

ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

1^η ΑΣΚΗΣΗ

Ακαδημαϊκό Έτος 2019-2020

Εξάμηνο 8^ο

Χρόνης Σάκος

03116168

7. Πειραματική Αξιολόγηση

Στα πλαίσια της εργασίας αυτής, θα διερευνηθεί η επίδραση των βασικότερων παραμέτρων ιεραρχίας κρυφής μνήμης στην επίδοση της εφαρμογής.

7.1 L1 cache

Για τα διάφορα benchmarks και όλες τις περιπτώσεις της L1 cache που εξετάζονται στο πείραμα αυτό, οι παράμετροι της L2 cache και του TLB θα διατηρηθούν σταθερές και συγκεκριμένα ίσες με:

L2_size = 1024 KB

L2_associativity = 8

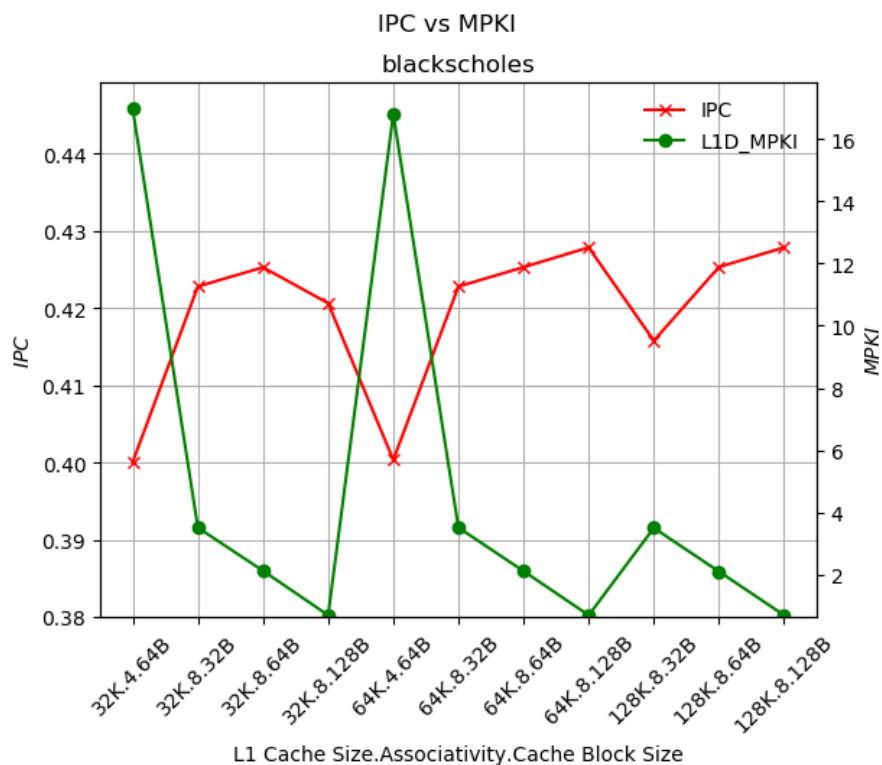
L2_block_size = 128 B

TLB_size = 64 entr.

TLB_associativity = 4

TLB_page size = 4096 B

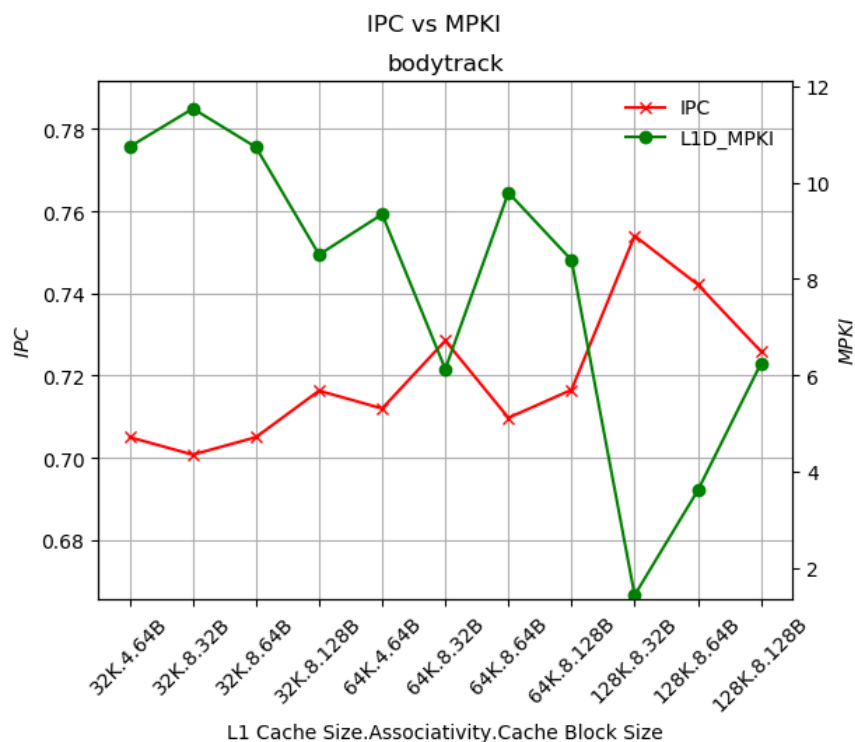
1) Blackscholes



Στο L1 blackscholes παρατηρούμε πως οι δύο βέλτιστοι συνδυασμοί είναι οι (64,8,128) και (128,8,128), με τον συνδυασμό (32,8,128) να ακολουθεί με ελάχιστα μικρότερη τιμή IPC. Πράγματι, οι δύο πρώτοι συνδυασμοί εμφανίζουν την ίδια απόδοση παρ' όλη τη διαφορά στο μέγεθος της L1 cache. Αυτό που φαίνεται να διαδραματίζει μεγαλύτερο ρόλο, σύμφωνα και με τους υπόλοιπους συνδυασμούς, είναι το associativity, καθώς για την τιμή 4 έχουμε χειρότερα αποτελέσματα. Η μεγάλη εξάρτηση από το associativity πιθανώς να δηλώνει πως υπάρχουν πολλά conflict misses. Επίσης, η αύξηση του cache size δεν φαίνεται να έχει μεγάλη επίδραση στα αποτελέσματα, ενώ η αύξηση του block size μειώνει τα misses και αυξάνει τα instructions per cycle.

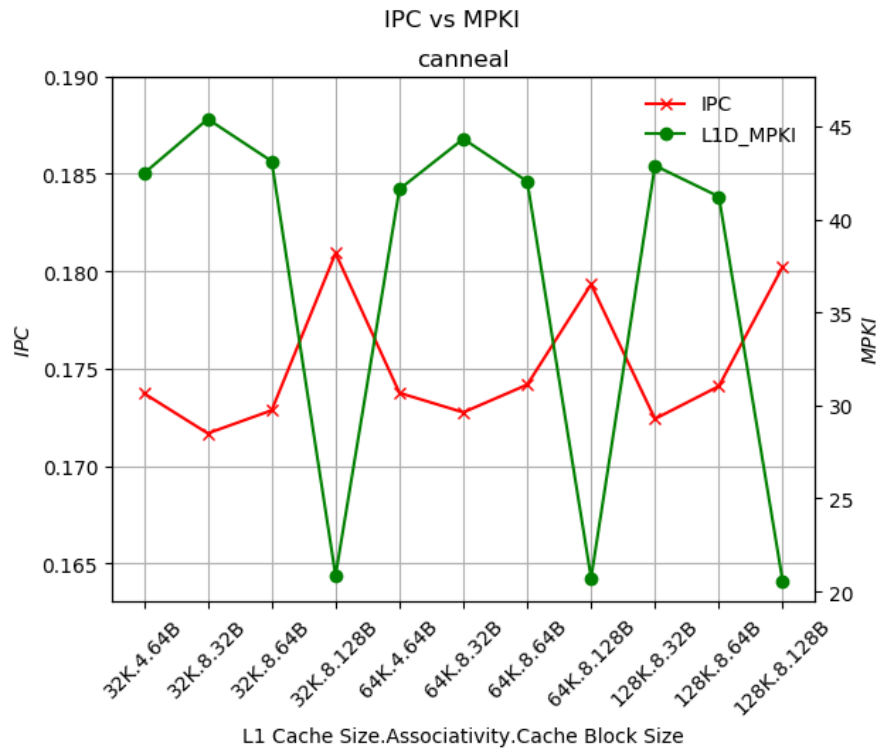
2) Bodytrack

Στο L1 bodytrack φαίνεται πως ο βέλτιστος συνδυασμός είναι ο (128,8,32) καθώς σε αυτόν παρατηρούμε την ελάχιστη τιμή MPKI και την μέγιστη τιμή σε IPC. Αναμφίβολα, προκύπτουν αρκετά скаμπανεβάσματα της απόδοσης για τις διάφορες περιπτώσεις της L1 cache, παρόλα αυτά η αύξηση του block size φαίνεται να συνδράμει στην βελτίωσή της. Το associativity δεν παρουσιάζει κάποια αξιοσημείωτη μεταβολή στην απόδοση, ενώ η μεταβολή του cache size σε 128K επέφερε θεαματική βελτίωση. Η απότομη πτώση στα misses για τον συνδυασμό (128,8,32) οφείλεται πιθανώς στην μείωση των capacity misses λόγω περισσότερων διαθέσιμων blocks.



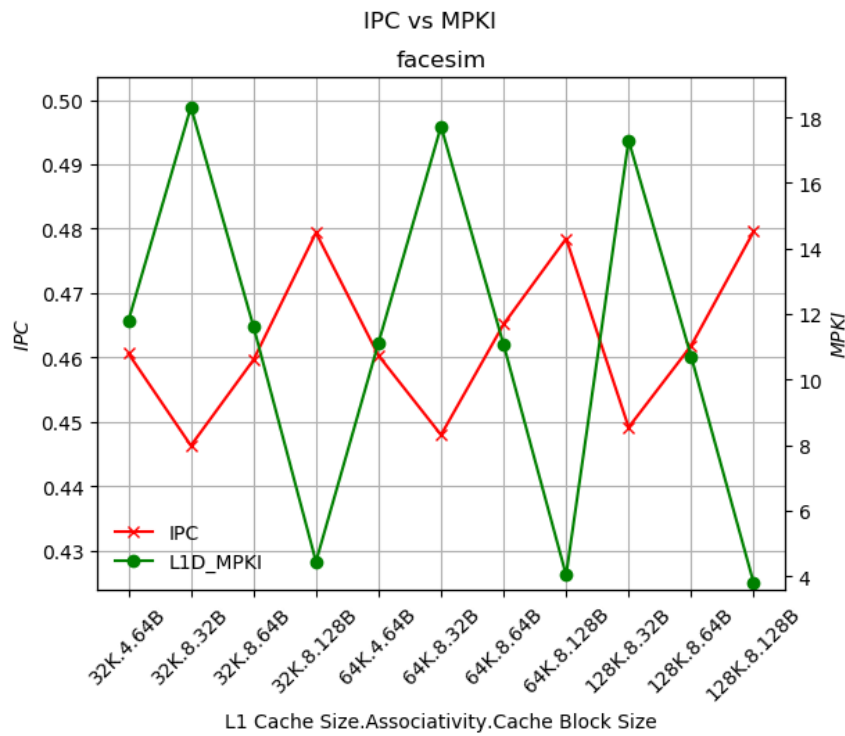
3) Canneal

Στο L1 canneal προκύπτουν τρεις βέλτιστοι συνδυασμοί, οι (32,8,128), (64,8,128) και (128,8,128). Πράγματι, παρατηρούμε πως για block size 128 B, ανεξαρτήτου cache size, έχουμε βέλτιστη απόδοση, κάτι που πιθανώς οφείλεται σε ισχυρή spatial locality. Επίσης, φαίνεται πως το cache size και το associativity δεν επιδρούν σημαντικά στα αποτελέσματα.



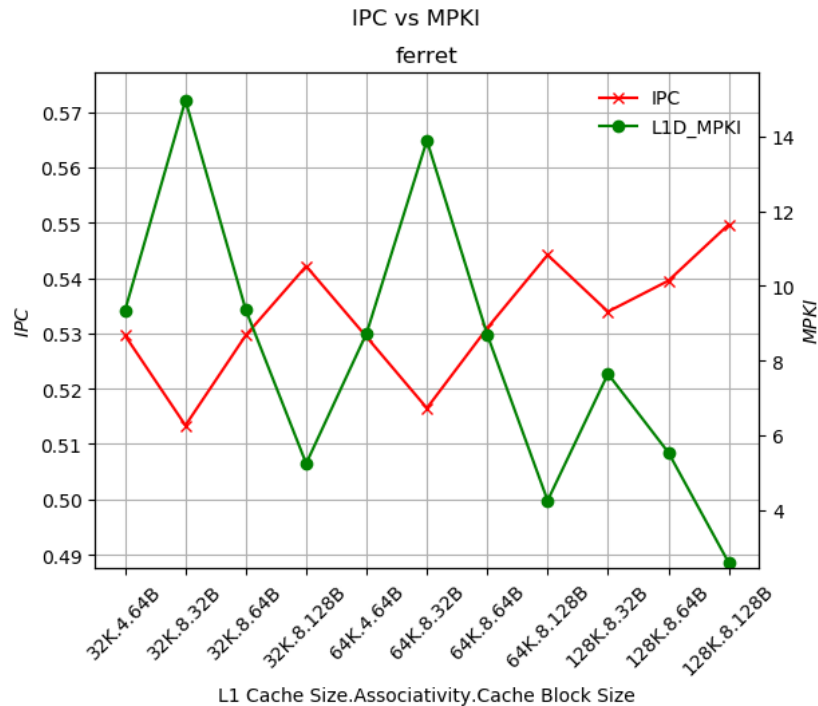
4) Facesim

Στο L1 facesim ως βέλτιστοι συνδυασμοί προκύπτουν οι (128,8,128), (64,8,128) και (32,8,128). Όπως και στο benchmark canneal, παρατηρούμε πως για block size 128 B, ανεξαρτήτου cache size, έχουμε βέλτιστη απόδοση. Επίσης, φαίνεται πως το cache size και το associativity έχουν αμελητέα επίδραση στα αποτελέσματα.



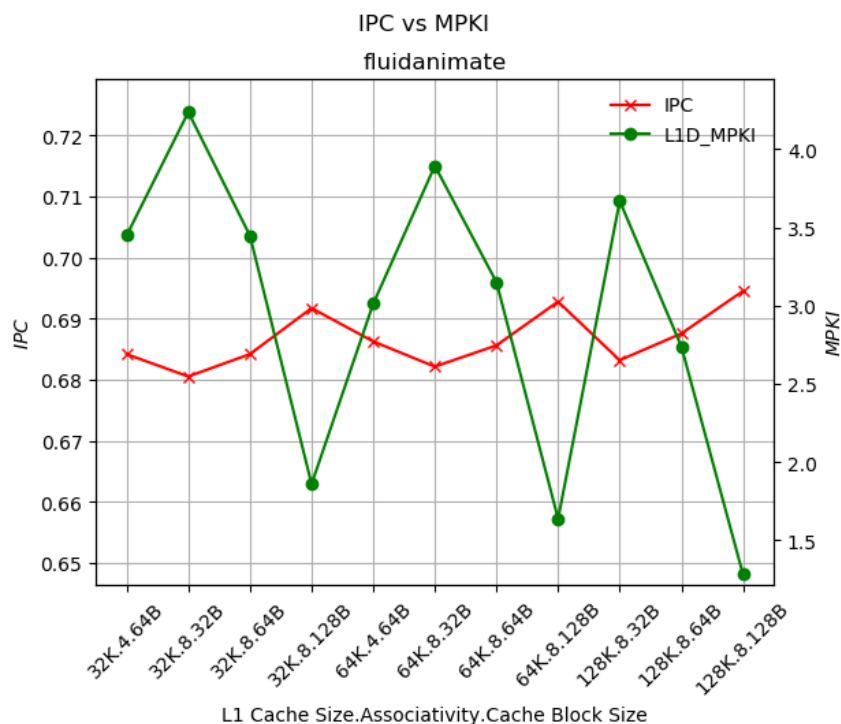
5) Ferret

Στο L1 ferret βέλτιστος συνδυασμός είναι αναμφίβολα ο (128,8,128). Συνολικά, παρατηρούμε ομοιότητες στα αποτελέσματα για τις διάφορες τιμές του cache size. Αυτό που έχει τη μέγιστη επίδραση στην απόδοση είναι εκ νέου το block size, καθώς η αύξηση του μειώνει σημαντικά τα misses και αυξάνει τα IPC.



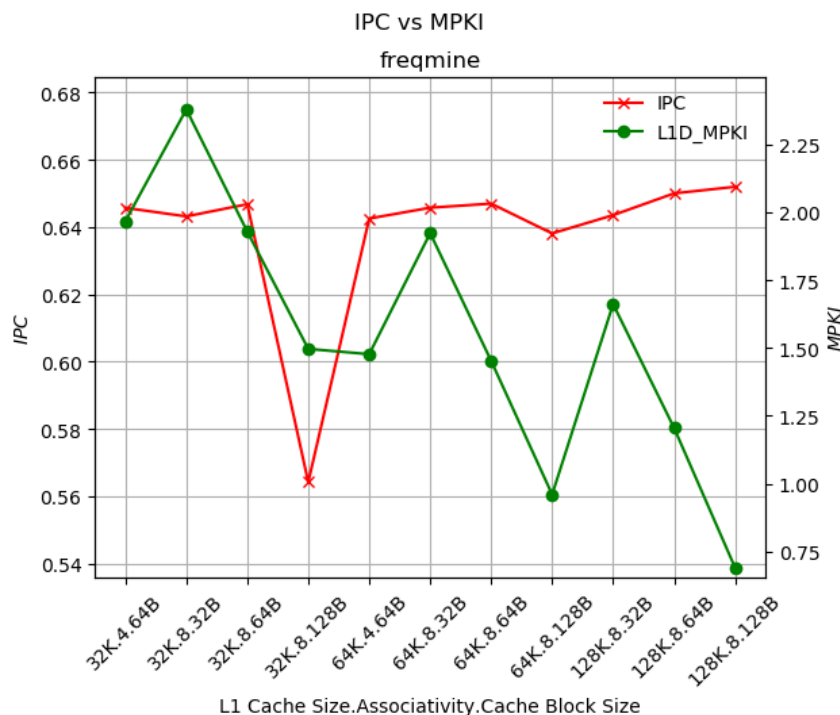
6) Fluidanimate

Στο L1 fluidanimate προκύπτει ως βέλτιστος συνδυασμός ο (128,8,128). Στα 4 τελευταία benchmark(canneal, facesim, ferret, fluidanimate) παρατηρούμε ομοιομορφία στα αποτελέσματα. Για ακόμη μία φορά, το block size διαδραματίζει τον μεγαλύτερο ρόλο στην μεταβολή της απόδοσης, ενώ και η αύξηση του cache size συνεισφέρει ελάχιστα στην βελτίωση της απόδοσης. Το associativity δεν φαίνεται να επιδρά σημαντικά.



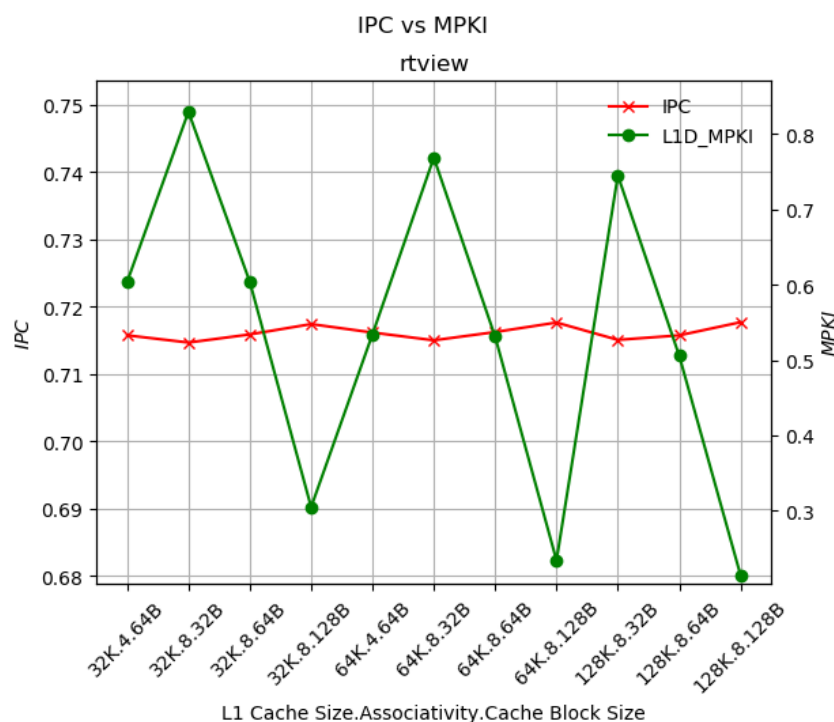
7) Freqmine

Στο L1 freqmine βέλτιστος συνδυασμός είναι ο (128,8,128). Αναμφίβολα, παρατηρούμε σταθερότητα στον αριθμό των εντολών ανά κύκλο (IPC), αν εξαιρέσουμε την απότομη πτώση της τιμής για τον συνδυασμό (32,8,128). Παρόλα αυτά, φαίνεται πως τόσο το cache size όσο και το block size επιδρούν σημαντικά στην μείωση των misses.



