ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΜΜΥ

Προηγμένα Θέματα
Αρχιτεκτονικής Υπολογιστών

1η Άσκηση
Ακ. έτος 2019-2020

Σκουρτσίδης Γιώργος
Α.Μ. : 03114307

14 Απριλίου 2020

Εισαγωγή

Σκοπός της άσκησης αυτής ήταν η χρήση και εξοικείωση με το λογισμικό PIN της Intel, σε συνδυασμό με τη μελέτη της επίδρασης διαφόρων παραμέτρων των μνημών L1, L2 και TLB cache στην απόδοση συγκεκριμένων benchmarks.Επιπλέον,προσομοιώσαμε προανάκληση (prefetching) στην L2 cache.

Γενικές Παρατηρήεις

Περιμένουμε πως η αύξηση του μεγέθους του μπλοκ θα οδηγήσει σε μείωση των compulsory misses και σε πιθανή αύξηση της απόδοσης.Γνωρίζουμε όμως πως υπερβολικά μεγάλο block size μπορεί να οδηγήσει σε αύξηση των conflict misses,εξαιτίας του λιγοστού αριθμού blocks. Επιπροσθέτως,η αύξηση της χωρητικότητας και το υψηλότερο associativity θεωρητικά πρέπει να οδηγήσουν σε μείωση των capacity και conflict misses αντίστοιχα.

Για όλες τις προσομοιώσεις κρατάμε σταθερές τις εξής παραμέτρους:

1. TLB hit: 0 cycles (η πρόσβαση πραγματοποιείται παράλληλα με την L1 cache)
2. TLB miss: 100 cycles
3. L1 hit: 1 cycle
4. L2 hit: 15 cycles
5. Main memory access: 250 cycles

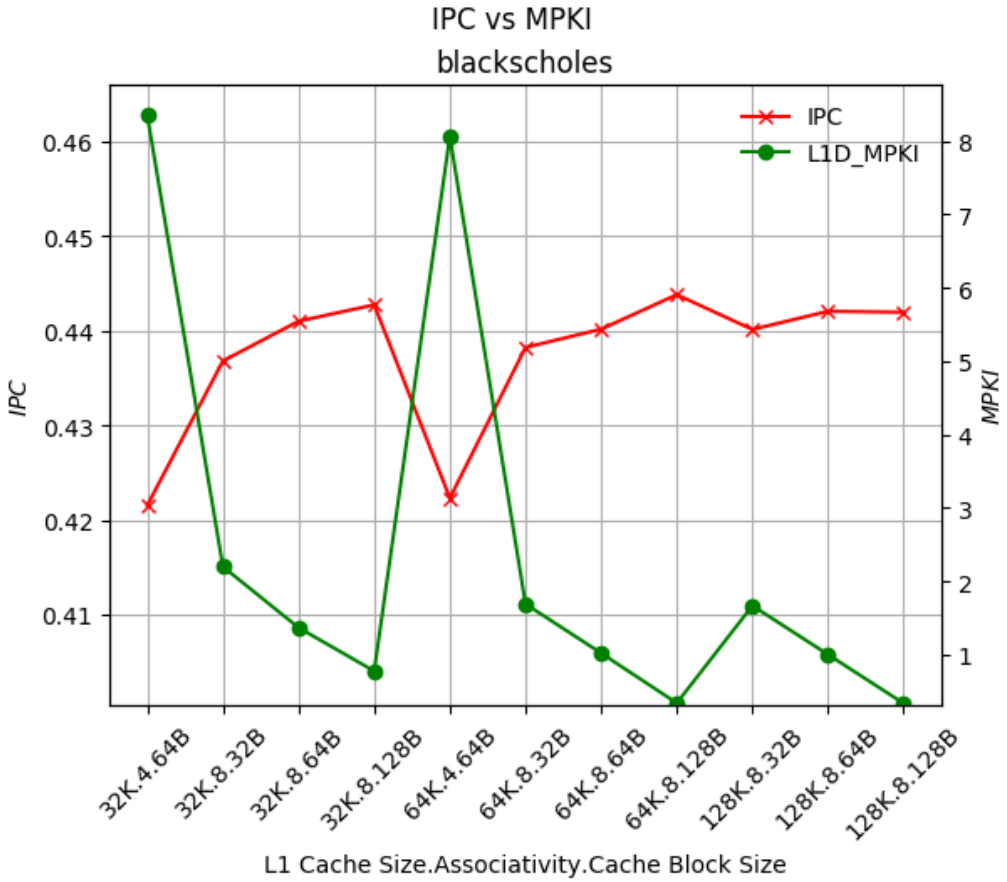
1 Ζητούμενο πρώτο

1.1 L1 cache

Για να αξιολογήσουμε μετρικές της L1 cache,κρατήθηκαν σταθερές παράμετροι της L2 cache και του TLB χρησιμοποιώντας διαφορετικά configuration για το μέγεθος,το associativity και το block size της L1 cache ως εξής:

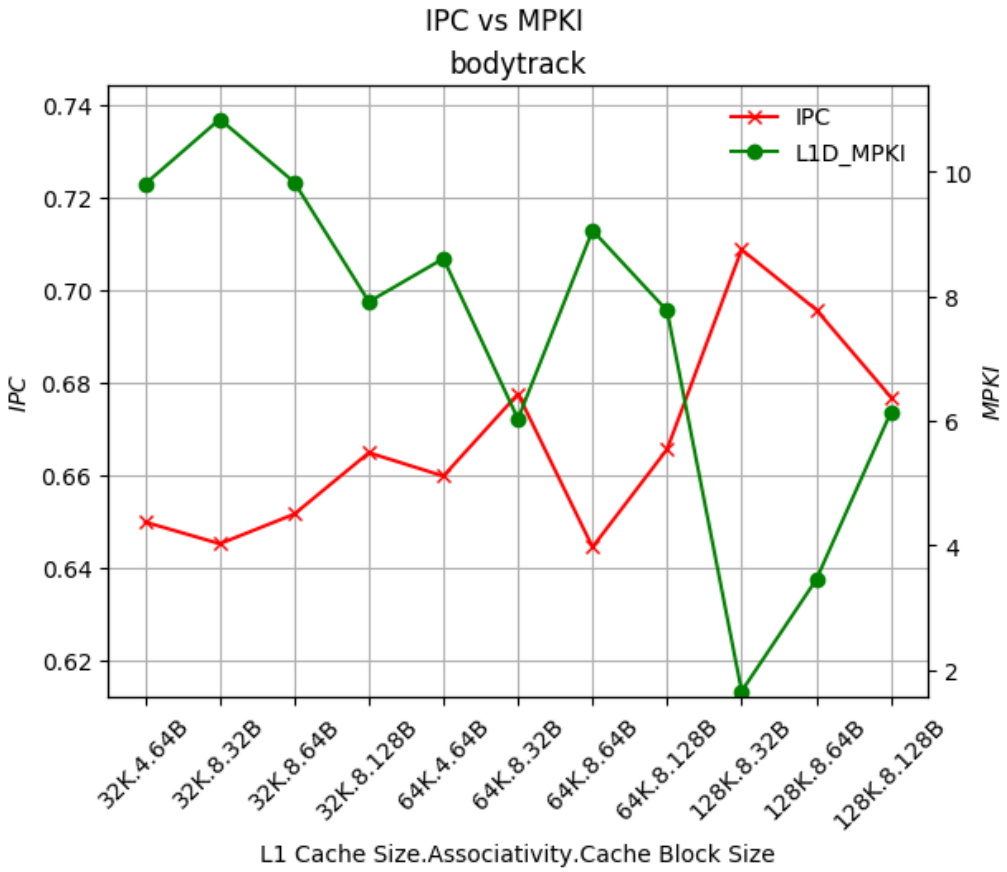
L1 size	L1 associativity	L1 cache block size
32KB	4	64B
32KB	8	32B, 64B, 128B
64KB	4	64B
64KB	8	32B, 64B, 128B
128KB	8	32B, 64B, 128B

1.1.1 blackscholes



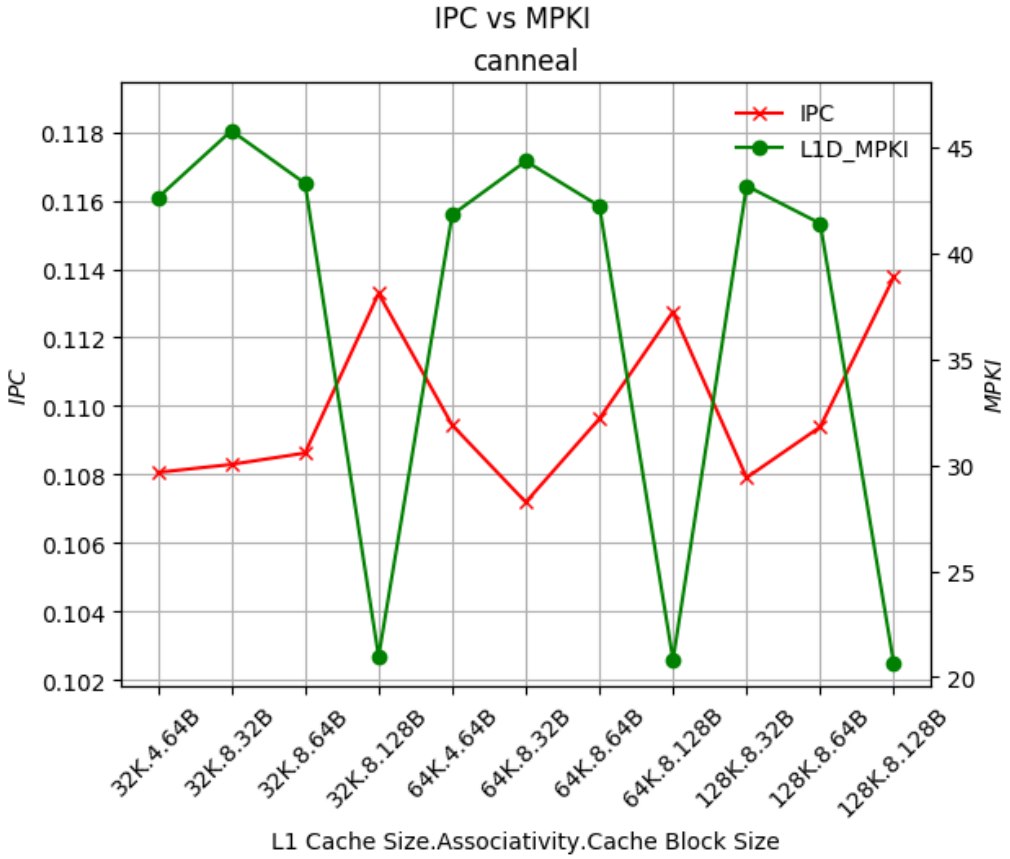
Παρατηρούμε πως καθώς αυξάνεται η χωρητικότητα της cache,το IPC αυξάνεται.Βέλτιστοι συνδυασμοί είναι οι (64,8,128),(128,8,128) , με τους συνδυασμούς (128,8,64),(32,8,128) και (64,8,64) να βρίσκονται αρκετά κοντά.Ο κοινός παρανομαστής όλων των συνδυασμών είναι το associativity που φαίνεται πως ειναι ο σημαντικότερος παράγοντας σε αυτην την προσομοίωση.Το block size δεν έχει μεγάλη επίδραση.Όλα τα παραπάνω μας οδηγούν στο συμπέρασμα πως υπάρχουν conflict misses.Τελικά επιλέγουμε το συνδυασμό (64,8,128) ή αν θέλουμε περισσότερη οικονομία τον (32,8,128).

1.1.2 bodytrack



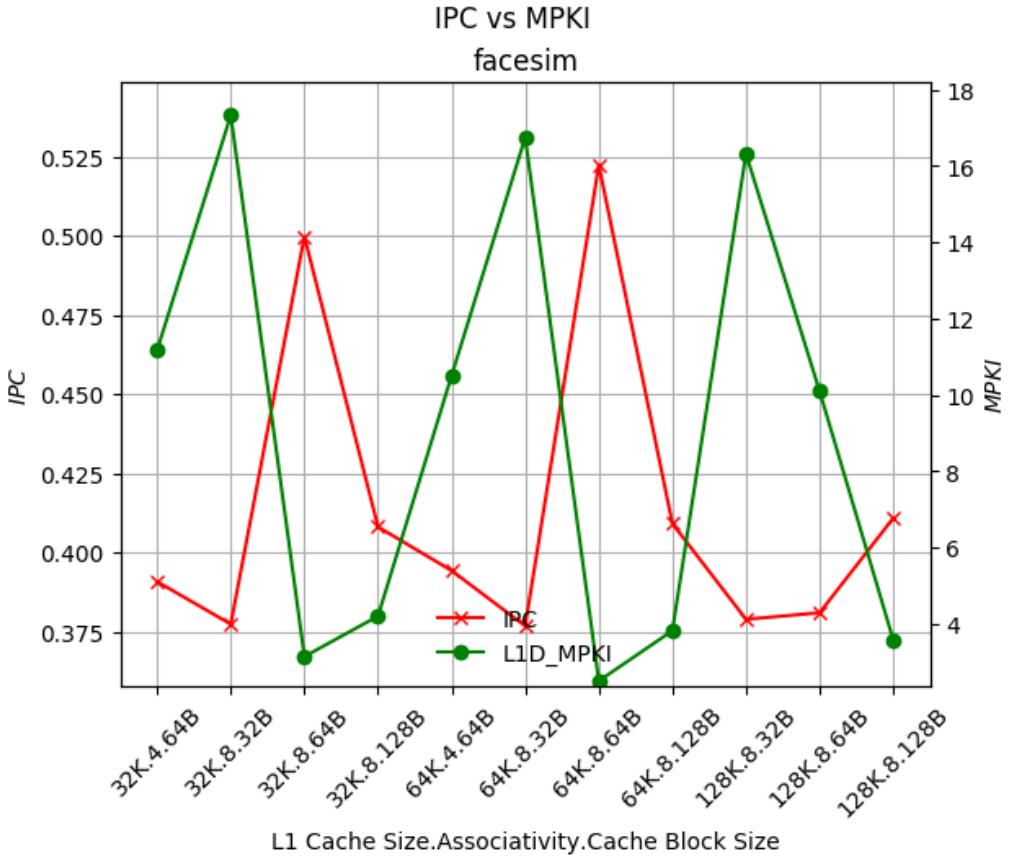
Στο L1 bodytrack ο βέλτιστος συνδυασμός είναι ο (128,8,32). Αρχικά παρατηρούμε πως για μικρό μέγεθος cache η αύξηση του block size επιδρά θετικά αφού (32,8,32) < (32,8,64) < (32,8,128). Όμως για τις μεγαλύτερες cache η αύξηση του block size παύει πια να έχει θετική επίδραση. Η μεταβολή του cache size πρόκειται για τον σημαντικότερο παράγοντα και τελικά η αύξηση σε 128K επέφερε θεαματική βελτίωση,σε συνδυασμό με το μικρό μέγεθος block.Άρα σε αυτήν την προσομοίωση υπήρχαν πολλαπλά capacity misses στις μικρότερες cache.Η υπερβολική αύξηση του block size στη συνέχεια οδηγεί σε conflicti misses.

