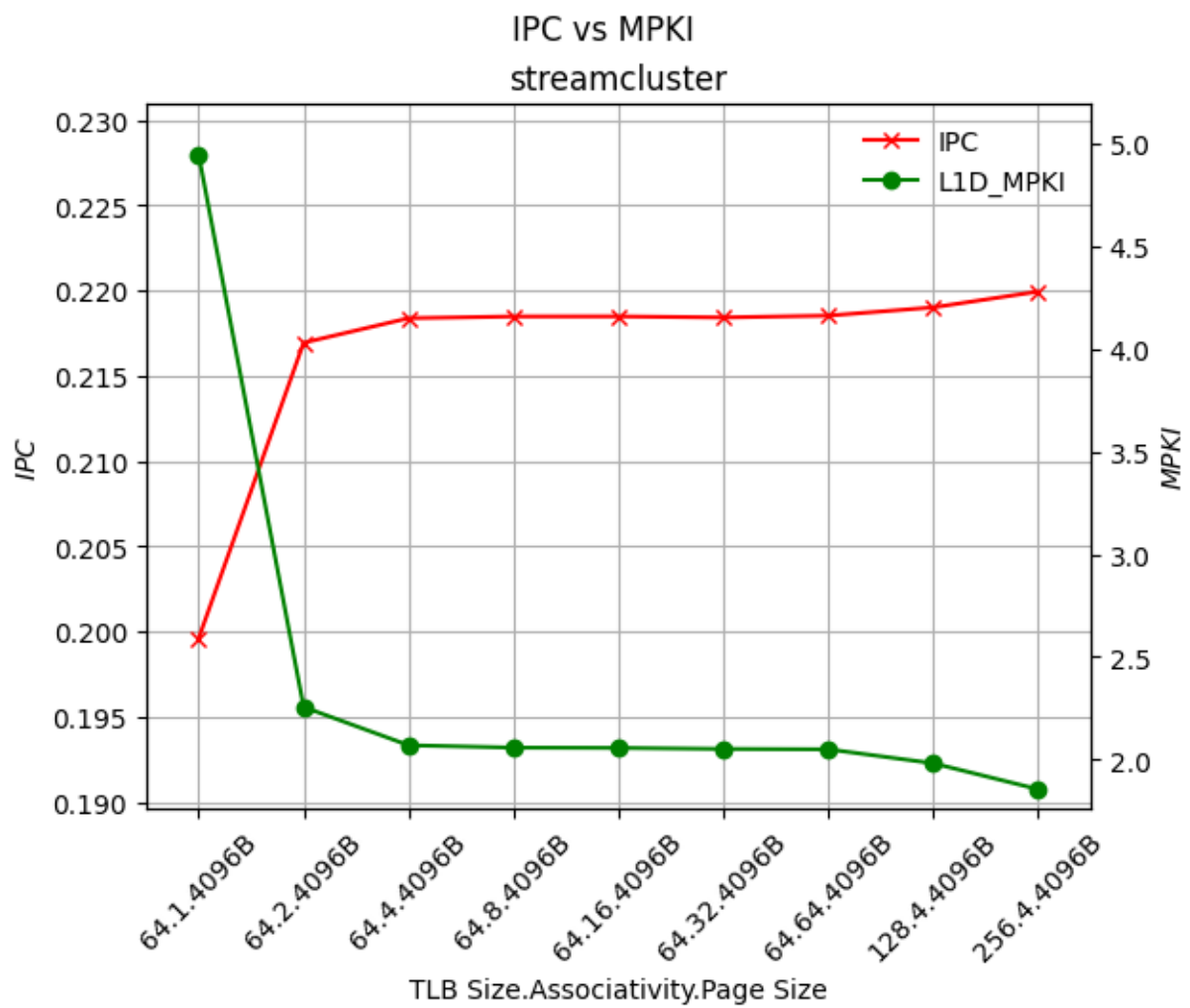


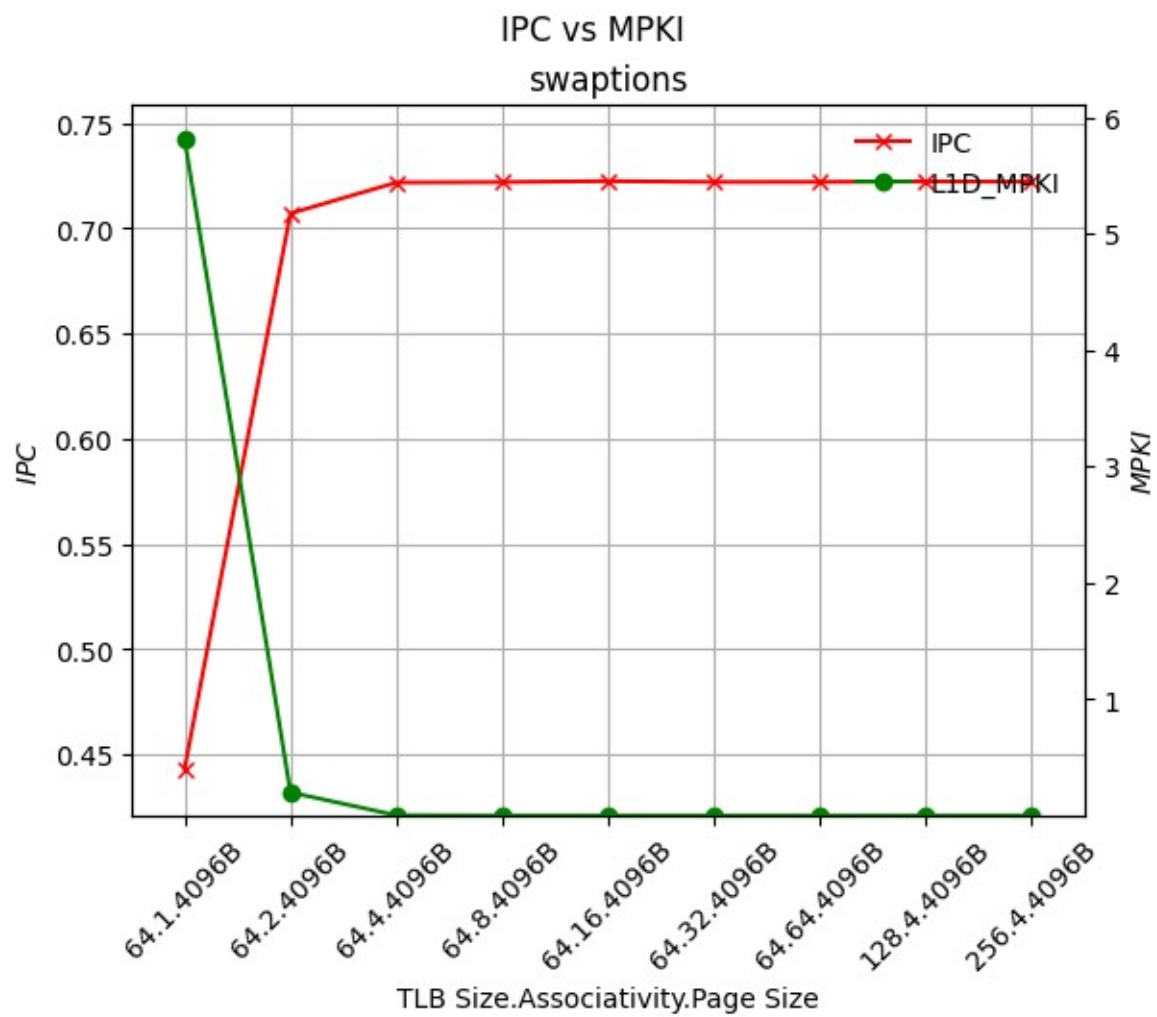


# IPC vs MPKI

rtview







### Συμπεράσματα για TLB

- Στο bodytrack, canneal, freqmine, rtview και streamcluster, η αύξηση του μεγέθους TLB από 64 σε 128 και 256 προσφέρει, έστω και μια μικρή, αύξηση στο IPC, λόγω των μειωμένων misses.
- Στο bodytrack, fluidanimate, streamcluster και swaptions το associativity αυξάνει το IPC για την πρώτη αύξηση του από 1 σε 2, ενώ στη συνέχεια περαιτέρω αυξήσεις δεν βελτιώνουν το IPC, ίσως διότι τα conflict misses είναι λίγα και έχουν επιλυθεί ήδη με την πρώτη αύξηση.
- Στο canneal, freqmine, rtview συμβαίνει το ίδιο με τα παραπάνω απλά σε μεταγενέστερη αύξηση, ωστόσο και αυτά φαίνεται μετά από μερικές αυξήσεις να εξυπηρετούν όλα τα conflict misses τους και να σταθεροποιούν το IPC.
- Το TLB page size δεν μπορούμε να εξετάσουμε εάν επηρεάζει κάπως το IPC, καθώς είναι ίσο με 4096B σε όλες τις μετρήσεις.
- Γενικότερα, το μέγεθος του TLB όσο και το associativity φαίνεται να αυξάνουν το IPC, ωστόσο κυρίως το associativity μέχρι ένα συγκεκριμένο σημείο, που για αρκετά benchmarks έρχεται σχετικά νωρίς.

## 1ο ζητούμενο

---

### Prefetching

Για αυτό το πείραμα θεωρούμε πως ο κύκλος ρολογιού, όπως και ο αριθμός εκτελούμενων εντολών παραμένει σταθερός και πως οι μεταβολές στο IPC που παρατηρούνται στα γραφήματα αντιστοιχούν σε διαφορές στην απόδοση. Θα διατηρήσουμε τις παραμέτρους της L1 και L2 cache, καθώς και του TLB σταθερές και ίσες με:

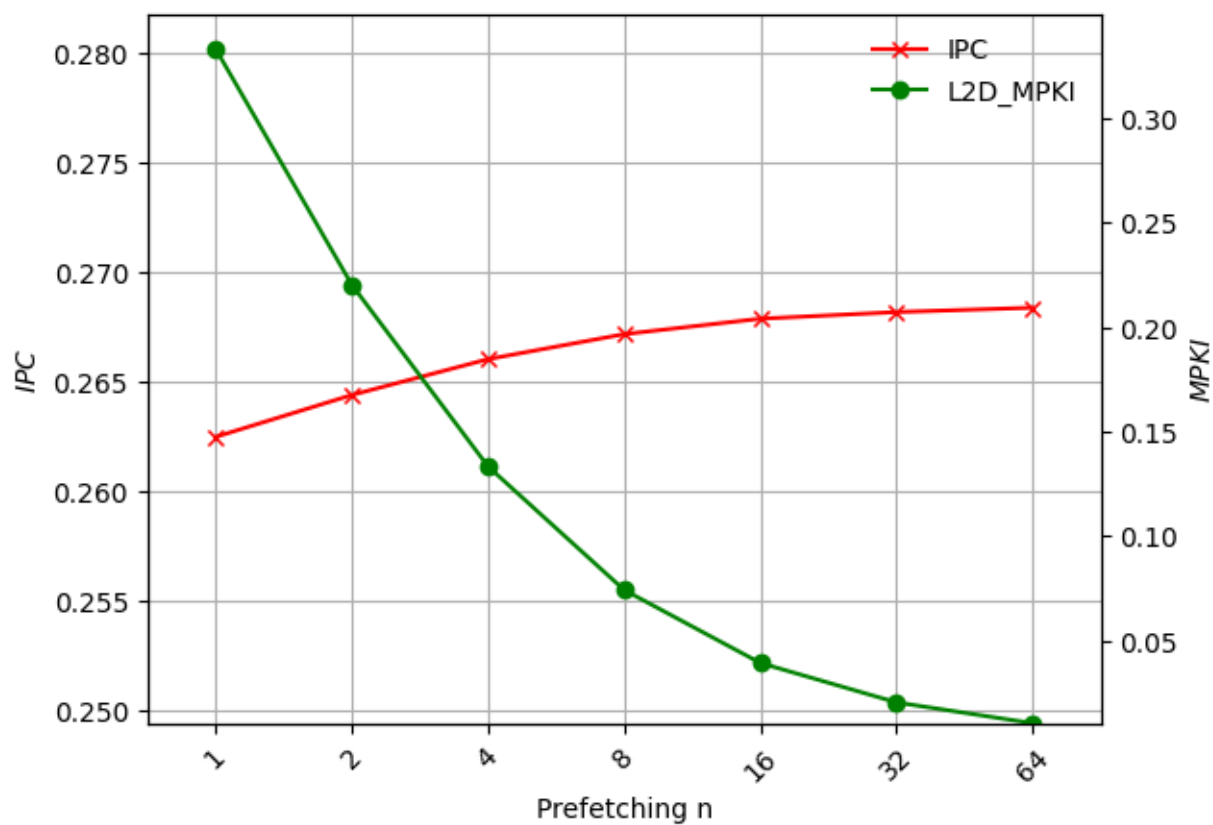
- L1 size = 32 KB
- L1 associativity = 8
- L1 block size = 64 B
- L2 size = 1024 KB
- L2 associativity = 8
- L2 block size = 128 B
- TLB size = 64 entries
- TLB associativity = 4
- TLB page size = 4096 B

Επίσης, χρησιμοποιήσαμε τις παρακάτω δυνατές τιμές για τα N blocks που θα φέρνει η L2 cache μαζί με το ζητούμενο block, λόγω του NEXT-N-LINE prefetching: 1, 2, 4, 8, 16, 32, 64.

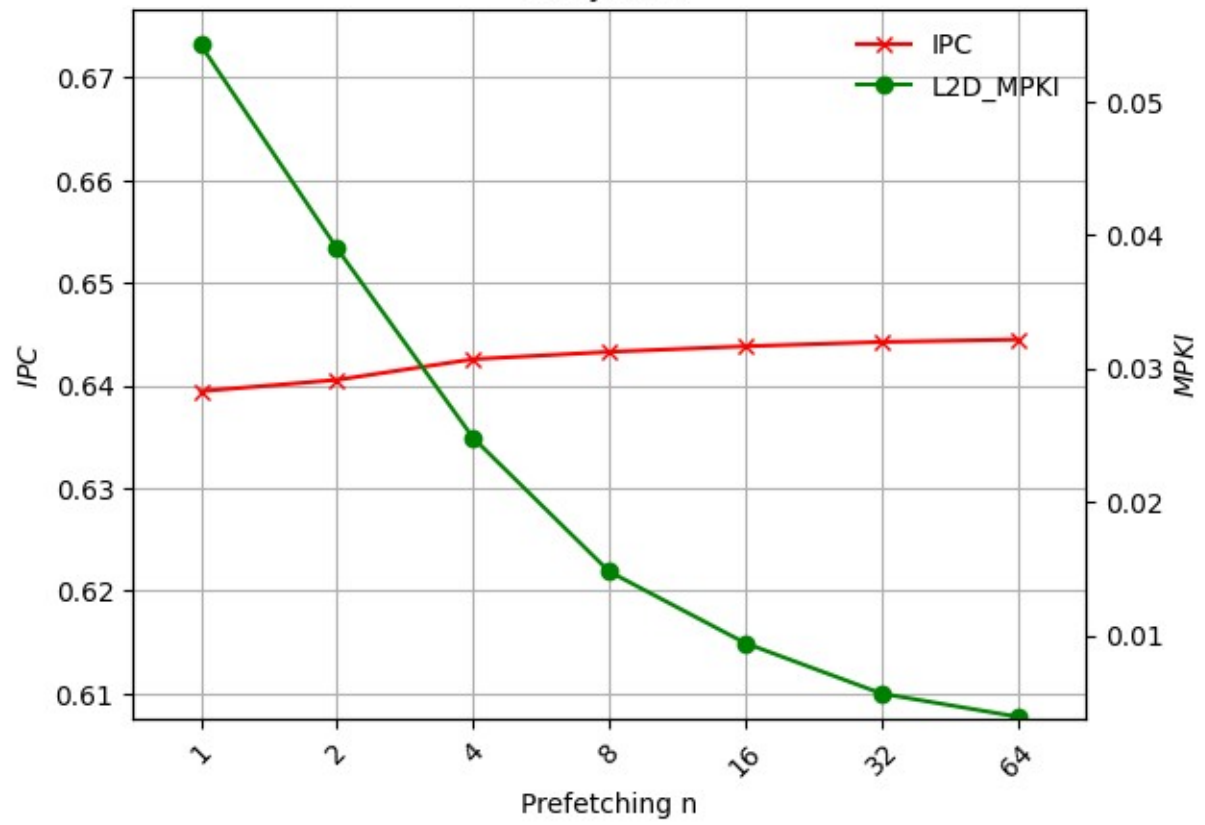
Ακολουθούν τα διαγράμματα που προέκυψαν, ένα για κάθε benchmark.