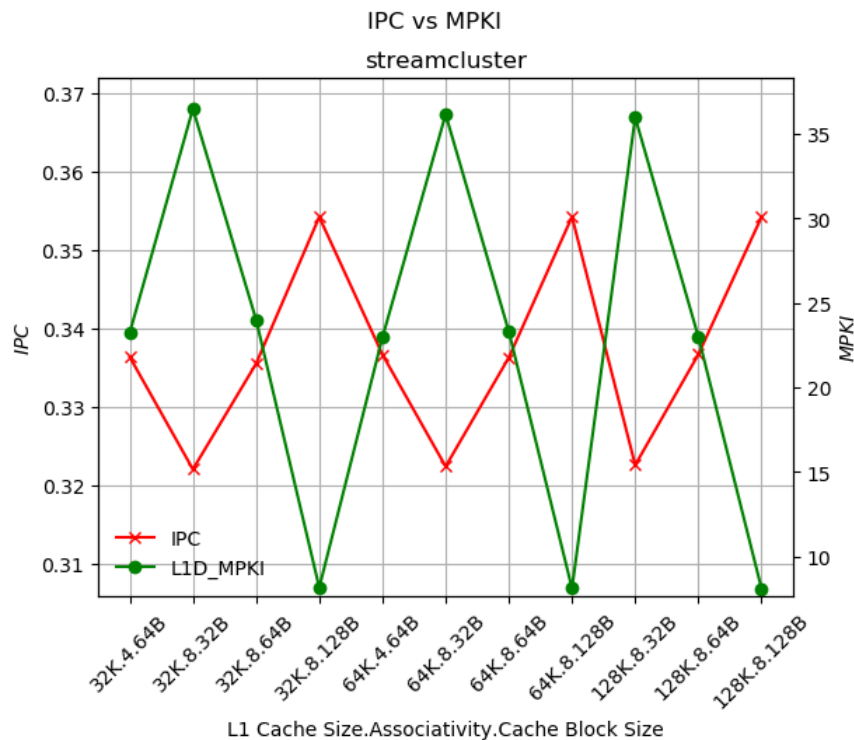
8) Rtview

Στο L1 rtview ο βέλτιστος συνδυασμός είναι ο (128,8,128) με τον (64,8,128) να ακολουθεί από κοντά. Αναμφίβολα, παρατηρούμε μεγάλη σταθερότητα στον αριθμό των εντολών ανά κύκλο (IPC). Όπως και προηγουμένως, αύξηση του cache size και του block size επιδρά θετικά στην μείωση των misses.



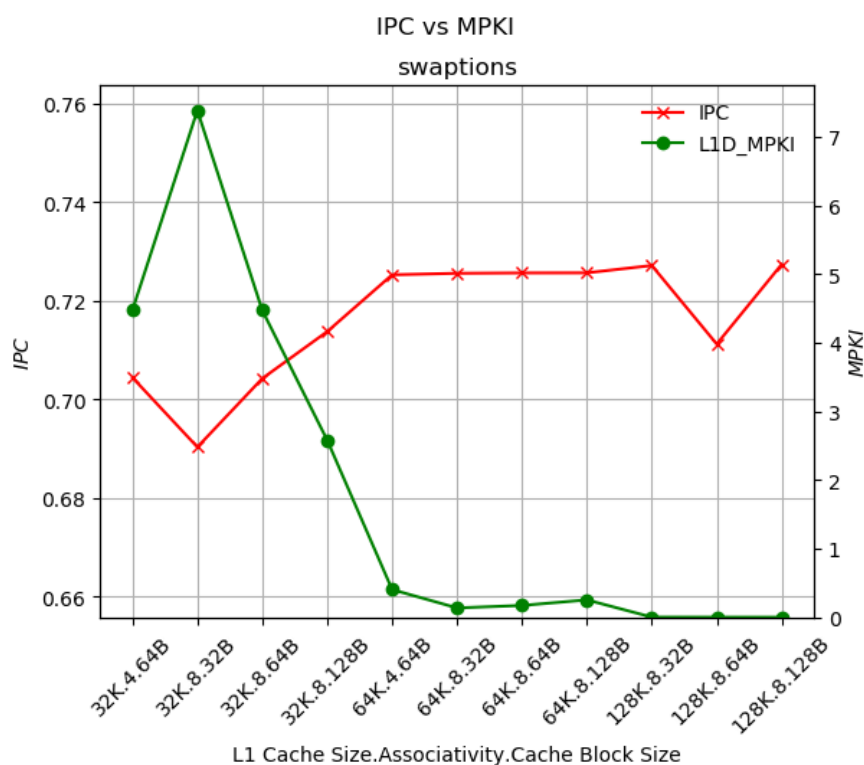
9) Streamcluster

Στο L1 streamcluster εντοπίζουμε τρεις βέλτιστους συνδυασμούς, τους (32,8,128), (64,8,128) και (128,8,128). Το block size είναι ο μόνος παράγοντας που έχει αξιοσημείωτη συνεισφορά στην βελτίωση της απόδοσης, με την τιμή 128 B να κάνει ουσιαστική διαφορά. Οι μεταβολές των cache size και associativity δε δείχνουν να προκαλούν αλλαγές.



10) Swaptions

Στο L1 swaptions βέλτιστοι συνδυασμοί, με ελάχιστη διαφορά από τους υπόλοιπους, είναι οι (128,8,32) και (128,8,128). Για cache size 32 K, φαίνεται πως σημαντικός παράγοντας είναι το block size, ενώ όταν φτάσει στα 64 K, η απόδοση σταθεροποιείται με ελάχιστες διαφορές μεταξύ των συνδυασμών.



7.1 L2 cache

Για τα διάφορα benchmarks και όλες τις περιπτώσεις της L2 που εξετάζονται στο πείραμα αυτό, οι παράμετροι της L1 cache και του TLB θα διατηρηθούν σταθερές και συγκεκριμένα ίσες με:

L1 size = 32 KB

L1 associativity = 8

L1 block size = 64 B

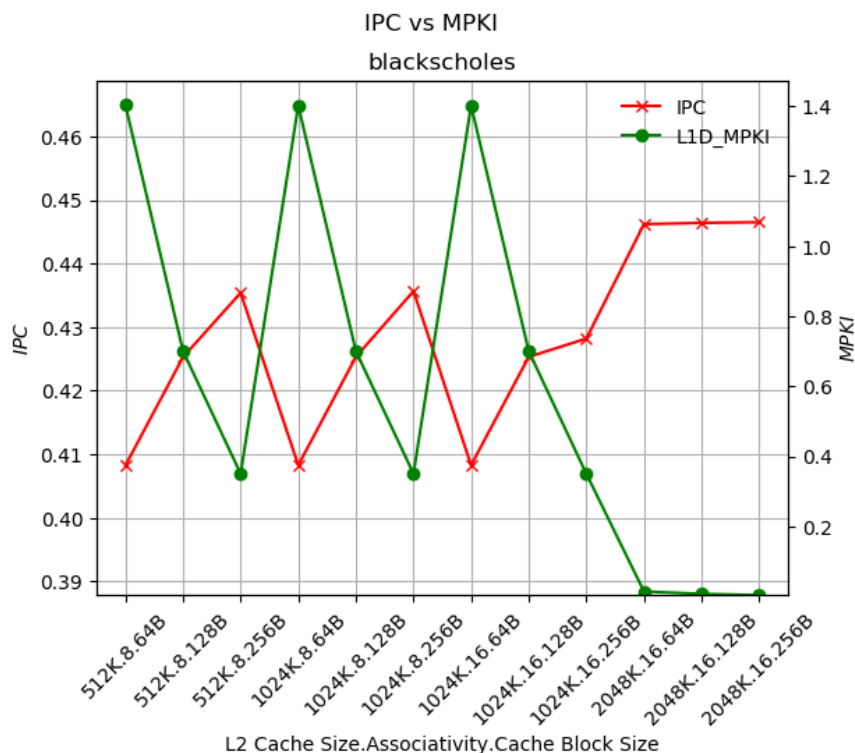
TLB size = 64 entr.

TLB associativity = 4

TLB page size = 4096 B

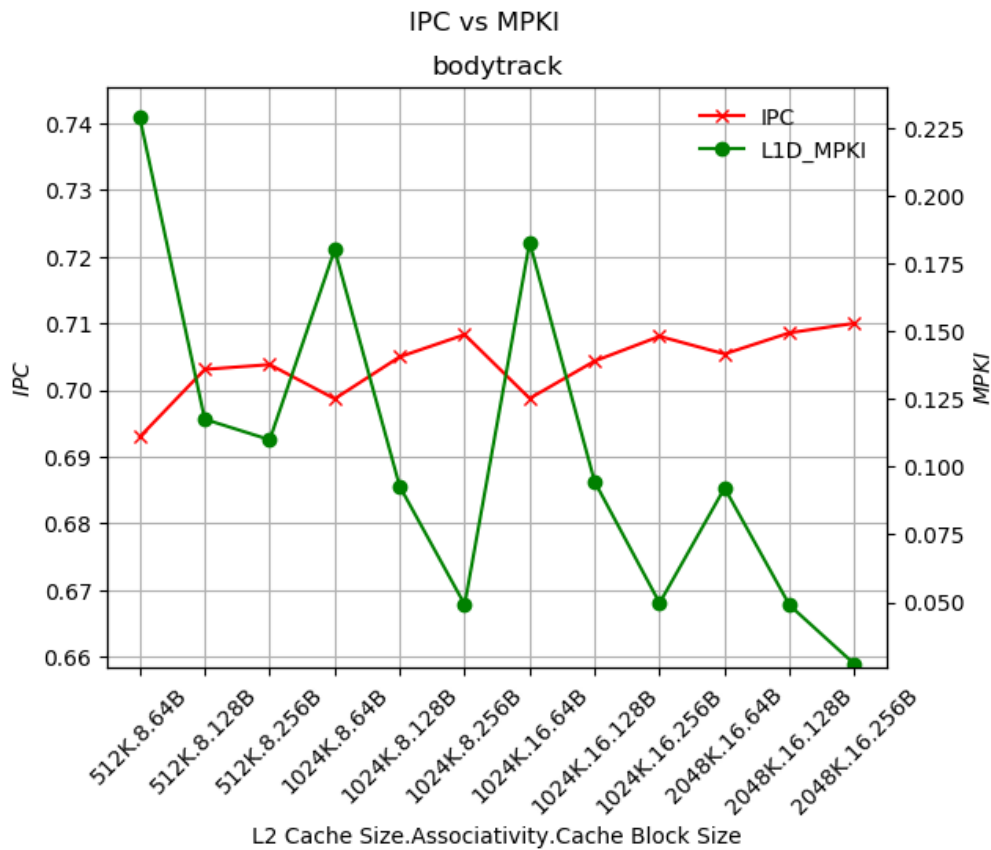
1) Blackscholes

Στο L2 blackscholes έχουμε τρεις βέλτιστους συνδυασμούς και είναι οι (2048,16,64), (2048,16,128) και (2048,16,256). Για cache size 512 K και 1024 K, παρατηρούμε όμοιες καμπύλες, καθώς αύξηση του block size επιφέρει αντίστοιχη βελτίωση της απόδοσης. Όμως, όταν το cache size λάβει την τιμή 2048 K επέρχεται πλήρης σταθερότητα της απόδοσης για τα διάφορα block sizes. Επίσης, αύξηση του associativity φαίνεται να μην έχει καμία επίδραση στο αποτέλεσμα.



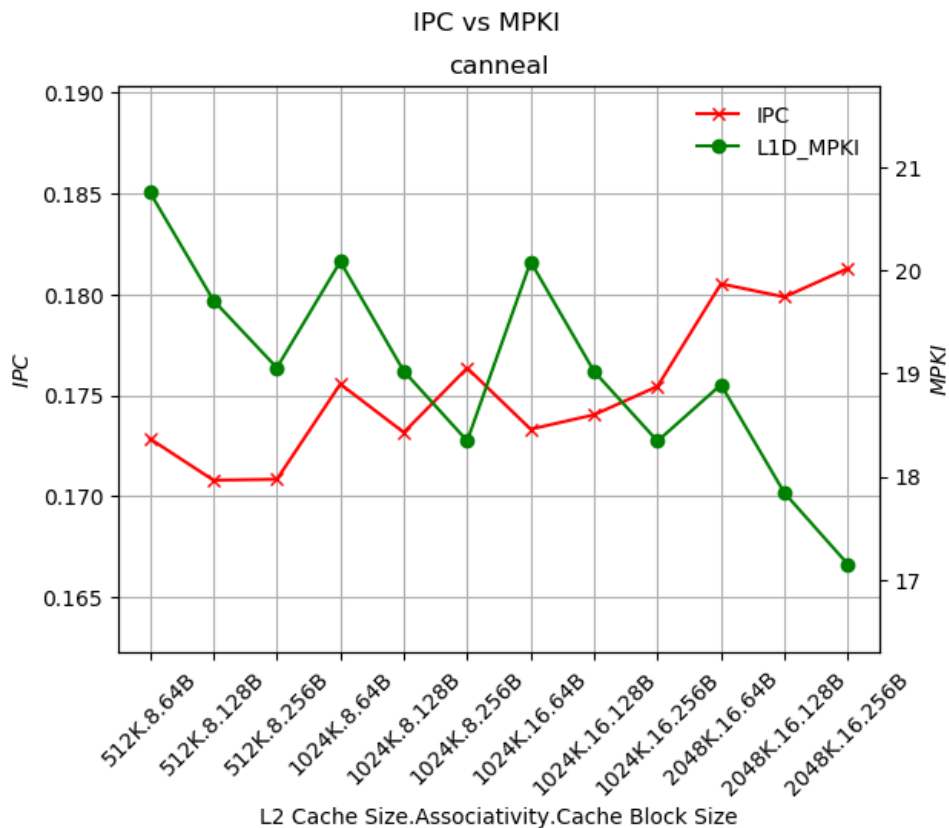
2) Bodytrack

Στο L2 bodytrack βέλτιστος συνδυασμός είναι ο (2048,16,256). Παρατηρούμε πως, ανεξαρτήτου associativity και cache size, αύξηση του block size επιφέρει βελτίωση της απόδοσης. Επίσης, φαίνεται πως όσο αυξάνεται το cache size, τόσο μειώνονται τα misses και αυξάνονται τα IPC, με τις βέλτιστες τιμές να επιτυγχάνονται για 2048 K.



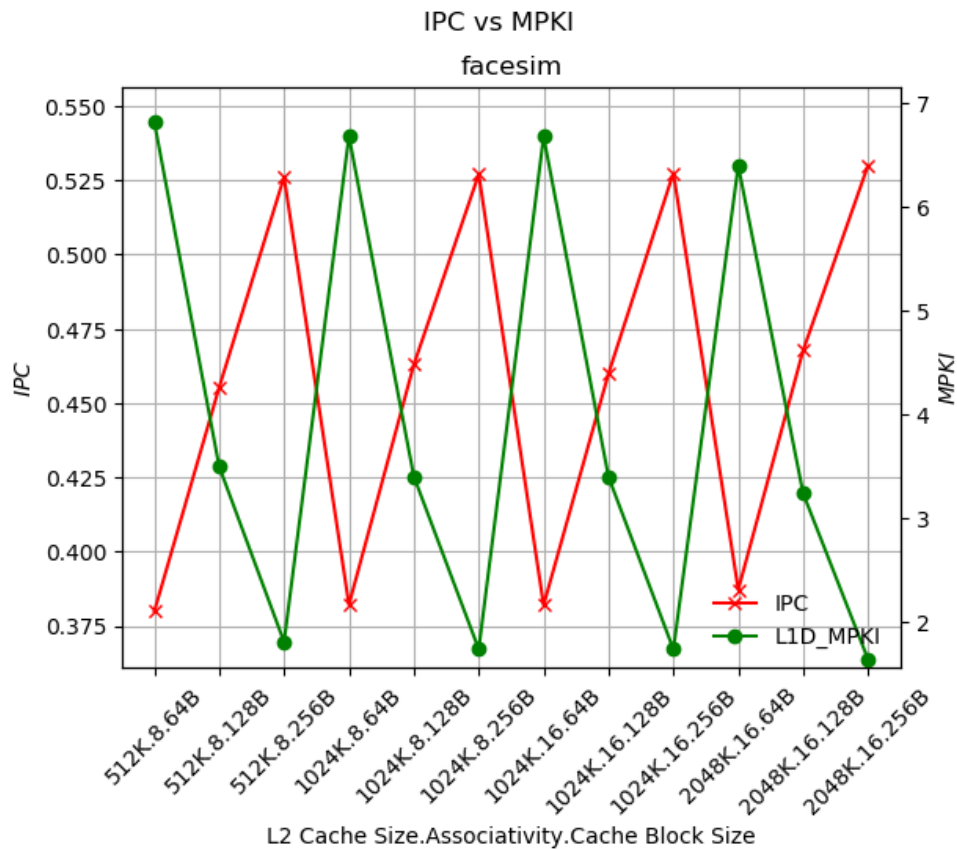
3) Canneal

Στο L2 canneal παρατηρούμε αναμφίβολα ως βέλτιστο συνδυασμό των (2048,16,256). Όπως και στο bodytrack, παρατηρούμε πως, ανεξαρτήτου associativity και cache size, αύξηση του block size επιφέρει βελτίωση της απόδοσης. Παρόλα αυτά, τα IPC δεν εμφανίζουν γραμμική αύξηση όπως προηγουμένως και φαίνεται να εξαρτώνται από το cache size.



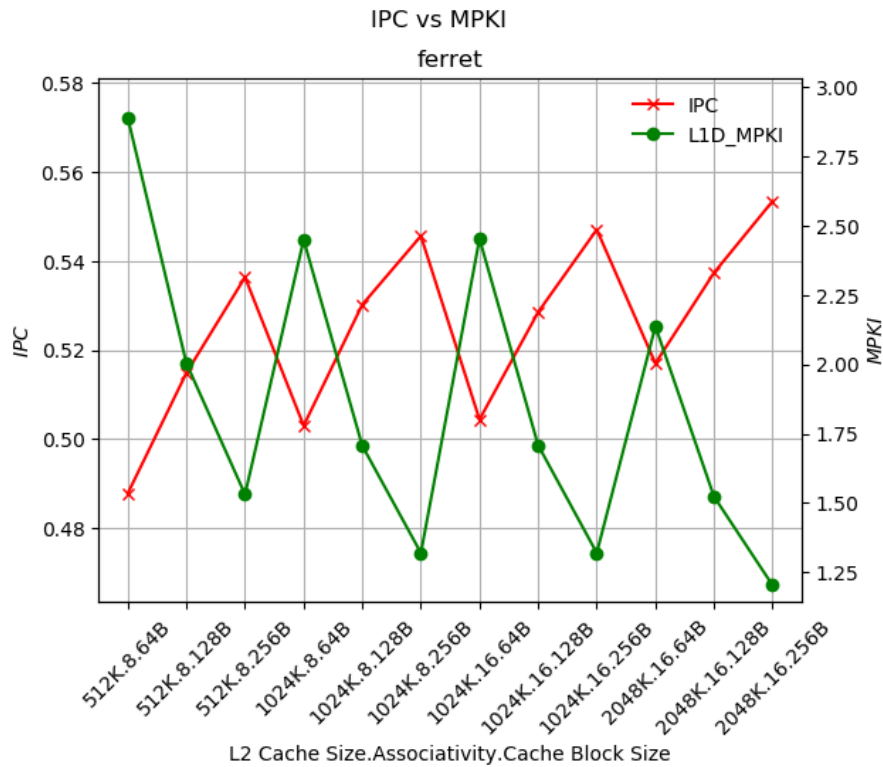
4) Facesim

Στο L2 facesim εντοπίζουμε τέσσερις βέλτιστους συνδυασμούς, με κοινό όρο μεταξύ τους τα 256 B σε block size. Γενικά, παρατηρούμε πως ούτε το cache size ούτε και το associativity έχουν σημαντική επίδραση στο αποτέλεσμα σε αντίθεση με το block size, η αύξηση του οποίου βελτιώνει αισθητά την απόδοση.



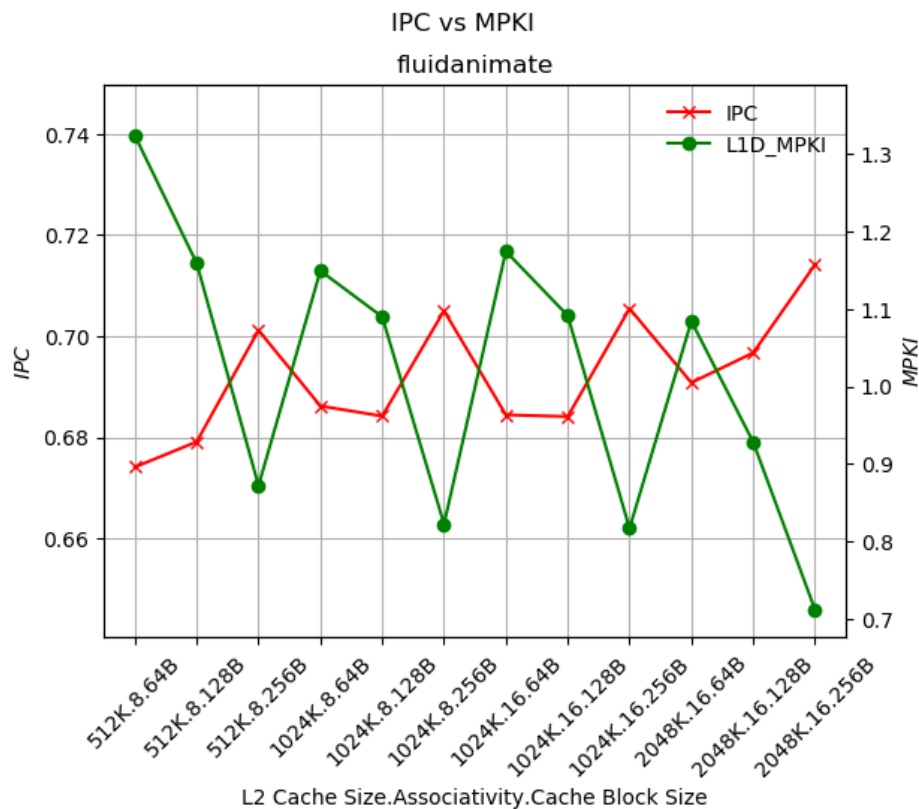
5) Ferret

Στο L2 ferret βέλτιστος συνδυασμός είναι ο (2048,16,256), με τους συνδυασμούς (1024,8,256) και (1024,16,256) να βρίσκονται αρκετά κοντά. Όπως είδαμε, στο bodytrack και το canneal, ανεξαρτήτου associativity και cache size, αύξηση του block size επιφέρει βελτίωση της απόδοσης. Επίσης, το associativity δεν επιδρά σημαντικά στο αποτέλεσμα, ενώ το cache size διαδραματίζει κάποιο ρόλο στην απόδοση, αφού αύξηση του συμβάλλει στην βελτίωσή της.