1.1.3 canneal



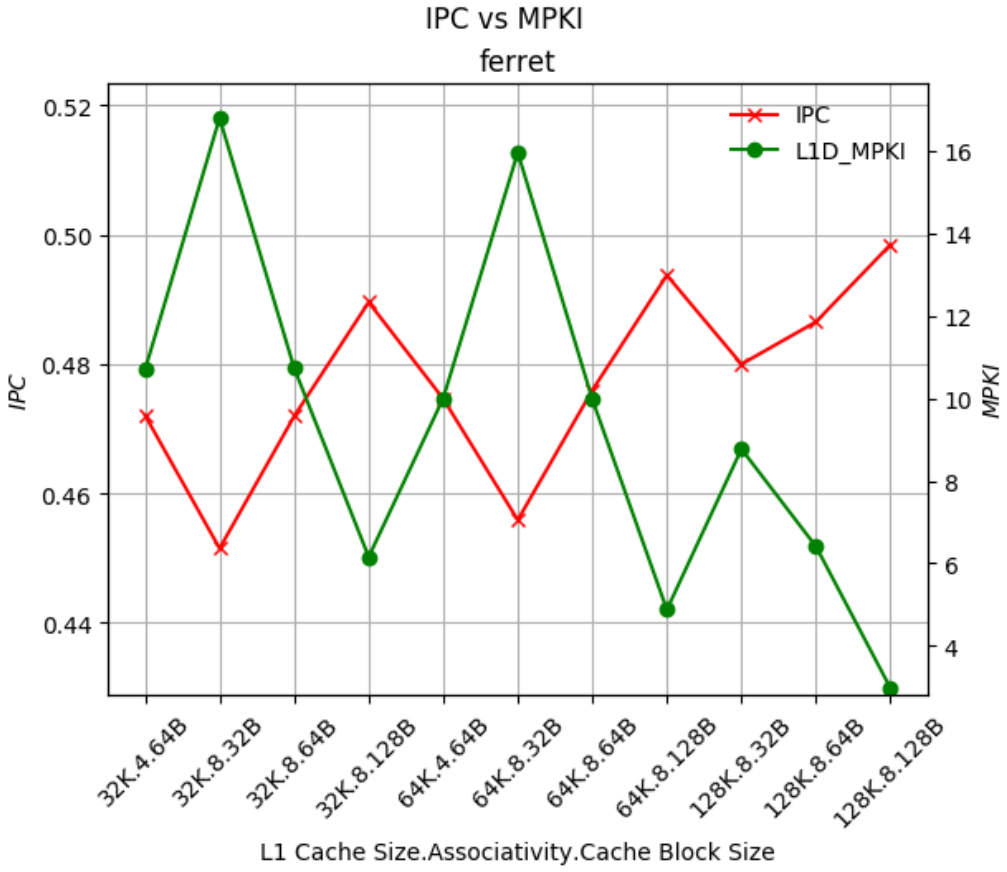
Τρεις συνδυασμοί δείχνουν να είναι βέλτιστοι , οι (32,8,128) , (64,8,128) και (128,8,128) . Είναι προφανές πως ο καθοριστικότερος παράγοντας είναι το block size , καθώς όλοι οι βέλτιστοι συνδυασμοί παρατηρούνται όταν το block size είναι 128 . Το cache size και το associativity έχουν αμελητέες επιδράσεις.Μάλλον υπάρχει ισχυρό spatial locality στη συγκεκριμένη προσομοίωση το οποίο πρέπει να εκμεταλλευτούμε,αυξάνοντας το μέγεθος του block.Τέλος,επιλέγεται ο συνδυασμός (32,8,128) ως βέλτιστος για λόγους οικονομίας.

1.1.4 facesim



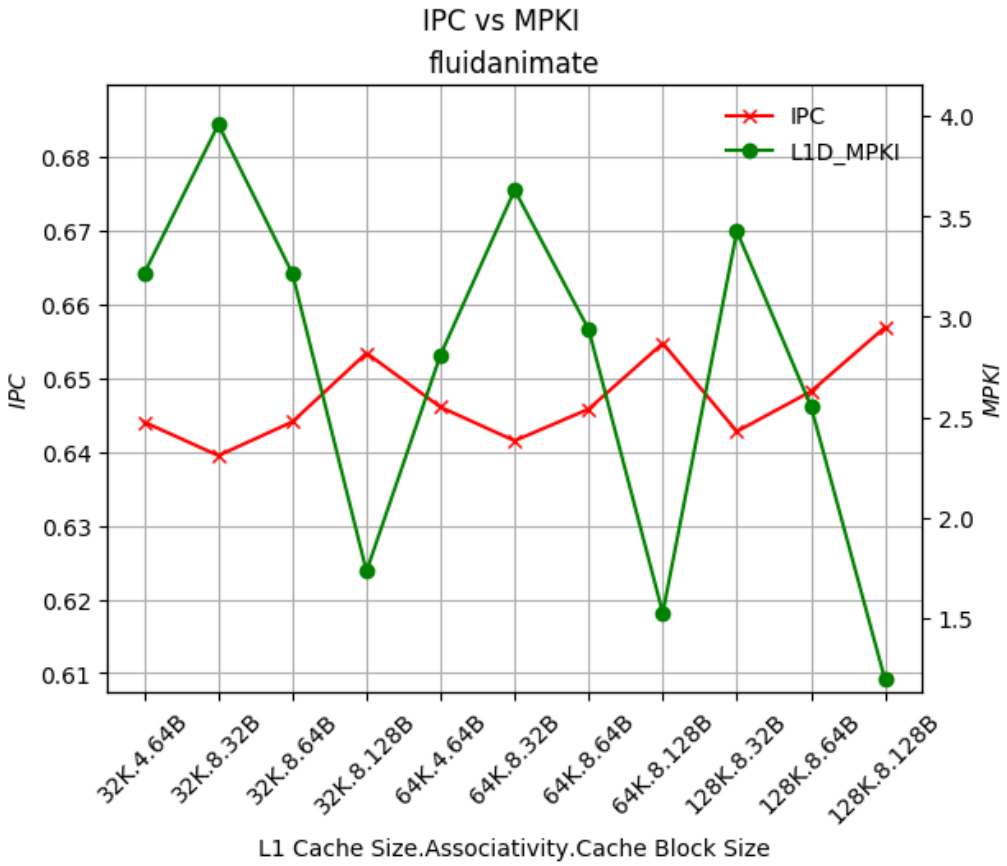
Η αύξηση του block size δείχνει να είναι ο καθοριστικός παράγοντας,με το associativity και το cache size να έχουν καθόλου και μικρές επιδράσεις αντίστοιχα.Η αύξηση απο 32B σε 64B στο block size αυξάνει θεαματικά το ipc και ρίχνει επίσης θεαματικά το mpki,όμως η υπερβολική του αύξηση σε 128B επιφέρει αρνητικές επιδράσεις.Βέλτιστος συνδυασμός είναι ο (64,8,64) και ακολουθεί ο (32,8,64).

1.1.5 ferret



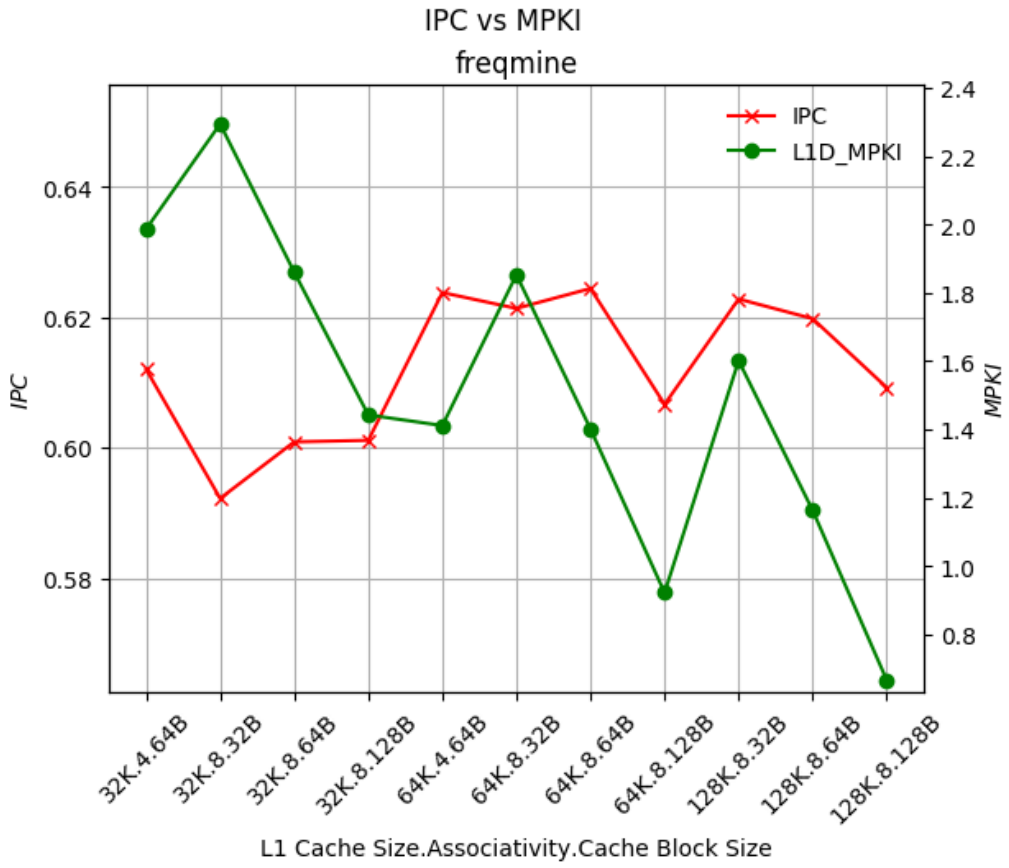
Σημαντικότερος παράγοντας είναι και πάλι η αύξηση του block size,αφού βλέπουμε πως όσο μεγαλύτερο,τόσο το καλύτερο. Ο ρόλος του associativity είναι περιορισμένος,ενώ η αύξηση του cache size επιφέρει θετικές επιδράσεις. Βέλτιστος συνδυασμός ο (128,8,128)

1.1.6 fluidanimate



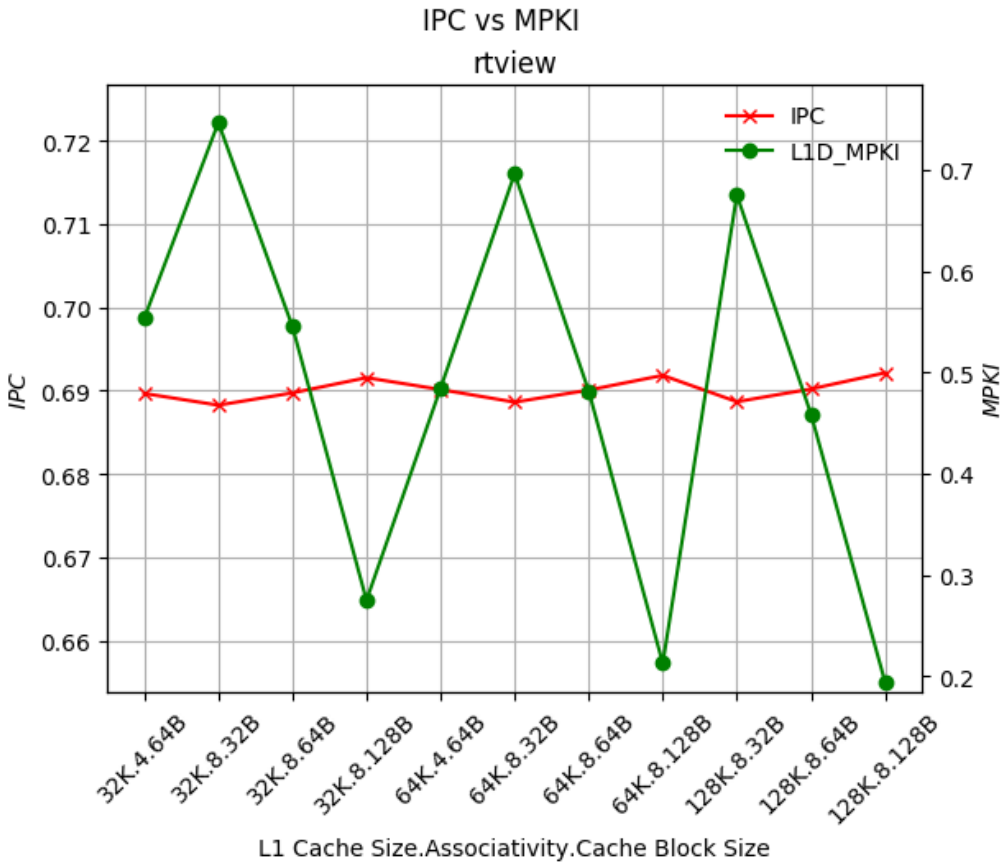
Σημαντικότερος παράγοντας είναι το block size.Το cache size , το οποίο επίσης όσο αυξάνεται βελτιώνει την απόδοση,έχοντας όμως δευτερεύουσα βαρύτητα.Το assiciativity δεν φαίνεται να επηρεάζει το αποτέλεσμα. Ανεξάρτητα από τις επιλογές των παραμέτρων , παρατηρούμε πως το IPC είναι αρκετά σταθερό. Βέλτιστος συνδυασμός είναι ο (128,8,128).

1.1.7 freqmine



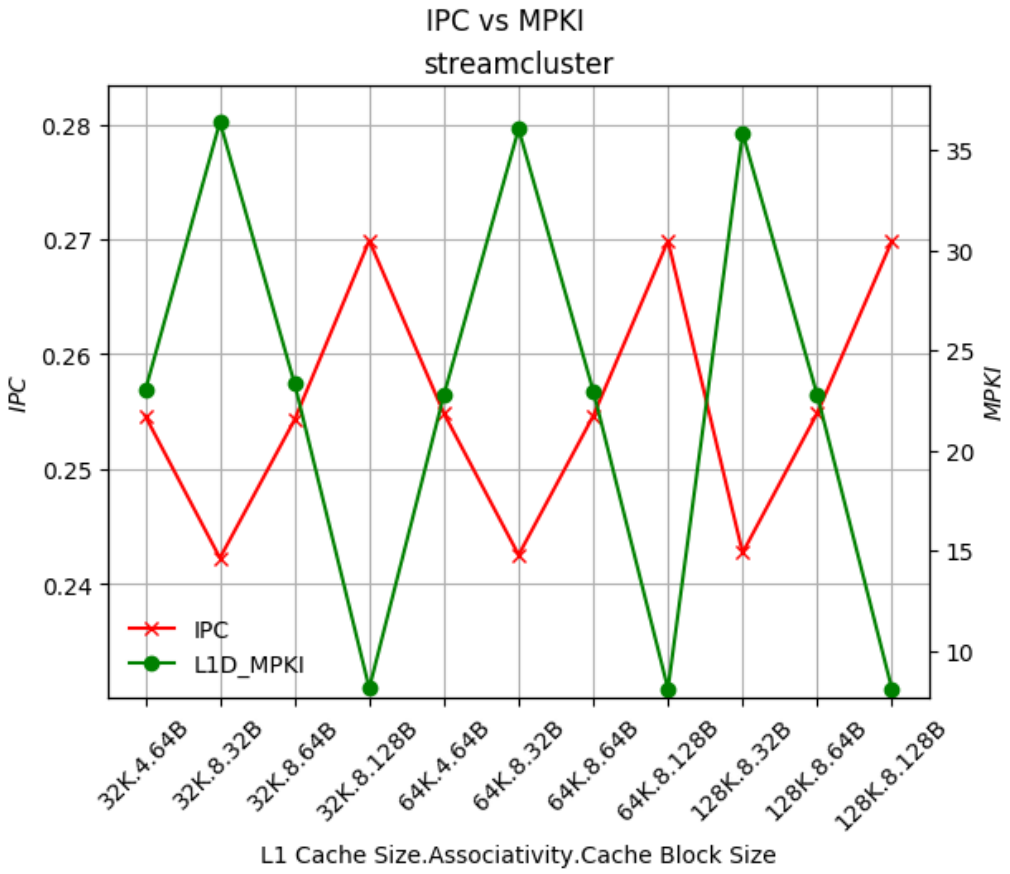
Στο παραπάνω διάγραμμα φαίνεται να εχουν σημαντική επίδραση και οι 3 παράγοντες.Είναι δύσκολη η ιεράρχιση κάποιου παράγοντα πάνω από κάποιον άλλον δίχως ένα ξεκάθαρο μοτίβο και πιθανώς χρειάζονται περαιτέρω προσομοιώσεις.Βέλτιστος ο συνδυασμός (128,8,128)