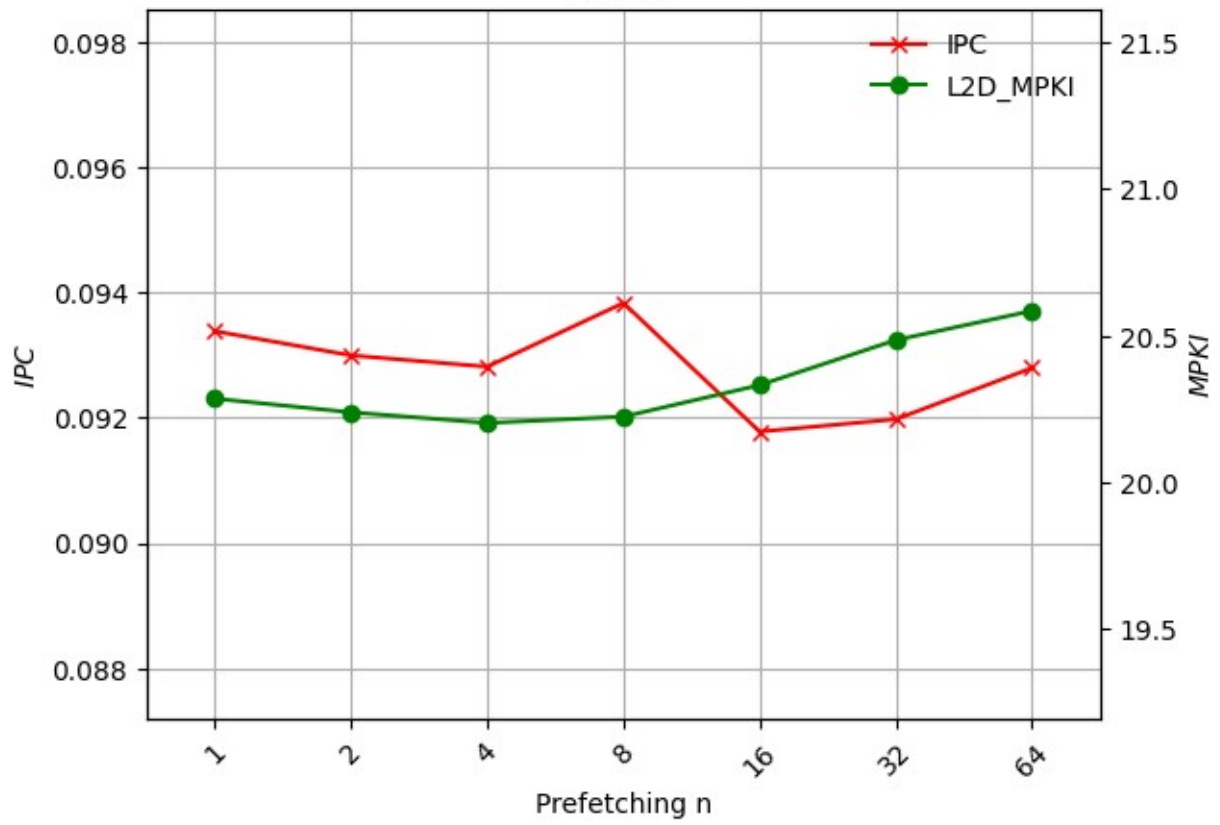
IPC vs MPKI  
blackscholes



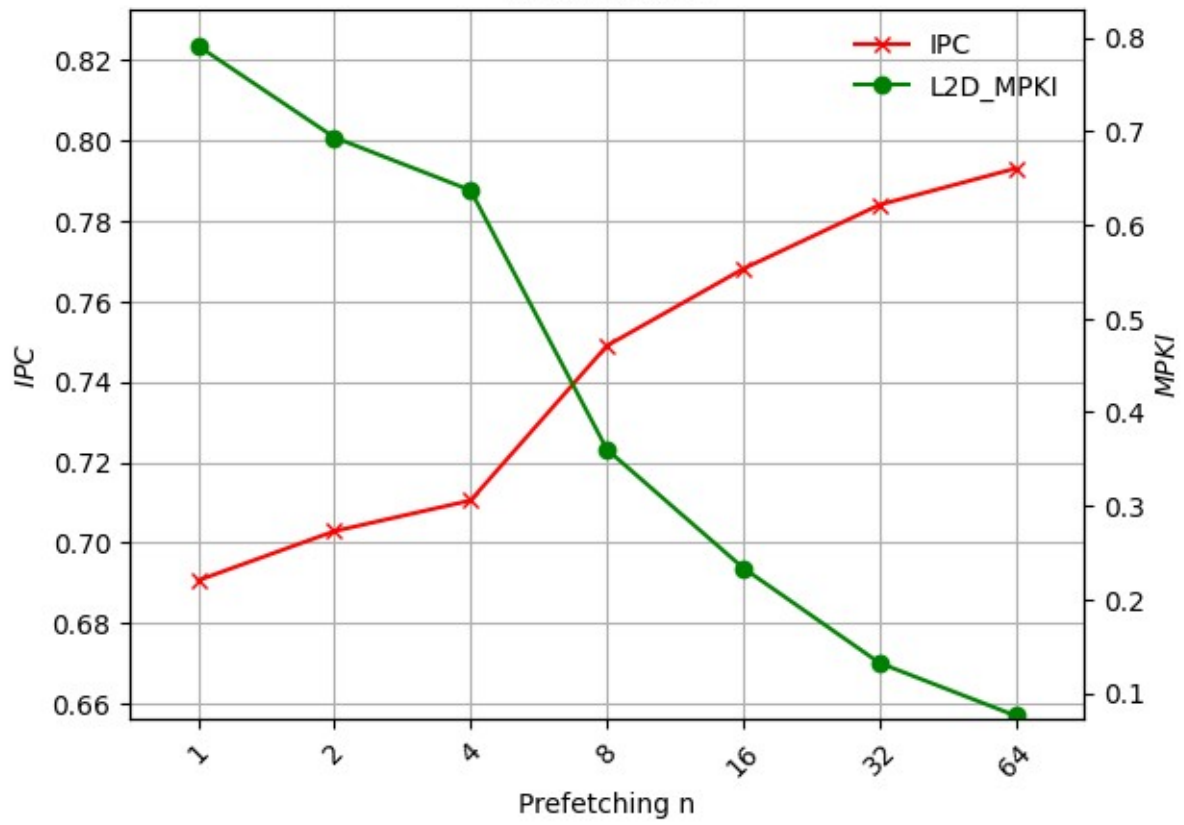
IPC vs MPKI  
bodytrack



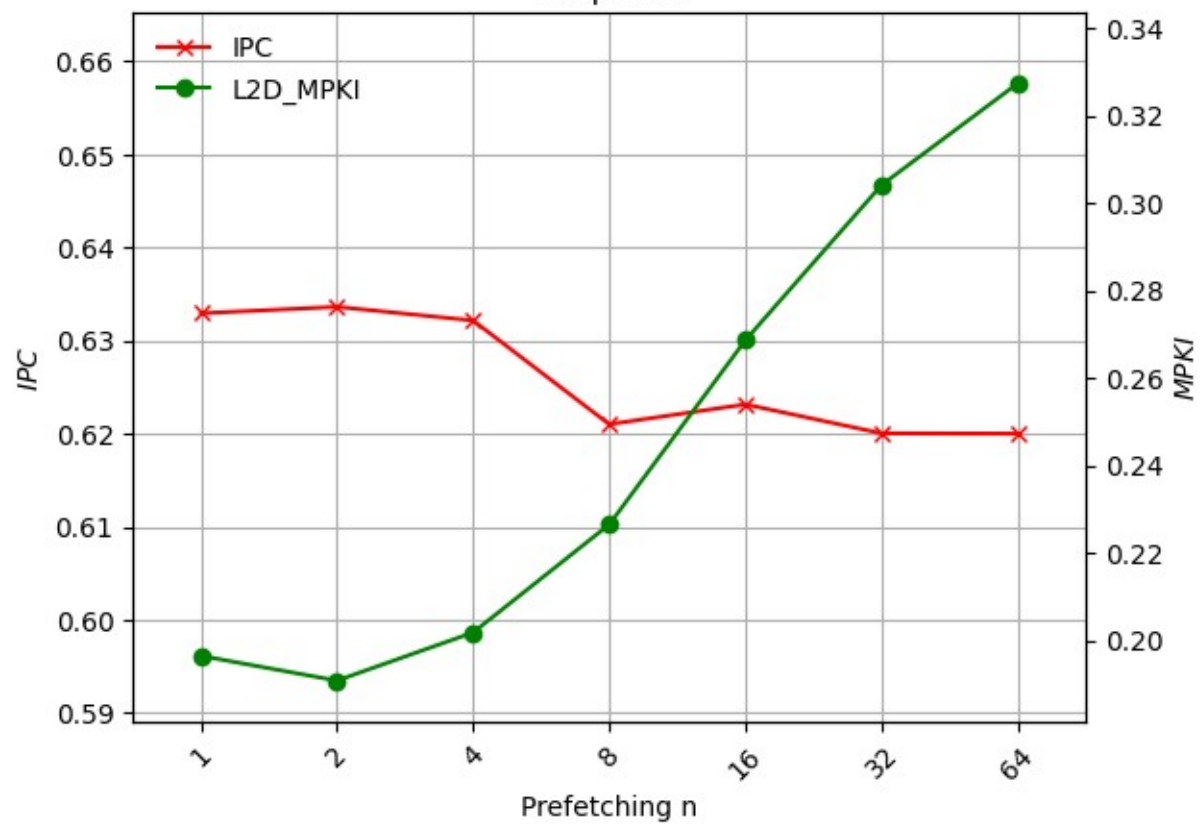
IPC vs MPKI  
canneal

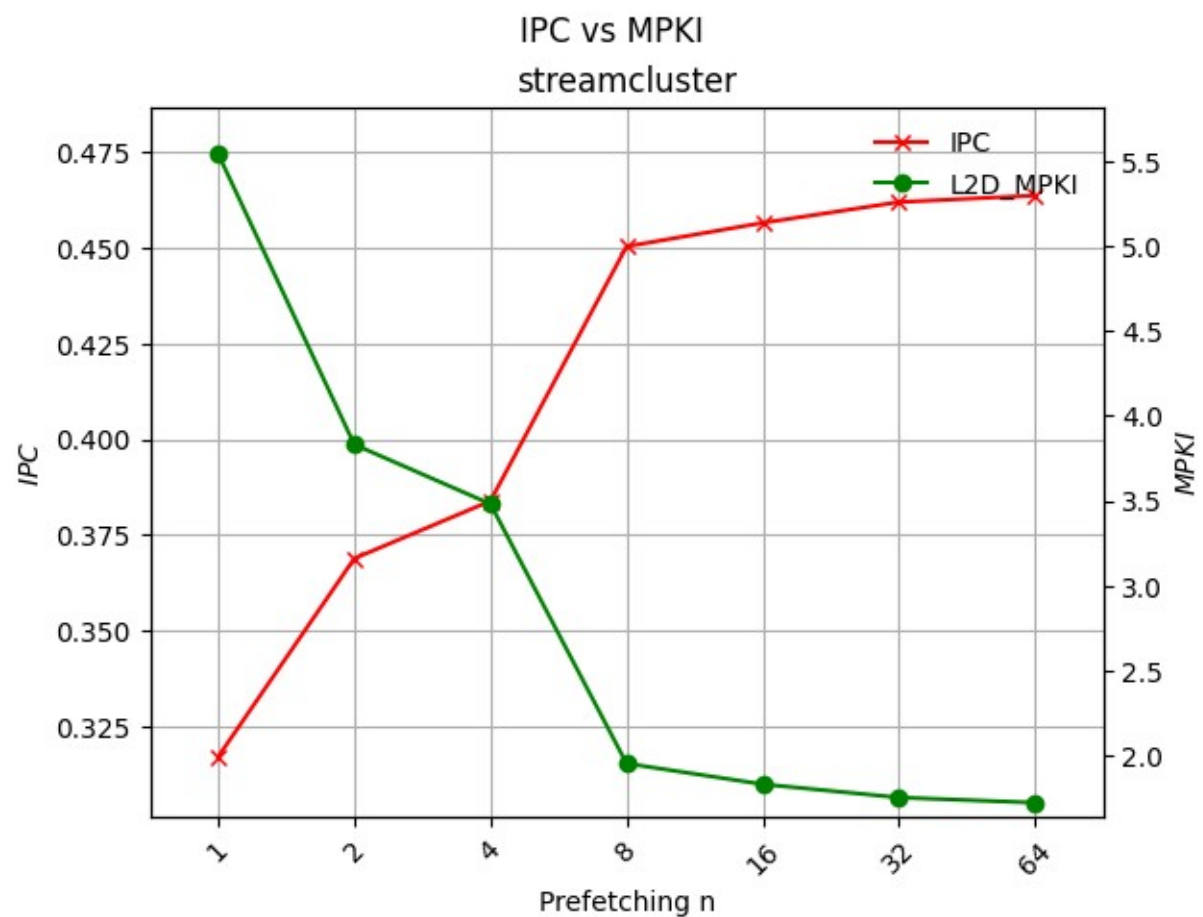
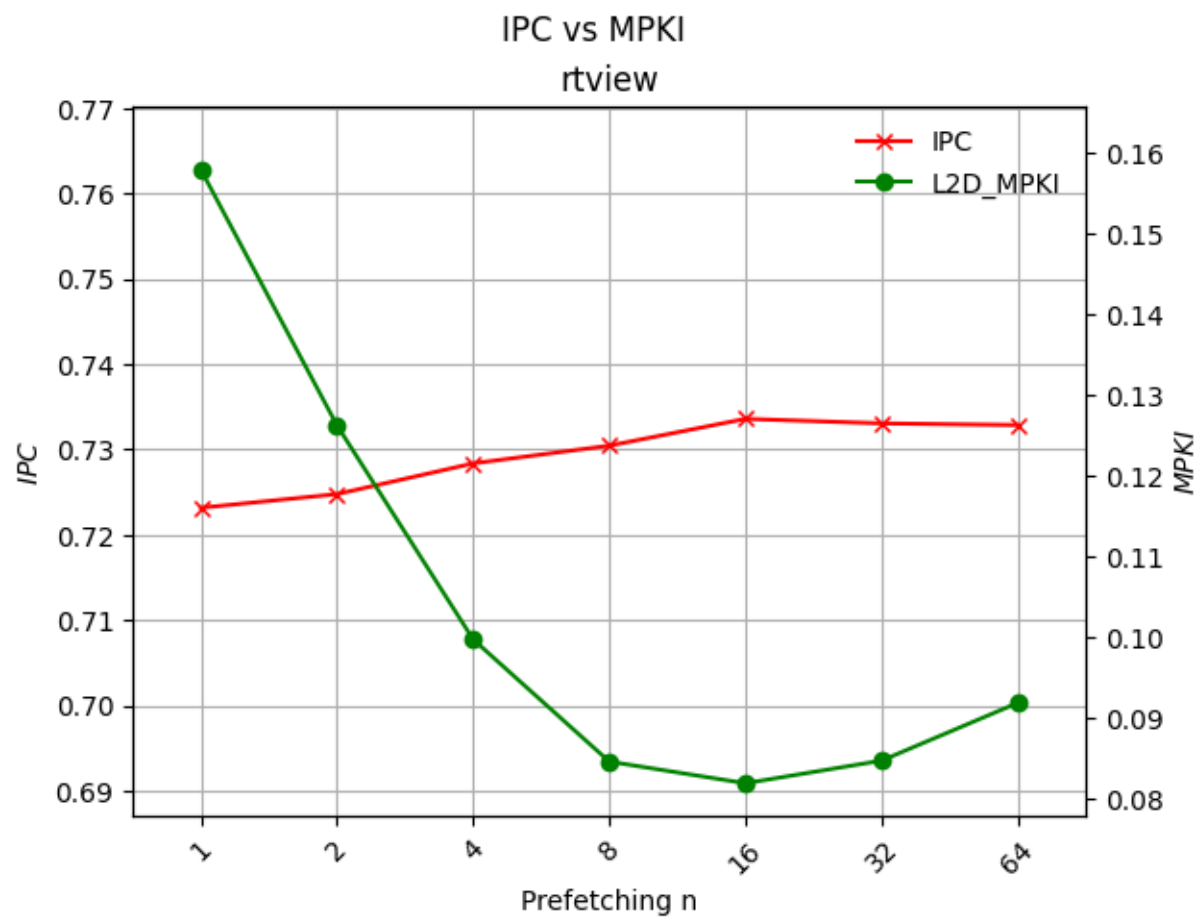