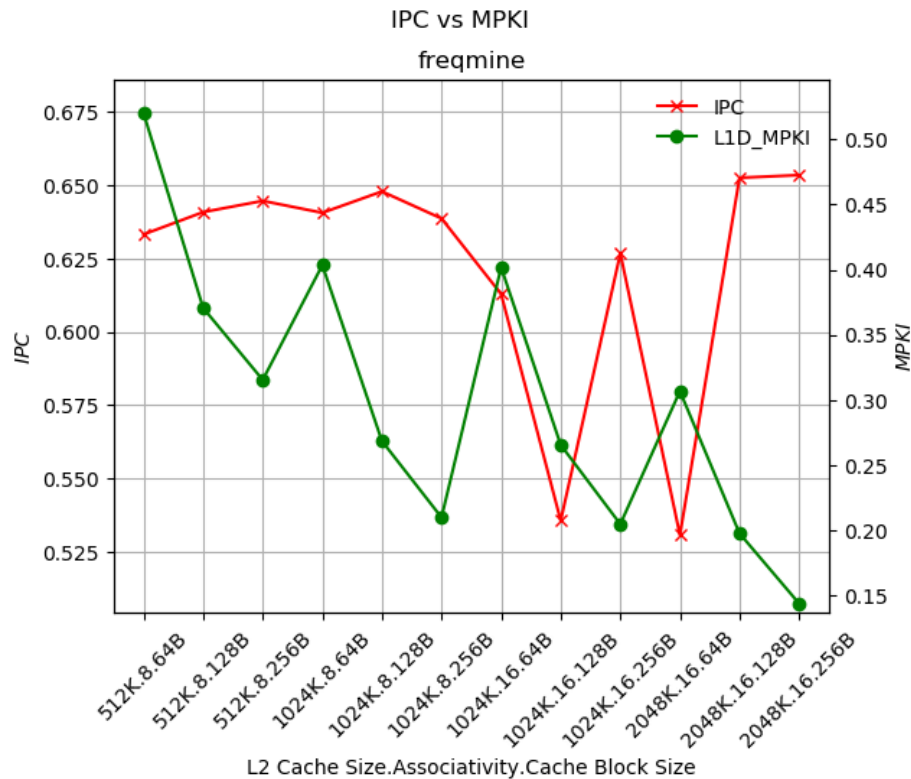
6) Fluidanimate

Στο L2 fluidanimate βέλτιστος συνδυασμός είναι ο (2048,16,256). Τα αποτελέσματα που παρατηρούμε είναι όμοια με εκείνα του ferret. Η μόνη διαφορά που εντοπίζουμε είναι για τις τιμές block size 64 B και 128 B, στις οποίες για cache size 1024 K παρατηρούμε μικρότερη μείωση του αριθμού των misses και σταθερό IPC.



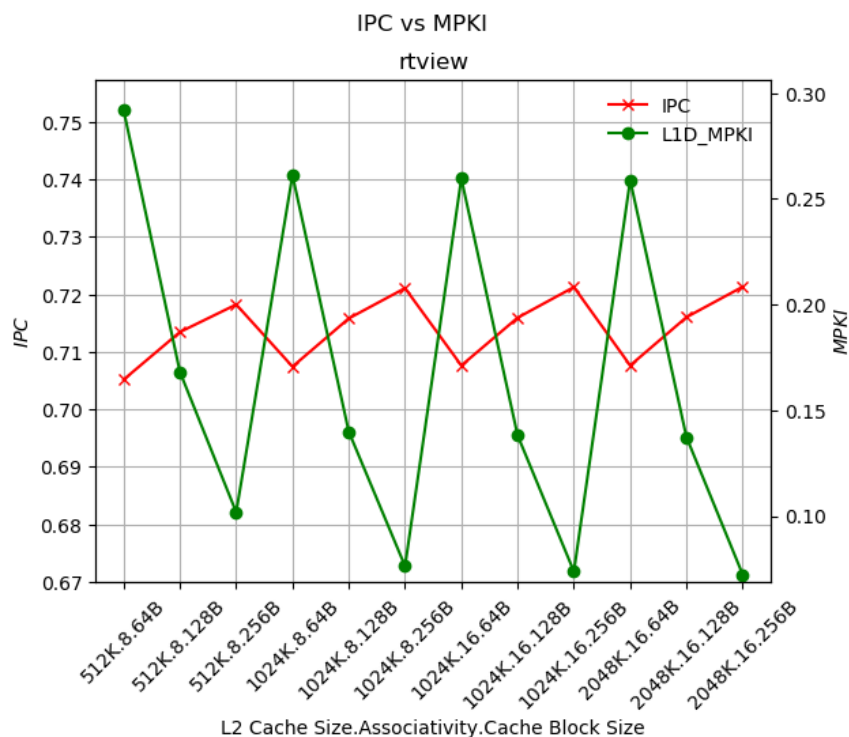
7) Freqmine

Στο L2 freqmine εντοπίζουμε ως βέλτιστο συνδυασμό τον (2048,16,256). Όσον αφορά τα MKPI, η καμπύλη είναι όμοια με εκείνη στα benchmark bodytrack, canneal και ferret, οπότε προκύπτουν τα ίδια συμπεράσματα. Όμως, στο IPC, ενώ στην αρχή υπάρχει σταθερότητα μέχρι και για cache size 1024 K και associativity 8, στην συνέχεια προκύπτουν αυξομειώσεις που δύσκολα εξηγούνται.



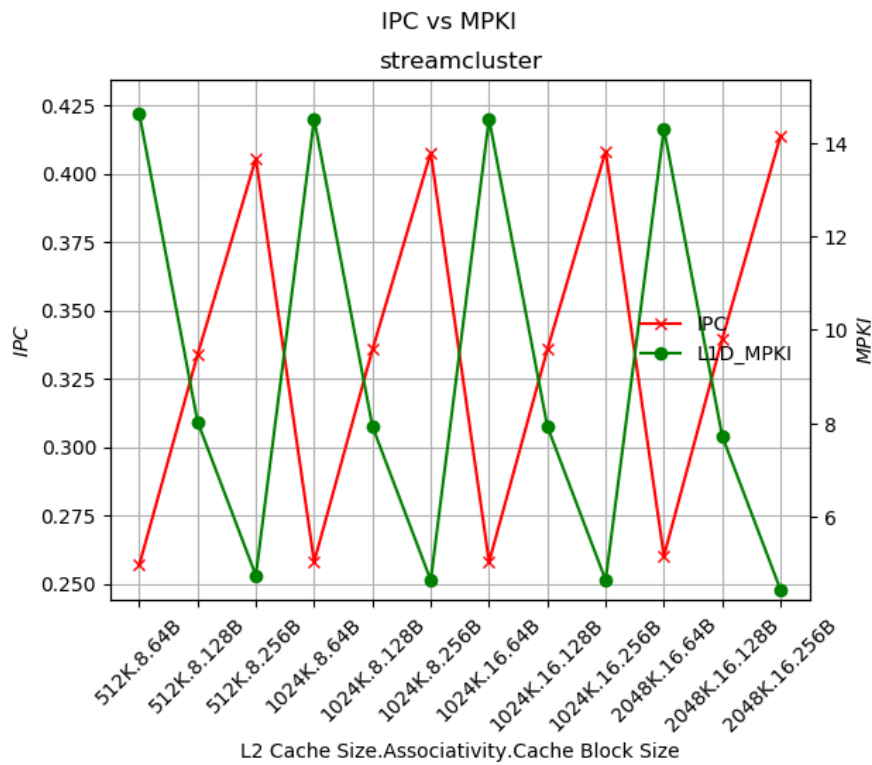
8) Rtview

Στο L2 rtview εντοπίζουμε τρεις βέλτιστους συνδυασμούς, τους (1024,8,256),(1024,16,256) και (2048,16,256). Αναμφίβολα οι καμπύλες εμφανίζουν γραμμικές σχέσεις που μοιάζουν σε εκείνες του benchmark ferret και του facesim.



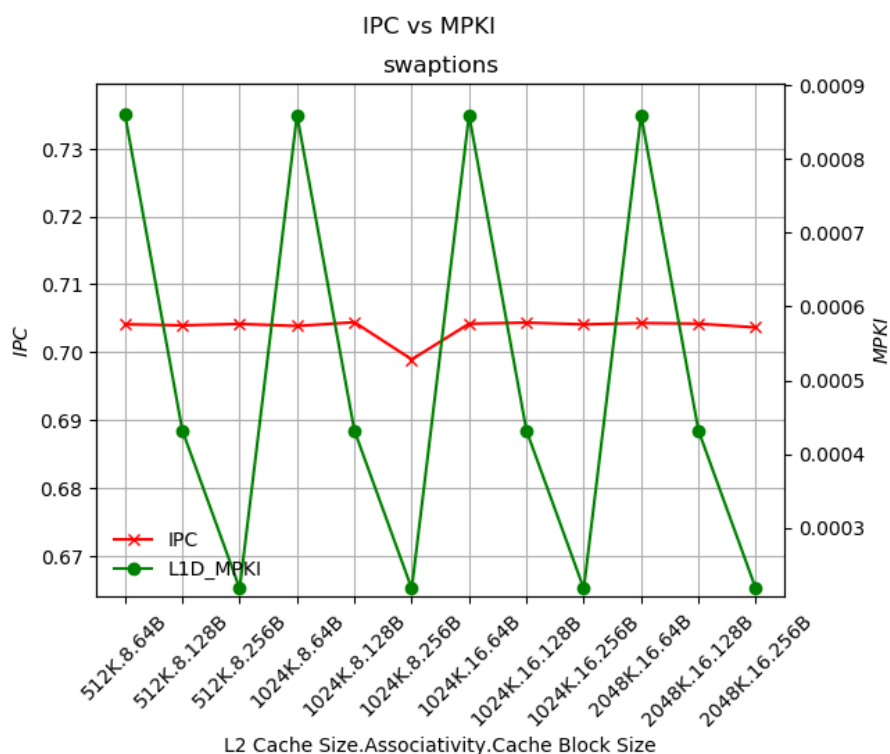
9) Streamcluster

Στο L2 streamcluster εντοπίζουμε τέσσερις βέλτιστους συνδυασμούς, με κοινό όρο μεταξύ τους τα 256 B σε block size, όπως και στο facesim. Γενικά, παρατηρούμε πως ούτε το cache size ούτε και το associativity έχουν σημαντική επίδραση στο αποτέλεσμα σε αντίθεση με το block size, η αύξηση του οποίου βελτιώνει αισθητά την απόδοση.



10) Swaptions

Στο L2 swaptions εντοπίζουμε τέσσερις βέλτιστους συνδυασμούς, με κοινό όρο μεταξύ τους τα 256 B σε block size, όπως και προηγουμένως. Τα αποτελέσματα του IPC είναι μία ευθεία γραμμή με μοναδική εξαίρεση τον συνδυασμό (1024,8,256).



7.1 TLB

Για τα διάφορα benchmarks και όλες τις περιπτώσεις του TLB που εξετάζονται στο πείραμα αυτό, οι παράμετροι των L1 και L2 cache θα διατηρηθούν σταθερές και συγκεκριμένα ίσες με:

L1_size = 32 KB

L1_associativity = 8

L1_block_size = 64 B

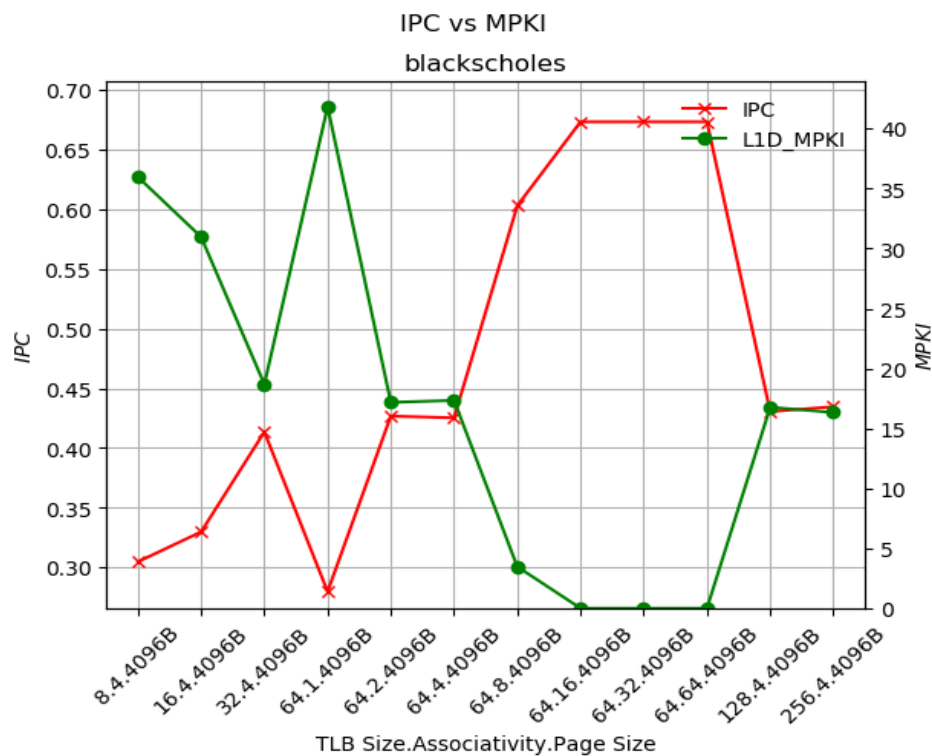
L2_size = 1024 entr.

L2_associativity = 8

L2_block_size = 128 B

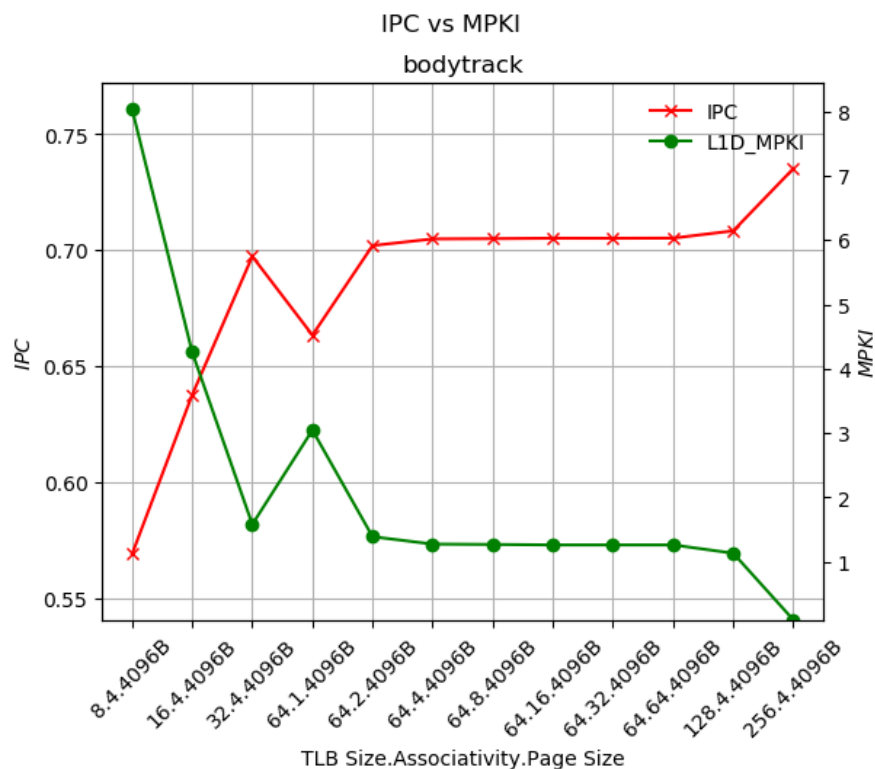
1) Blackscholes

Στο TLB blackscholes εντοπίζουμε τρεις βέλτιστους συνδυασμούς, τους (64,16,4096), (64,32,4096) και (64,64,4096). Τόσο το TLB size όσο και το associativity είναι πολύ σημαντικοί παράγοντες, καθώς η αύξησή τους βελτιώνει την απόδοση. Για το associativity συγκεκριμένα, παρατηρούμε μεταβολή όταν αυξάνεται από 1 σε 2 και μετά ξανά όταν αυξάνεται από 4 σε 8.



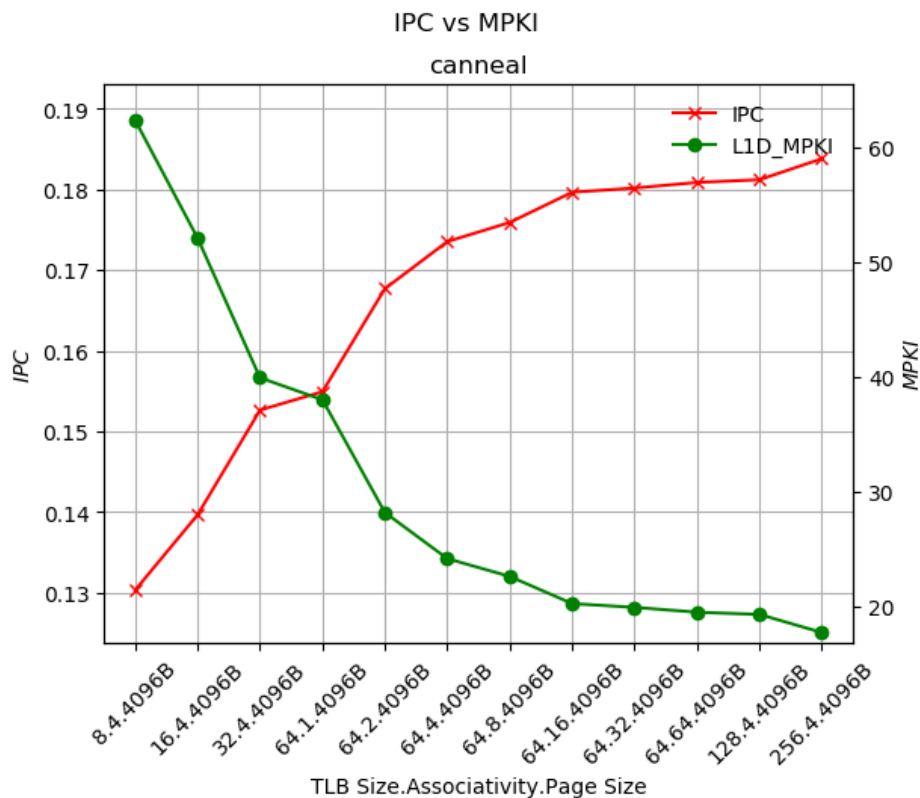
2) Bodytrack

Στο TLB bodytrack βέλτιστος συνδυασμός είναι με διαφορά ο (256,4,4096). Παρατηρούμε πως, σημαντικότερος παράγοντας είναι το TLB size, καθώς κάθε αύξησή του βελτιώνει θεαματικά την απόδοση. Το associativity δεν συνεισφέρει ιδιαίτερα, αν και κάθε αύξησή του επιφέρει και μια μικρή βελτίωση. Μοναδική εξαίρεση για associativity=1, όπου είχαμε πολύ μειωμένη απόδοση.



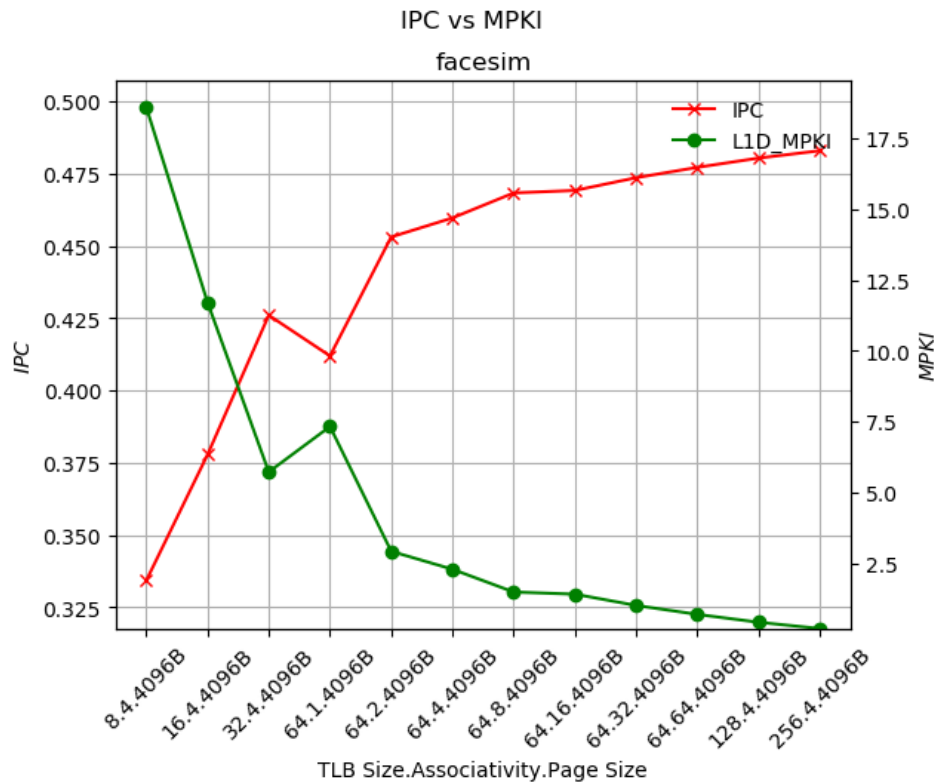
3) Canneal

Στο TLB canneal βέλτιστος συνδυασμός είναι ο (256,4,4096). Γενικά, όπως και στα προηγούμενα benchmark, παρατηρούμε πως κάθε αύξηση του TLB size και του associativity επιδρά στην βελτίωση της απόδοσης, ιδιαίτερα έως και τον συνδυασμό (64,2,4096).