1.1.8 rtview



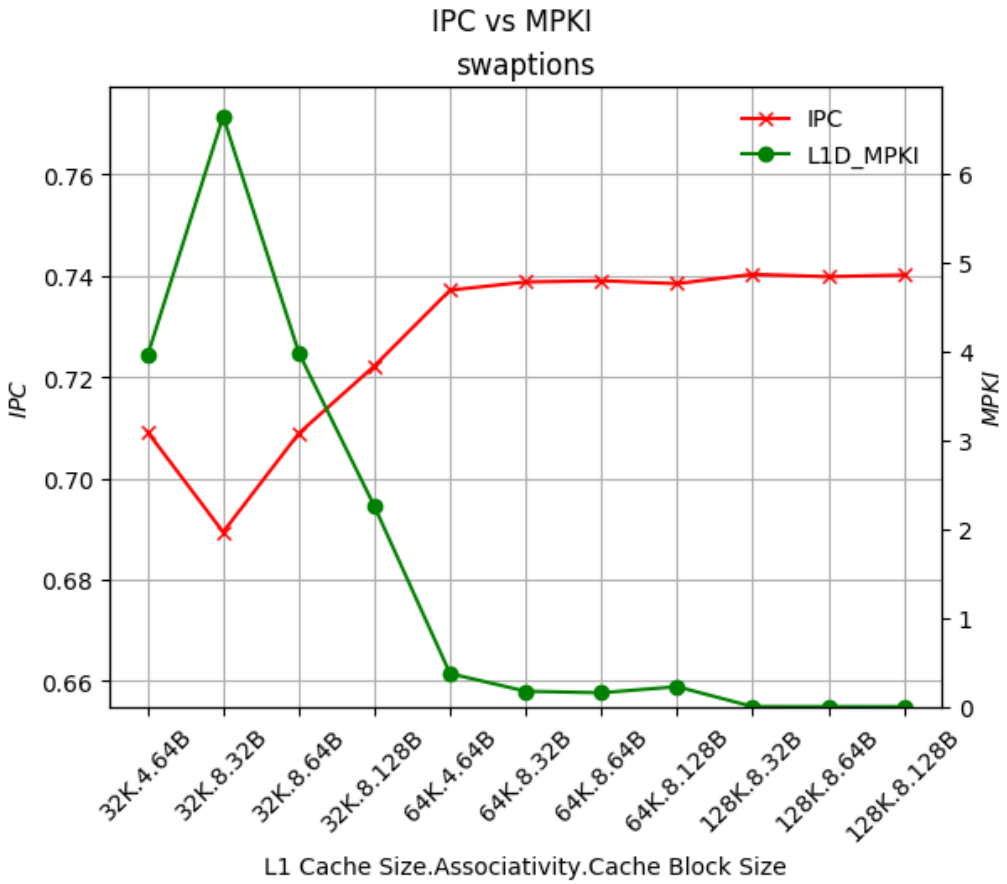
Στο L1 rtview .Γενικά η απόδοση είναι πάρα πολύ σταθερή , ανεξάρτητα από τις επιλογές των παραμέτρων . Το associativity δεν φαίνεται να επιδρά . Το cache size και το block size έχουν μια μικρή επίδραση , καθώς κάθε αύξησή τους οδηγεί σε ελαφρώς καλύτερα αποτελέσματα. Βέλτιστος συνδυασμός είναι ο (128,8,128),αλλά λαμβάνοντας υπ'όψην την οικονομία μπορούμε να επιλέξουμε τον (64,8,128)

1.1.9 streamcluster



Το block size είναι ο καθοριστικός παράγοντας σε αυτήν την περίπτωση. Οι μεταβολές των cache size και associativity δε δείχνουν να προκαλούν αλλαγές και είναι ήσσονος σημασίας μπροστά στο block size. Βέλτιστοι συνδυασμοί οι (32,8,128), (64,8,128), (128,8,128).

1.1.10 swaptions



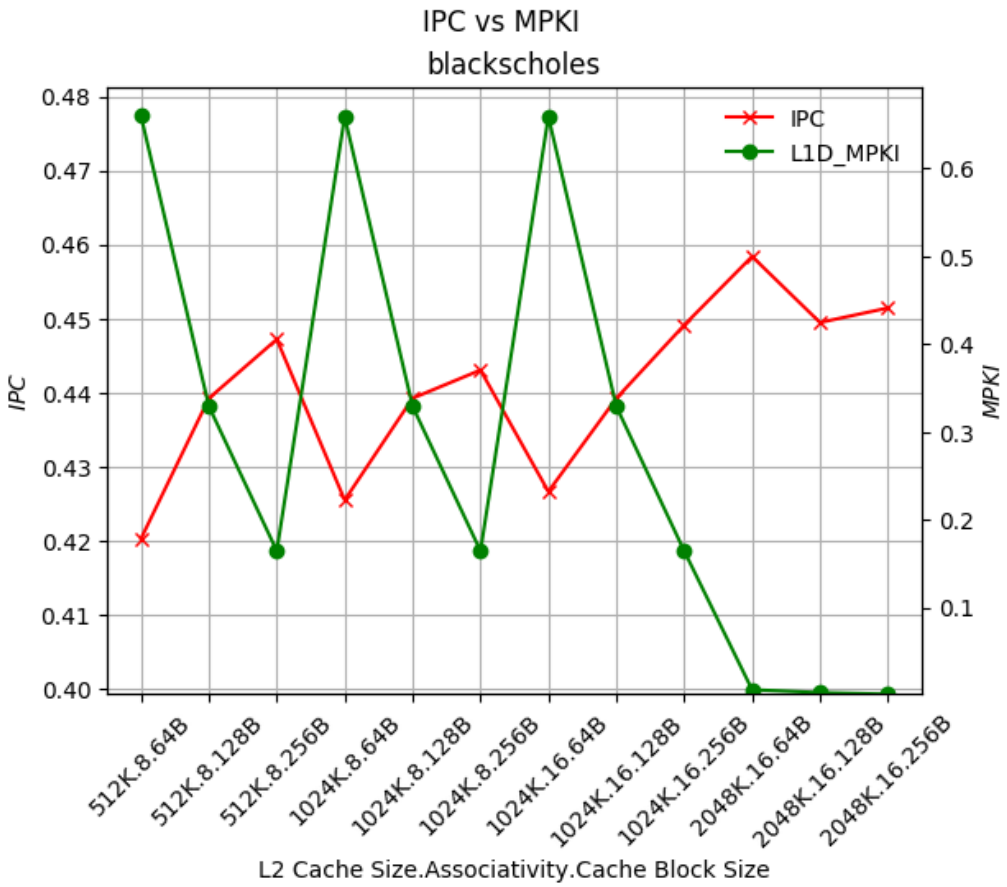
Σημαντικοί παράγοντες είναι το cache size και το block size , καθώς κάθε αύξησή τους συνεπάγεται και αύξηση της απόδοσης.Οι μεταβολές ήταν μεγάλες μέχρι το cache size να φτάσει τα 64K , αλλά στη συνέχεια η απόδοση σχεδόν σταθεροποιήθηκε,πράγματ που μας δείχνει πως 64K είναι αρκετά για τις ανάγκες αυτής της εφαρμογής.Βέλτιστος συνδυασμός για λόγους οικονομίας ο (64,8,32).

1.2 L2 cache

Για να αξιολογήσουμε μετρικές της L2 cache,κρατήθηκαν σταθερές παράμετροι της L1 cache και του TLB χρησιμοποιώντας διαφορετικά configuration για το μέγεθος,το associativity και το block size της L2 cache ως εξής:

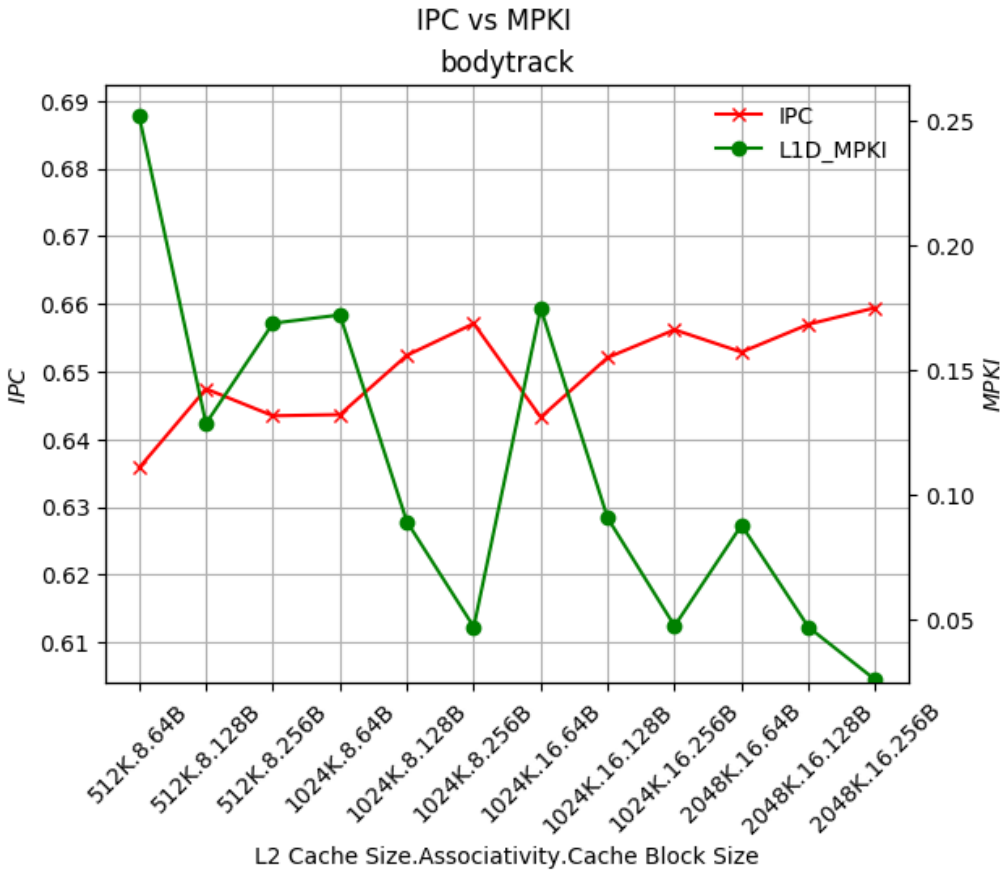
L2 size	L2 associativity	L2 cache block size
512 KB	8	64B, 128B, 256B
1024 KB	8, 16	64B, 128B, 256B
2048 KB	16	64B, 128B, 256B

1.2.1 blackscholes



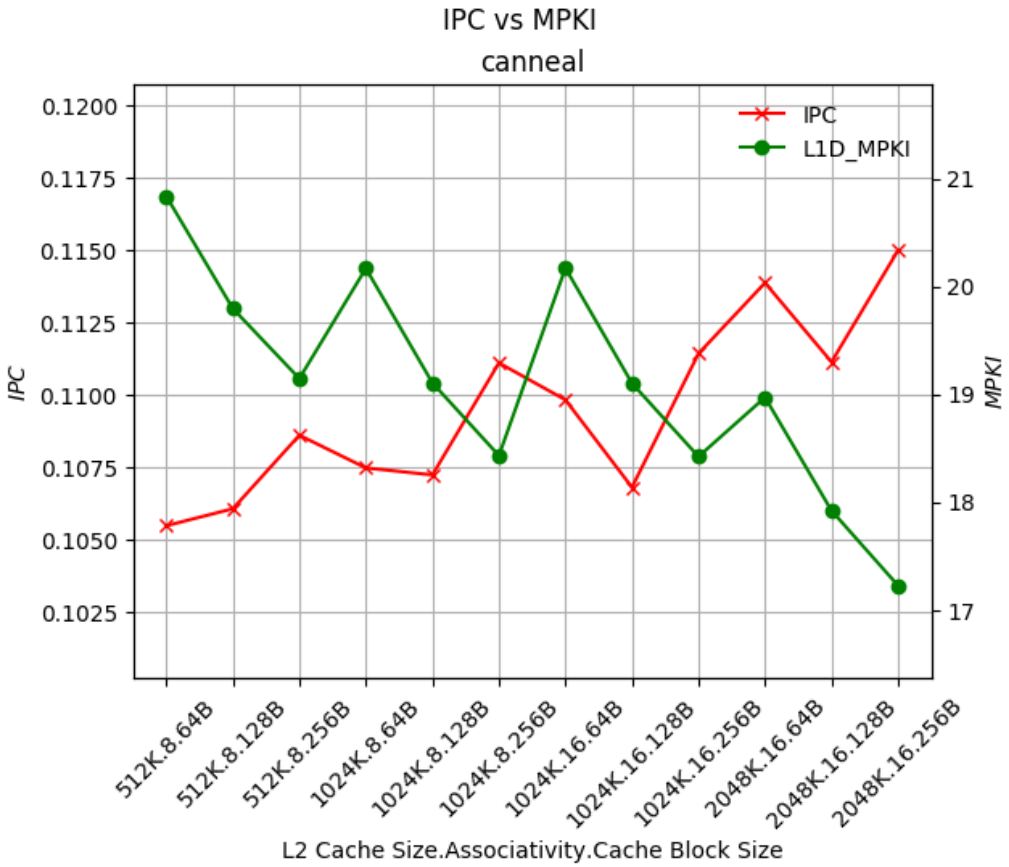
Βέλτιστος συνδυασμός είναι ο (2048,16,64) για λάγους απόδοσης και οικονομίας.Όλοι οι παράγοντες δείχνουν να είναι σημαντικοί,αλλά το cache size φαίνεται να έχει μικρότερη σημασία μπροστά στο associativity και το block size.

1.2.2 bodytrack



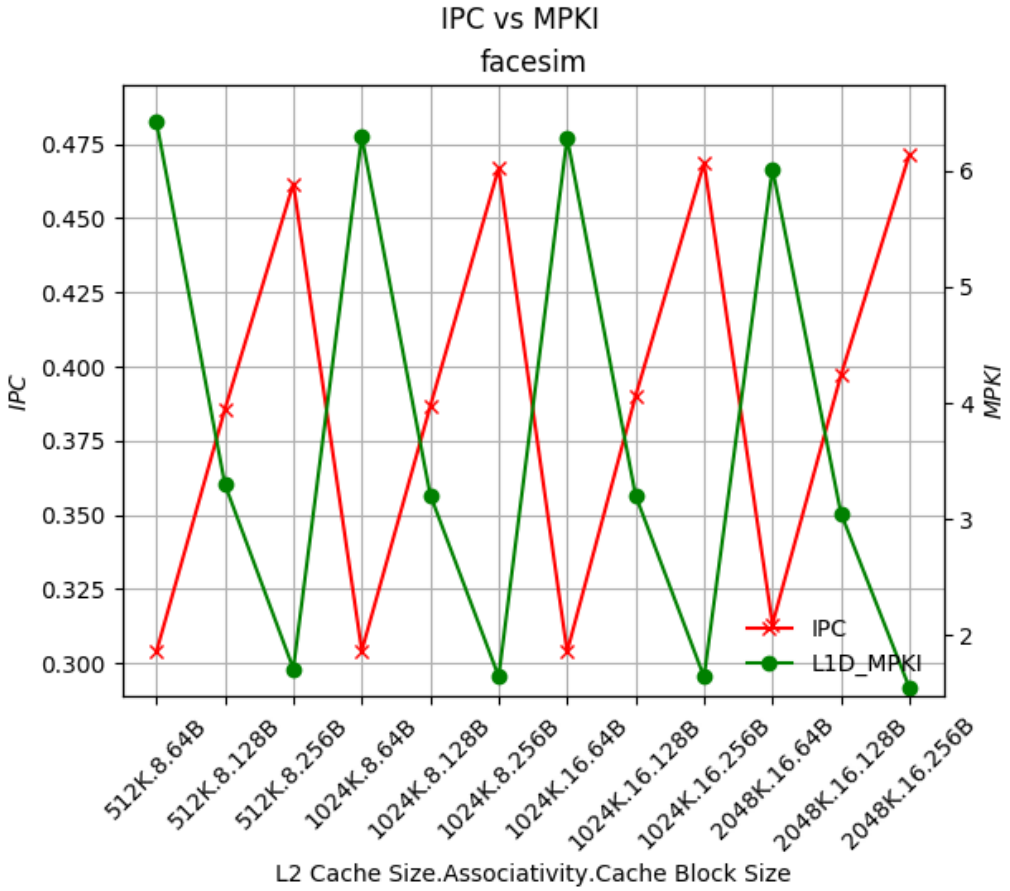
Βέλτιστος συνδυασμός ο (2048,16,256).Το associativity δεν δείχνει να είναι σημαντικό και η βελτίωση της απόδοσης έγκειται στο συνδυασμό αύξηση του cache size και block size.