IPC vs MPKI  
fluidanimate

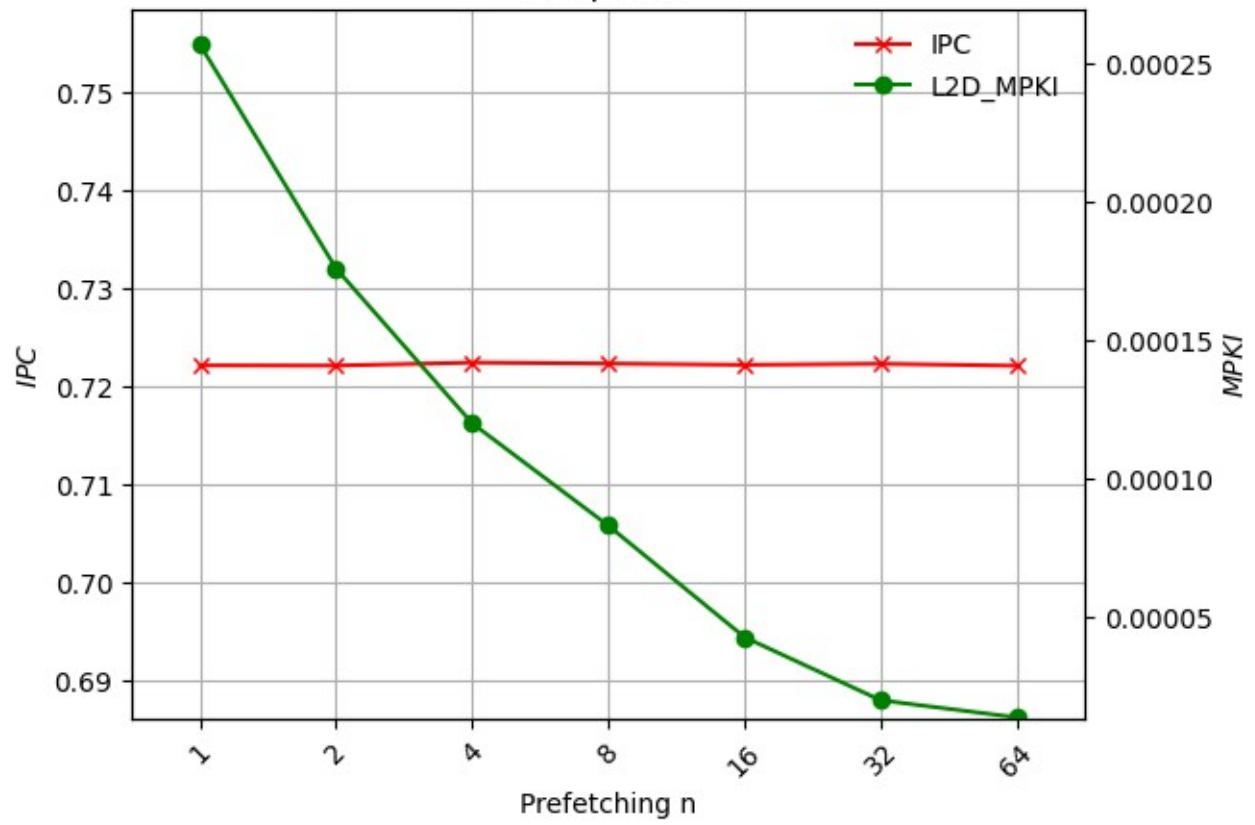


IPC vs MPKI  
freqmine





IPC vs MPKI  
swaptions



### Συμπεράσματα για Prefetching

- Στο blackscholes, bodytrack, fluidanimate, rtview και streamcluster η αύξηση των N blocks του prefetching φαίνεται να αυξάνει και το IPC, ωστόσο μετά από ένα σημείο, αυτή η αύξηση παύει να ισχύει και σχεδόν σταθεροποιείται το IPC.
- Στο canneal, η αύξηση του N δεν επιφέρει κάποιο σταθερό αποτέλεσμα στο IPC, αυξάνοντας και μειώνοντας το άλλες φορές και με τυχαίο τρόπο. Καταλαβαίνουμε, επομένως, πως τα δεδομένα, ή έστω μέρος αυτών, δεν χρησιμοποιούνται ή ανακτώνται με κάποιο γραμμικό τρόπο
- Στο freqmine, η αύξηση του N μειώνει το IPC, κάτι που σημαίνει πως το prefetching δεν βοηθάει καθόλου, αντιθέτως αντικαταστεί χρήσιμα προηγούμενα blocks, τα οποία πιθανότατα να οδηγήσουν σε περαιτέρω misses.
- Γενικότερα, βλέπουμε ότι στην πλειοψηφία η αύξηση του αριθμού N των prefetching blocks βοηθάει τα benchmarks, με λίγες εξαιρέσεις και ιδιορρυθμίες.