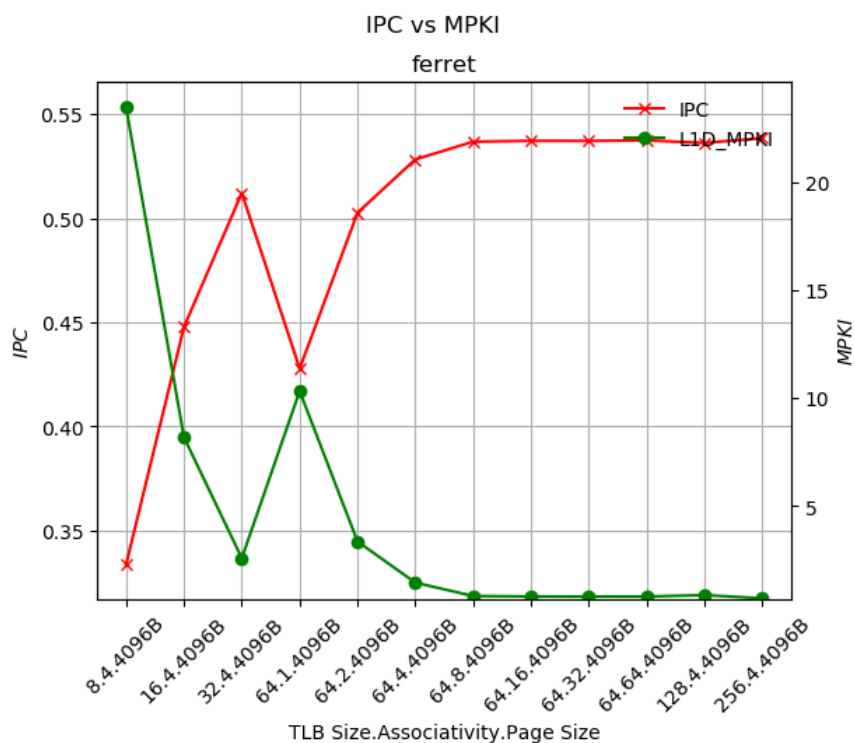
4) Facesim

Στο TLB facesim βέλτιστος συνδυασμός είναι ο (256,4,4096), με όχι κάποια ιδιαίτερη διαφορά. Και σε αυτό το benchmark παρατηρούμε όμοια αποτελέσματα, με μόνη εξαίρεση για associativity=1, όπου πέφτει η απόδοση.



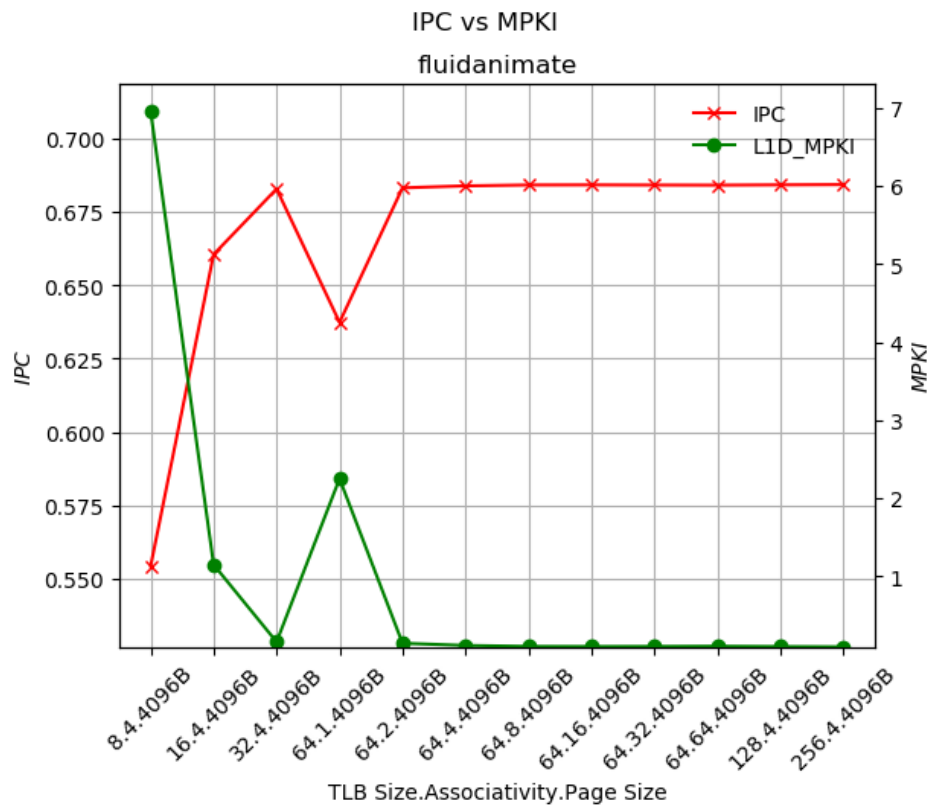
5) Ferret

Στο TLB ferret βέλτιστος συνδυασμός είναι ο (256,4,4096), με μηδαμινή διαφορά από τους υπόλοιπους συνδυασμούς για TLB size ≥ 64 και associativity=4. Όπως και προηγουμένως, για associativity=1, παρατηρούμε πολύ μειωμένη απόδοση.



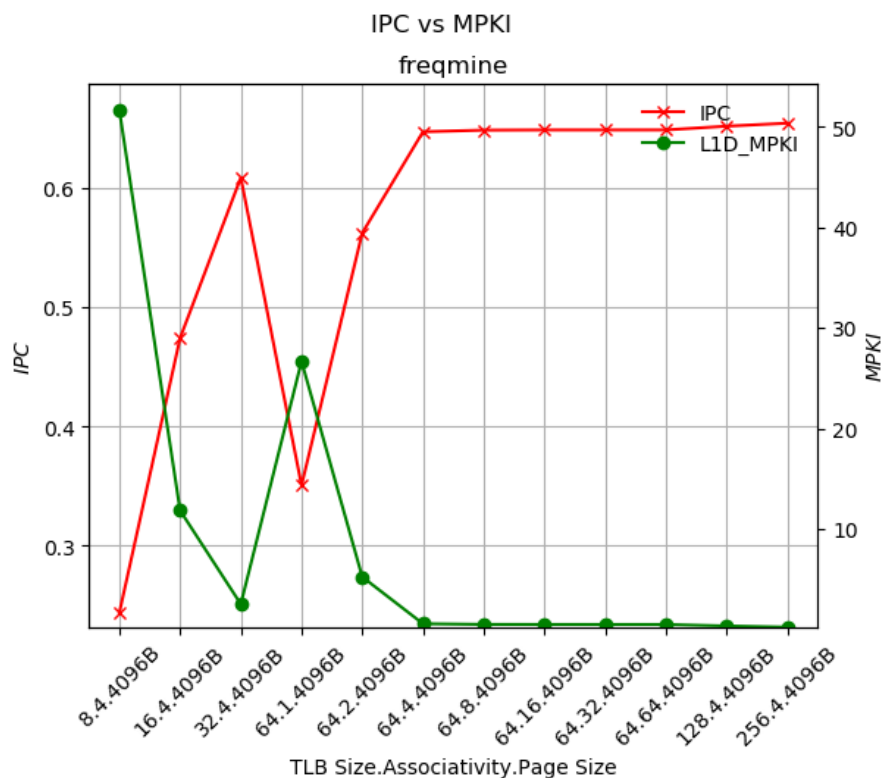
6) Fluidanimate

Στο TLB fluidanimate προκύπτει βέλτιστη απόδοση για κάθε συνδυασμό από τον (64,2,4096) και μετά. Γενικά παρατηρούμε όμοια αποτελέσματα με όλα τα προηγούμενα benchmark.



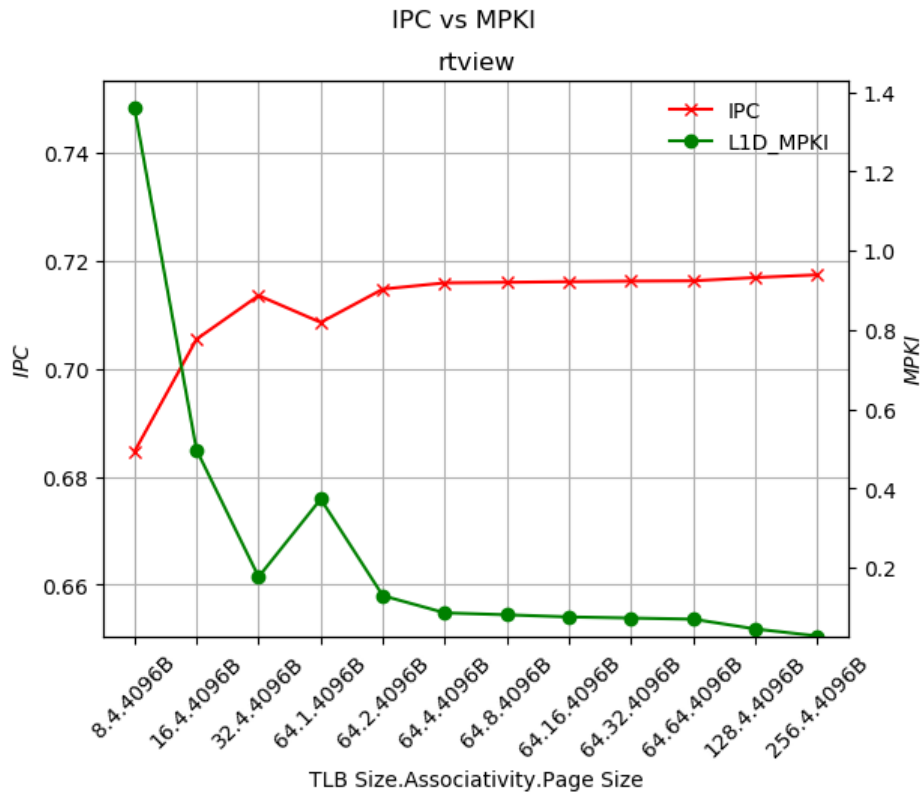
7) Freqmine

Στο TLB freqmine παρατηρούμε σταθερά βέλτιστη απόδοση για τους συνδυασμούς μετά και τον (64,4,4096). Μια αξιοσημείωτη παρατήρηση είναι πως για associativity=1, η απόδοση πέφτει δραματικά, ενώ και αρχικά το TLB size επιδρά στο αποτέλεσμα.



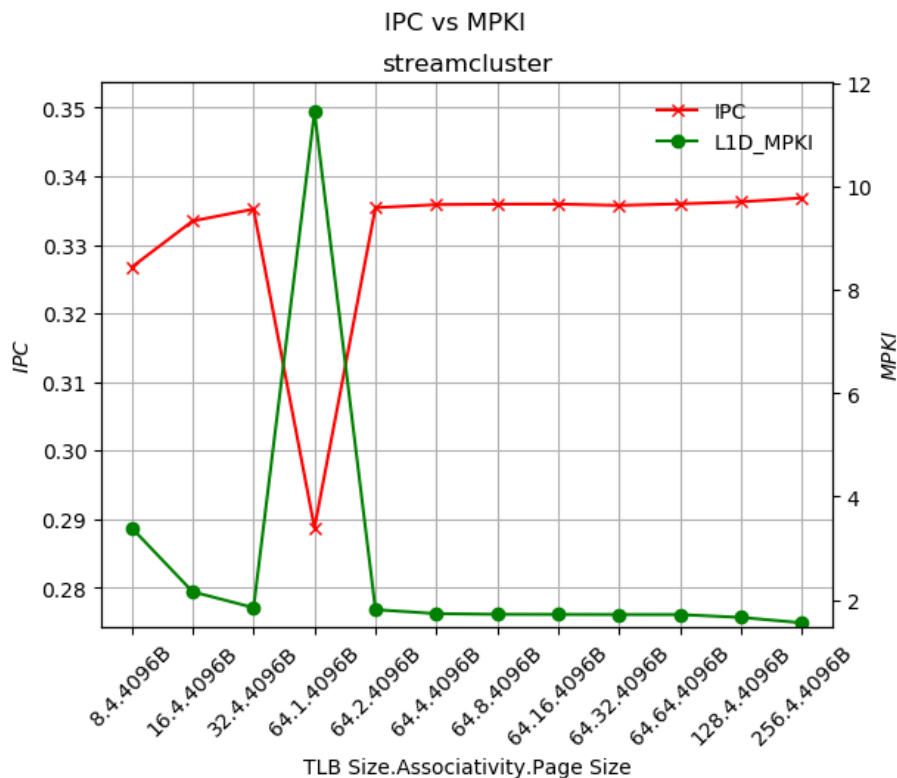
8) Rtview

Στο TLB rtview παρατηρούμε ίδια αποτελέσματα με όλα τα προηγούμενα benchmark, μόνο που η πτώση της απόδοσης για associativity=1 είναι φανερά ανεπαίσθητη.



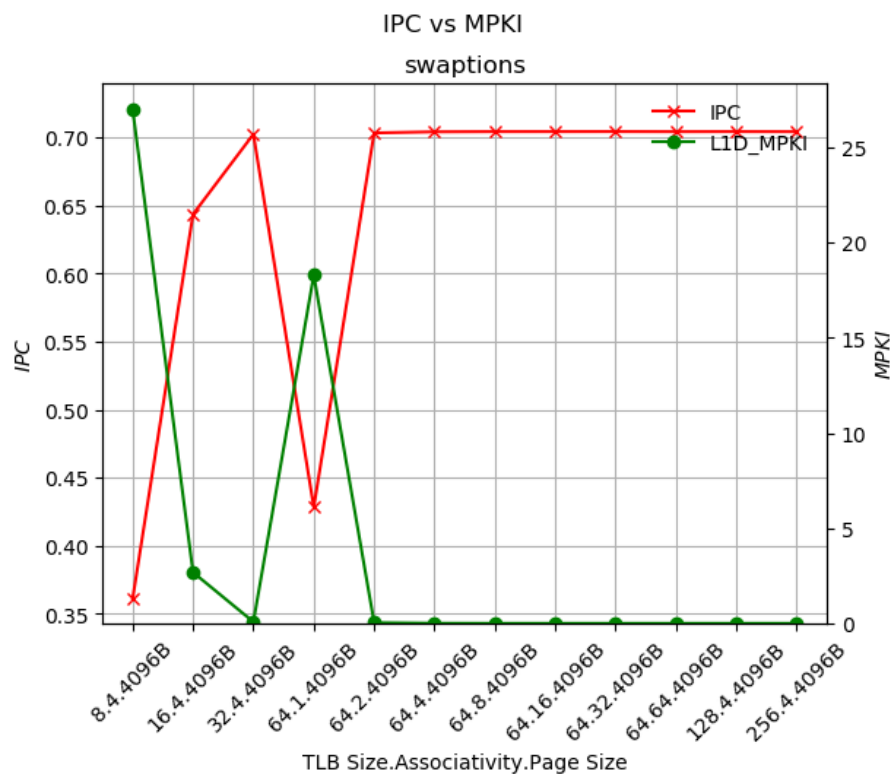
9) Streamcluster

Στο TLB streamcluster παρατηρούμε τέλεια απόδοση για σχεδόν όλους τους συνδυασμούς. Εξάιρεση αποτελεί ο συνδυασμός με associativity=1, που παρουσιάζει την μεγαλύτερη πτώση απόδοσης σε σύγκριση με κάθε άλλο benchmark.



10) Swaptions

Στο TLB swaptions παρατηρούμε όμοια αποτελέσματα με όλα τα προηγούμενα benchmark.



7.1 prefetching

Εκτελούμε τα benchmarks για $n = 1, 2, 4, 8, 16, 32, 64$. Οι παράμετροι των L1, L2, TLB θα διατηρηθούν σταθερές και συγκεκριμένα ίσες με:

L1 size = 32 KB

L2 size = 1024KB

L1 block size = 64 B

L2 block size = 128 B

L1 associativity = 8

L2 associativity = 8

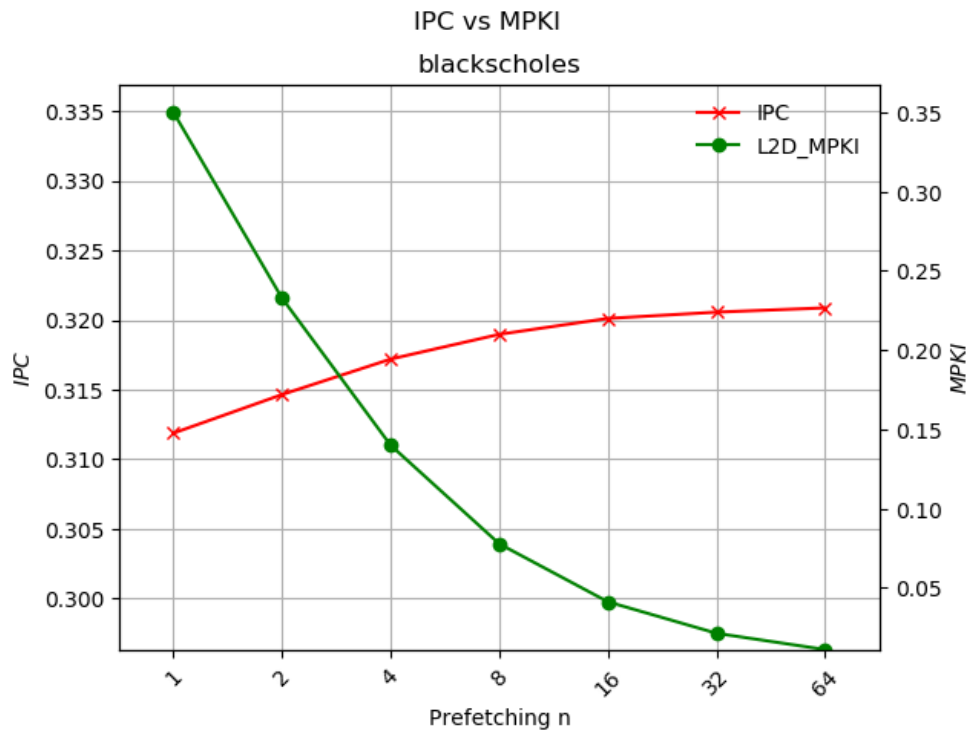
TLB size = 64 entr.

TLB associativity = 4

TLB page size = 4096 B

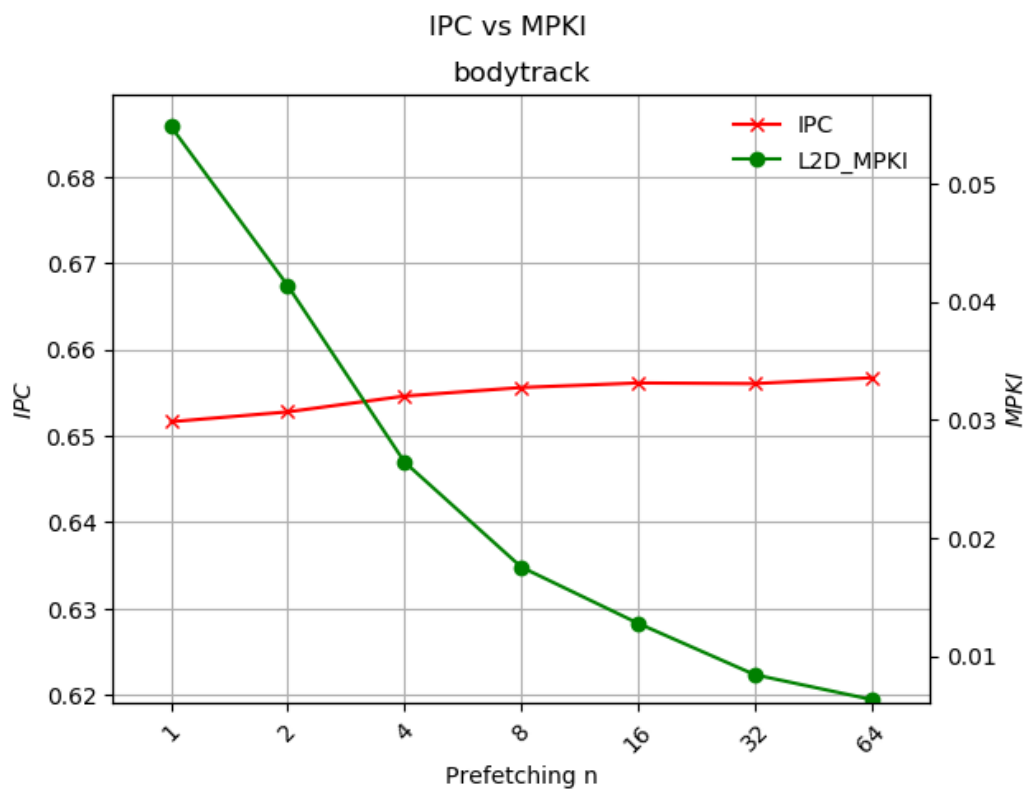
1) Blackscholes

Στο blackscholes παρατηρούμε ότι βέλτιστη επιλογή για το prefetching είναι το 64. Όσο αυξάνεται το prefetching, βελτιώνεται και η απόδοση.