1.2.3 canneal



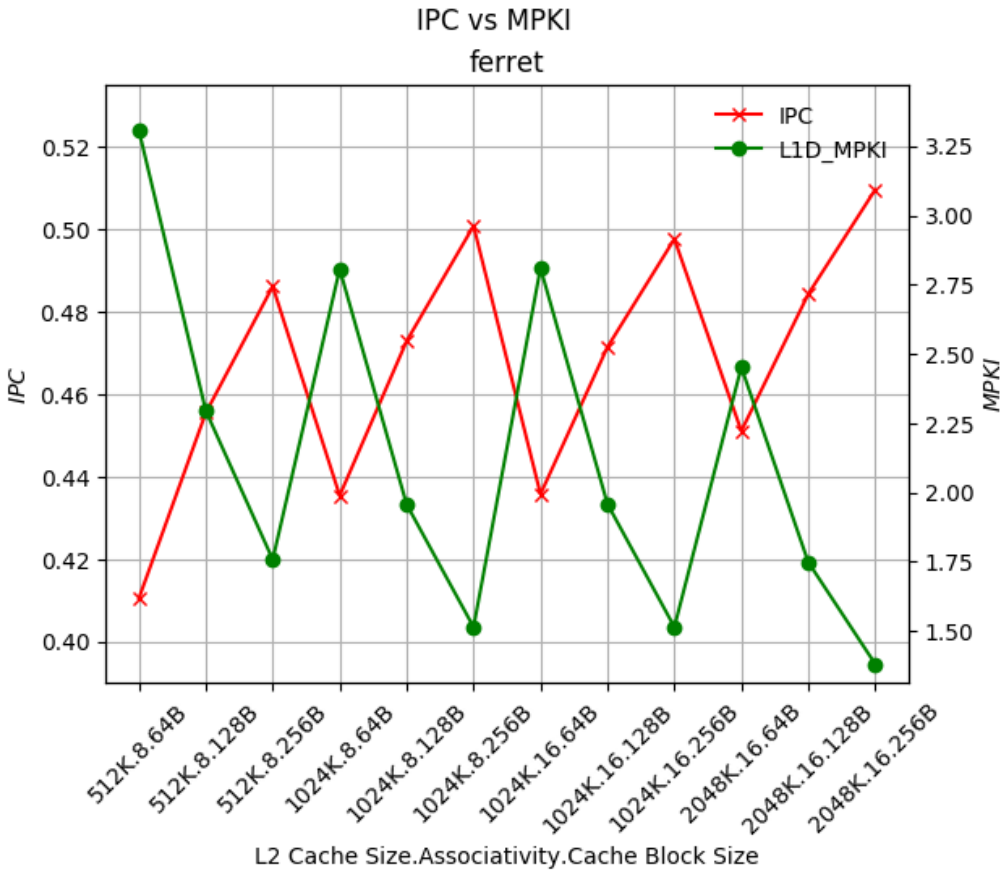
Βέλτιστος συνδυασμός είναι ο (2048,16,256). Σημαντικοί παράγοντες είναι το cache size και το block size .Κάθε φορά που αυξάνεται το cache size, βελτιώνεται η απόδοση . Αντίθετα , ενώ είναι προφανές πως το block size επηρεάζει το αποτέλεσμα , ο τρόπος που συμβαίνει αυτό δεν είναι τόσο ξεκάθαρος. Το associativity επηρεάζει ελάχιστα το αποτέλεσμα.

1.2.4 facesim



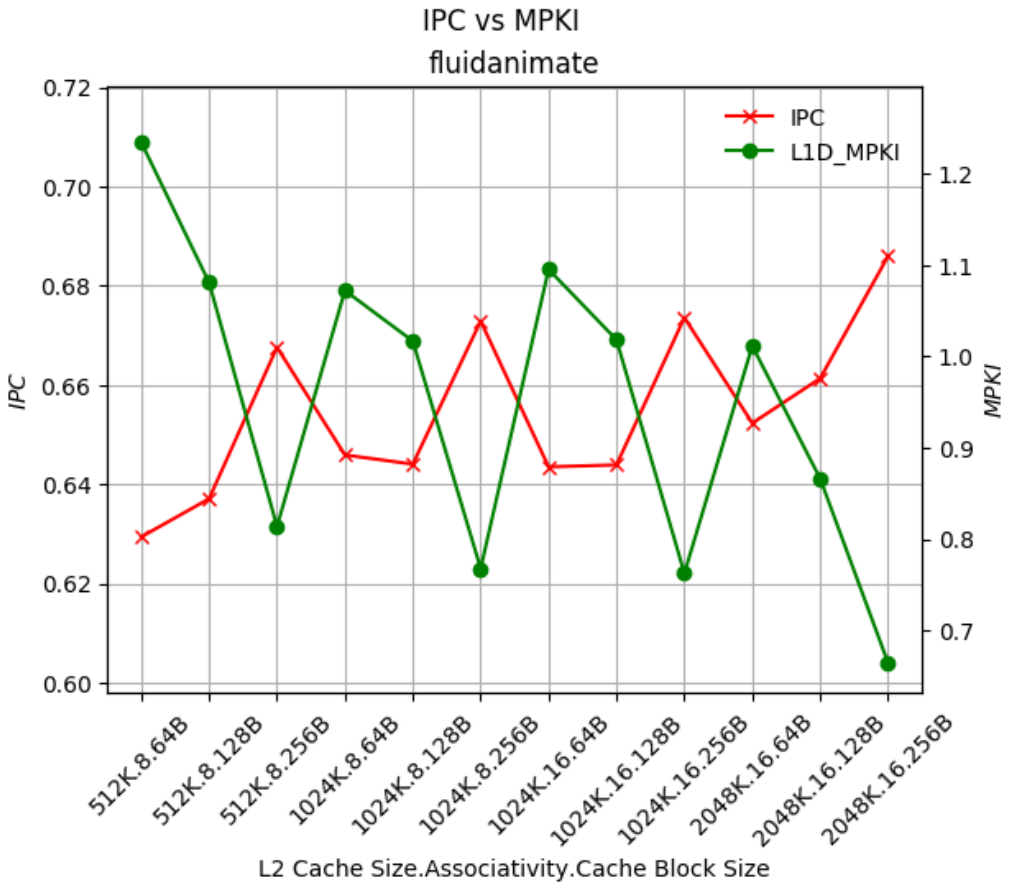
Βέλτιστος συνδυασμός είναι ο (512,8,128),για λόγους όχι μόνο απόδοσης,αλλά και οικονομίας,αφού ο συνδασμός (2048,16,256) προσφέρει ελάχιστη αύξηση της απόδοσης,με αρκετά αυξημένη χρήση υλικών.
Είναι εμφανές ότι καθοριστικότερος παράγοντας είναι το block size . Κάθε αύξηση του block size οδηγεί σε πολύ μεγάλη αύξηση της απόδοσης. Όσον αφορά το cache size , ισχύει περίπου ό,τι και στις προηγούμενες περιπτώσεις.Κάθε αύξησή του βελτιώνει λίγο την απόδοση . Το associativity δεν φαίνεται να επιδρά με κάποιον τρόπο.

1.2.5 ferret



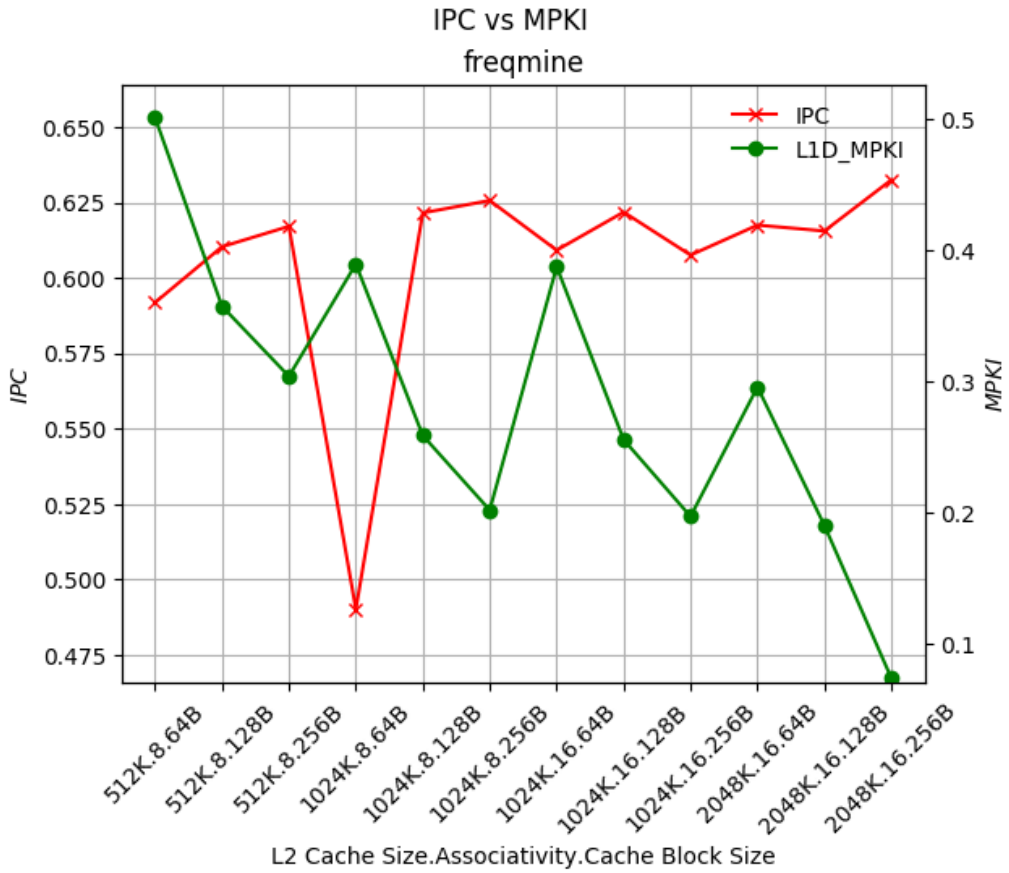
Βέλτιστος συνδυασμός είναι ο (2048,16,256).Υπάρχει μεγάλη ομοιότητα στους παράγοντες που επηρεάζουν το αποτέλεσμα με το ”L2 facesim”

1.2.6 fluidanimate



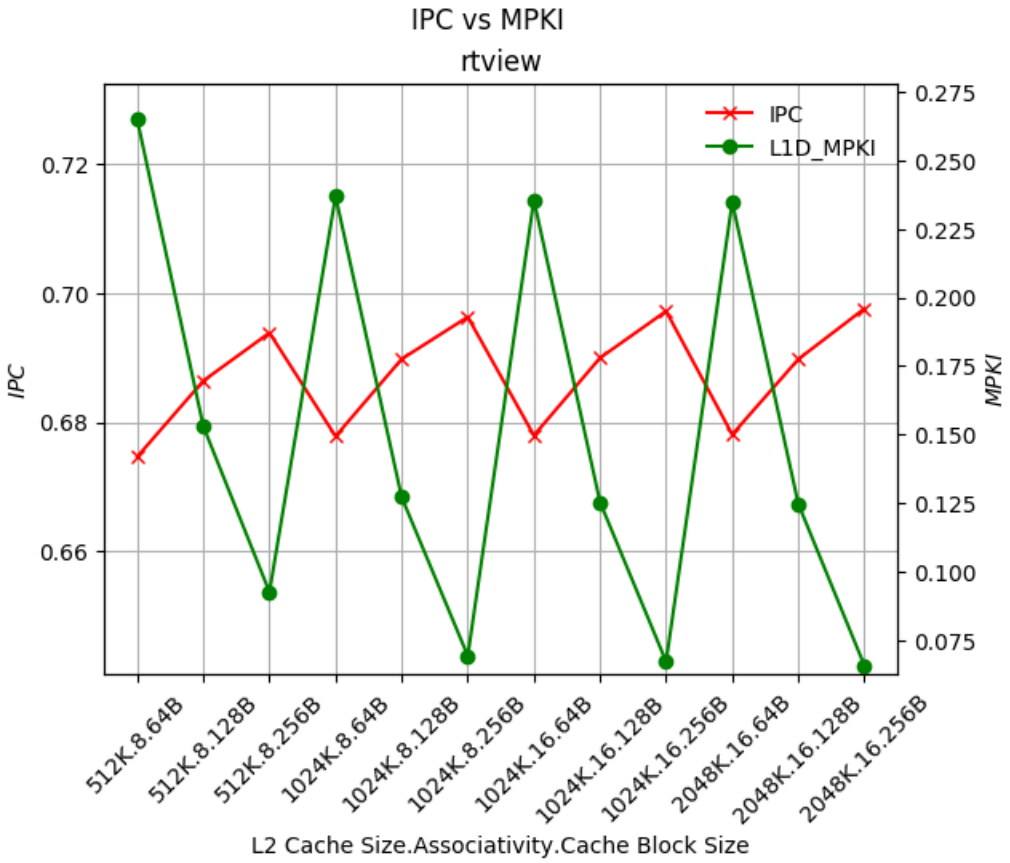
Βέλτιστος συνδυασμός είναι ο (2048,16,256).Σημαντικότερος παράγοντας είναι πάλι το μέγεθος του μπλοκ,ενώ κάθε αύξηση της cache επιφέρει κάποια αύξηση και στην απόδοση.Συγκρίνοτας τους συνδυασμούς (1024,8,256) και (1024,16,256) καταλήγουμε στο συμπέρασμα πως το associativity επηρεάζει ελάχιστα ή καθόλου το αποτέλεσμα.

1.2.7 freqmine



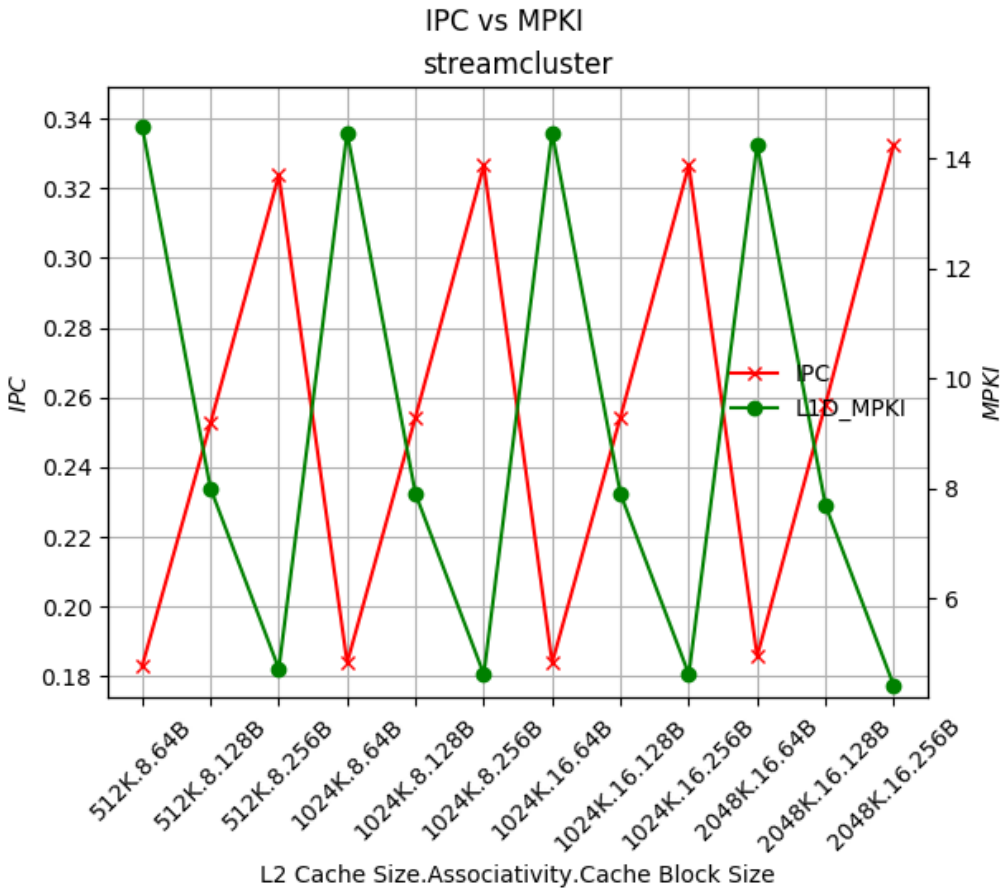
Βέλτιστος συνδυασμός είναι ο (2048,16,256).Εντύπωση κάνει ο συνδυασμός (1024,8,64) όπου παρατηρείται τρομακτική μείωση της απόδοσης παρά την αύξηση του cache size.Η εξήγηση έγκειται στην μείωση του block size.Συνεπώς,σημαντικότερος παράγοντας είναι πάλι το μέγεθος του μπλοκ,ενώ γενικά ισχύει πως κάθε αύξηση της cache επιφέρει αύξηση και στην απόδοση.