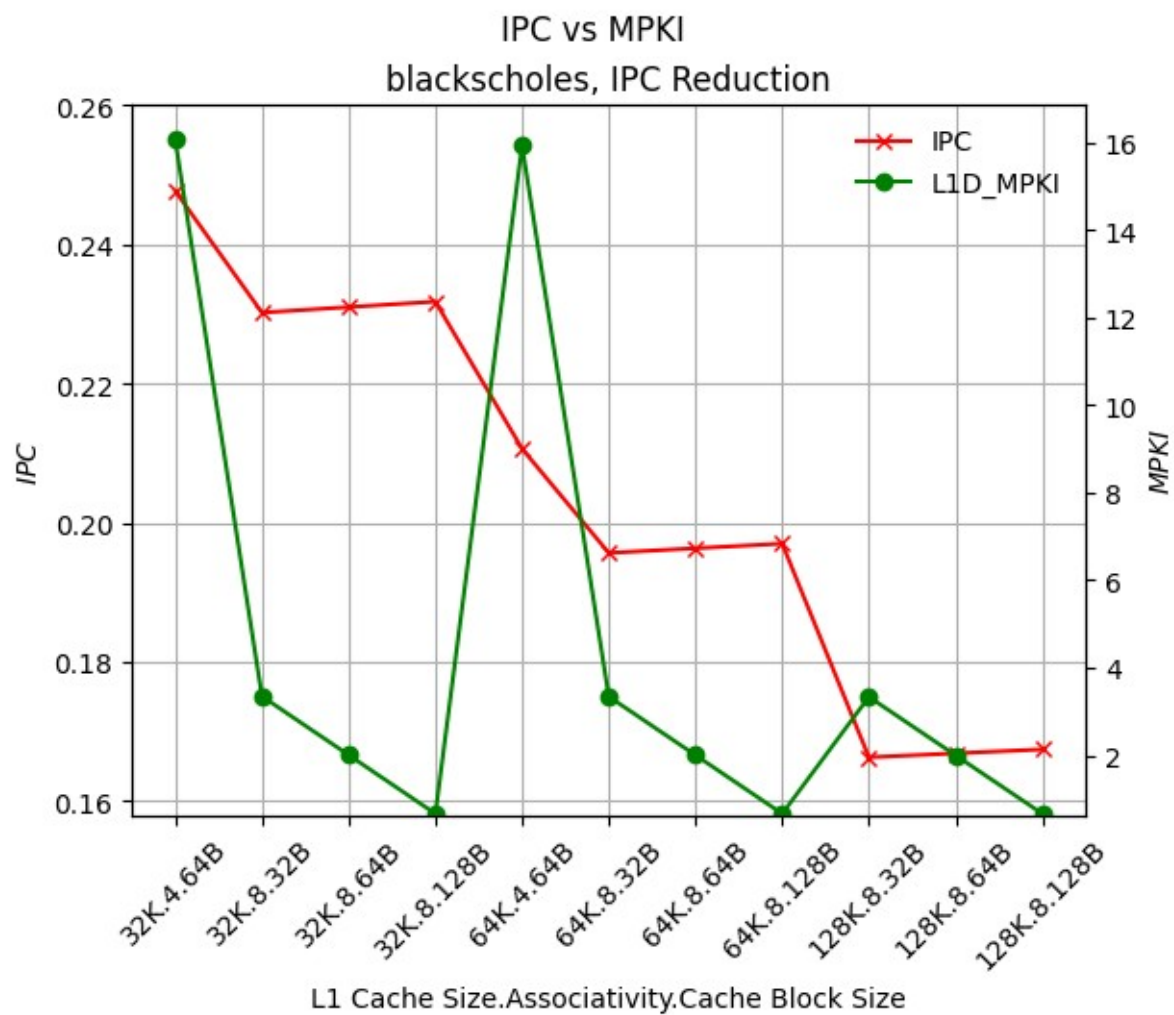


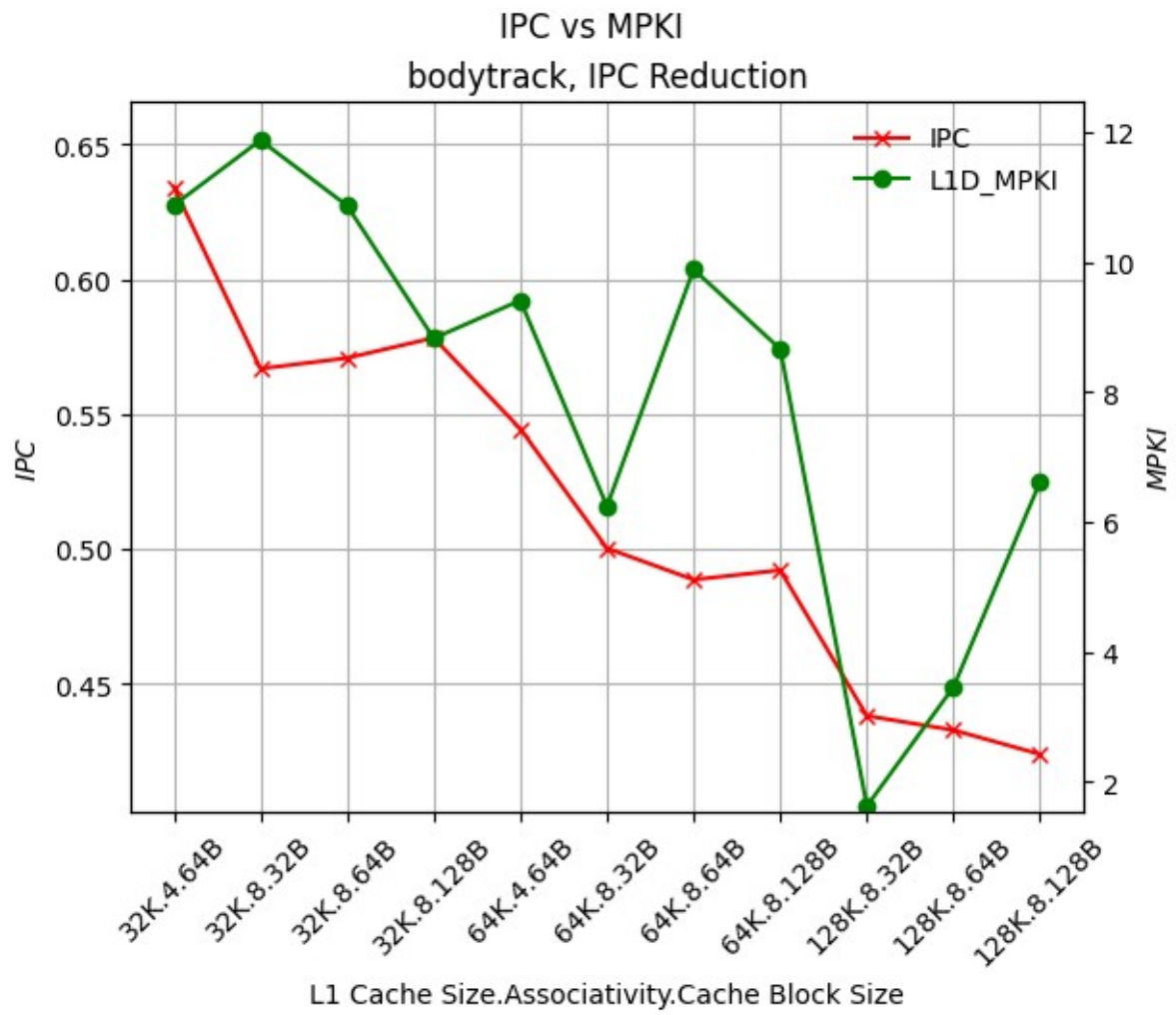
## 2ο ζητούμενο

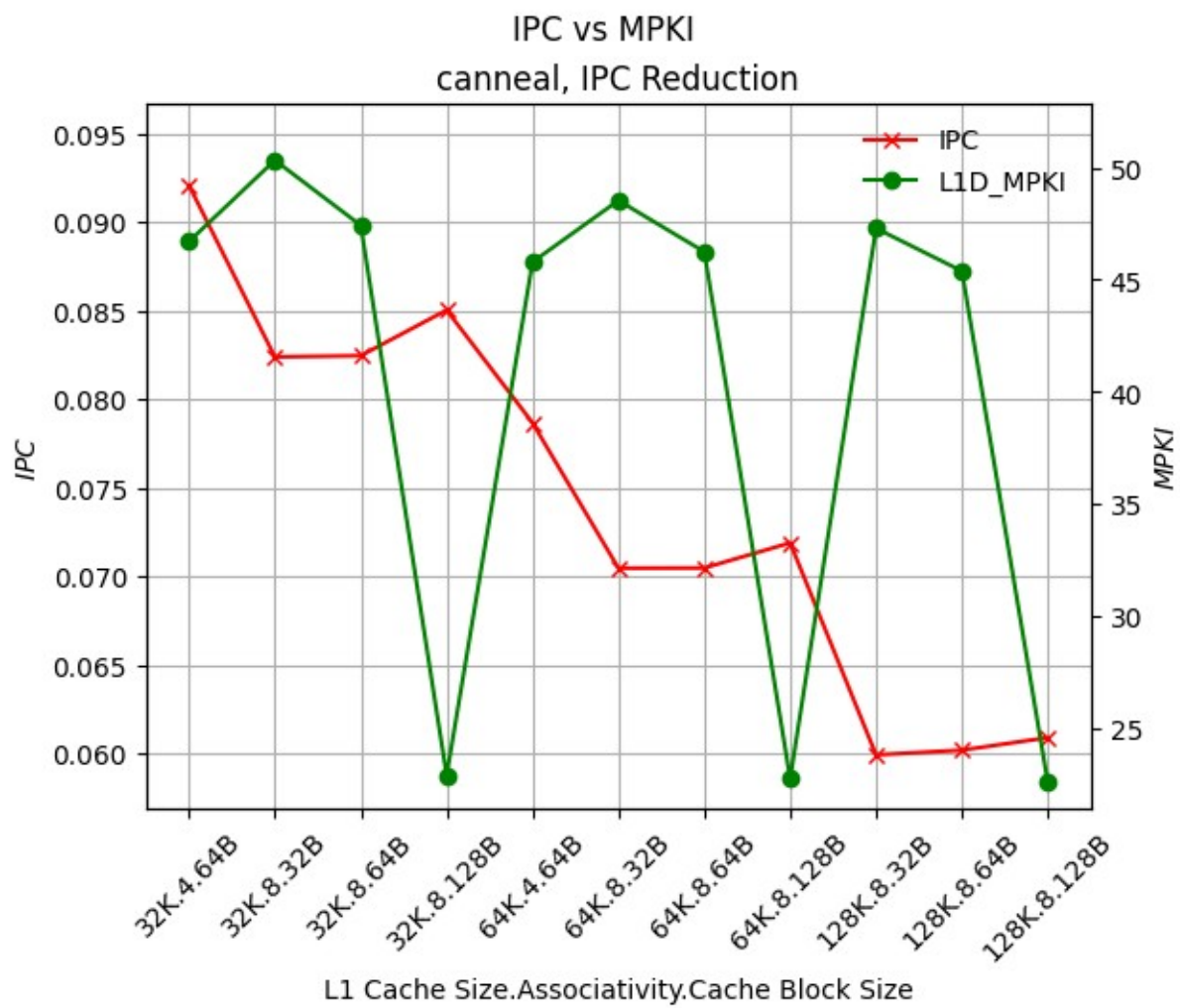
---

### L1 Cache

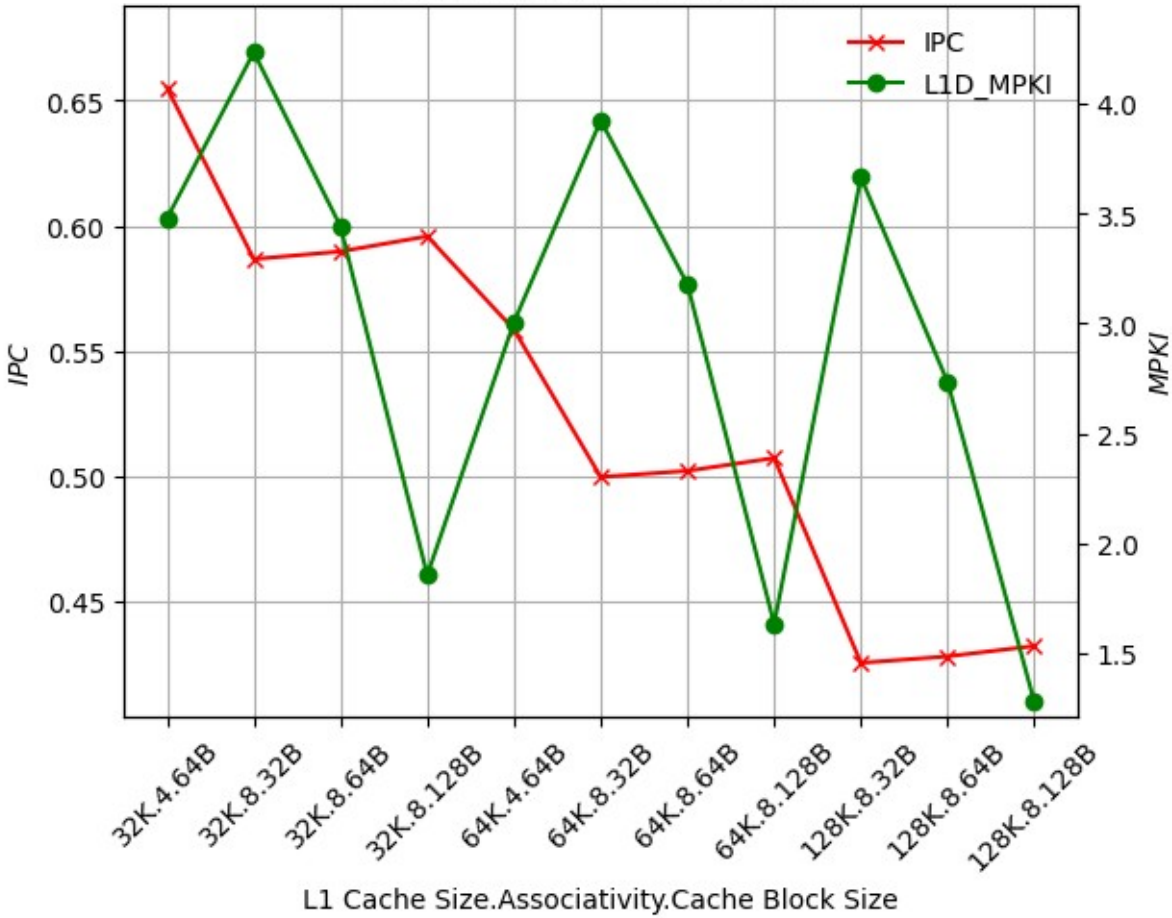
Για αυτό το πείραμα θεωρούμε πως κάθε διπλασιασμός του associativity ή του size προκαλεί αύξηση του κύκλου ρολογιού κατά 10% ή 15% αντίστοιχα. Ακολουθούν τα νέα γραφήματα που προέκυψαν με αρχικό σημείο αναφοράς την πρώτη προσομοίωση, ένα για κάθε benchmark.