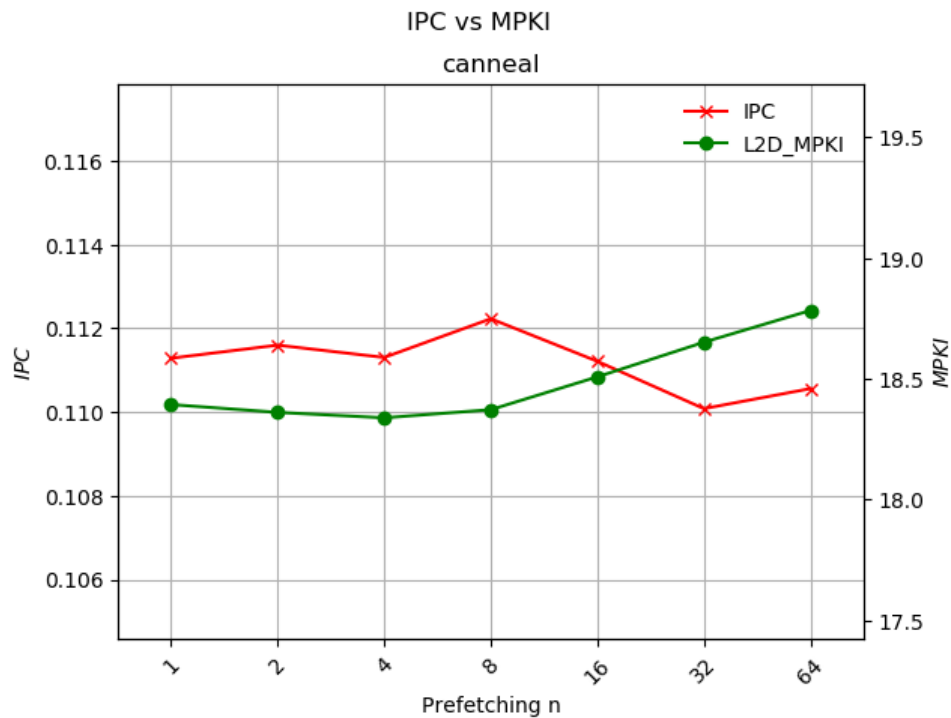
2) Bodytrack

Στο bodytrack βέλτιστη επιλογή είναι το 64. Διαπιστώνουμε πως για κάθε αύξηση του prefetching έχουμε και βελτίωση στην απόδοση. Οι μεταβολές, όμως, είναι πολύ μικρές και όσο αυξάνεται το prefetching είναι όλο και μικρότερες.



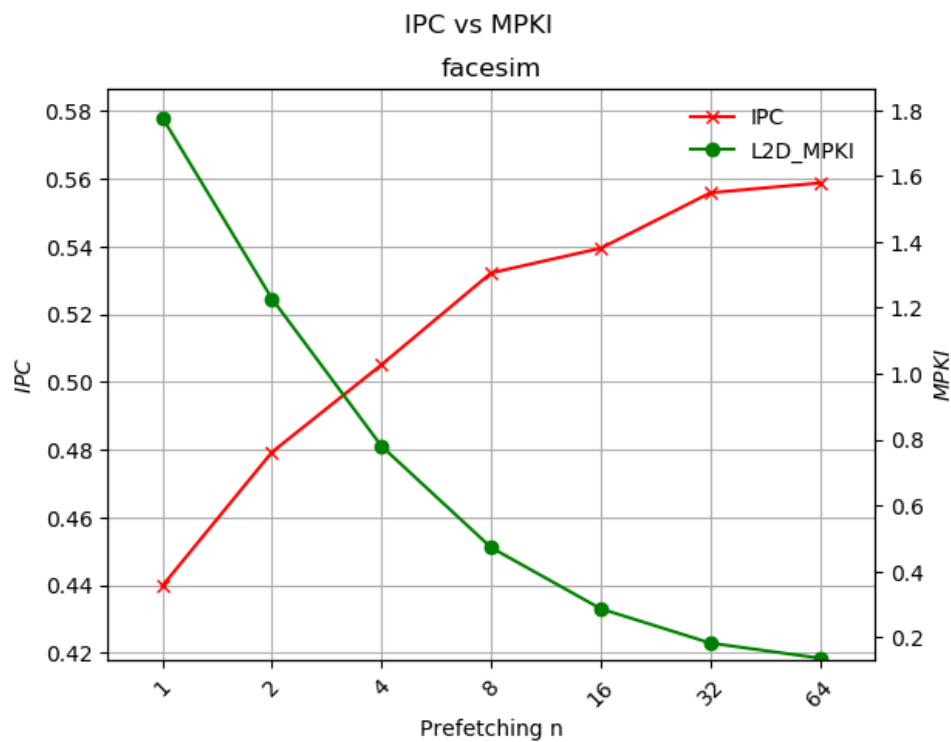
3) Canneal

Στο canneal βέλτιστη επιλογή είναι το 8. Αντίθετα με τις προηγούμενες περιπτώσεις, δεν υπάρχει προβλέψιμη μεταβολή της απόδοσης.



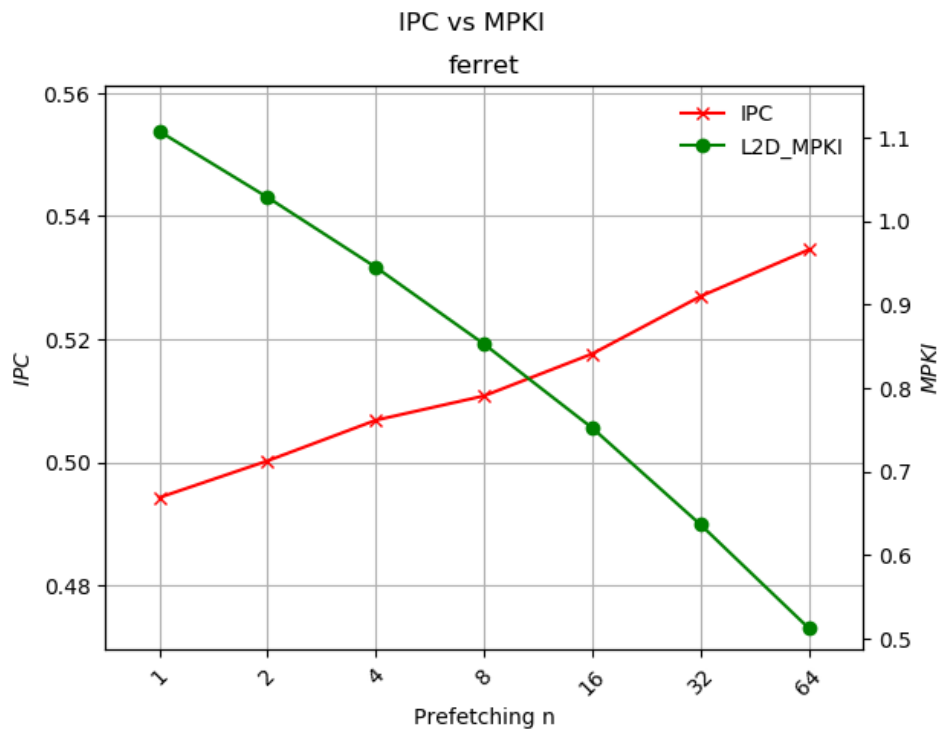
4) Facesim

Στο facesim βέλτιστη επιλογή είναι το 64. Κάθε αύξηση του prefetching βελτιώνει την απόδοση, ωστόσο με φθίνοντα ρυθμό.



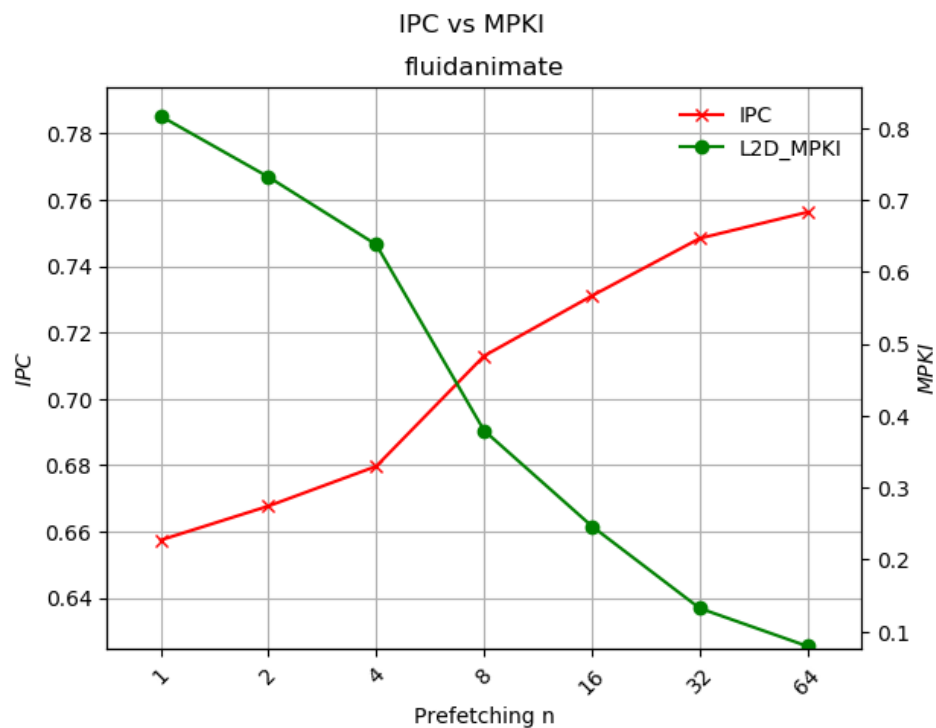
5) Ferret

Στο ferret βέλτιστη επιλογή είναι το 64. Παρατηρούμε γραμμική μεταβολή της απόδοσης, με ελάχιστα πιο απότομες μεταβολές για μεγαλύτερες τιμές prefetching.



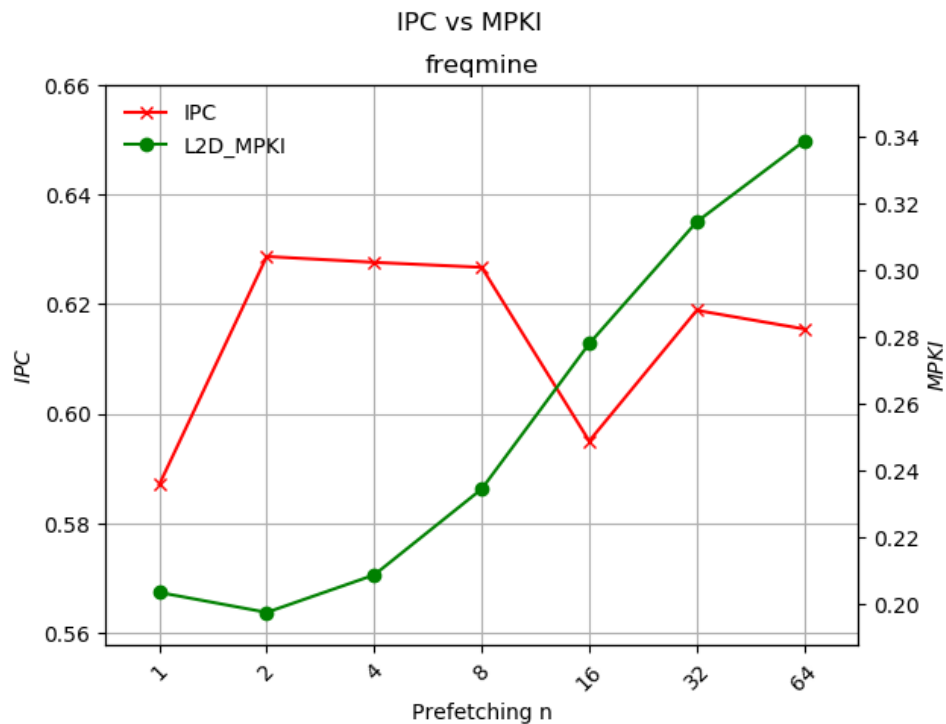
6) Fluidanimate

Στο fluidanimate βέλτιστη επιλογή είναι το 64. Κάθε αύξηση του prefetching για ακόμα μια φορά βελτιώνει την απόδοση. Η πιο απότομη μεταβολή πραγματοποιείται όταν το prefetching αυξάνεται από 4 σε 8.



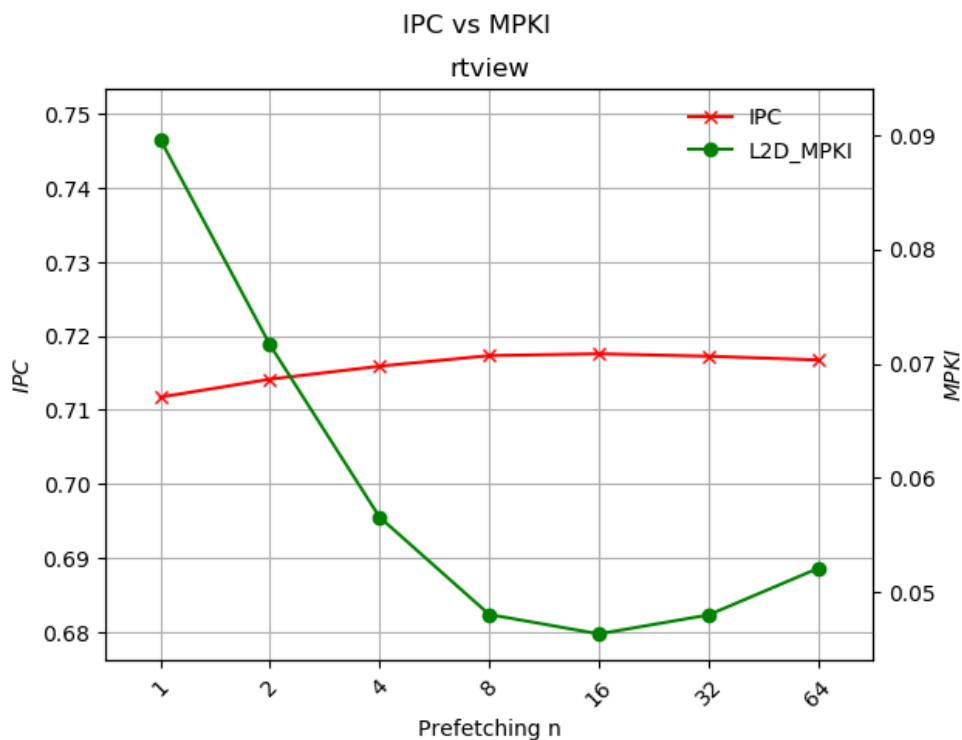
7) Freqmine

Στο freqmine βέλτιστη επιλογή είναι το 2. Παρατηρούμε πως για $n > 2$ υπάρχει ελάχιστη πτώση της απόδοσης, με εξαίρεση για $n=16$, στο οποίο η απόδοση πέφτει απότομα.



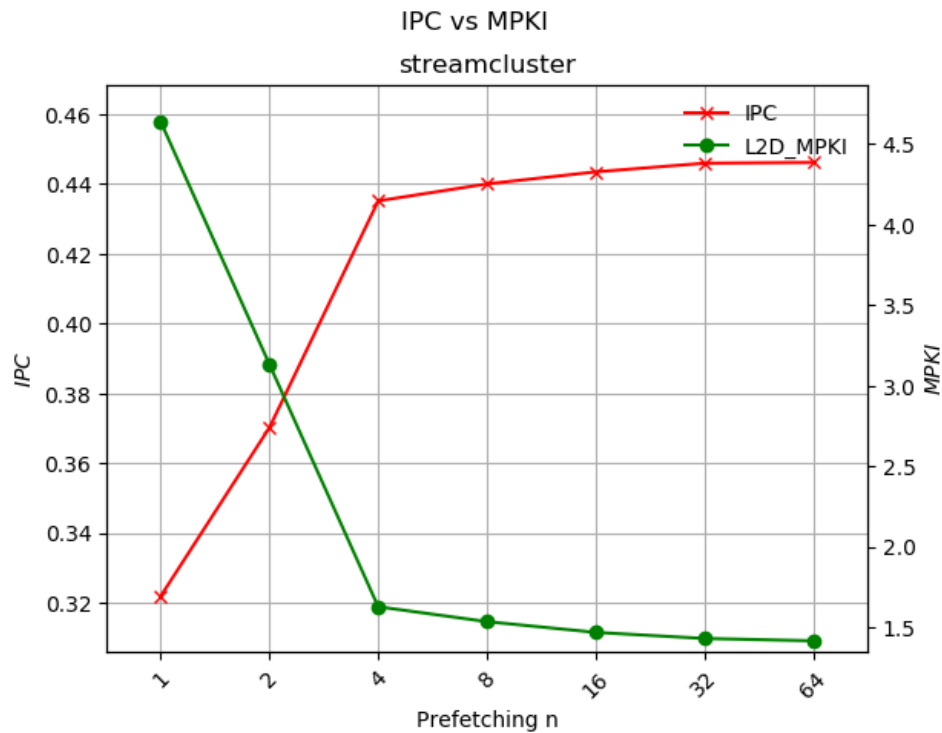
8) Rtview

Στο rtview βέλτιστη επιλογή είναι το 16, με το 8 και το 32 να πλησιάζουν αρκετά κοντά. Όπως και για τα benchmark blackscholes και facesim κάθε αύξηση του prefetching βελτιώνει ελάχιστα και την απόδοση.



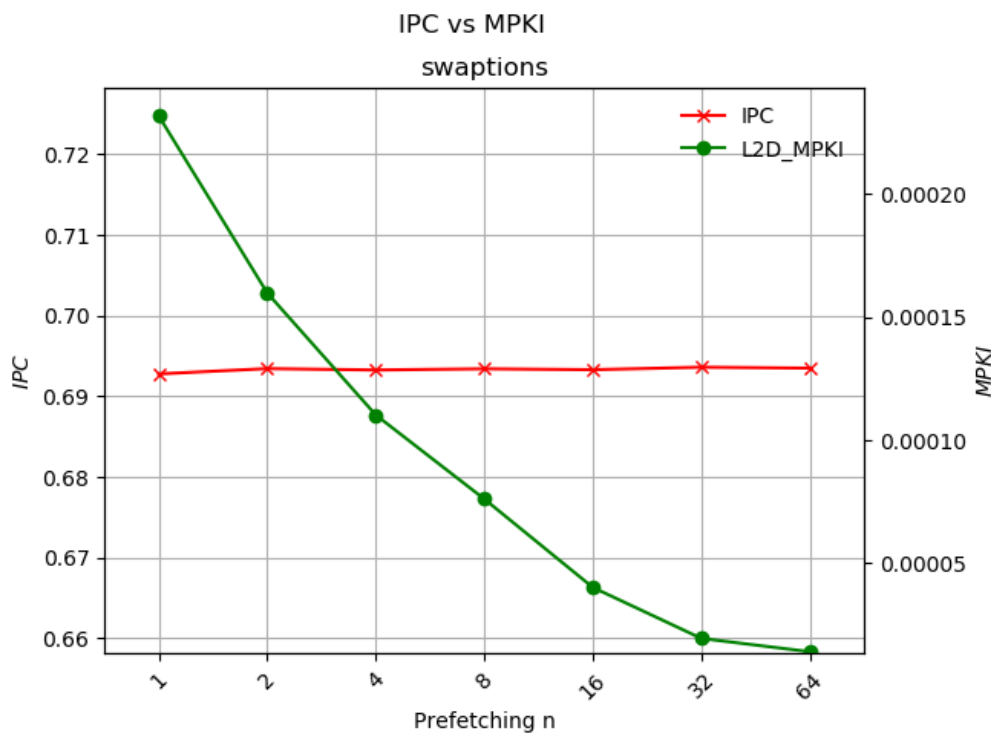
9) Streamcluster

Στο streamcluster βέλτιστη επιλογή είναι το 64. Η αύξηση του prefetching βελτιώνει την απόδοση . Στην αρχή παρατηρούμε θεαματικές αλλαγές , μέχρι το prefetching να φτάσει στην τιμή 4, όπου οι μεταβολές είναι μικρότερες.



10) Swaptions

Στο swaptions όλες οι επιλογές φαίνεται να έχουν σχεδόν το ίδιο αποτέλεσμα , με εξαίρεση την τιμή 1. Με πολύ μικρή διαφορά θα λέγαμε ως βέλτιστη επιλογή το 64.