1.2.8 rtview



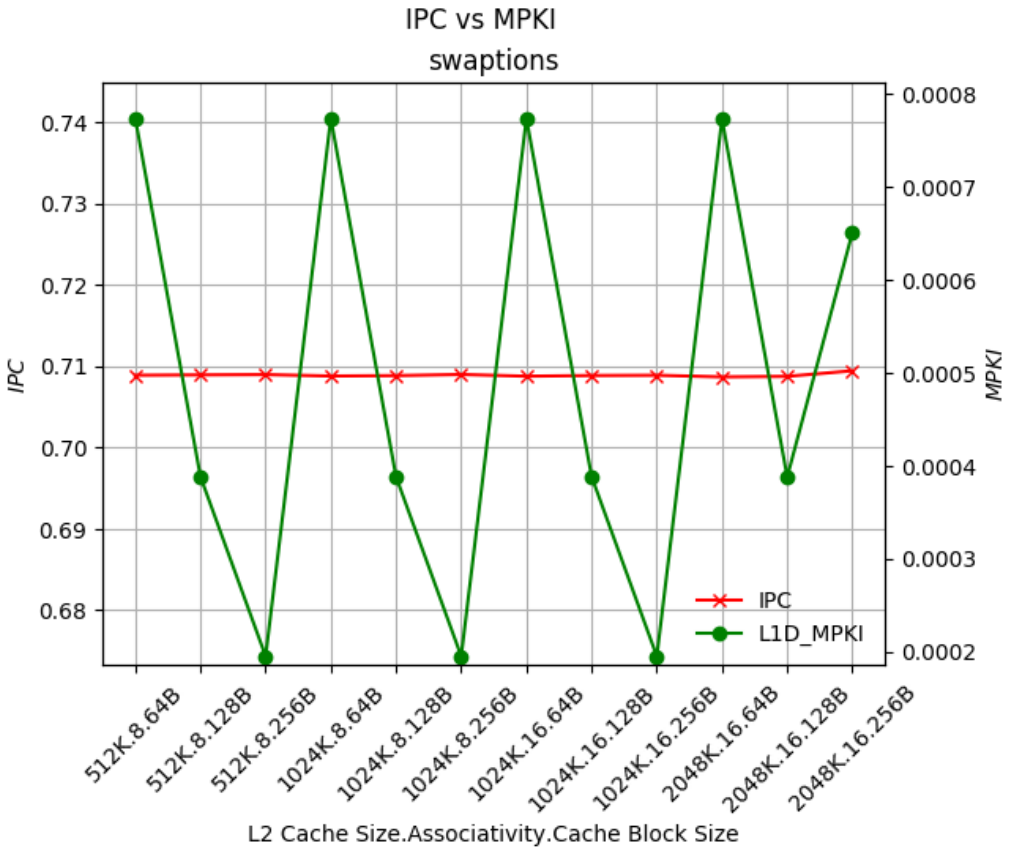
Βέλτιστος συνδυασμός είναι ο (1024,8,128) για λόγους απόδοσης και οικονομίας.Έχουμε άλλη μια περίπτωση στην οποία το block size είναι ο καταλυτικότερος παράγοντας και η αύξησή του οδηγεί σε καλύτερη απόδοση.Κάθε αύξηση του cache size επίσης επιφέρει μια μικρή βελτίωση. Η αύξηση του associativity προκαλεί μια πάρα πολύ μικρή βελτίωση , σχεδόν αμελητέα.

1.2.9 streamcluster



Βέλτιστος συνδυασμός είναι ο (512,8,256) για λόγους απόδοσης και οικονομίας,αφού ο συνδασμός (2048,16,256) προσφέρει ελάχιστη αύξηση της απόδοσης,με αρκετά αυξημένη χρήση υλικών.Ισχύει ότι και στο rtview για τους παράγοντες που επηρεάζουν το αποτέλεσμα.

1.2.10 swaptions



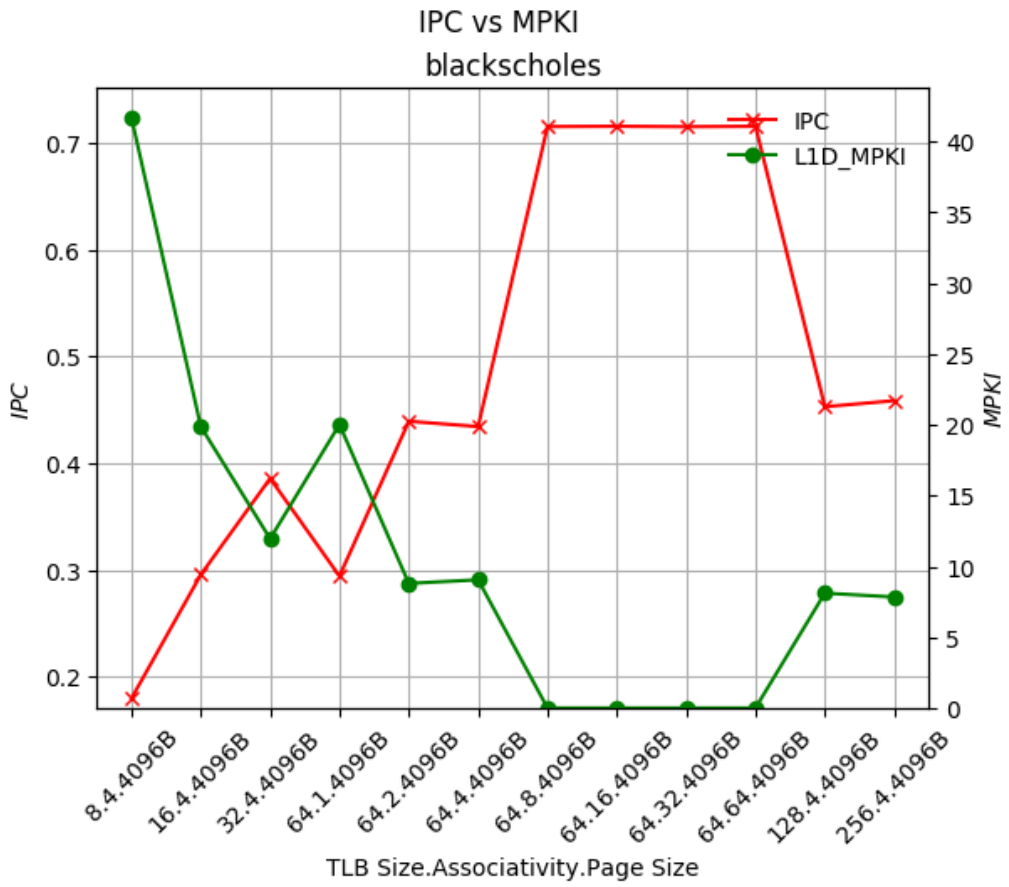
Βέλτιστος συνδυασμός είναι ο (512,8,128) για λόγους οικονομίας.Κανένας παράγοντας δε φαίνεται να επηρεάζει την απόδοση,πράγμα που μας υποδηλώνει πως στη συγκεκριμένη προσομοίωση η L2 cache χρησιμοποιείται ελάχιστα ή καθόλου.

1.3 TLB

Στο τμήμα αυτό της άσκησης μελετήσαμε την επίδραση της μεταβολής του μεγέθους και του associativity της L2 cache στην εκτέλεση των benchmarks. Τα χαρακτηριστικά που θέσαμε για την L2 φαίνονται στον ακόλουθο πίνακα. Ακόμη η L1 cache είχε μέγεθος 32K και cache line size 64B.

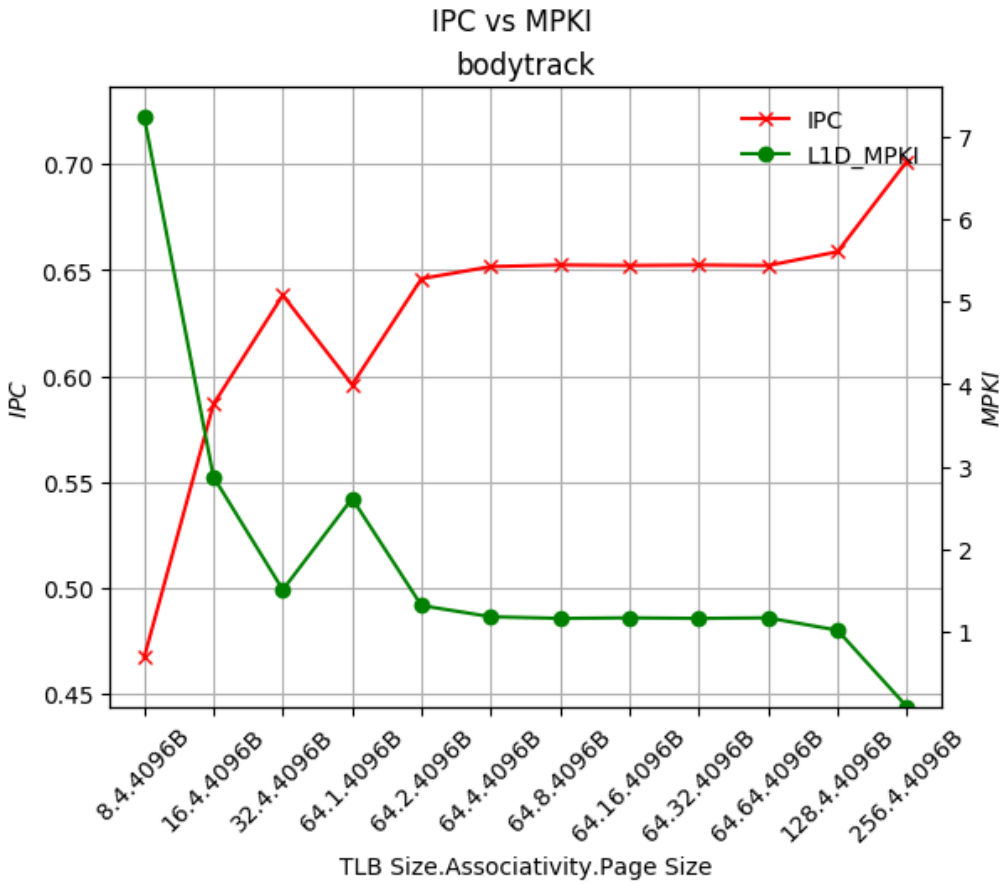
size	associativity	page size
8	4	4096B
16	4	4096B
32	4	4096B
64	1, 2, 4, 8, 16, 32, 64	4096B
128	4	4096B
256	4	4096B

1.3.1 blackscholes



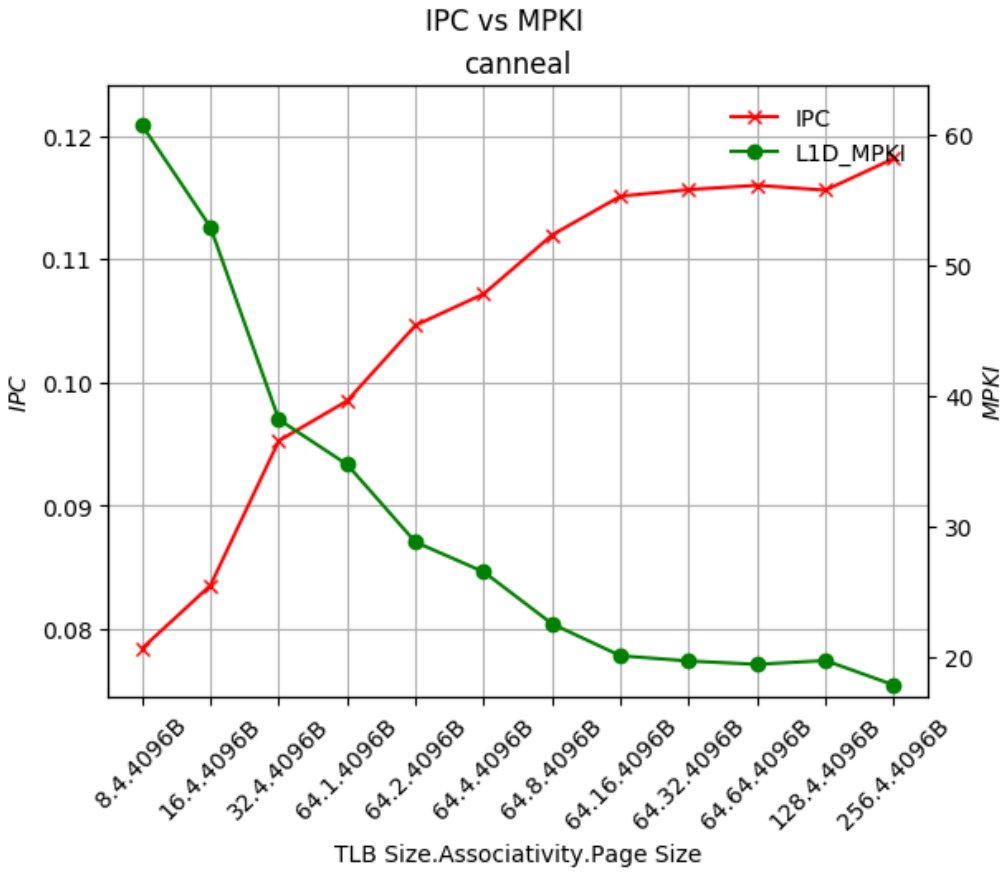
Βέλτιστοι συνδυασμοί είναι οι (64,8,4096) , (64,16,4096),(64,32,4096) και (64,64,4096).Η αύξησή του TLB size και του associativity βελτιώνει την απόδοση.

1.3.2 bodytrack



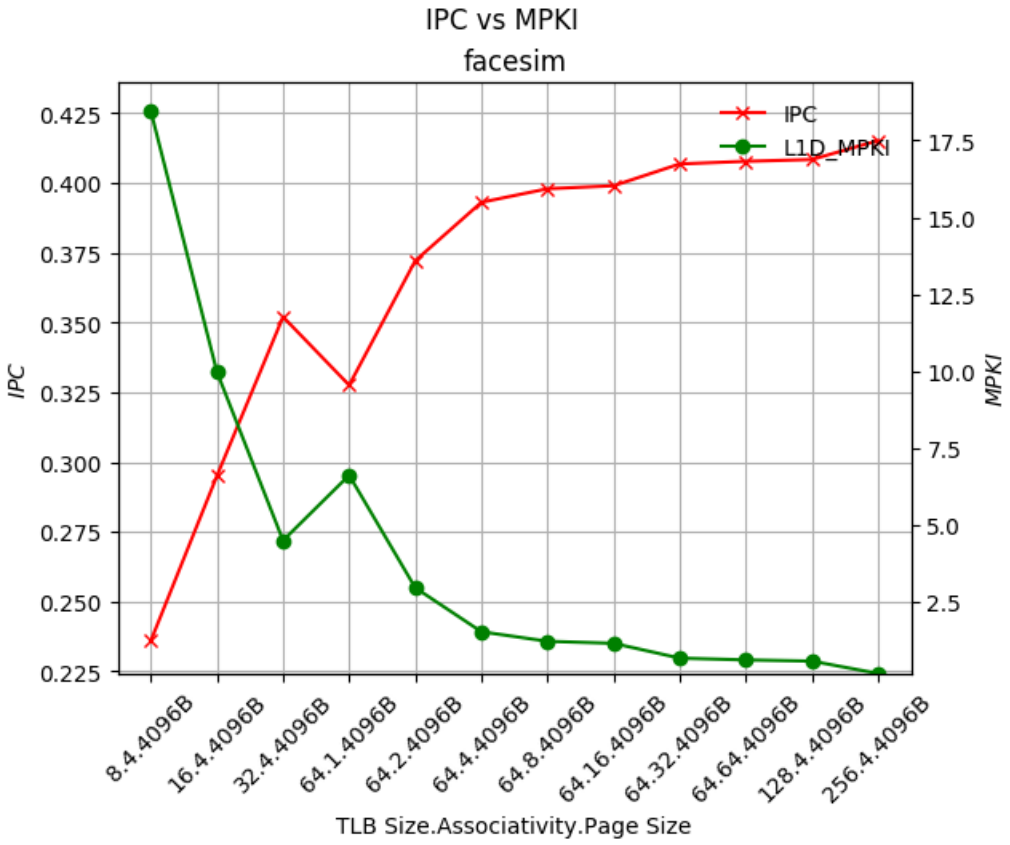
Βέλτιστος συνδυασμός είναι ο (256,4,4096).Το TLB size είναι ο καθοριστικός παράγοντας.Το associativity δεν επιδρά σημαντικά για τιμές μεγαλύτερες ή ίσες του 2.