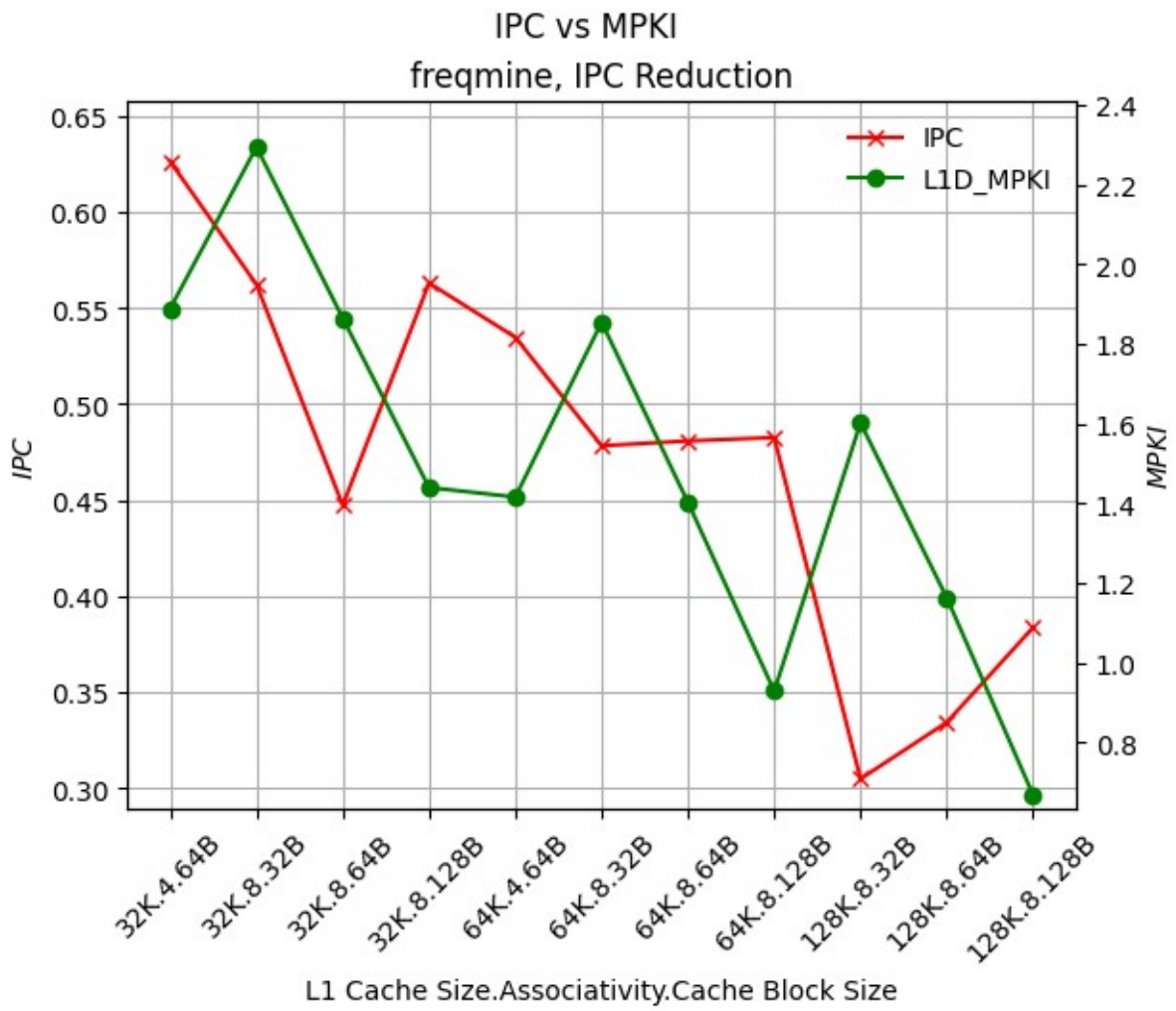


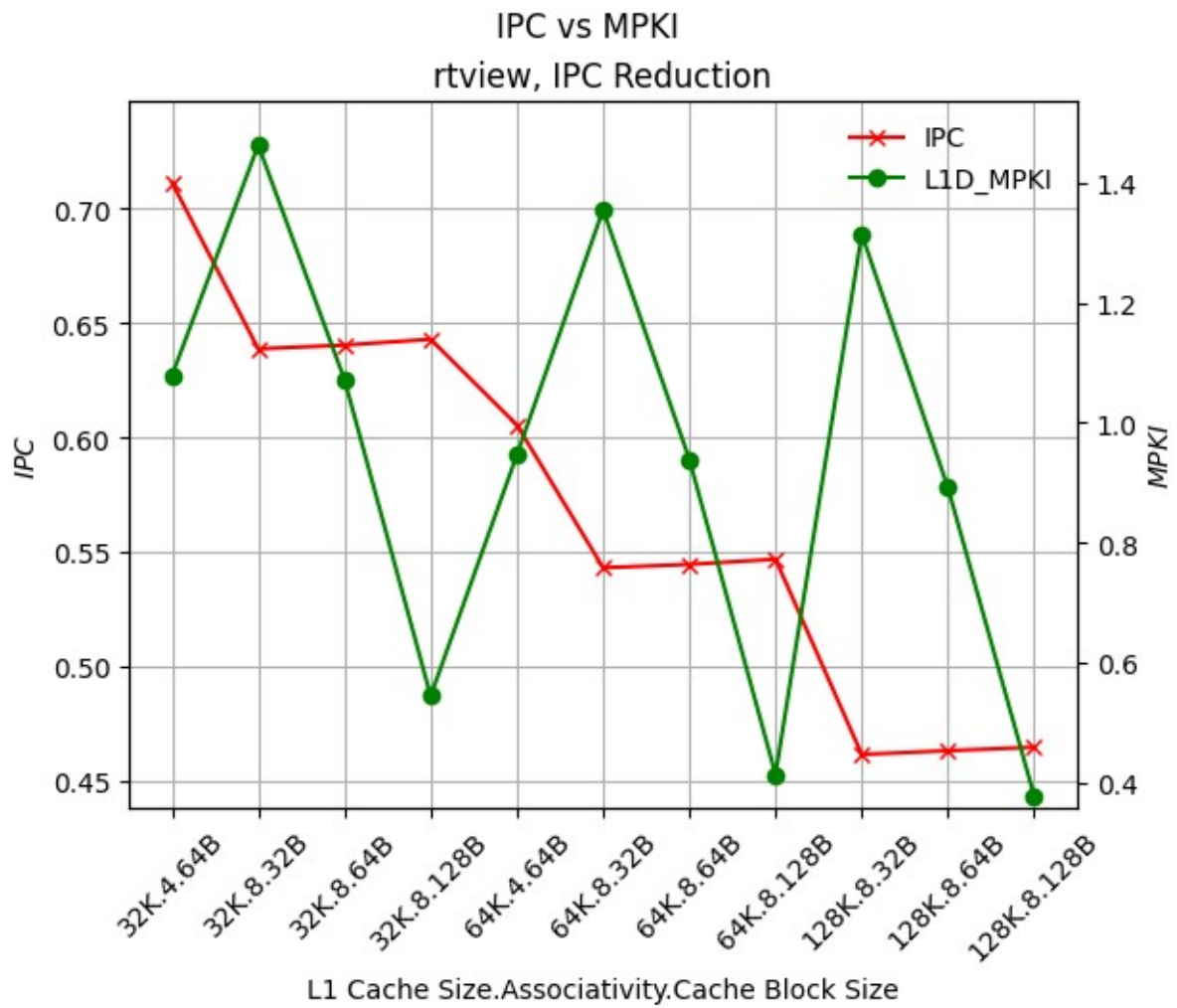


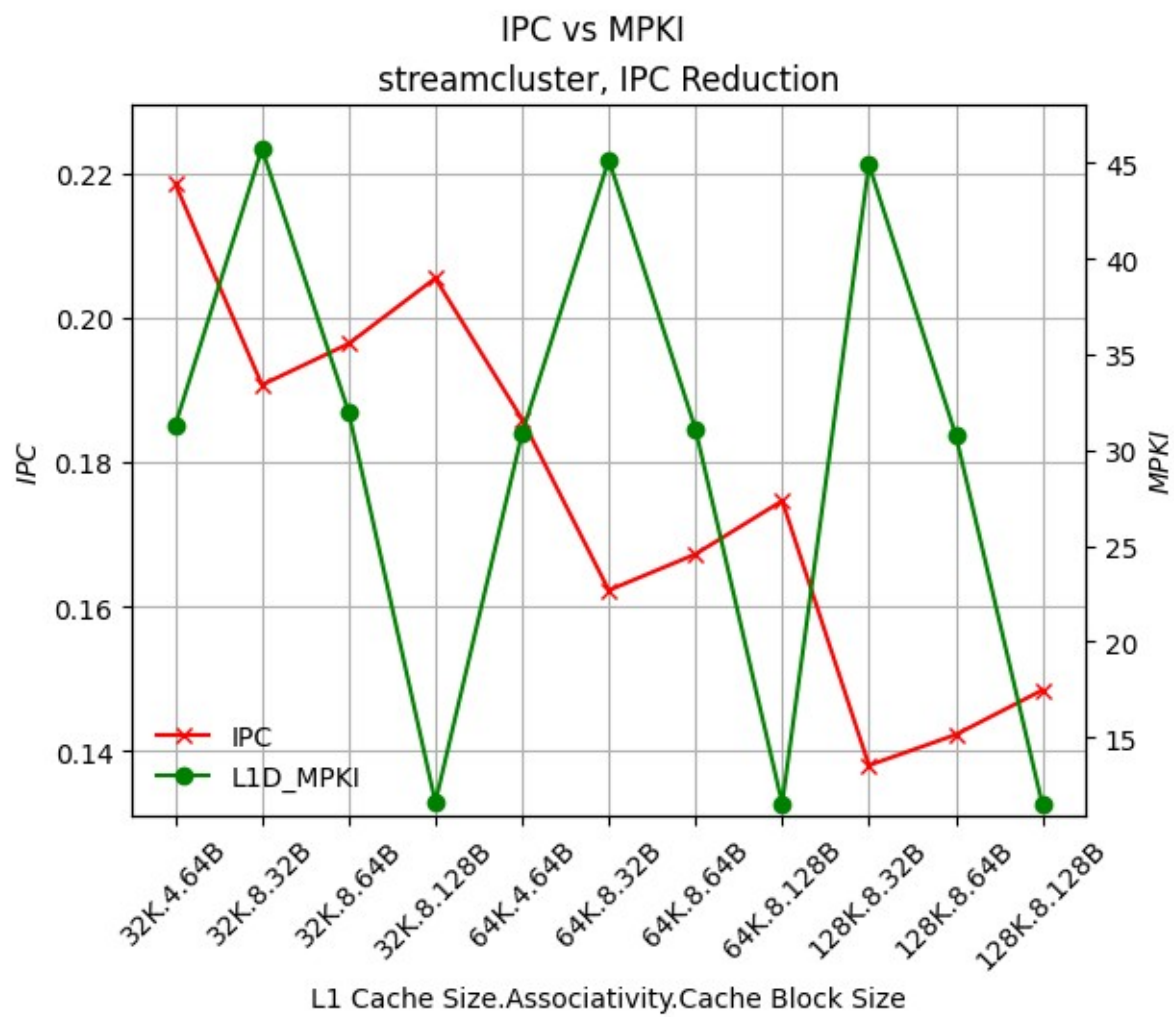


## fluidanimate, IPC Reduction

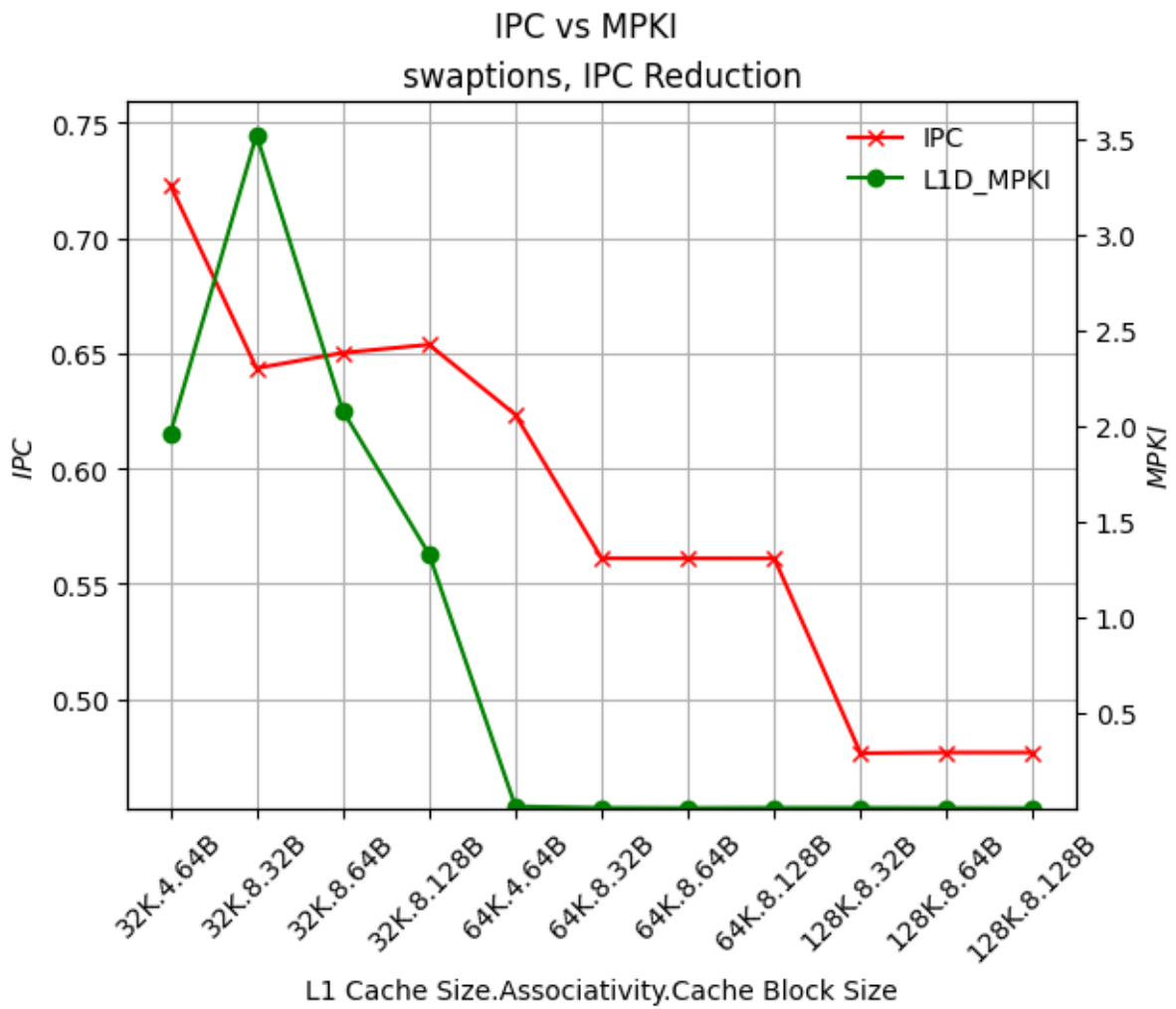












## 2ο ζητούμενο

---

### Συμπεράσματα για L1 cache

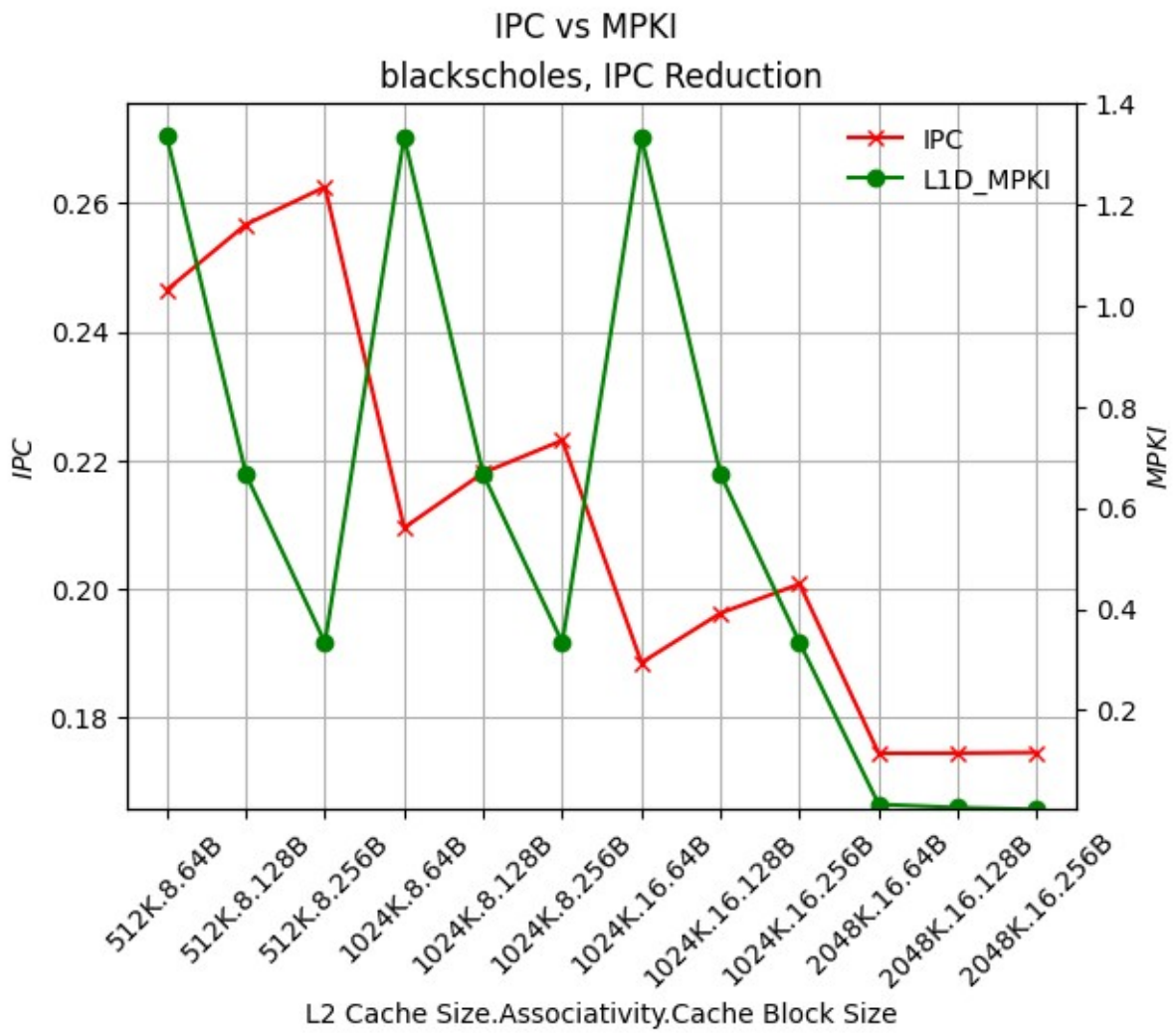
- Πλέον, δεν ισχύει η αντιστρόφως ανάλογη σχέση των IPC και MPKI.
- Παρατηρούμε μία γενικότερη μείωση του IPC στην πλειοψηφία των benchmarks, όσο αυξάνονται οι διάφοροι παράμετροι της L1 cache. Αυτό συμβαίνει προφανώς, διότι τα οφέλη των αυξήσεων αυτών δεν αντισταθμίζουν το penalty που υπάρχει λόγω της αύξησης του κύκλου ρολογιού.

## 2ο ζητούμενο

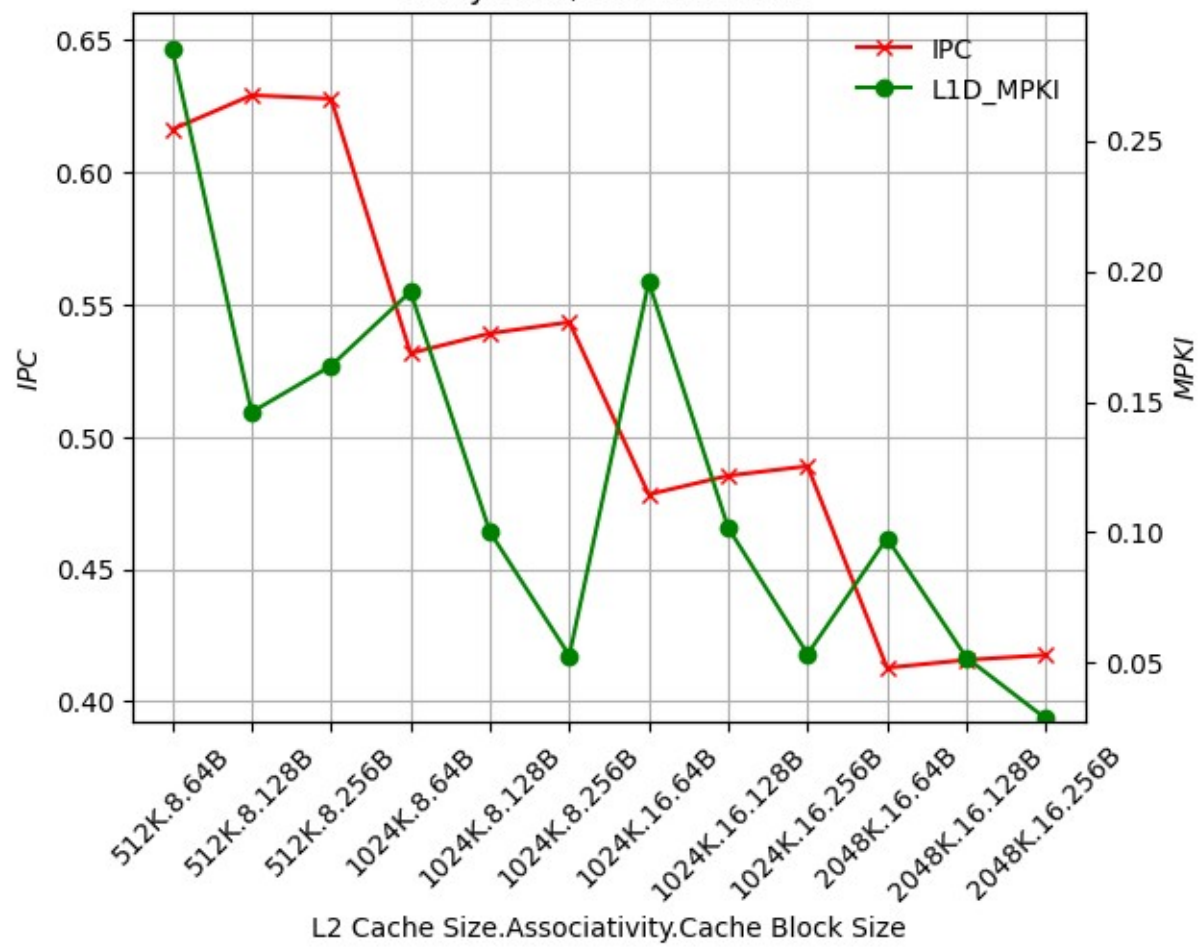
---

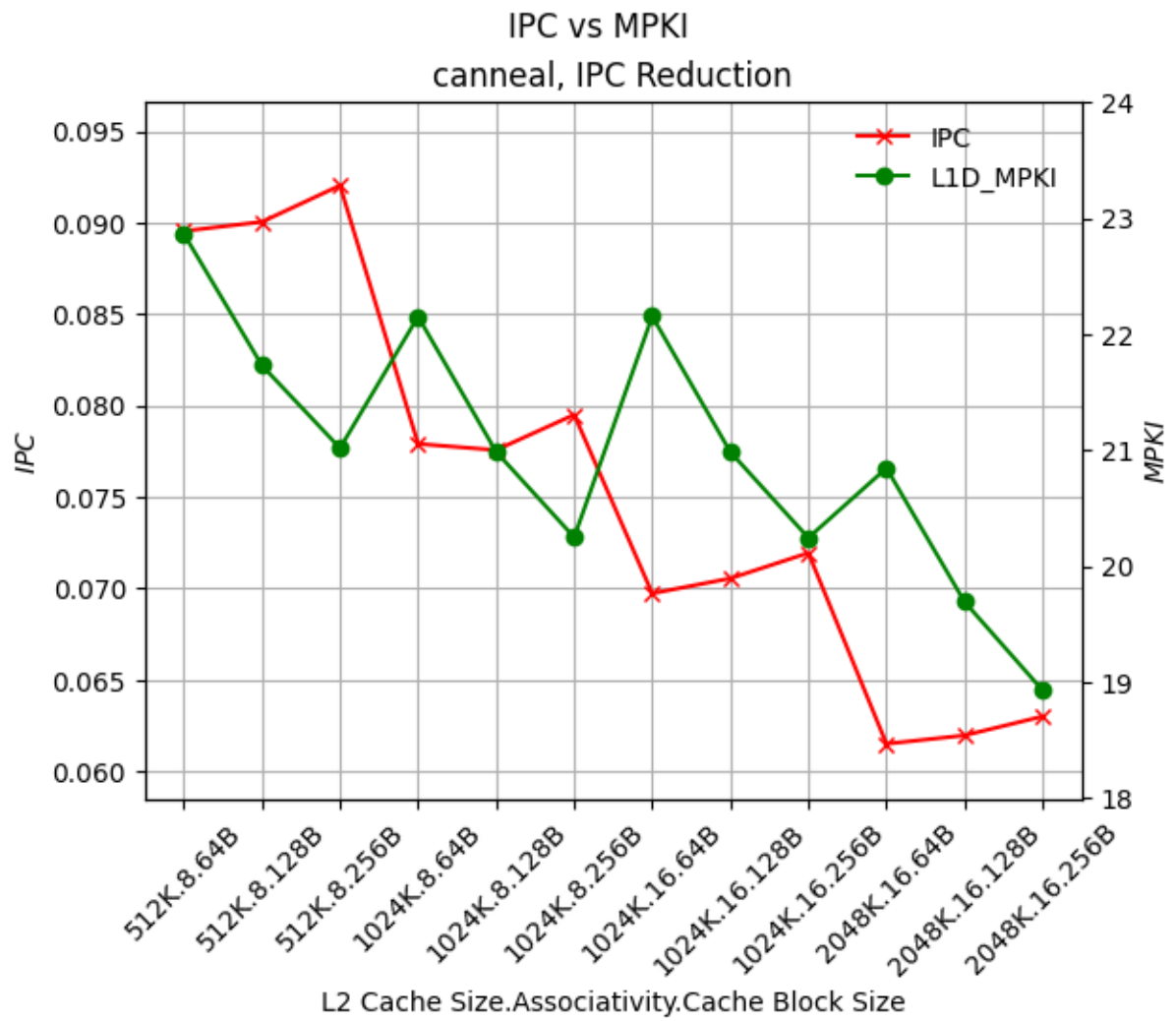
### L2 Cache

Για αυτό το πείραμα θεωρούμε πως κάθε διπλασιασμός του associativity ή του size προκαλεί αύξηση του κύκλου ρολογιού κατά 10% ή 15% αντίστοιχα. Ακολουθούν τα νέα γραφήματα που προέκυψαν με αρχικό σημείο αναφοράς την πρώτη προσομοίωση, ένα για κάθε benchmark.