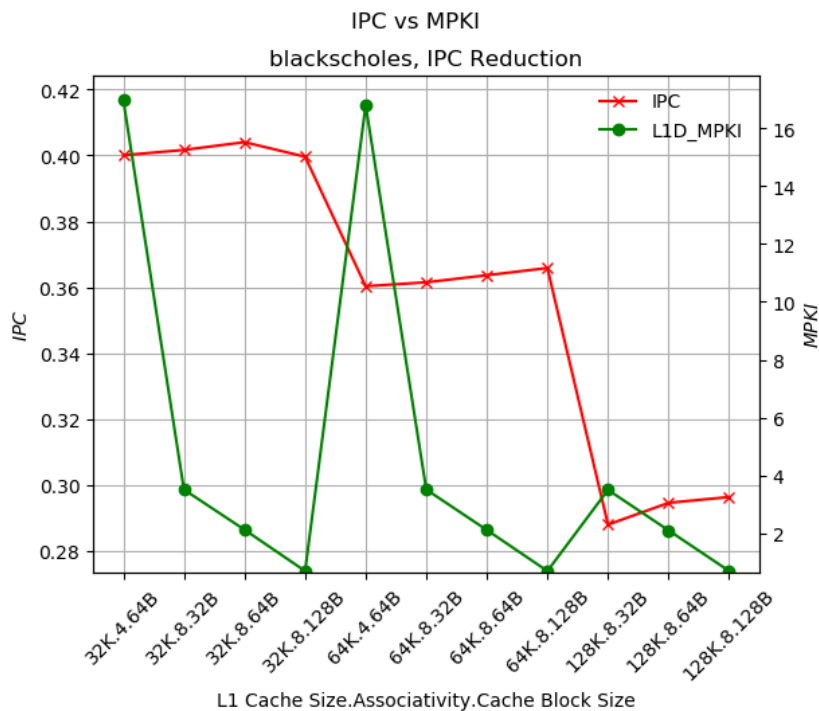


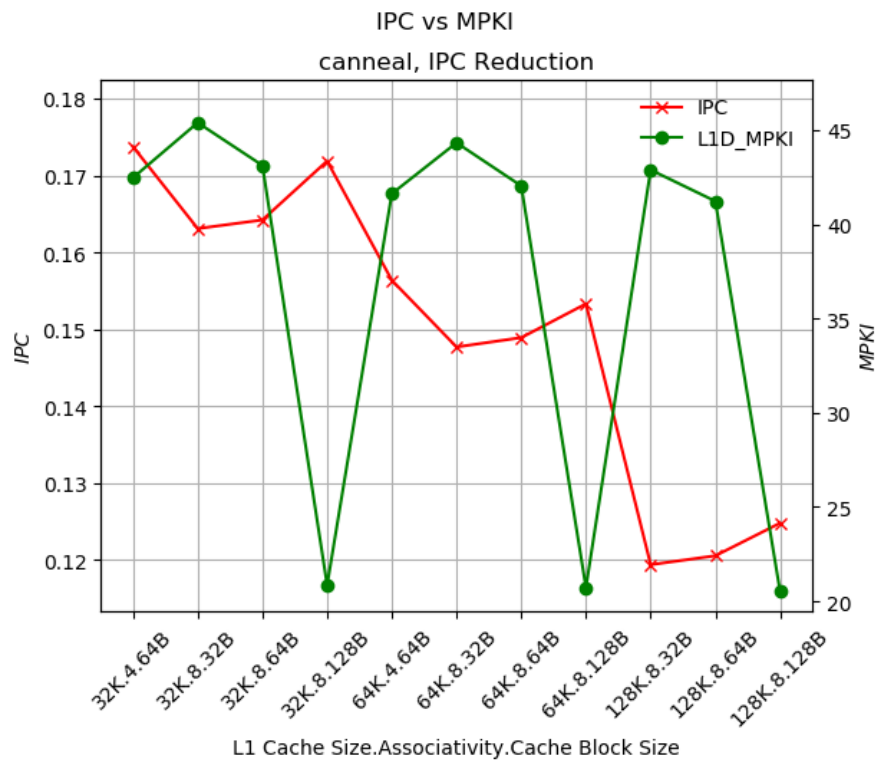
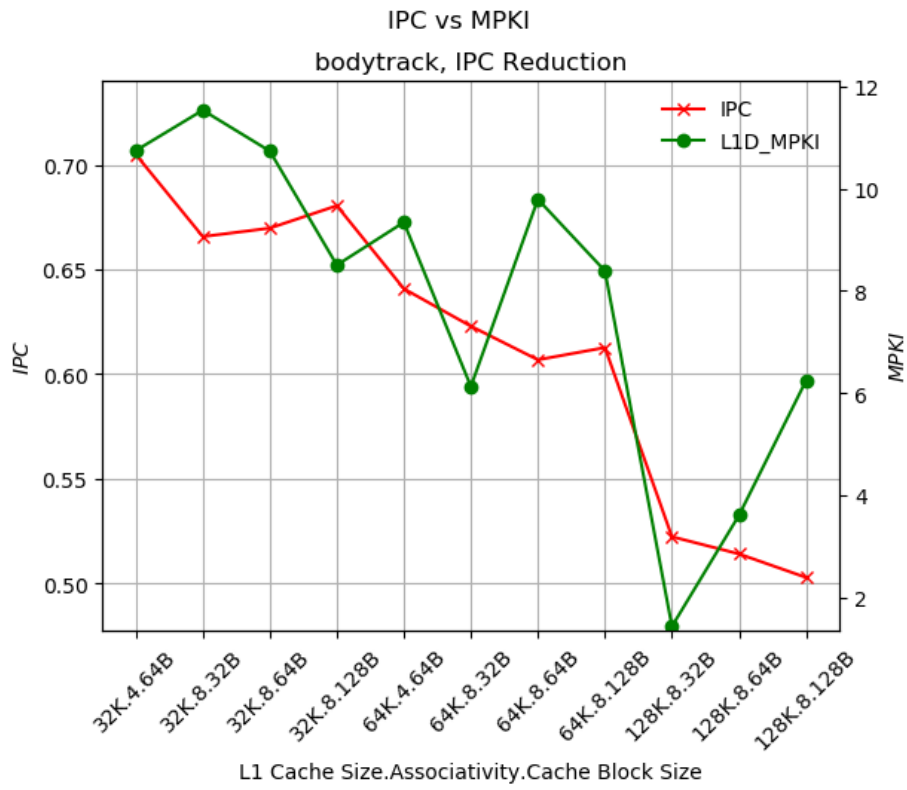
ΣΥΜΠΕΡΑΣΜΑΤΑ

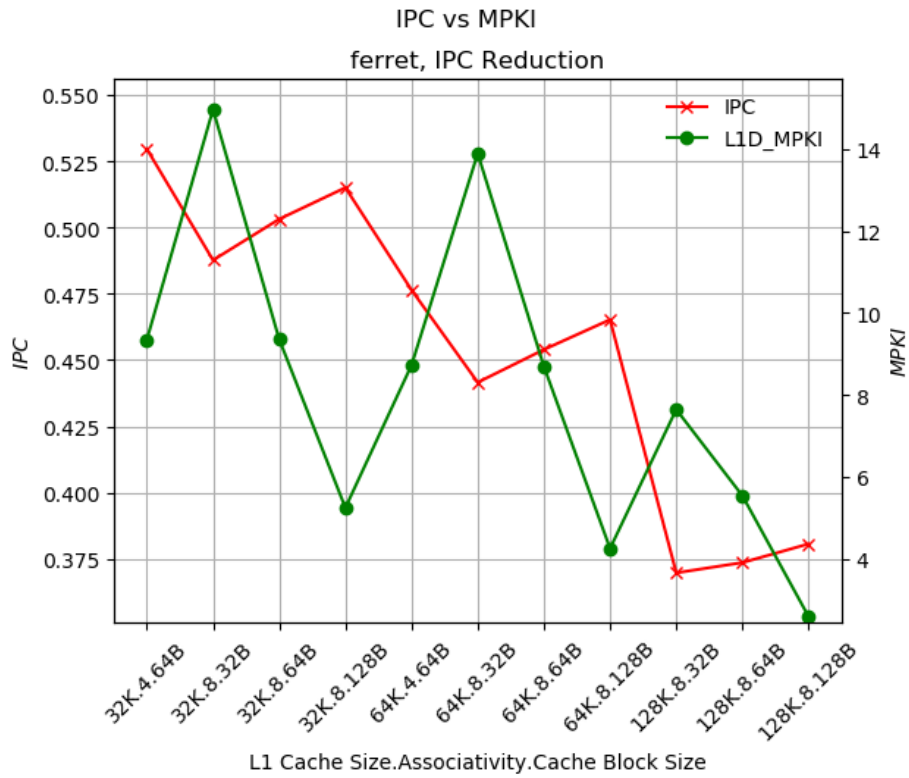
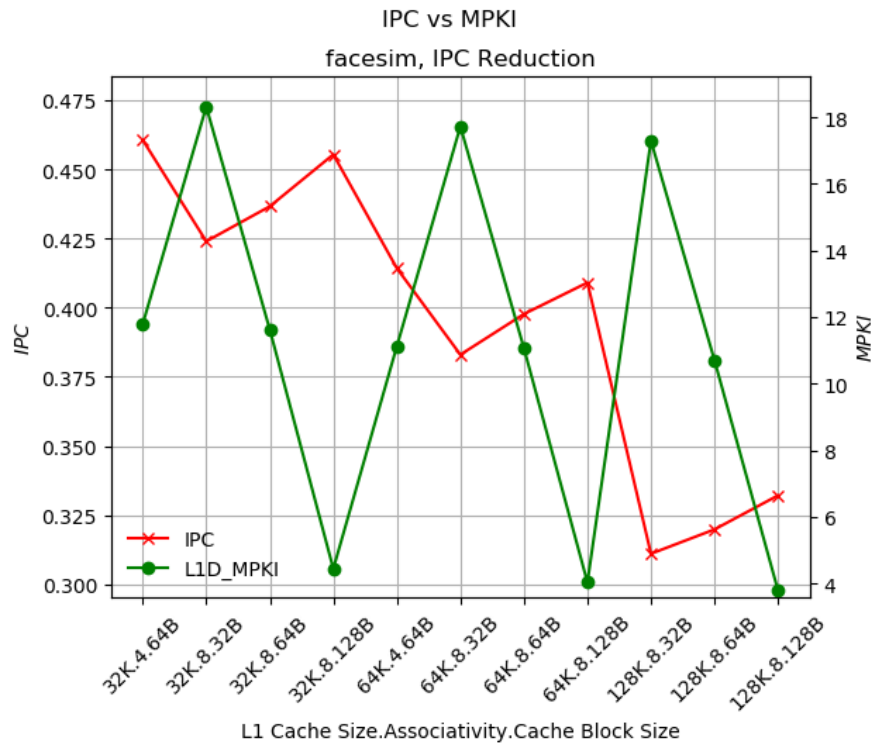
- 1) **L1 cache** : Λαμβάνοντας υπόψιν τα 10 benchmark , παρατηρούμε ότι τις περισσότερες φορές καθοριστικότερος παράγοντας είναι το block size . Ο ρόλος του cache size επίσης είναι σημαντικός, ενώ το associativity δεν επιδρά ιδιαίτερα στο αποτέλεσμα.
- 2) **L2 cache** : Από τα παραπάνω benchmark διαπιστώνουμε πως τον σημαντικότερο ρόλο διαδραματίζει είναι το block size, ενώ σημαντικό είναι και το cache size. Αντίθετα, το associativity δείχνει να μην επηρεάζει το αποτέλεσμα. Τελικά, για βέλτιστο αποτέλεσμα, χρειαζόμαστε όσο το δυνατόν μεγαλύτερο block και cache size.
- 3) **TLB** : Παρατηρούμε ότι για το TLB απαιτείται όσο το δυνατό μεγαλύτερο TLB size και associativity.
- 4) **Prefetching** : Συμπεραίνουμε πως, όταν αυξάνεται το prefetching βελτιώνεται και η απόδοση. Συνήθως η βελτίωση πραγματοποιείται με φθίνοντα ρυθμό.

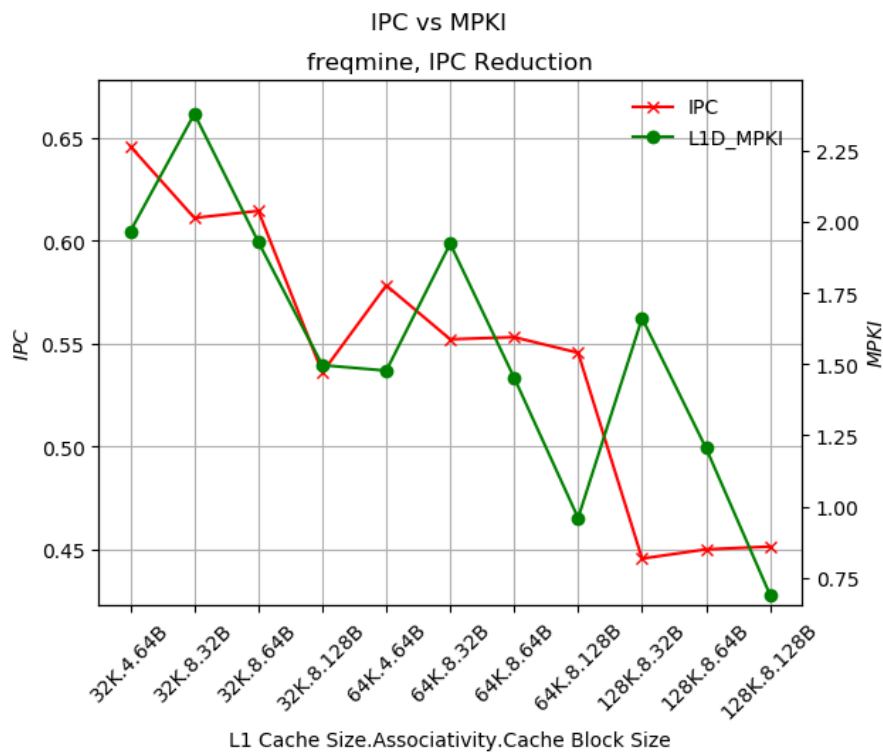
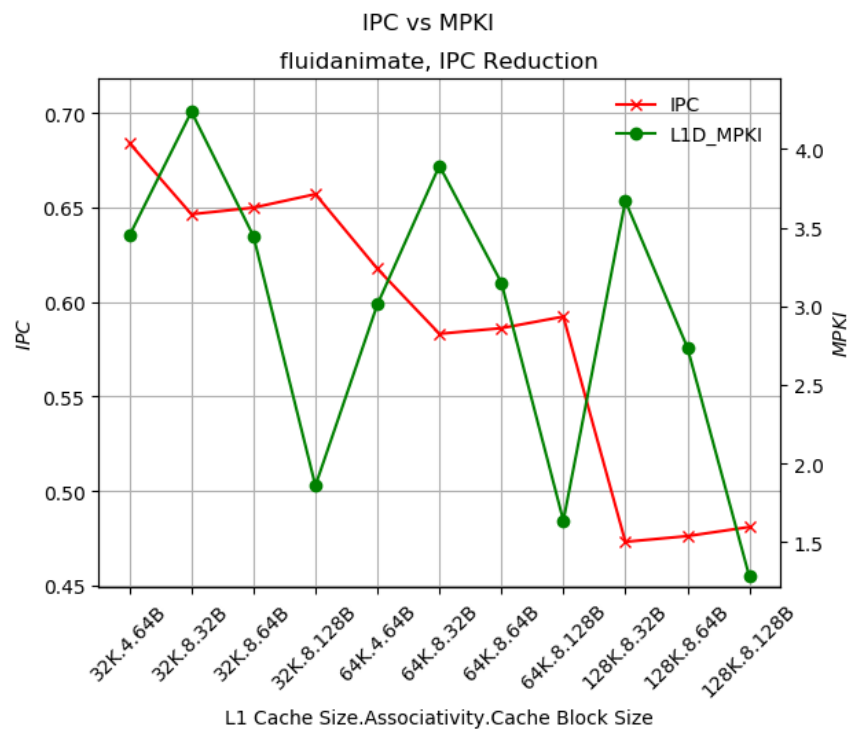
7.2 L1 cache

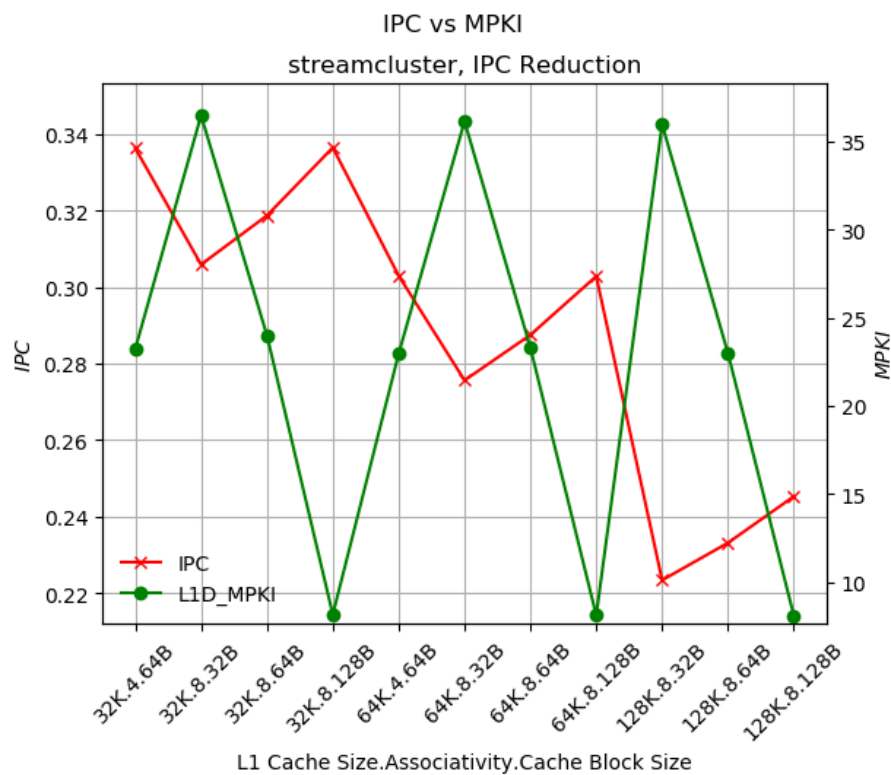
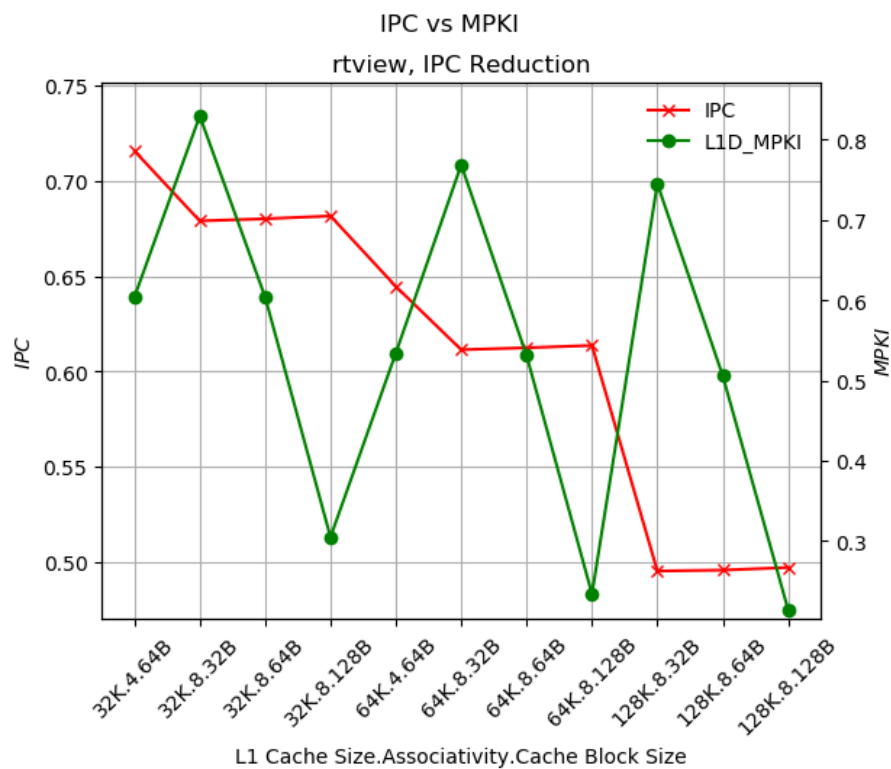
Θεωρώντας κάθε φορά ως αρχικό σημείο αναφοράς τη πρώτη προσομοίωση που εκτελέσαμε στο κάθε ένα benchmark, μελετάμε πως επηρεάζονται τα αποτελέσματα αν κάθε διπλασιασμός του associativity ή του μεγέθους προκαλεί αύξηση του κύκλου κατά 5% ή 10% αντίστοιχα.