1.3.3 canneal



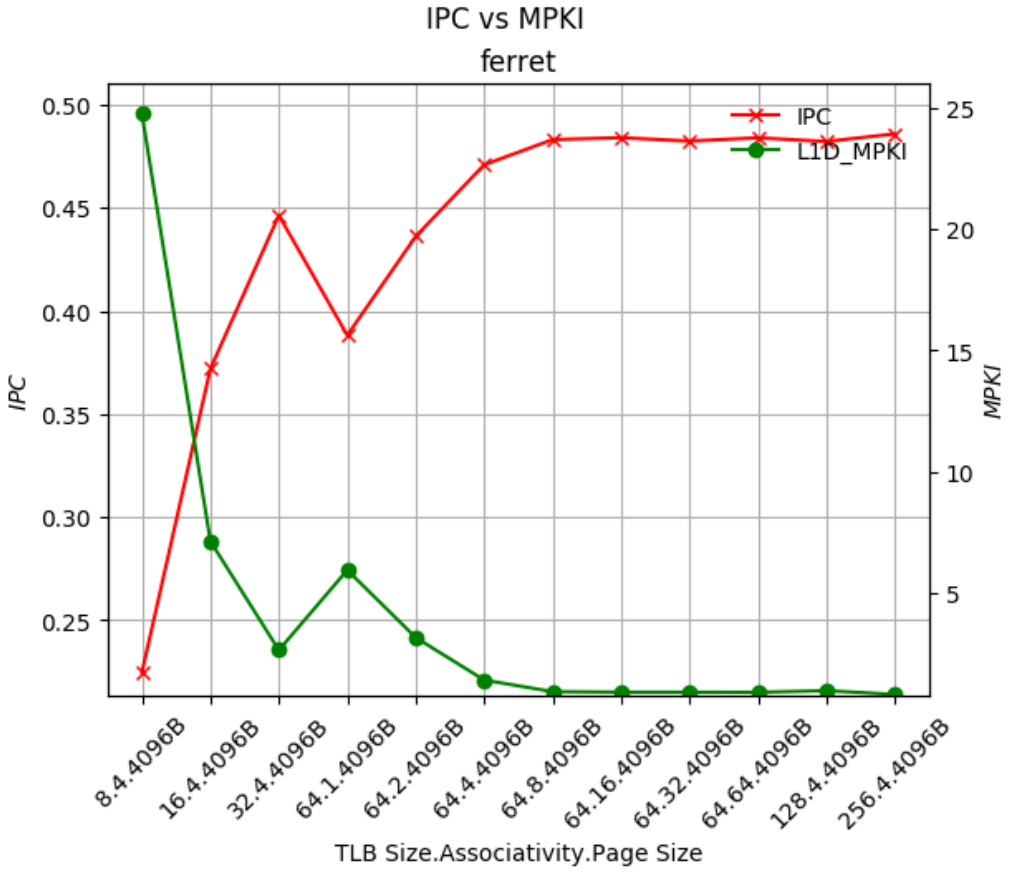
Βέλτιστος συνδυασμός είναι ο (256,4,4096).Οι σημαντικότεροι παράγοντες είναι το TLB size και το associativity.

1.3.4 facesim



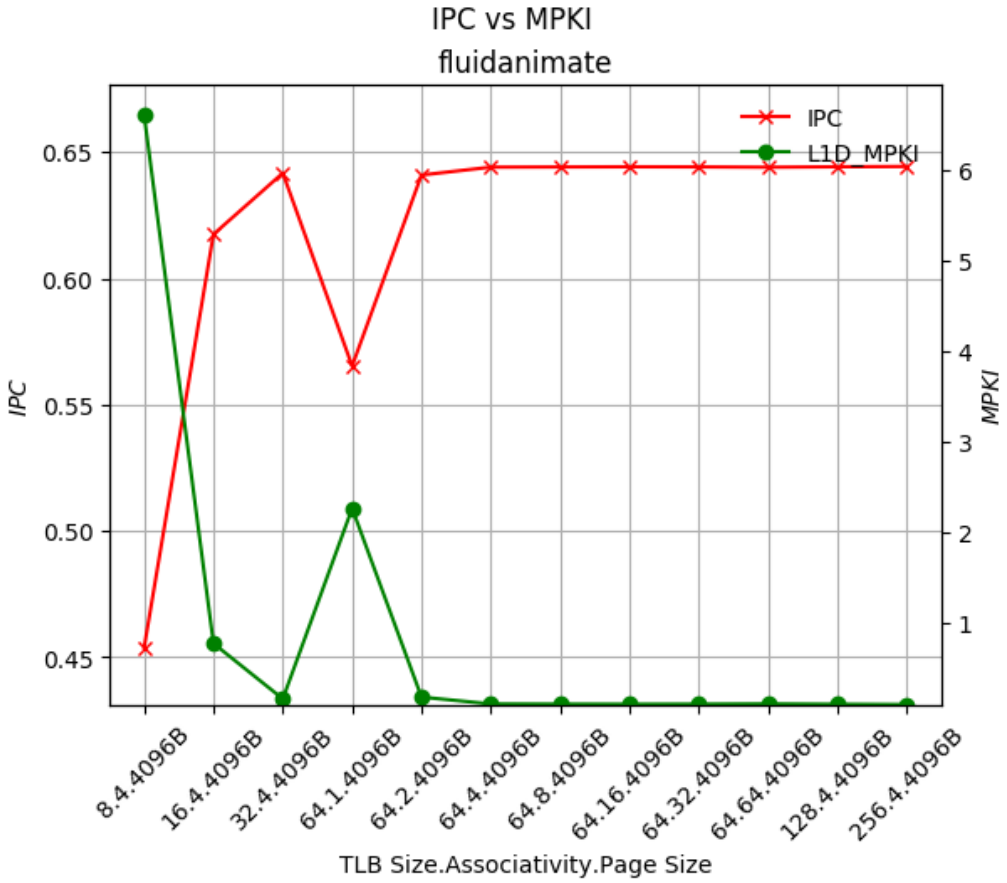
Στο TLB facesim βέλτιστος συνδυασμός είναι ο (256,4,4096).Οι σημαντικότεροι παράγοντες είναι (ξανά) το TLB size και το associativity.Το associativity δεν επιδρά σημαντικά,πρέπει όμως να έχει τιμή τουλάχιστον 2 διότι εάν είναι ίσο με 1 τότε επιδρά αρνητικά.

1.3.5 ferret



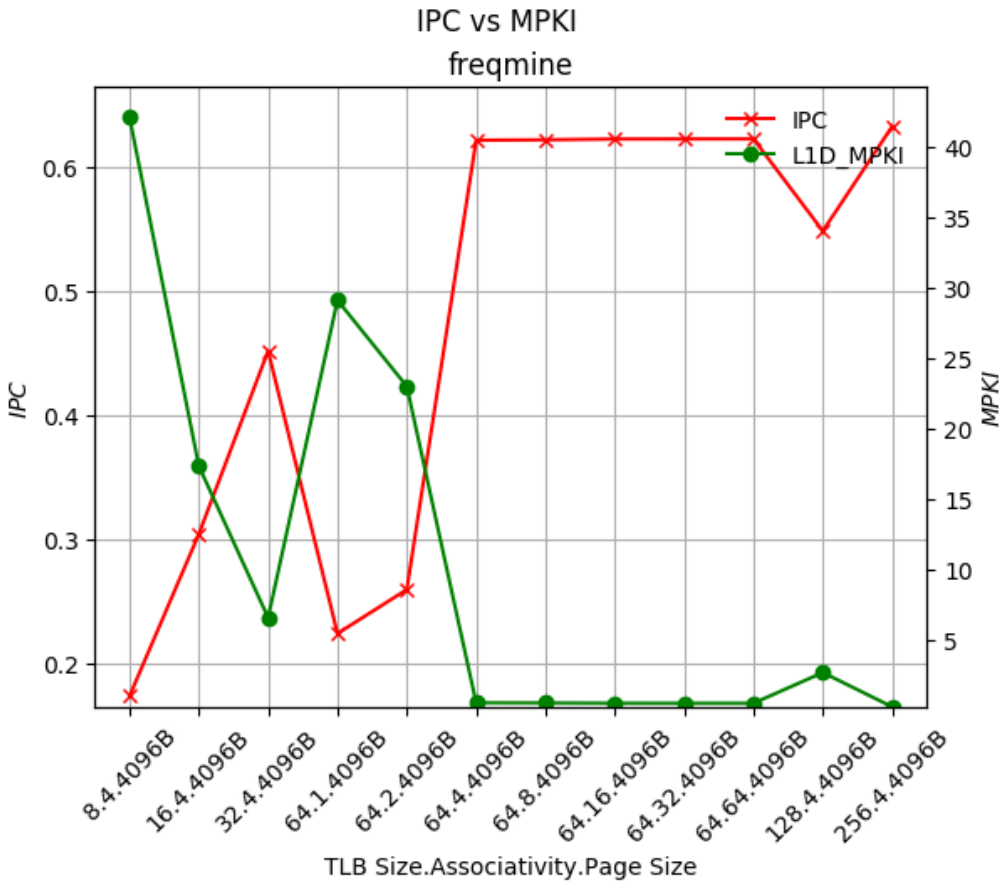
Βέλτιστος συνδυασμός είναι το (256,4,4096).Οι αυξήσεις των TLB size και associativity βελτιώνουν την επίδοση.Για να έχουμε καλές επιδόσεις , το associativity πρέπει να είναι τουλάχιστον 4 και το block size τουλάχιστον 16.

1.3.6 fluidanimate



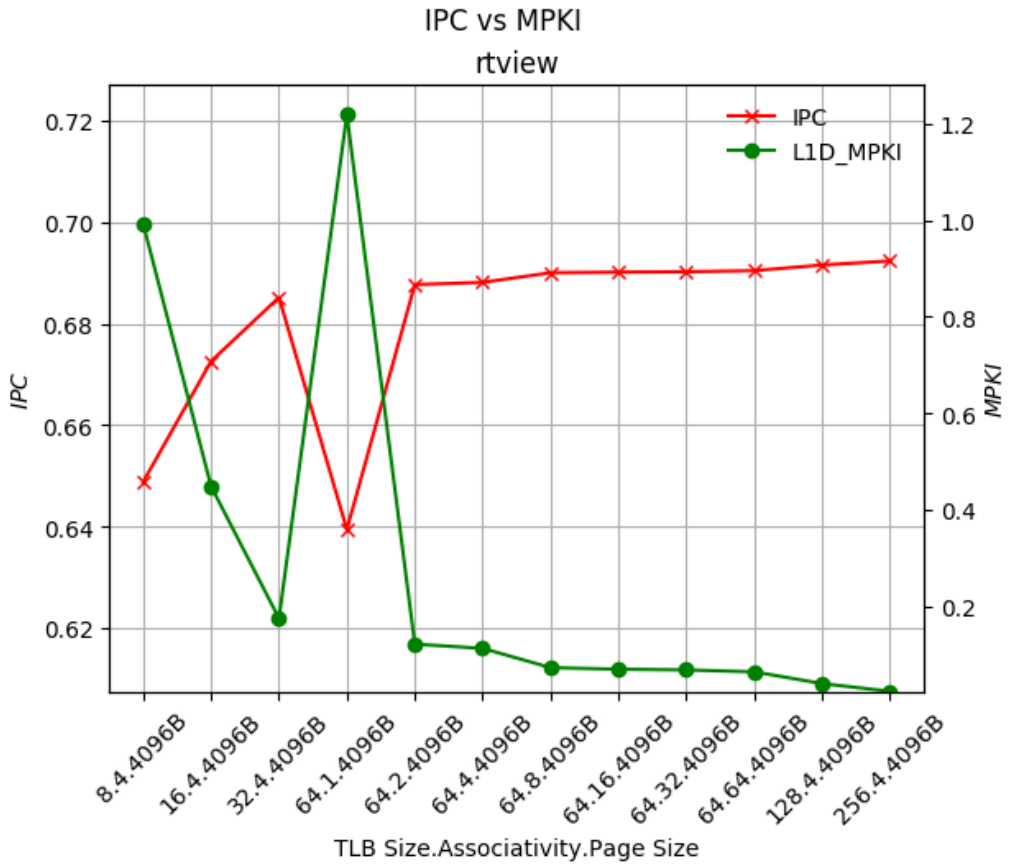
Βέλτιστοι συνδυασμοί είναι οι(64,4,4096),(64,8,4096),(64,16,4096), (64,32,4096),(64,64,4096),(128,4,4096) όσο αυξάνεται το TLB size και το associativity , αυξάνεται και η απόδοση. Κάθε αύξηση του TLB size πάνω από 64 και κάθε αύξηση του associativity πάνω από 8 δεν έχει σχεδόν καμία επίδραση όμως.