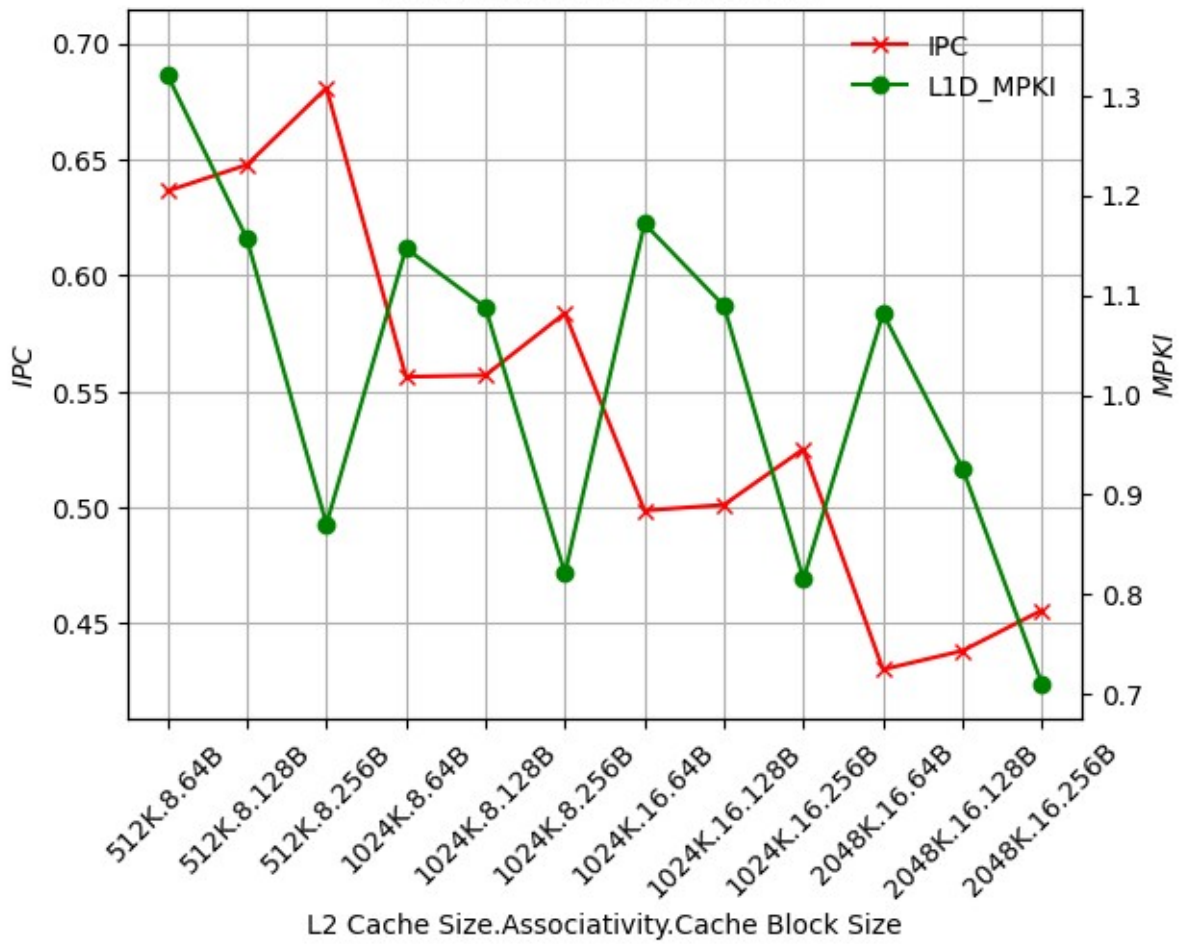


IPC vs MPKI  
bodytrack, IPC Reduction

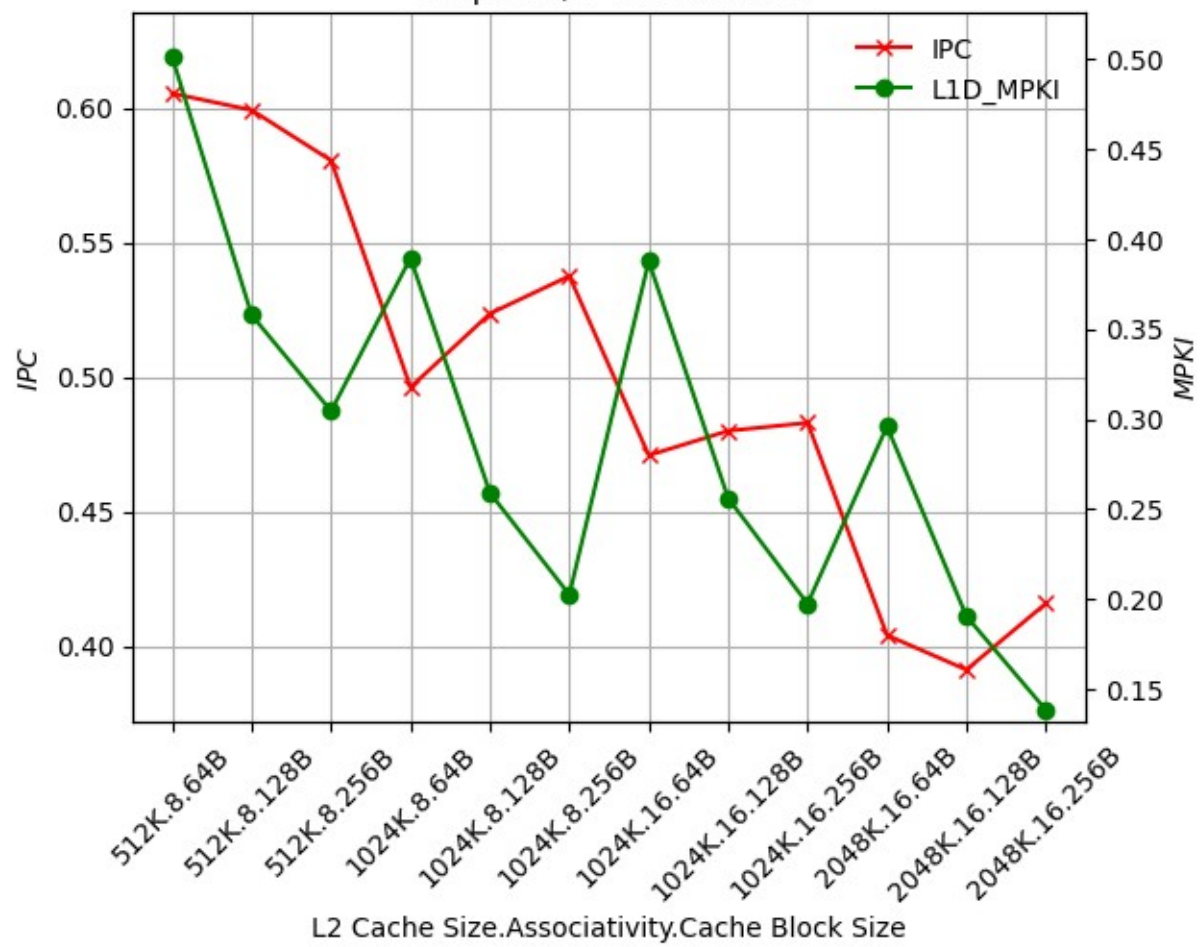




IPC vs MPKI  
fluidanimate, IPC Reduction

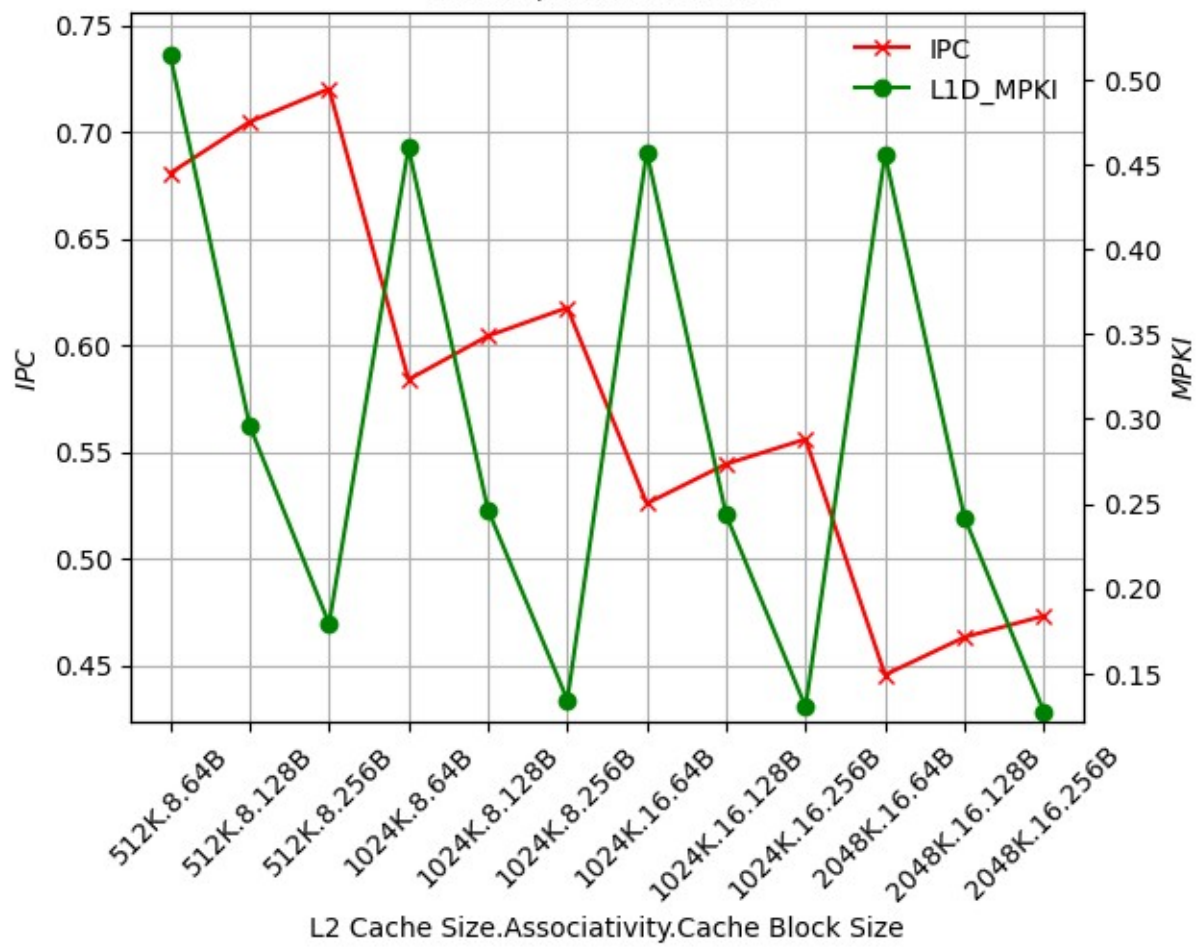


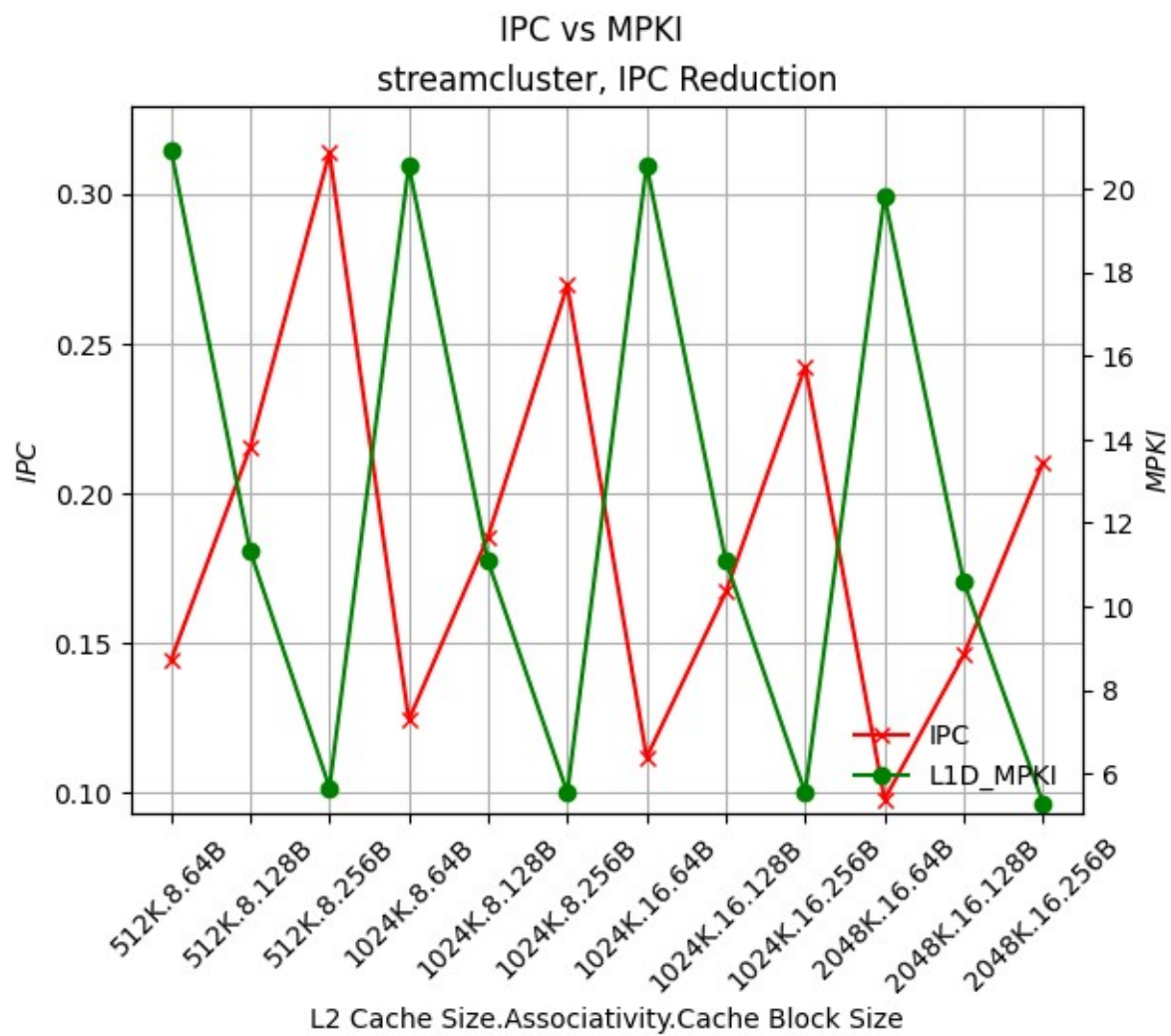
IPC vs MPKI  
freqmine, IPC Reduction

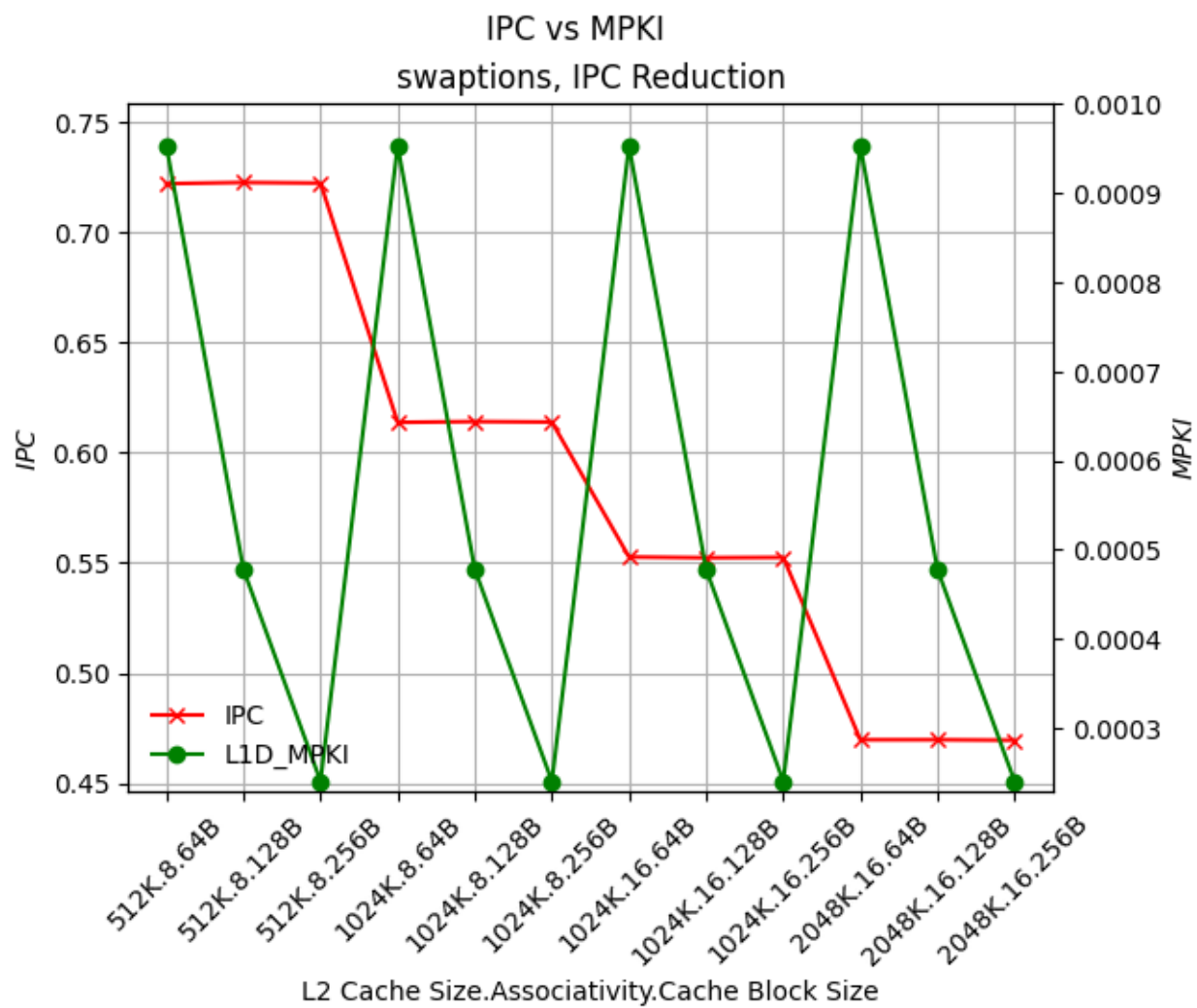




IPC vs MPKI  
rtview, IPC Reduction







## 2ο ζητούμενο

---

### Συμπεράσματα για L2 cache

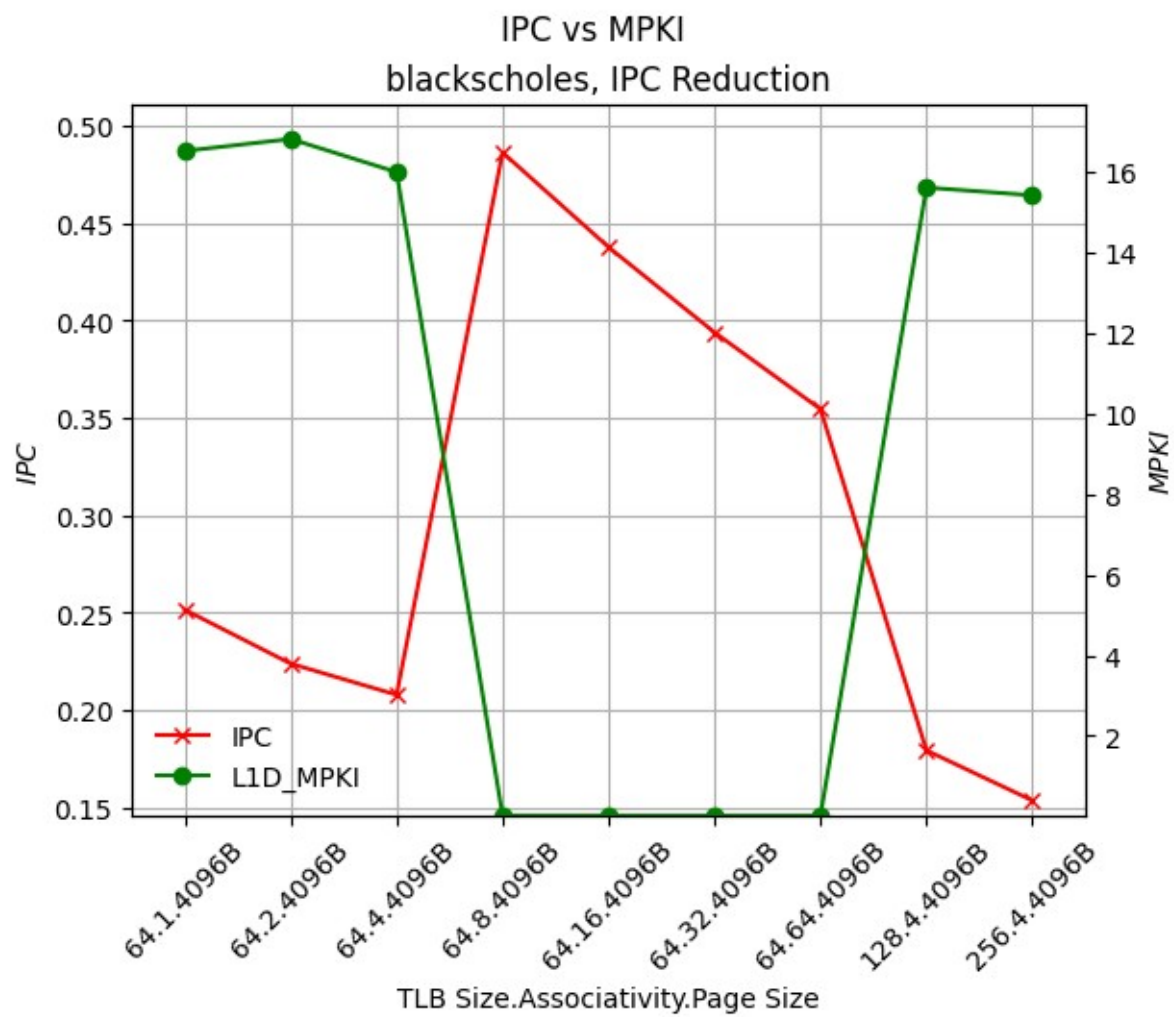
- Πλέον, δεν ισχύει η αντιστρόφως ανάλογη σχέση των IPC και MPKI.