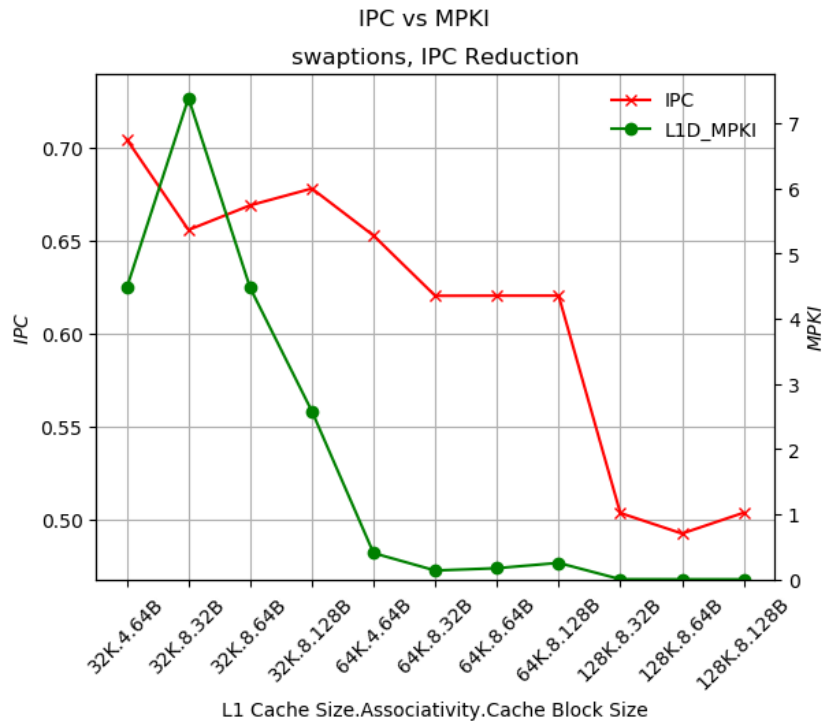






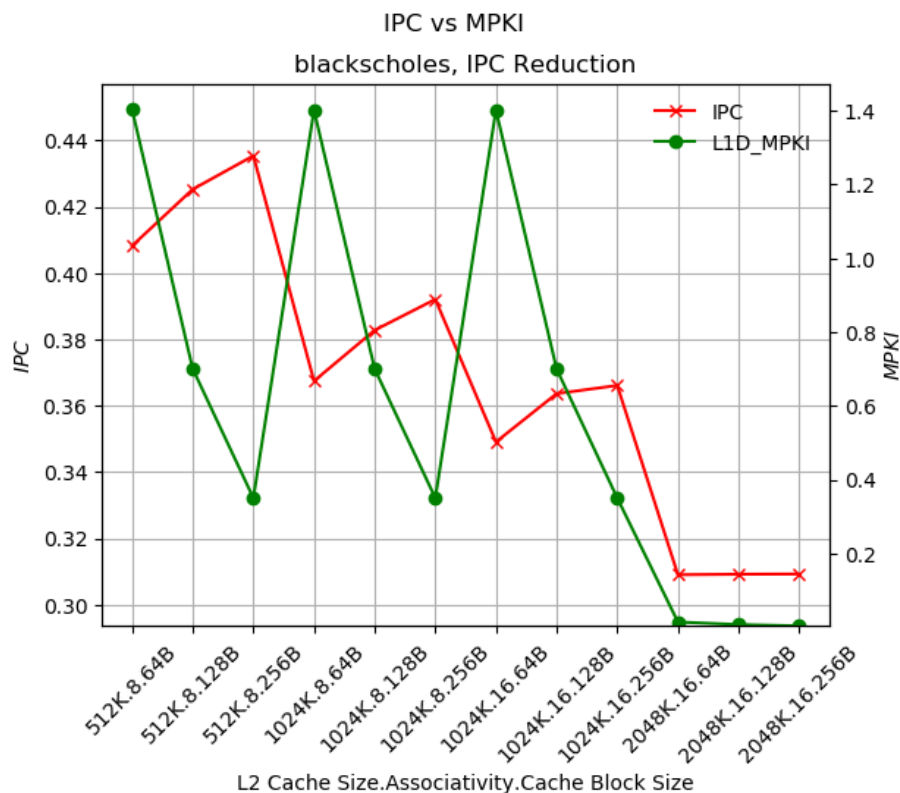


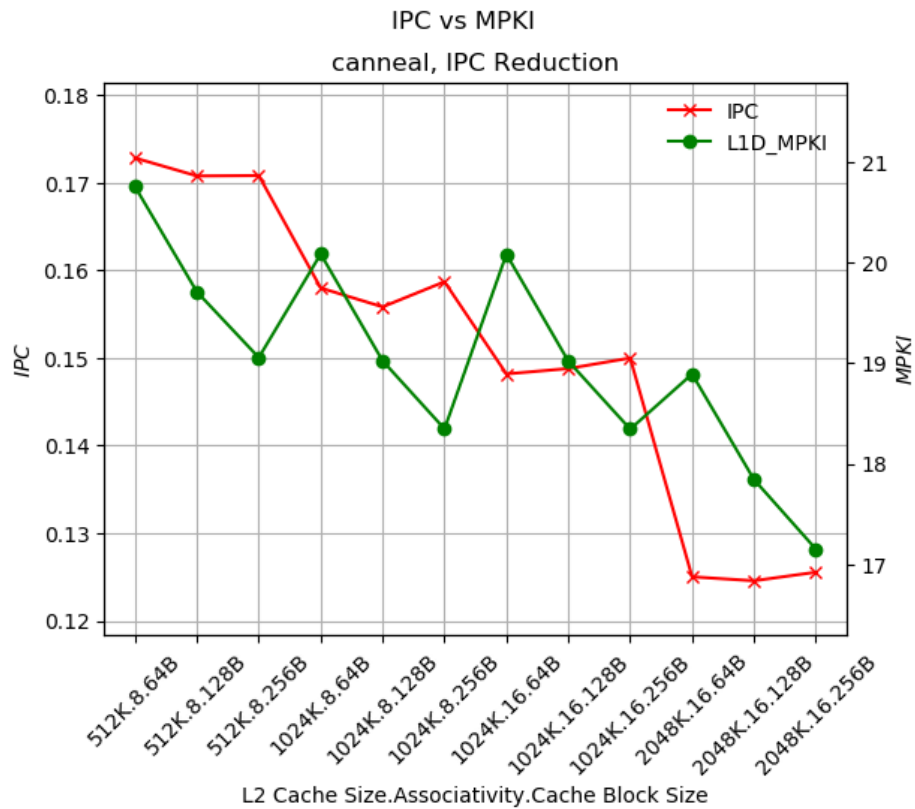
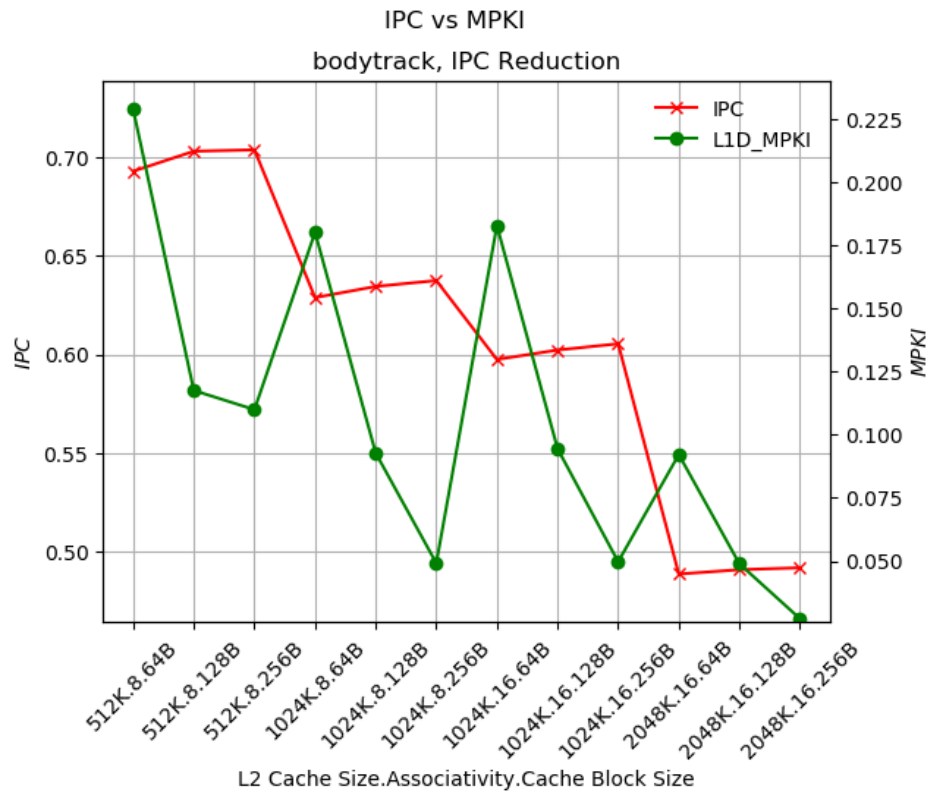


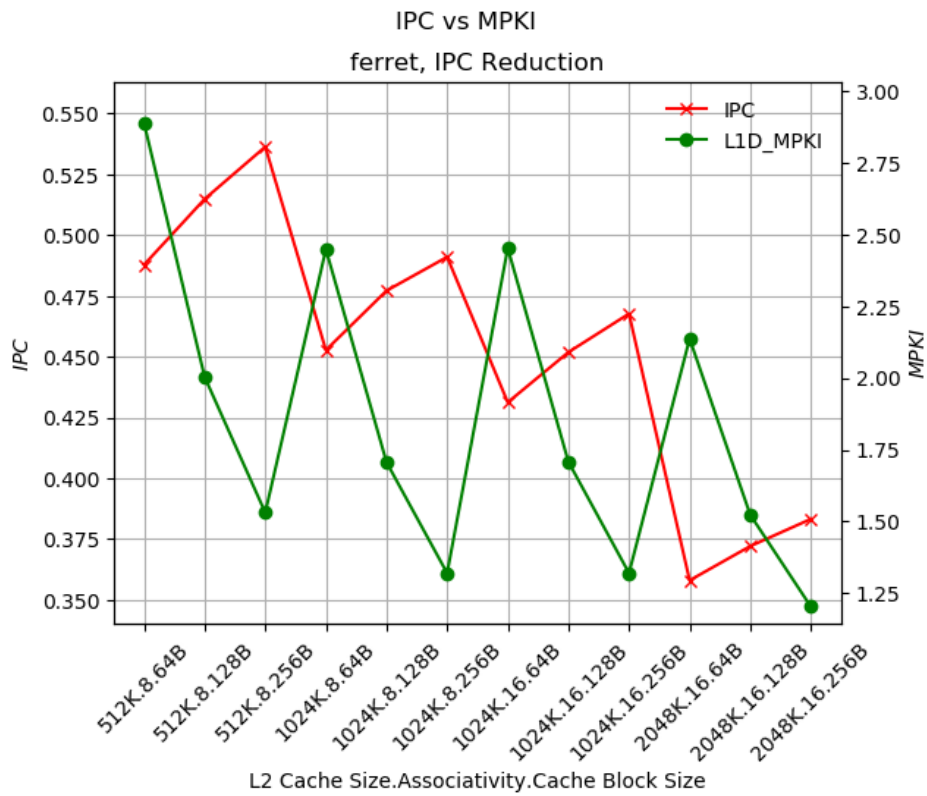
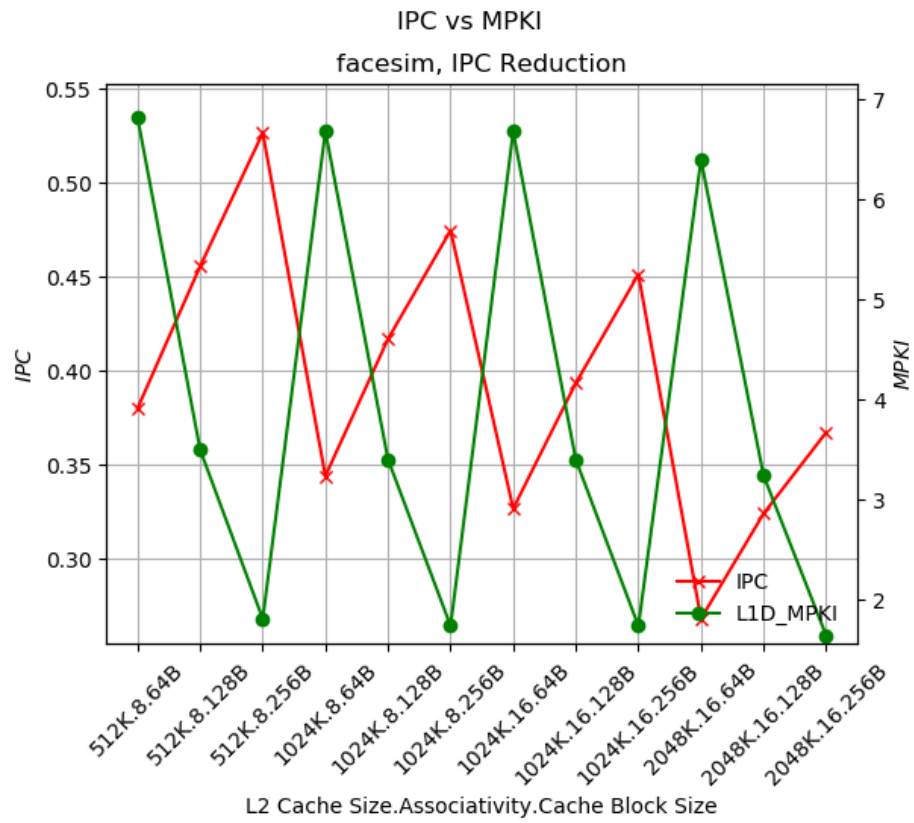


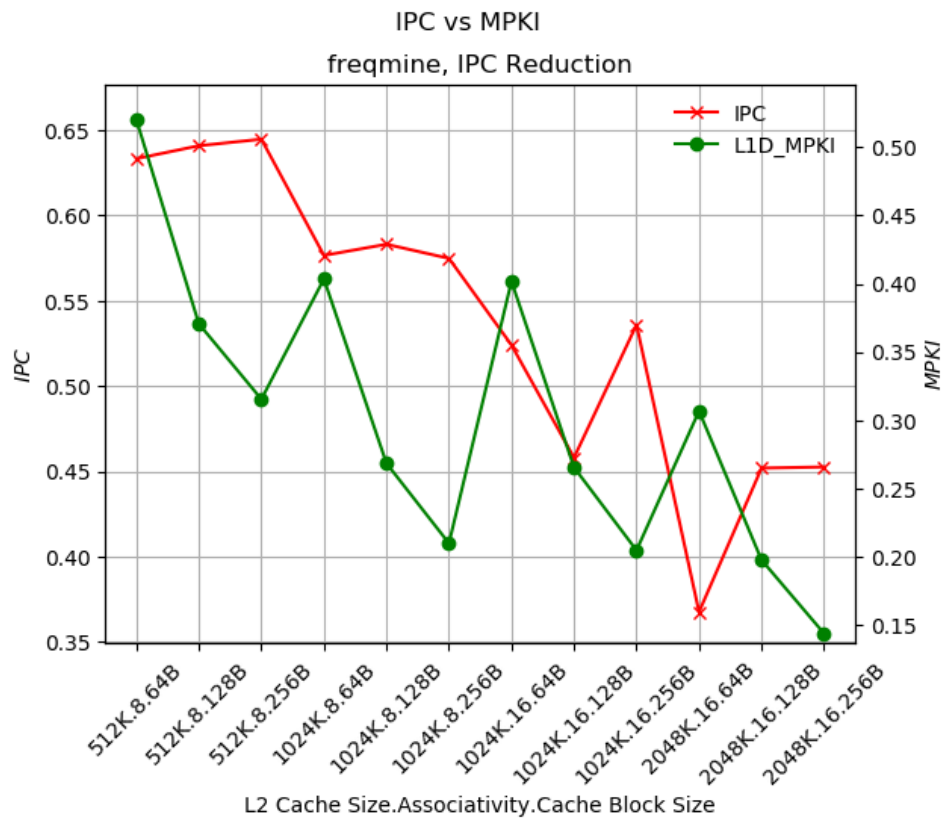
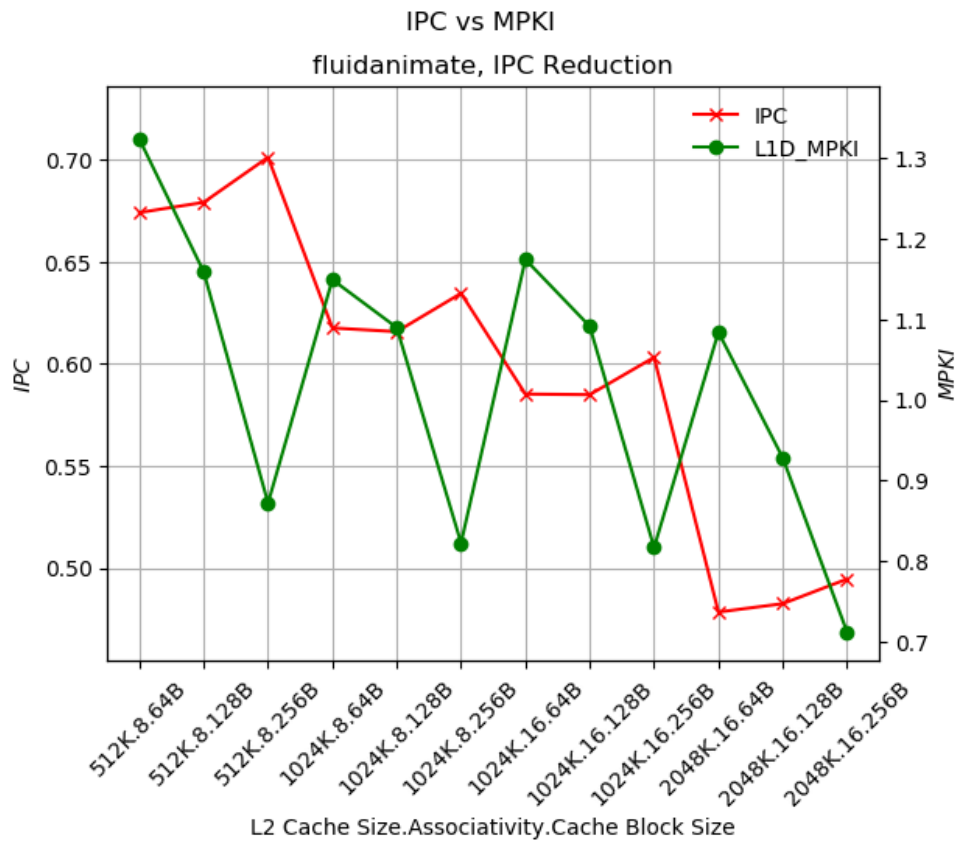
Συγκεντρωτικά, σε όλα τα benchmark, αν κάθε διπλασιασμός του associativity ή του μεγέθους προκαλεί αύξηση του κύκλου κατά 5% ή 10% αντίστοιχα, παρατηρούμε πως η απόδοση πέφτει ραγδαία, ειδικά κάθε φορά που αυξάνεται το cache size. Από την άλλη, η αύξηση του block size, αυξάνει λίγο και την απόδοση, αλλά ο σημαντικότερος παράγοντας είναι η cache size. Το συμπέρασμα αυτό εκλαμβάνεται από όλα τα γραφήματα.

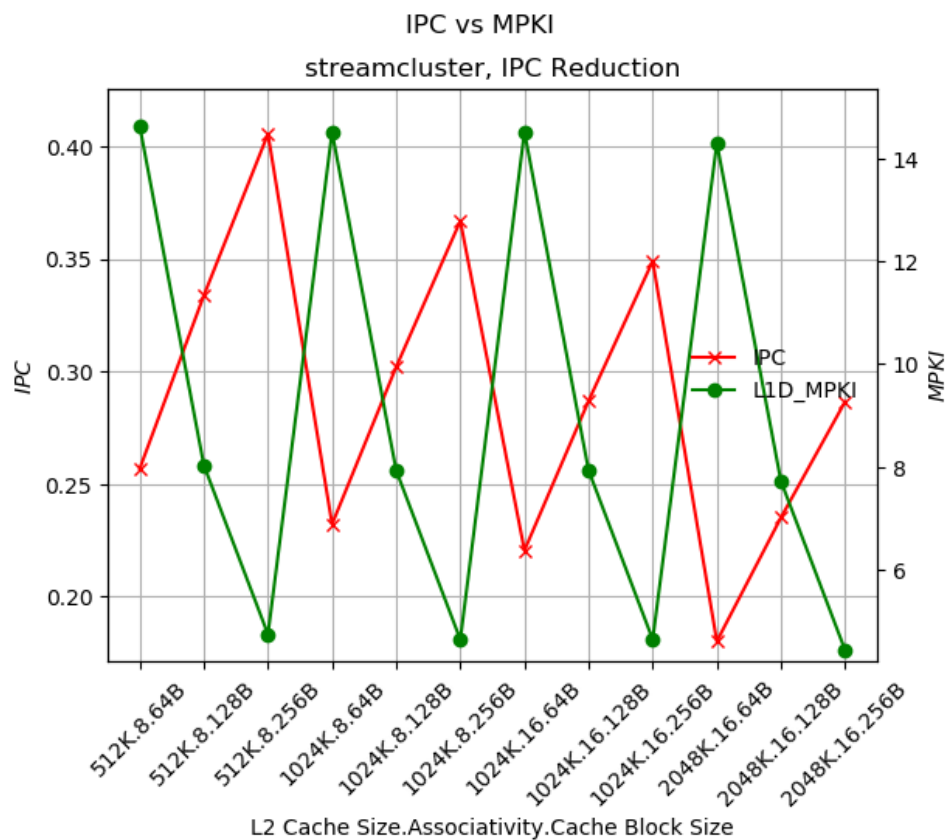
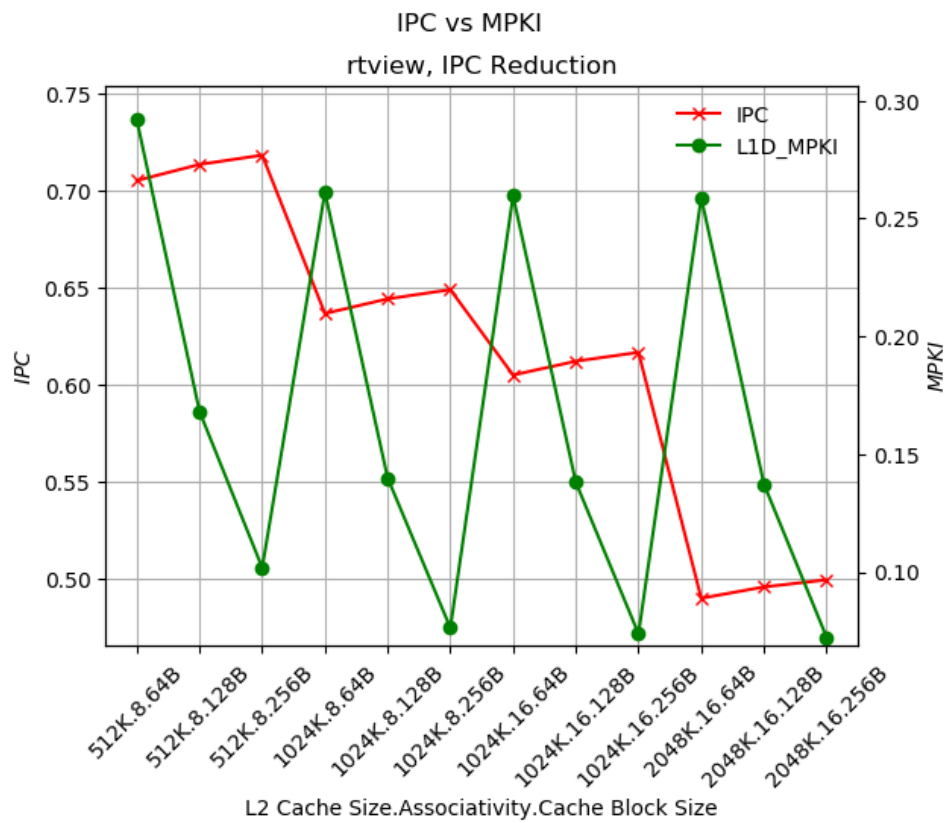
7.2 L2 cache