1.3.7 freqmine



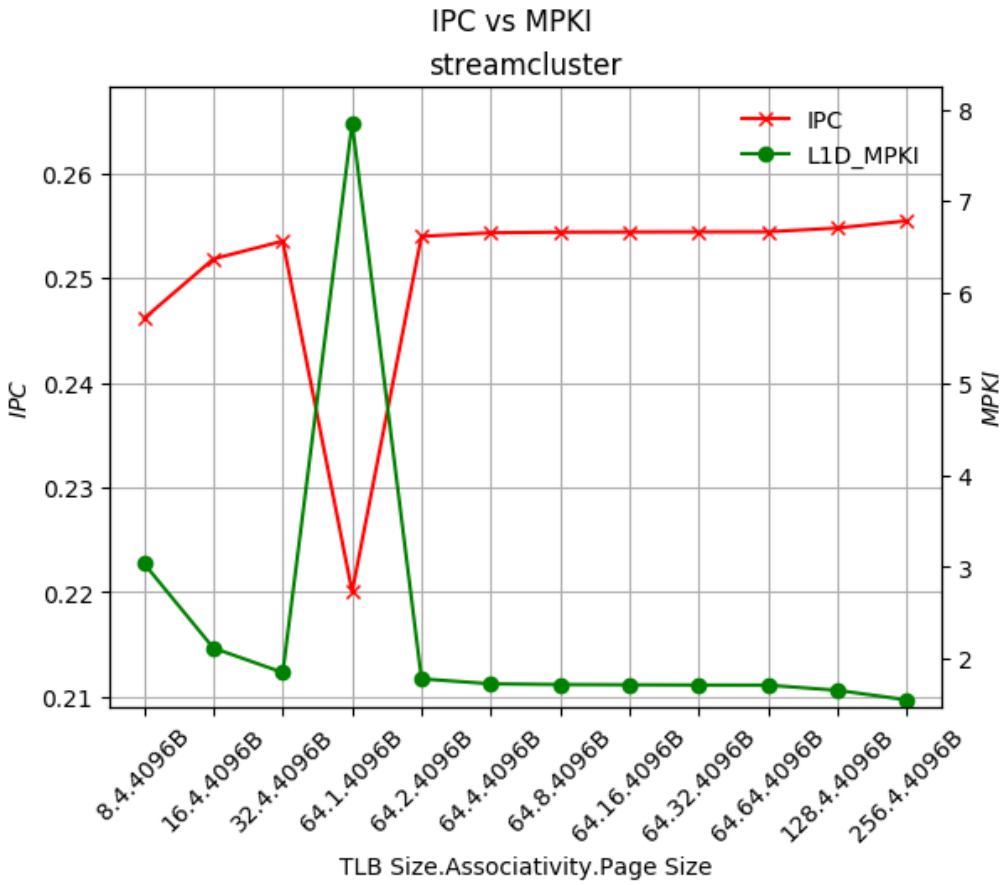
Βέλτιστος συνδυασμός είναι ο (256,4,4096).Όσο αυξάνεται το TLB size ,βελτιώνεται και η απόδοση

1.3.8 rtview



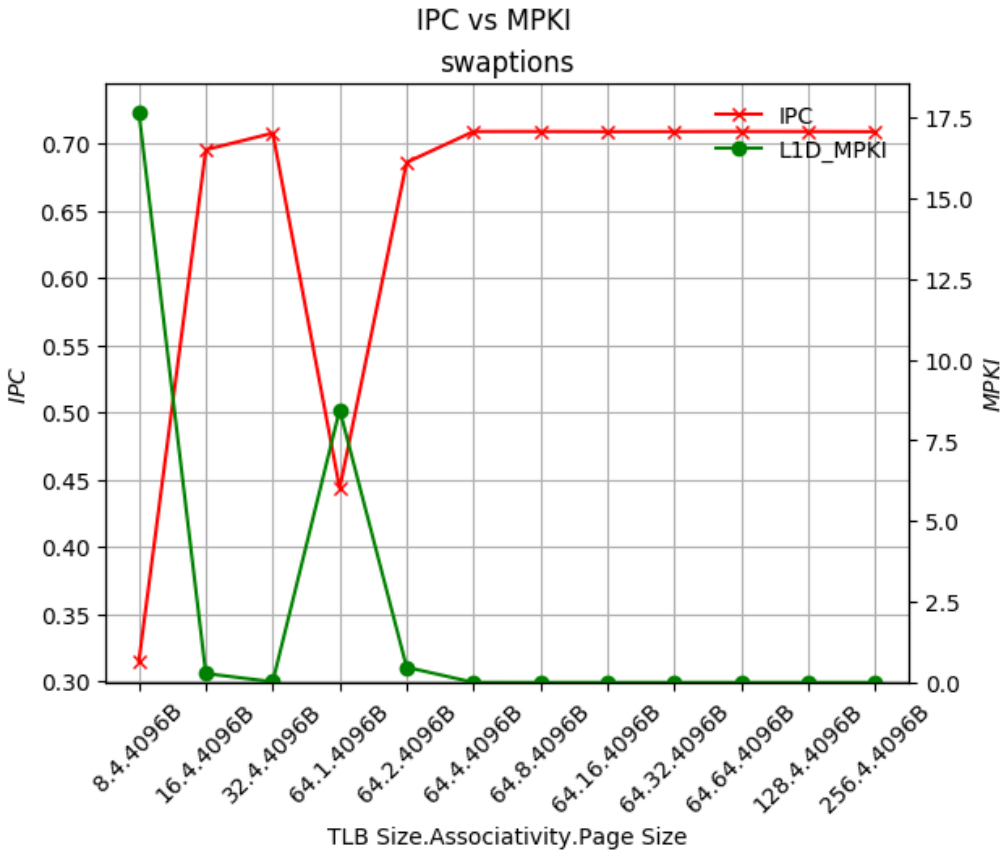
Βέλτιστος συνδυασμός είναι ο (256,4,4096).Όσο αυξάνεται το TLB size και το associativity, βελτιώνεται και η απόδοση. Βλέπουμε πως το TLB size πρέπει να είναι τουλάχιστον 32 και το associativity τουλάχιστον 2 για να έχουμε ικανοποιητικά αποτελέσματα.

1.3.9 streamcluster



Βέλτιστος συνδυασμός είναι ο (256,4,4096).Όλοι οι συνδυασμοί δίνουν ικανοποιητικά αποτελέσματα πλην της επιλογής όπου associativity = 1. Κάθε αύξηση του TLB size και του associativity βελτιώνει ελάχιστα την απόδοση.

1.3.10 swaptions



Βέλτιστοι συνδυασμοί είναι οι (64,4,4096) , (64,8,4096),(64,16,4096), (64,32,4096),(64,64,4096) , (128,4,4096) και (256,4,4096).Και σε αυτήν την περίπτωση , όσο αυξάνεται το TLB size και το associativity , βελτιώνεται η απόδοση.

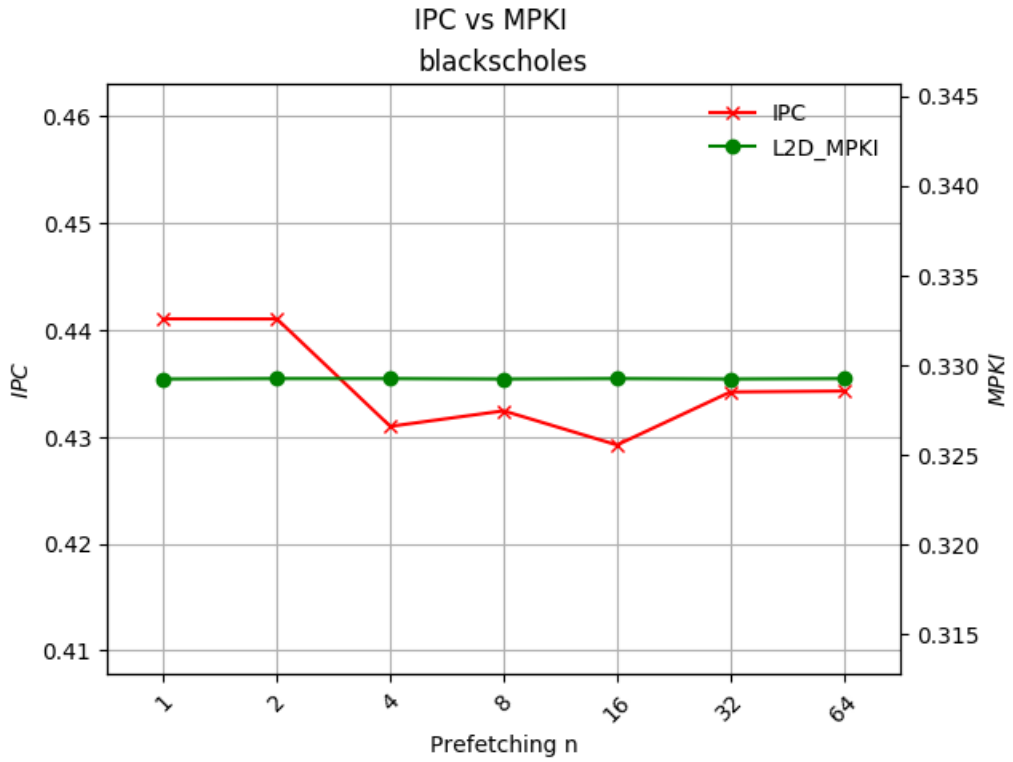
1.4 PRF

Τέλος μεταβάλλουμε το το μέγεθος του cache-line και το μέγεθος της L2 cache σύμφωνα με τον παρακάτω πίνακα κρατώντας τα χαρακτηριστικά της L1 όπως ίδια με τις προηγούμενες εκτελέσεις.

Size	line size	lines	associativity
512K	64	8192	8
512K	256	2048	8
1024K	64	16384	8
1024K	256	4096	8
2048K	64	32768	8
2048K	256	8192	8

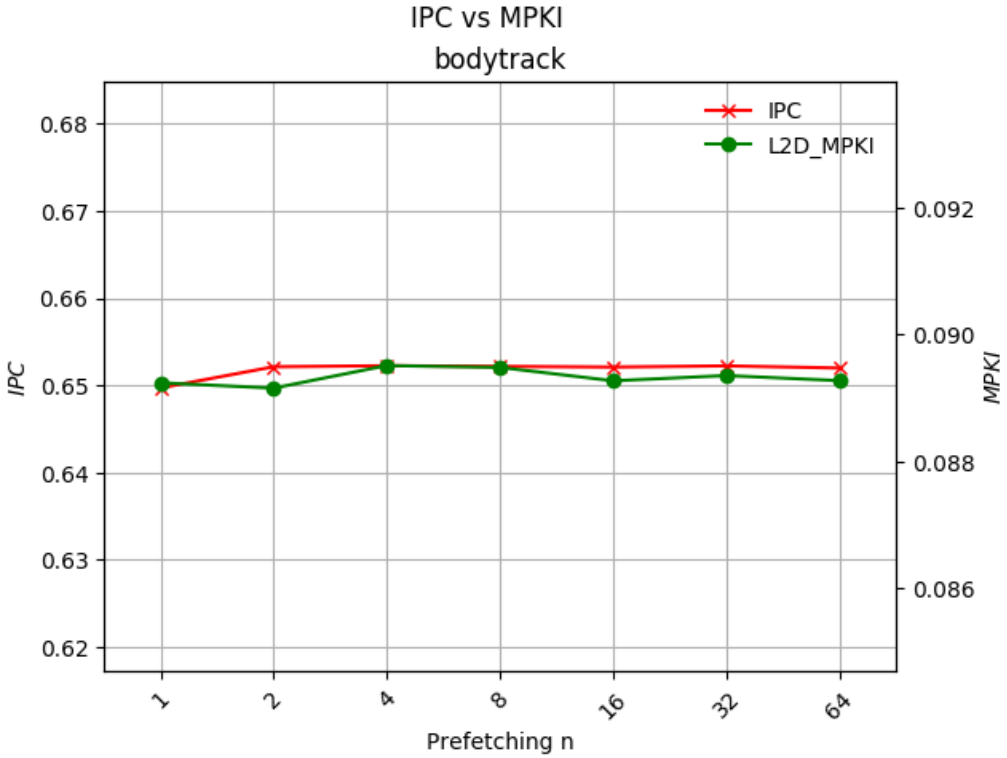
Τα αποτελέσματα ακολουθούν χωρισμένα ανά μέγεθος cache-line. Παρατηρούμε πως το IPC βελτιώνεται κατά πολύ όσο μεγαλώνει τόσο το μέγεθος της cache όσο και του cache-line. Τα equake και gzip δείχνουν να μην επηρεάζονται από τις αλλαγές στην L2 cache πράγμα που δηλώνει πως έχουν μικρό αριθμό προσβάσεων στην L2.

1.4.1 blackscholes



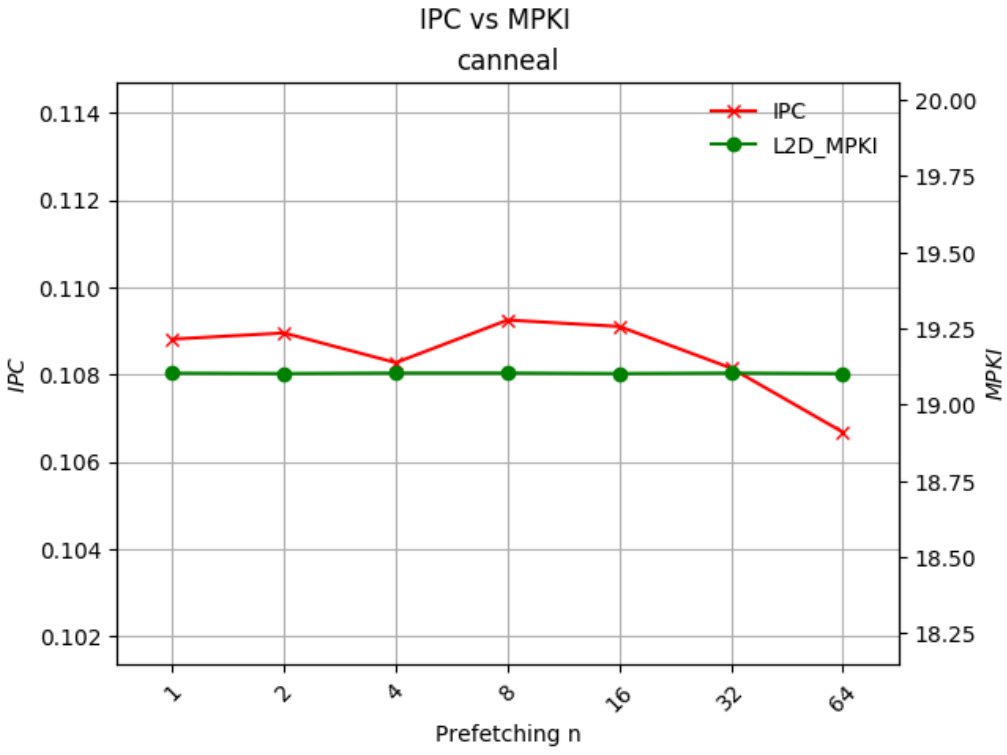
Βέλτιστη επιλογή για το prefetching είναι το 1 και 2. Καθώς το prefetching αυξάνεται,η απόδοση ελαττώνεται.Πιθανός λόγος είναι πως το συγκεκριμένο μετροπρόγραμμα δεν έχει spatial locality στην L2 cache.

1.4.2 bodytrack



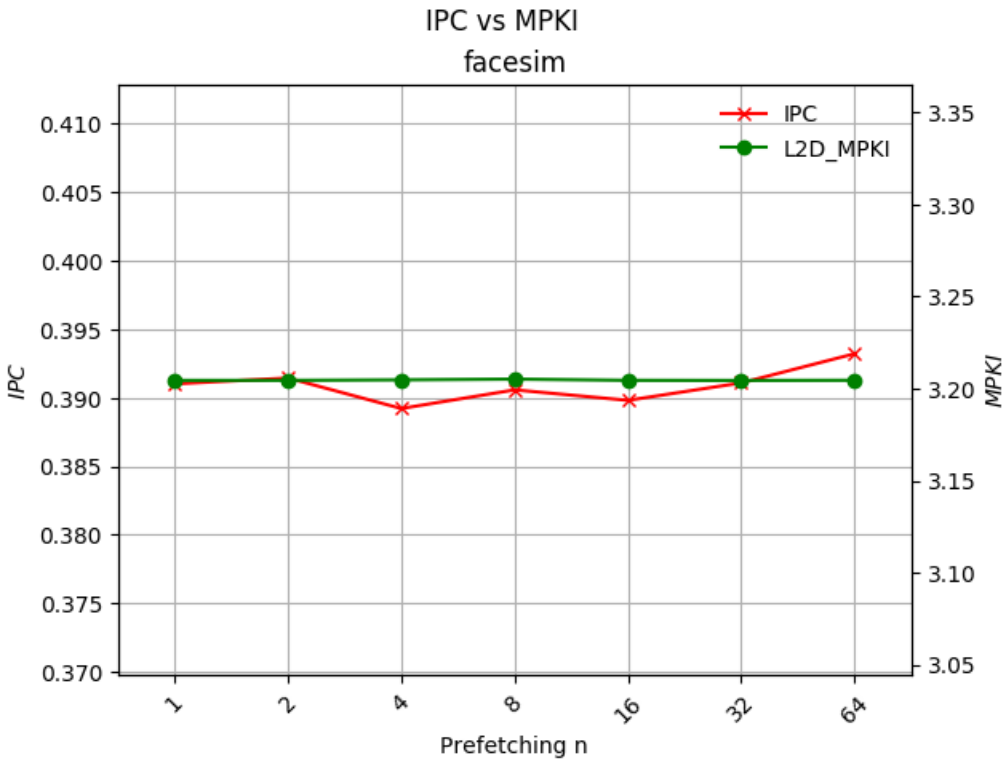
Οι βέλτιστοι συνδυασμοί είναι 2,4,...,64 οι οποίοι μας δίνουν την ίδια απόδοση.Λογικά η εφαρμογή δεν εκμεταλεύεται παραπάνω απο 2 blocks από το αρχικό στην L2 cache (ισχυρό spatial locality).