- Παρατηρούμε μία γενικότερη μείωση του IPC στην πλειοψηφία των benchmarks, όσο αυξάνονται οι διάφοροι παράμετροι της L2σ cache. Αυτό συμβαίνει προφανώς, διότι τα οφέλη των αυξήσεων αυτών δεν αντισταθμίζουν το penalty που υπάρχει λόγω της αύξησης του κύκλου ρολογιού.

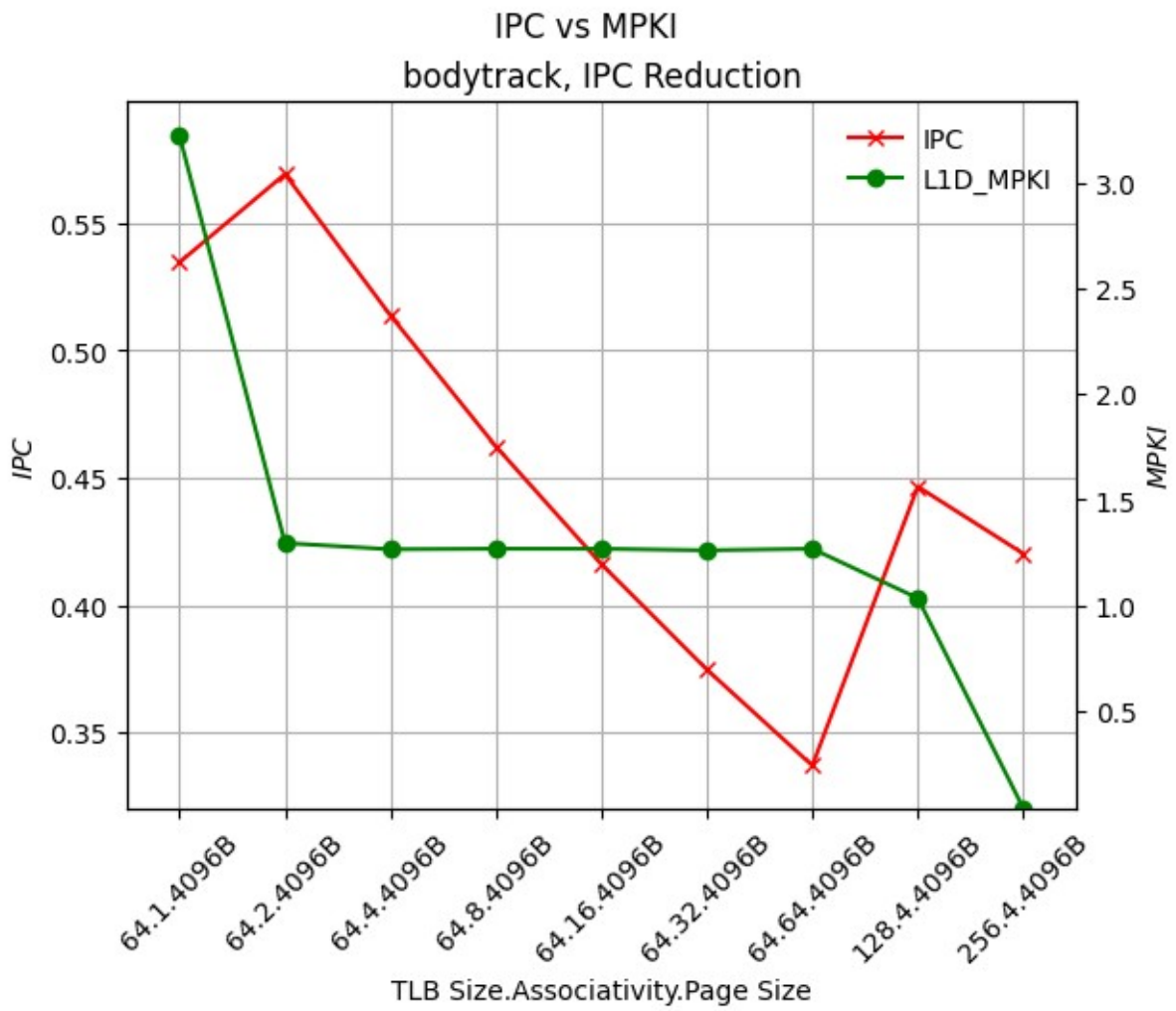
## 2ο ζητούμενο

---

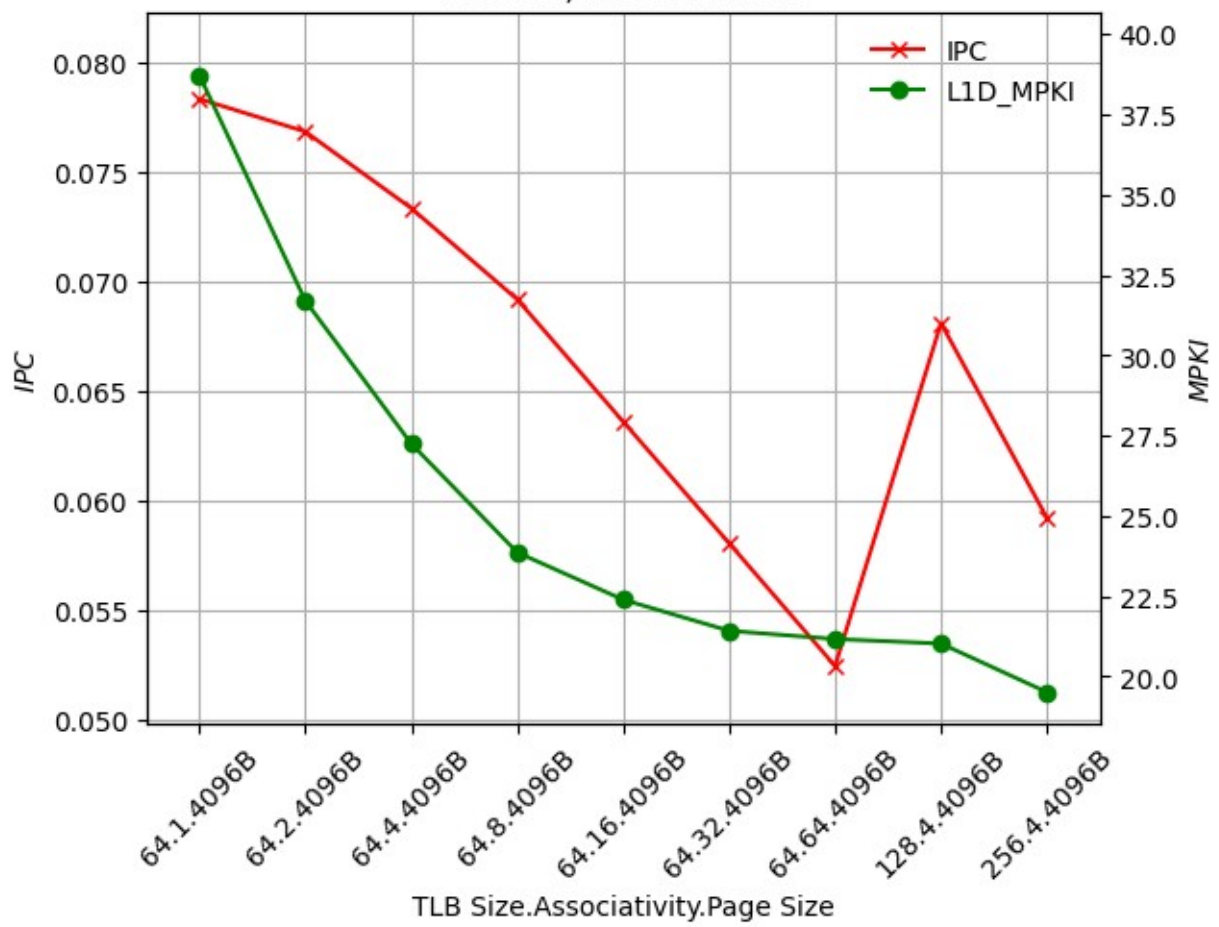
### TLB

Για αυτό το πείραμα θεωρούμε πως κάθε διπλασιασμός του associativity ή του size προκαλεί αύξηση του κύκλου ρολογιού κατά 10% ή 15% αντίστοιχα. Ακολουθούν τα νέα γραφήματα που προέκυψαν με αρχικό σημείο αναφοράς την πρώτη προσομοίωση, ένα για κάθε benchmark.