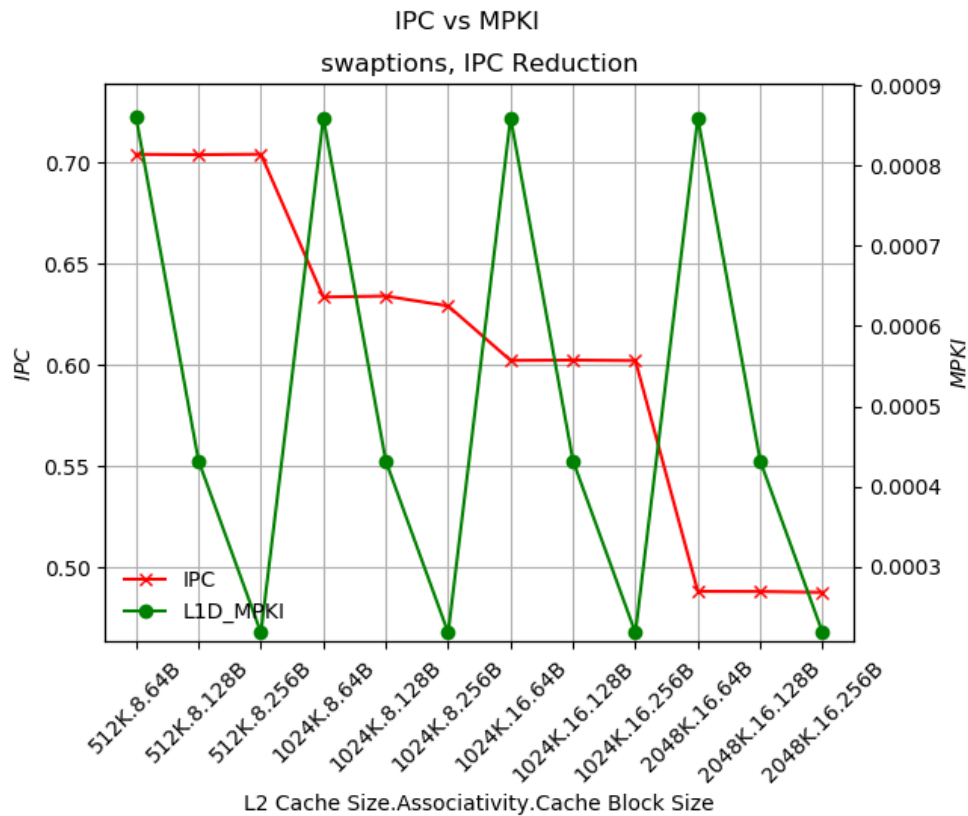






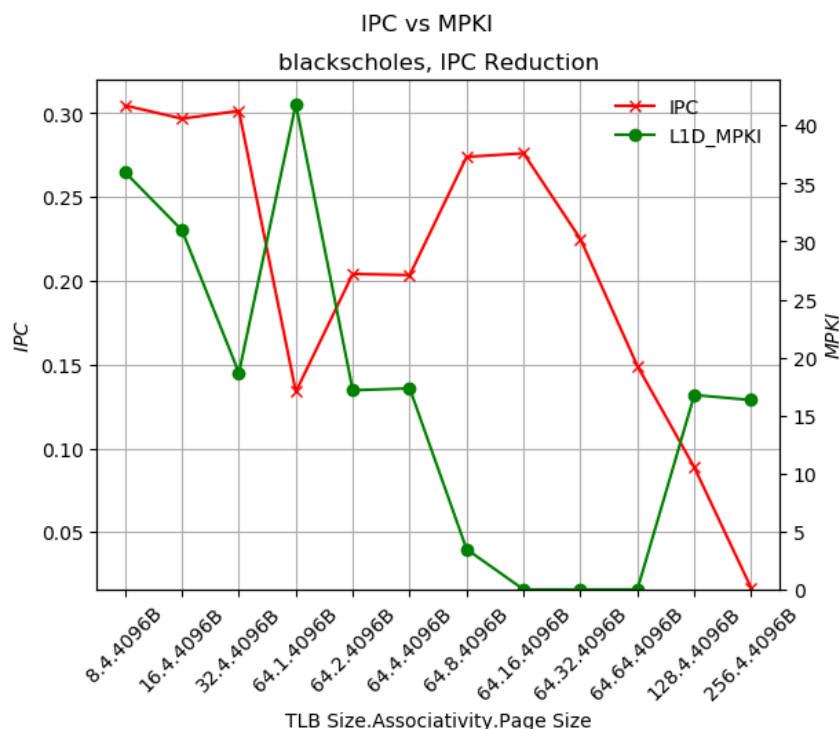


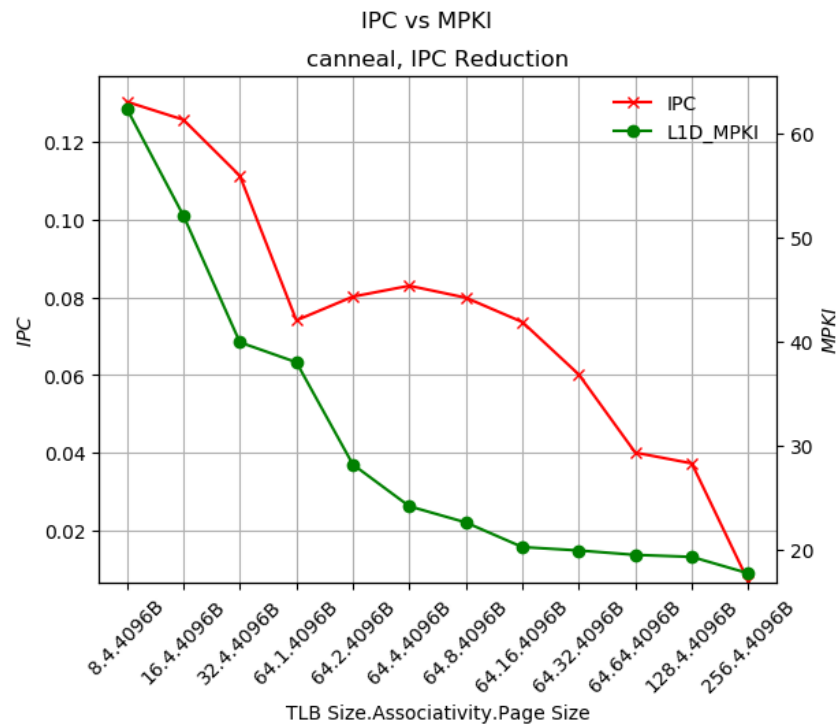
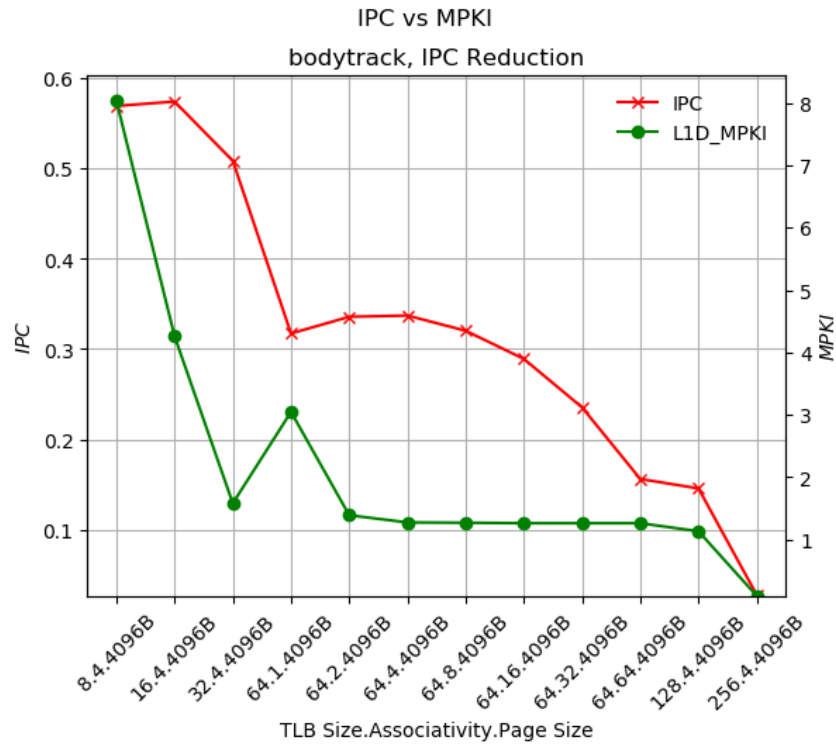


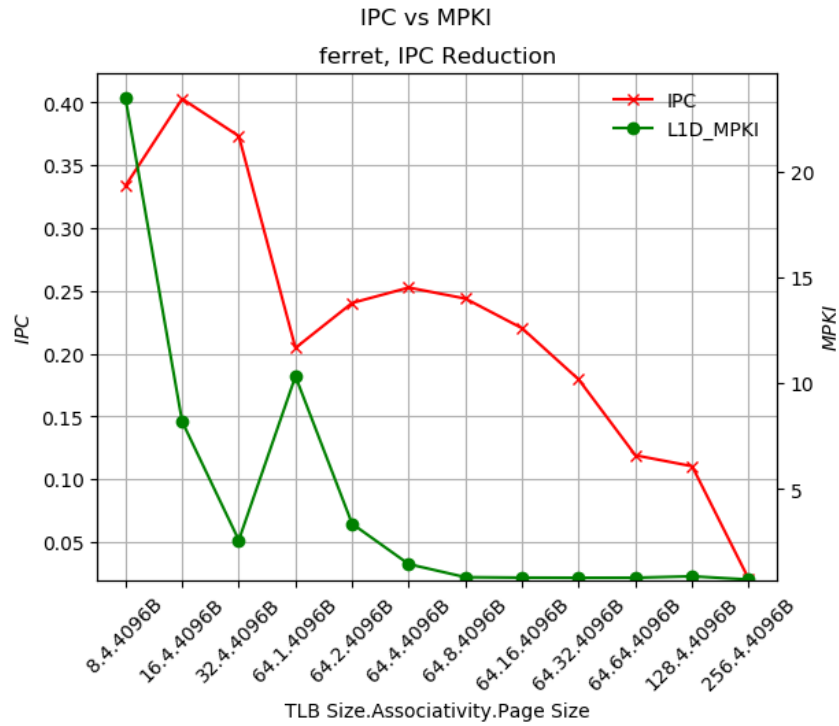
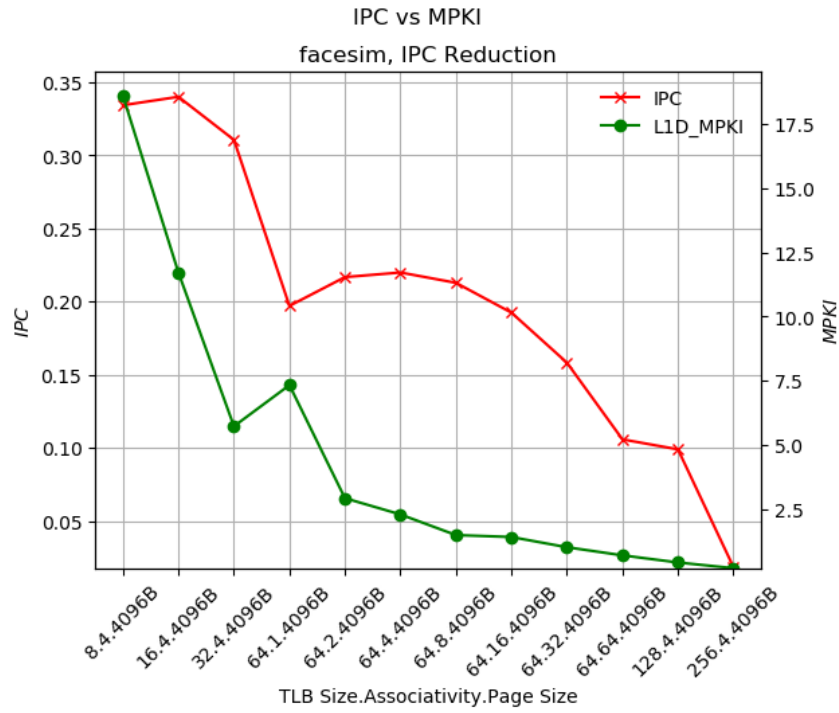


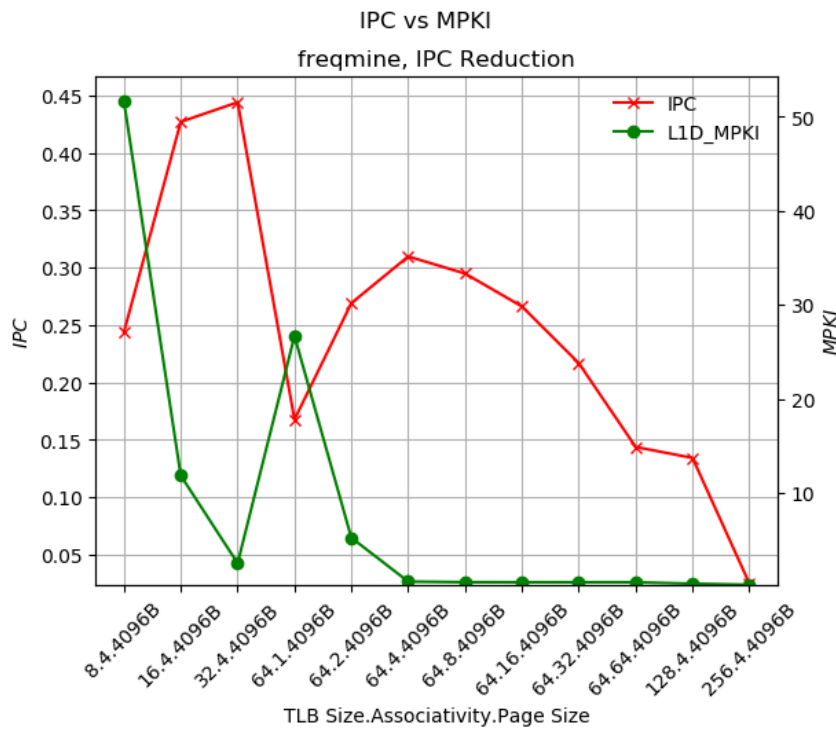
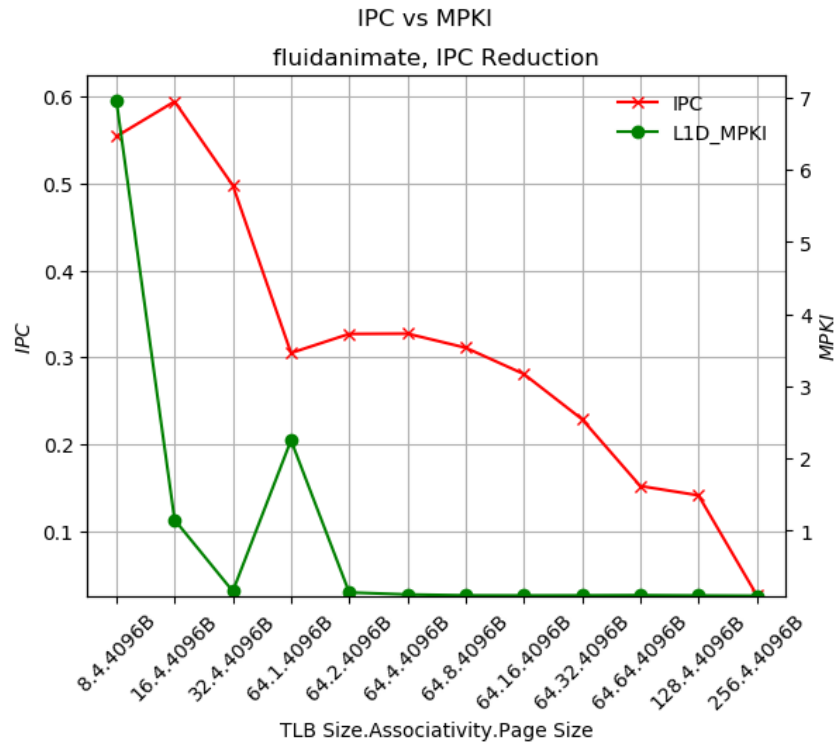
Ουσιαστικά, η βασική διαφορά που παρατηρούμε είναι ότι το cache size επηρεάζει σημαντικότερα την απόδοση και την ρίχνει απότομα κάθε φορά που αυξάνεται. Η βασική τοπολογία των διαγραμμάτων είναι ίδια με εκείνη της πρώτης προσομοίωσης, δηλαδή υπάρχουν οι ίδιες “κορυφές”. Το μόνο που αλλάζει είναι το ύψος κάθε κορυφής, που μειώνεται εξαιτίας της αύξησης του cache size και σε μικρότερο βαθμό του associativity.

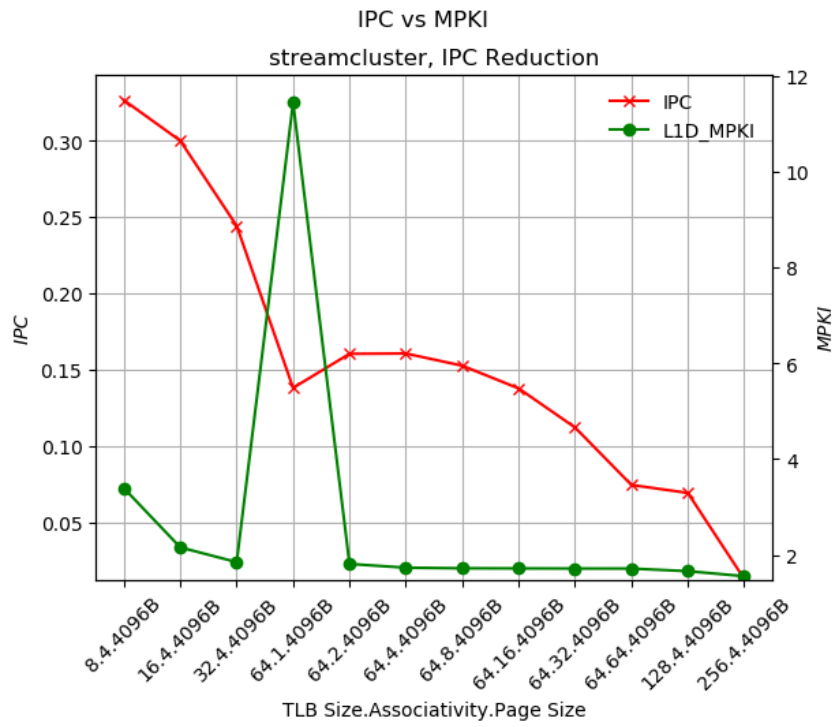
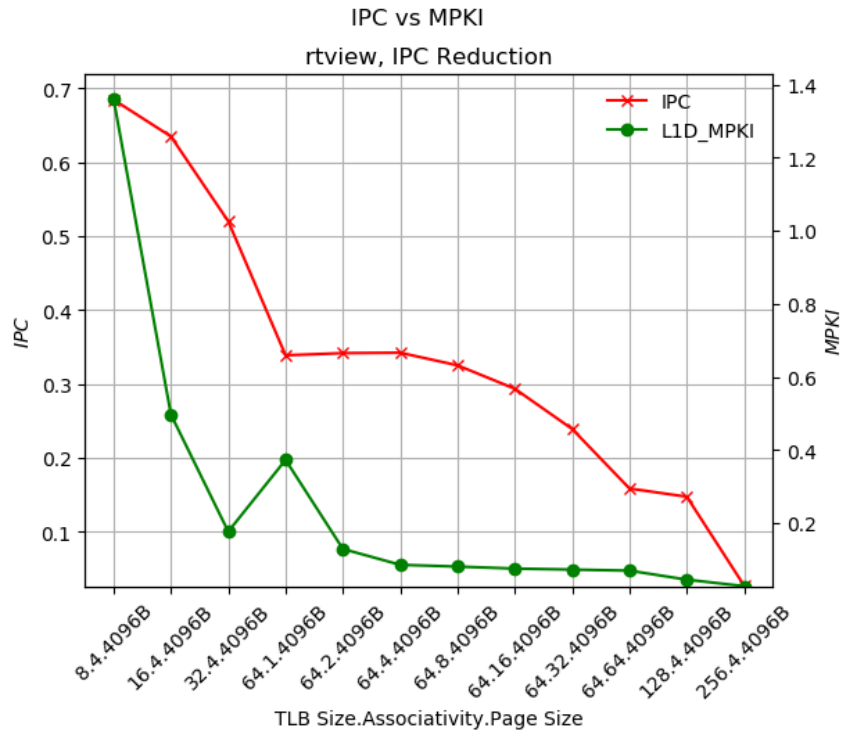
7.2 TLB

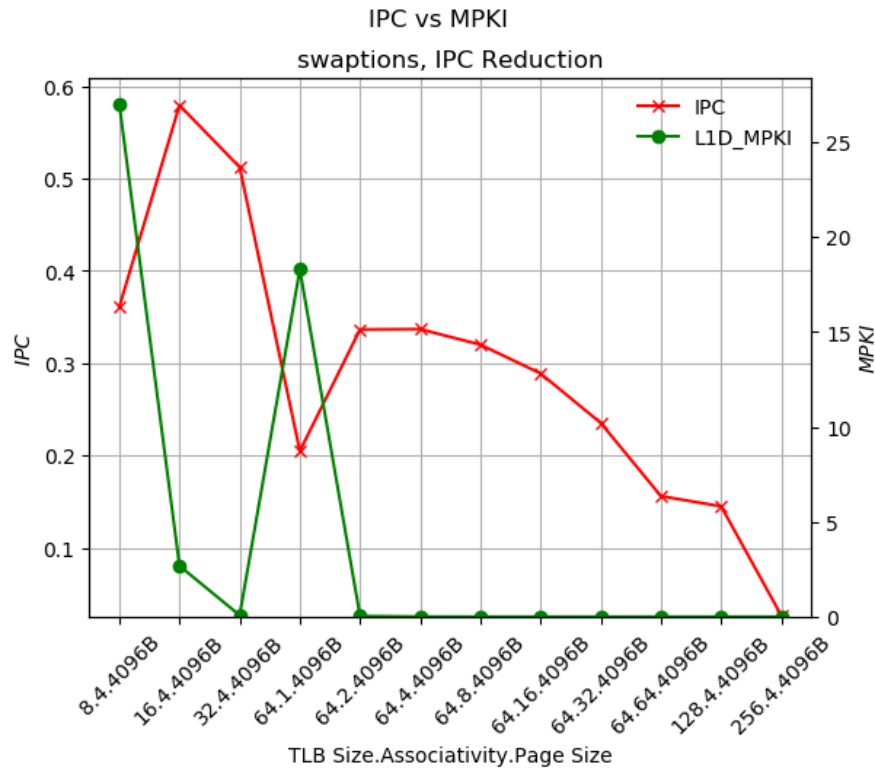












Όπως και για τις L1 και L2 caches, έτσι και για το TLB, η βασική διαφορά που παρατηρείται σε σχέση με την πρώτη προσομοίωση, είναι η ραγδαία πτώση της απόδοσης κάθε φορά που αυξάνεται το cache size. Βέβαια, στο TLB η σημαντική πτώση οφείλεται και στην αύξηση του associativity, σε αντίθεση με τις L1 και L2 caches.

ΣΥΜΠΕΡΑΣΜΑ

Η πραγματική απόδοση μειώνεται σε σημαντικό βαθμό όταν αυξάνονται οι τιμές των παραμέτρων, ιδίως του cache size και του associativity.