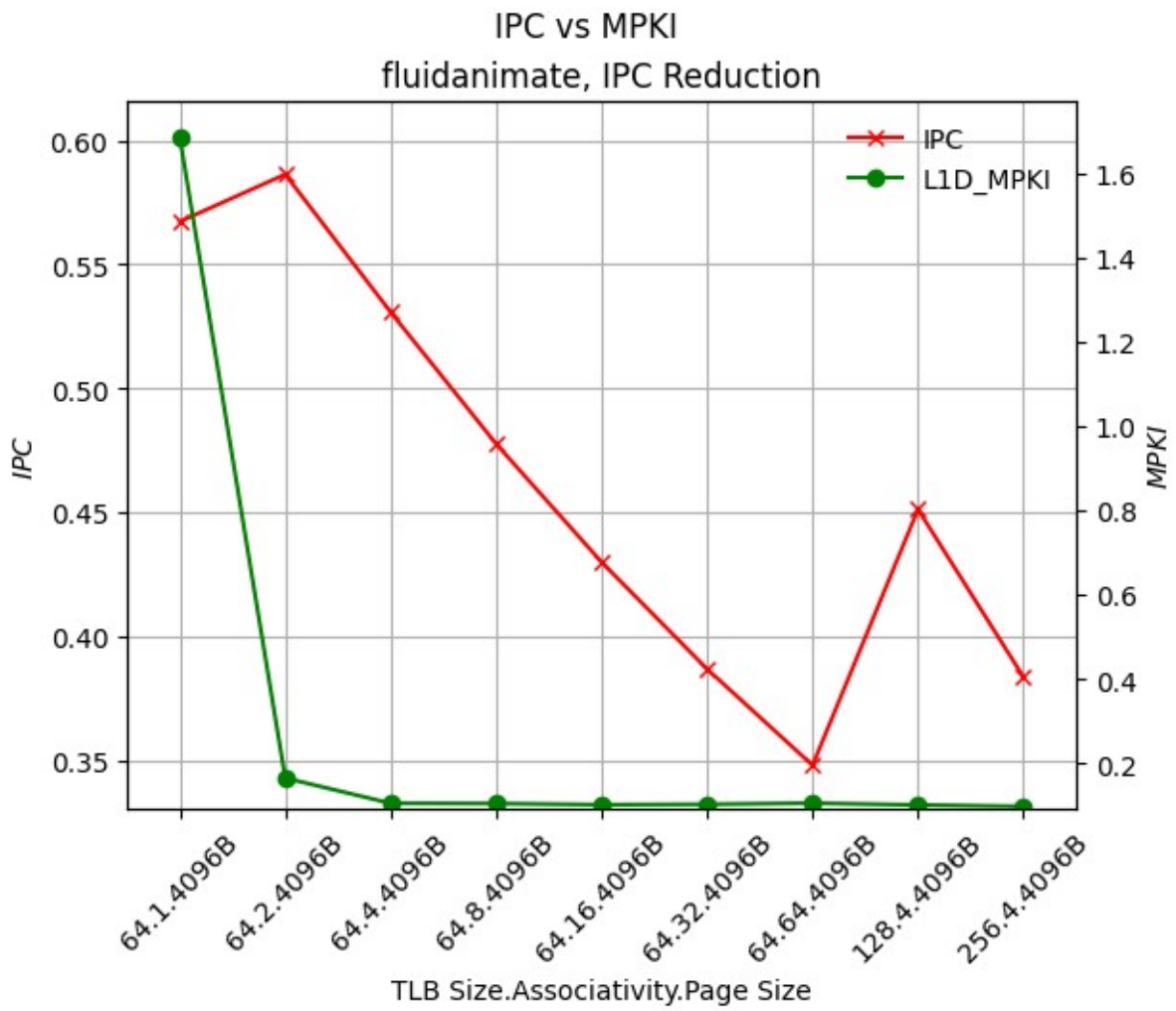


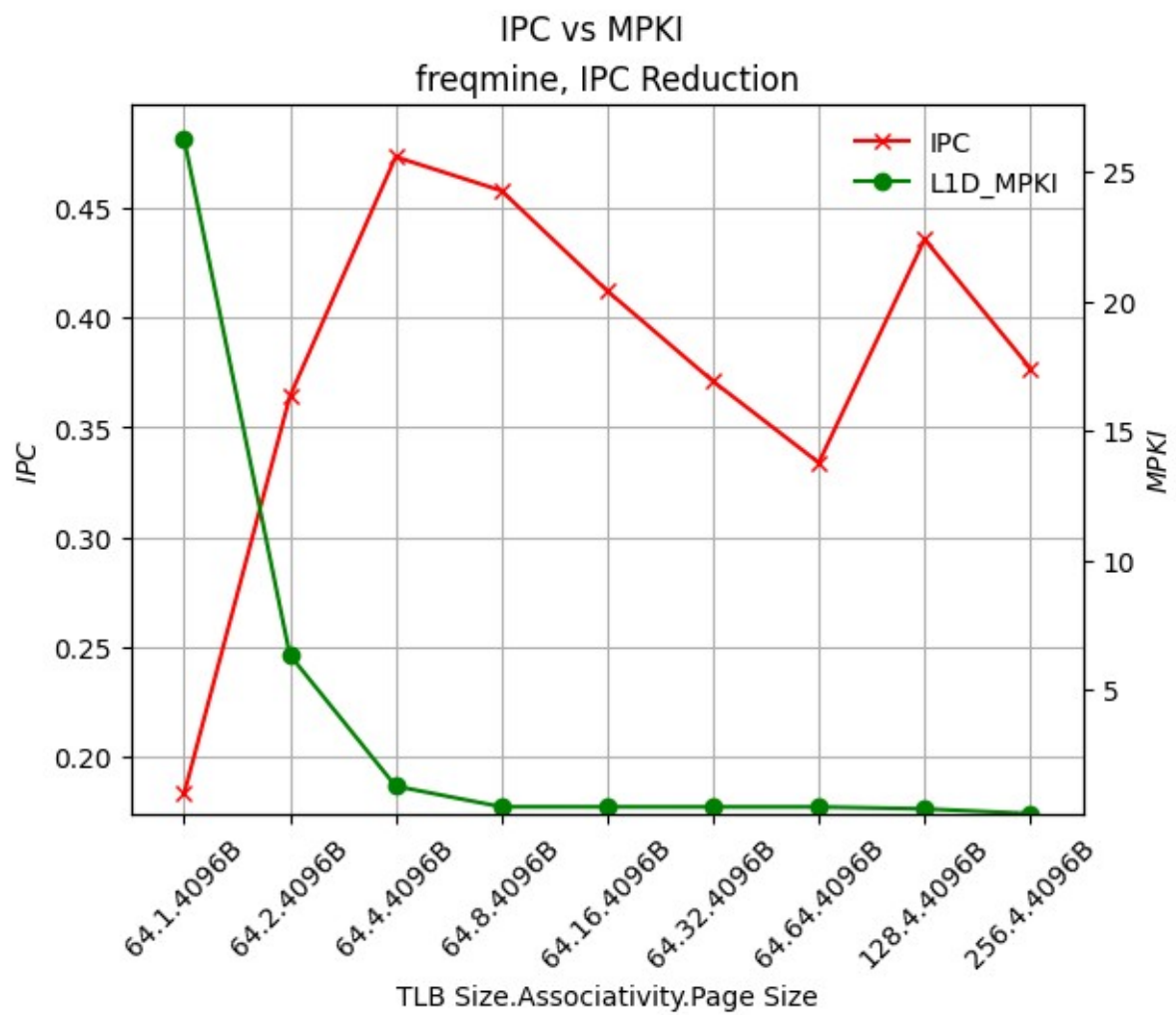


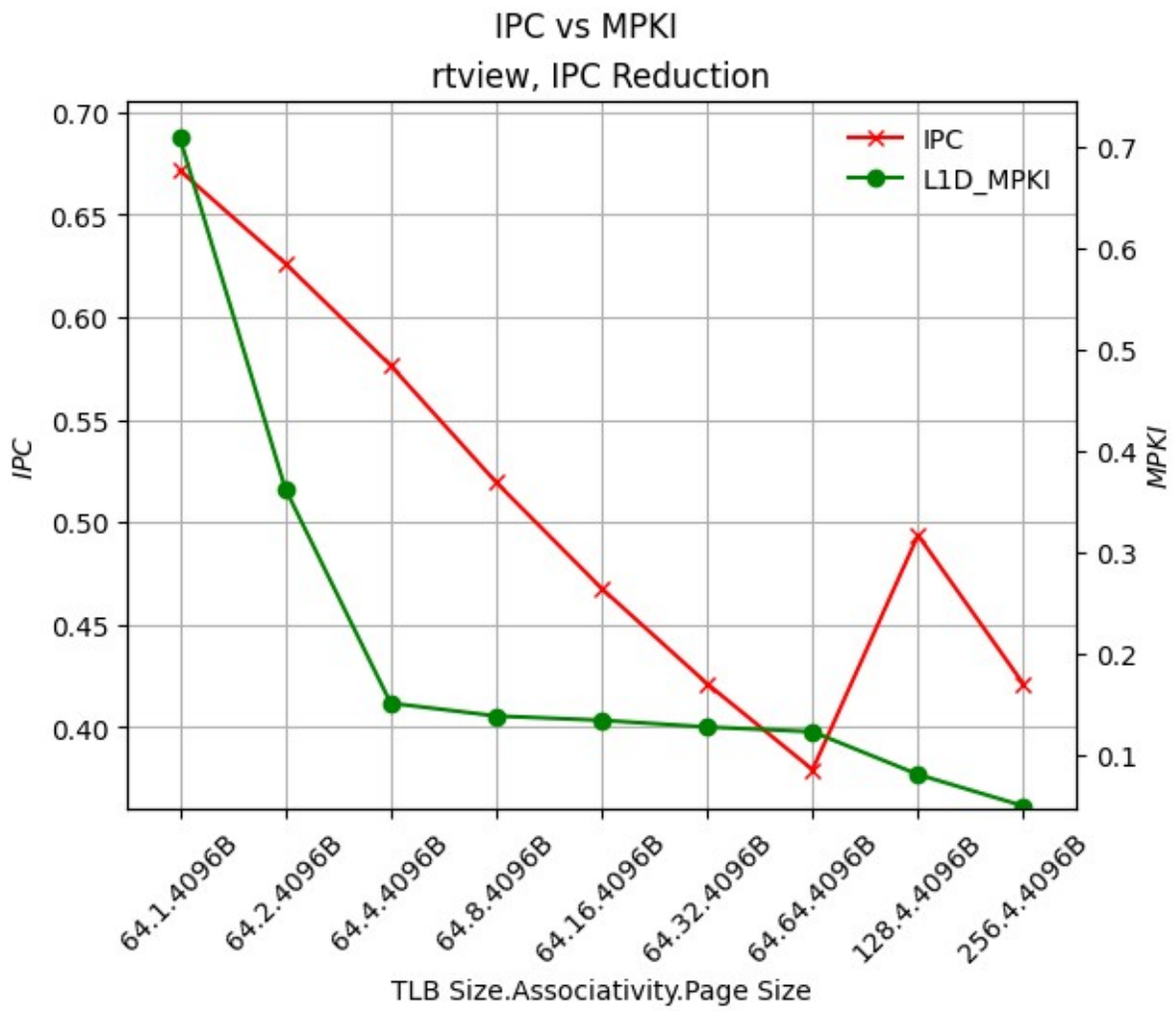
IPC vs MPKI  
canneal, IPC Reduction

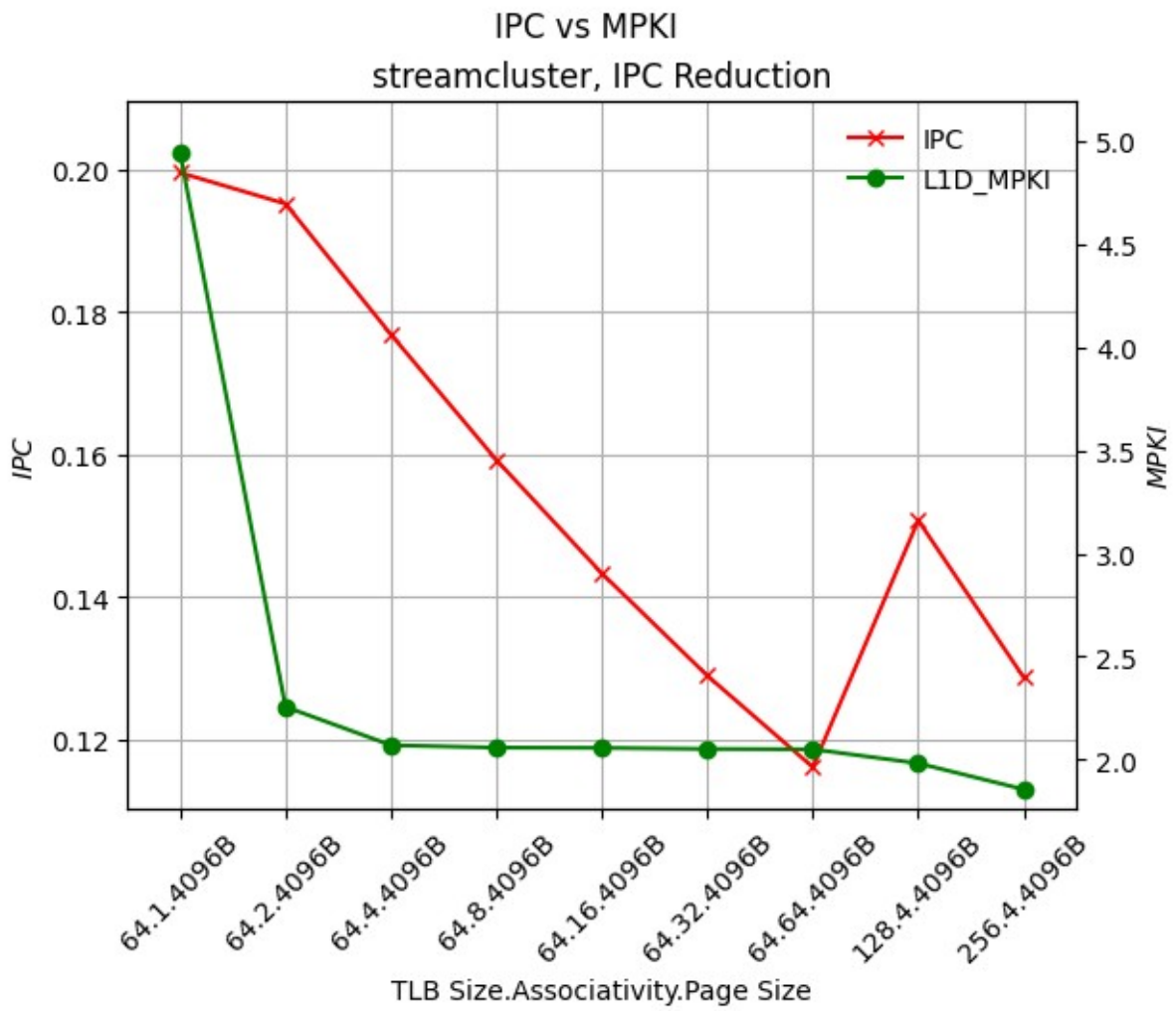


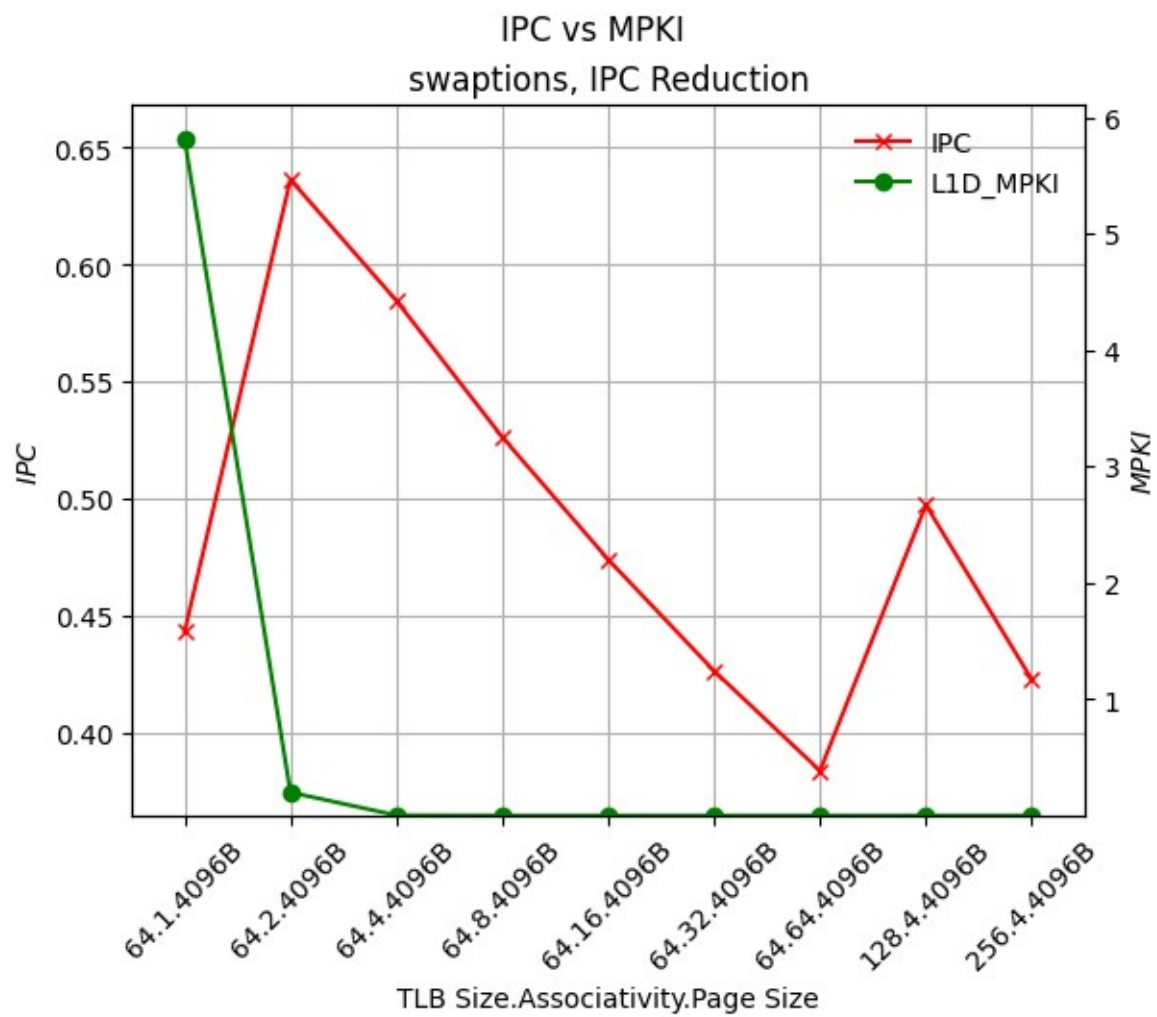












## 2ο ζητούμενο

---

### Συμπεράσματα για TLB

- Πλέον, δεν ισχύει η αντιστρόφως ανάλογη σχέση των IPC και ΜΡΚΙ.
- Υπάρχει μία μεικτή συμπεριφορά για το TLB, καθώς άλλες φορές αυξάνεται το IPC και άλλες μειώνεται.
- Αρχικά φαίνεται να υπάρχει μία μείωση για μέγεθος TLB 64, ωστόσο στο τέλος, αυξάνεται και πάλι το IPC λόγω αύξησης του μεγέθους TLB.