1.4.3 canneal



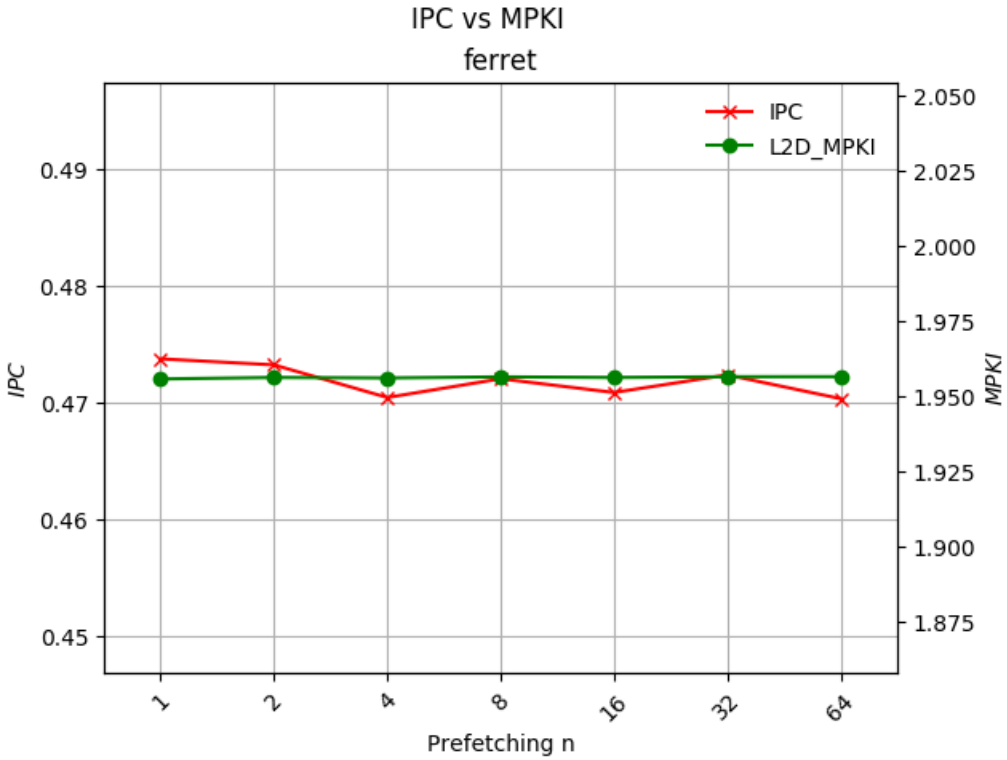
Η επιλογή 8 δίνει τα καλύτερα αποτελέσματα.Παρατηρούμε πως η απόδοση ελαττώνεται για τιμές μεγαλύτερες απο 8.

1.4.4 facesim



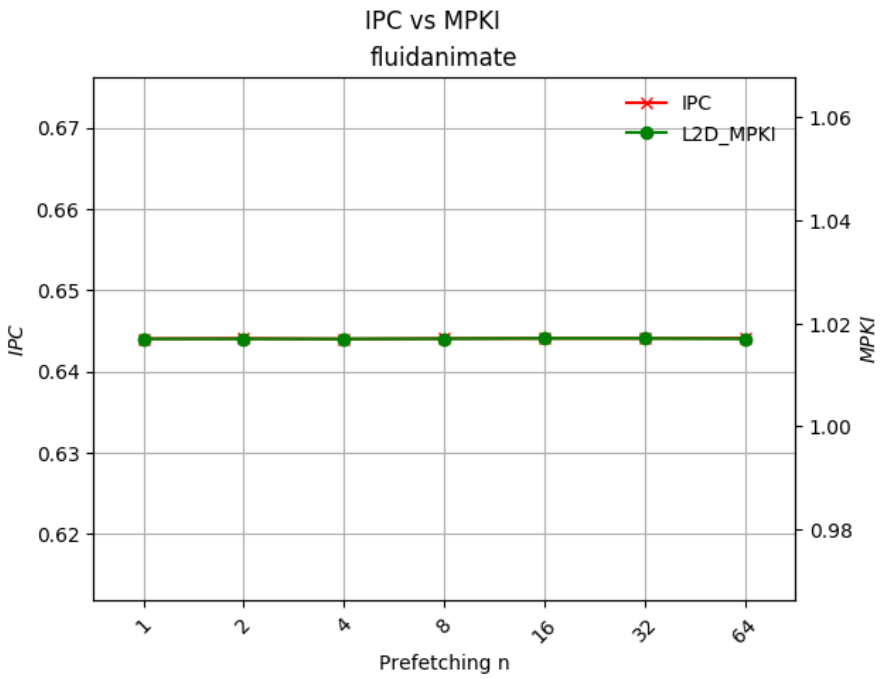
Βέλτιστη επιλογή για το prefetching είναι το 64.Οι διαφορές κάθε τιμής είναι ελάχιστες.

1.4.5 ferret



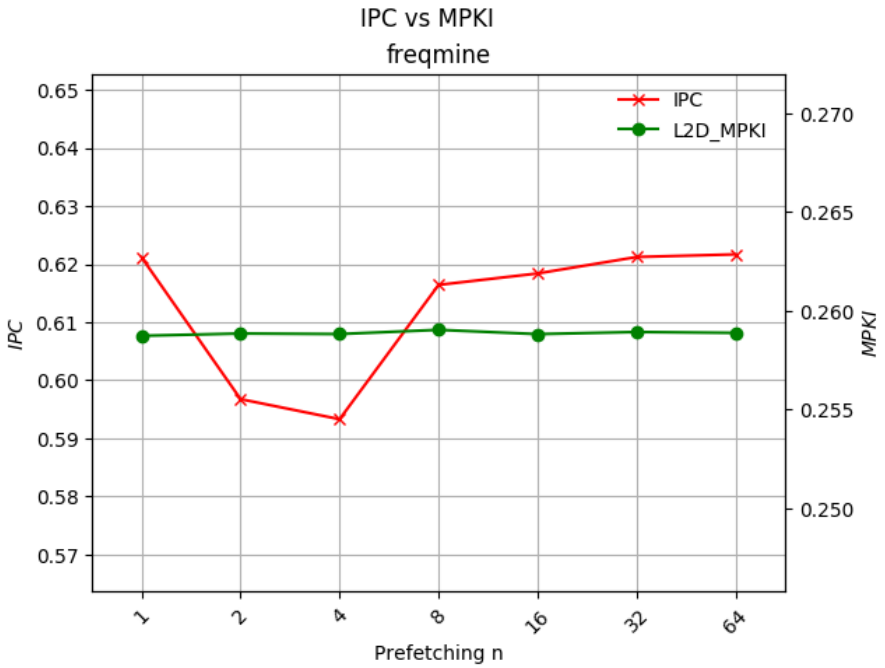
Βέλτιστη επιλογή για το prefetching είναι το 1.Η απόδοση πέφτει ελάχιστα για όλες τις άλλες επιλογές.

1.4.6 fluidanimate



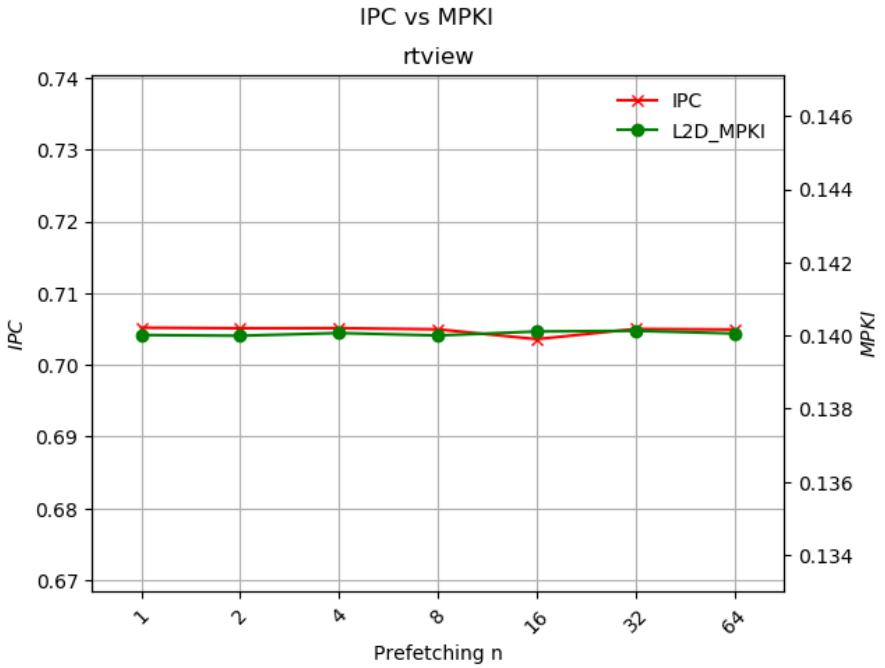
Το fluidaminate μένει ανεπηρέαστο απο το L2 prefetching,καθώς όλες οι επιλογές έχουν την ίδια απόδοση.

1.4.7 freqmine



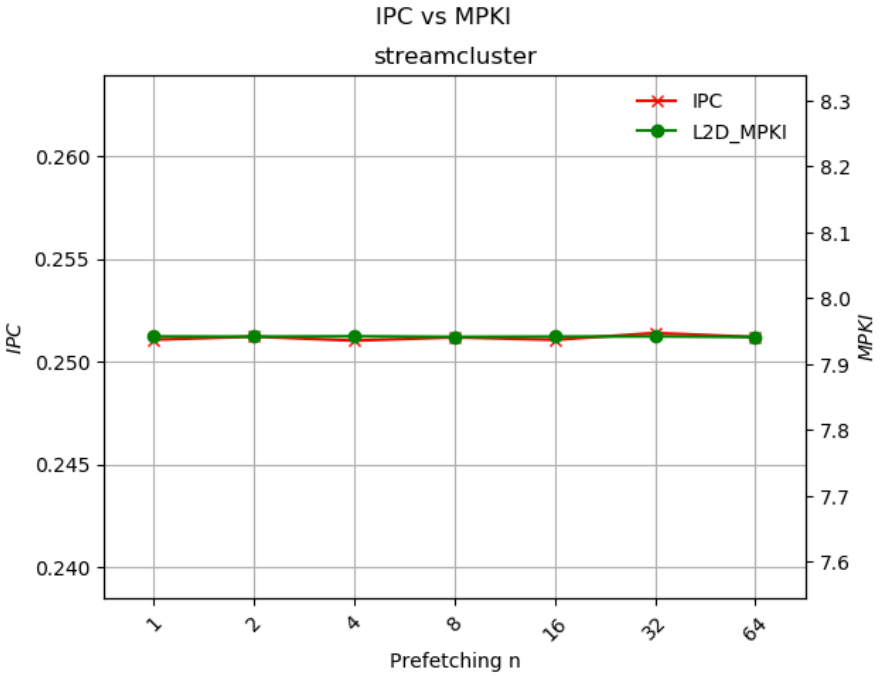
Βέλτιστη επιλογή είναι το 64. Η απόδοση μειώνεται για τις τιμες 2 και 4 και ύστερα αυξάνεται.

1.4.8 rtview



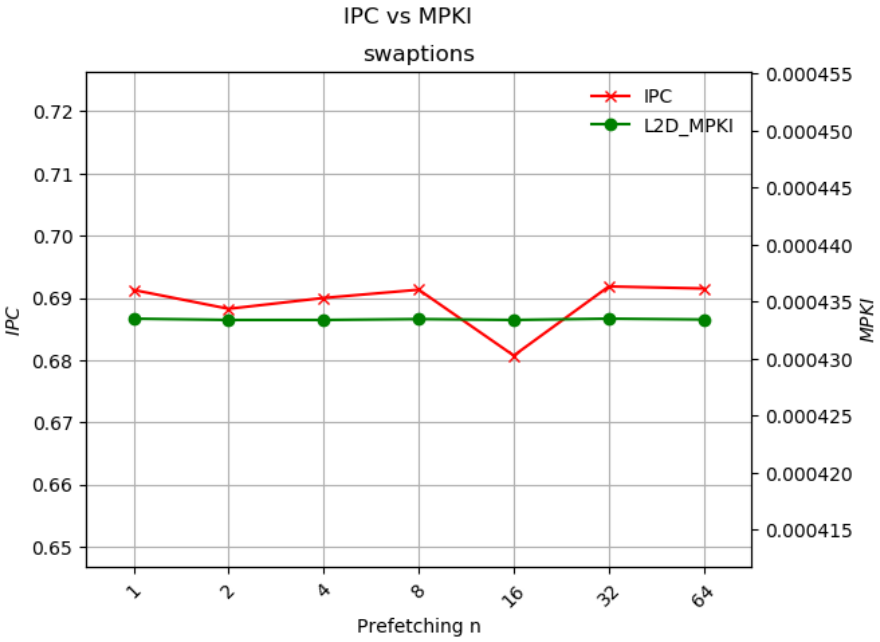
Το rtview μένει ανεπηρέαστο απο το L2 prefetching,καθώς όλες οι επιλογές έχουν την ίδια απόδοση.

1.4.9 streamcluster



Το streamcluster μένει ανεπηρέαστο απο το L2 prefetching,καθώς όλες οι επιλογές έχουν την ίδια απόδοση.

1.4.10 swaptions



Όλες οι επιλογές δίνουν περίπου την ίδια απόδοση πλην την επιλογής 16.

1.4.11 Σχολιασμός Prefetching

Κατά μέσο όρο το prefetching δεν οδήγησε σε μεγάλες μεταβολές της απόδοσης,με εξαίρεση το freqmine και το blackscholes.Σε κάποιες περιπτώσεις όταν αυξήθηκε το prefetching, βελτιώθηκε και η απόδοση.Στις περιπτώσεις όπου η απόδοση μειώθηκε κατά την αύξηση του prefetching είναι πιθανό τα μπλοκ που αφαιρέθηκαν να περιείχαν απαραίτητες για το πρόγραμμα πληροφορίες.

2 Ζητούμενο δεύτερο

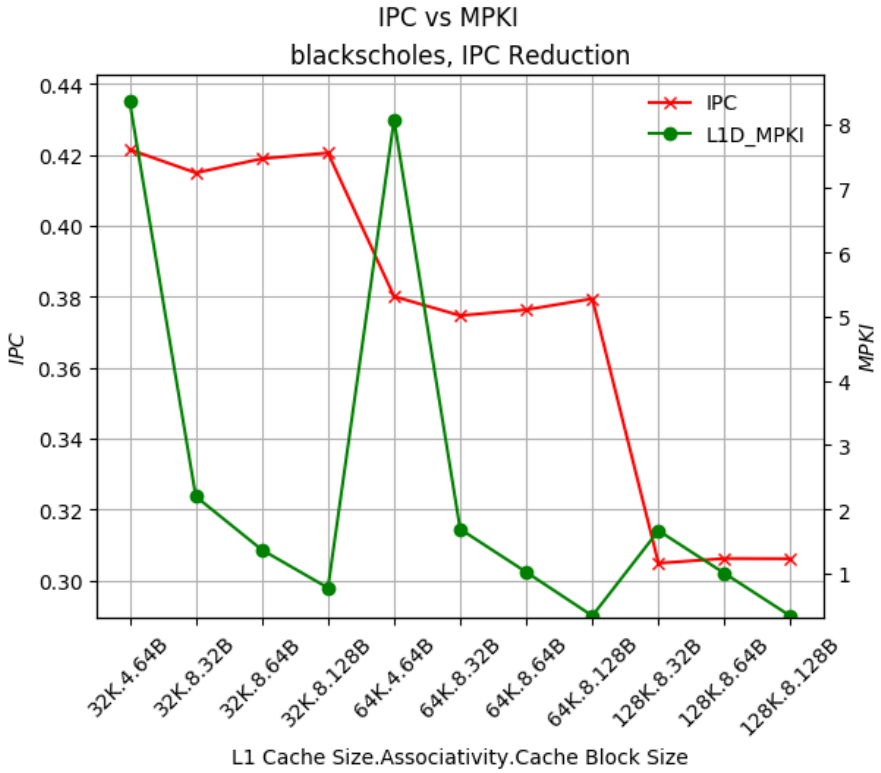
Οι διάφορες τροποποιήσεις στα μικροαρχιτεκτονικά χαρακτηριστικά του επεξεργαστή επιφέρουν συνήθως αλλαγές και στην διάρκεια του κύκλου.Για τις προσομοιώσεις που εκτελέσαμε προηγουμένως,θα θεωρήσουμε σε αυτό το σημείο πως κάθε διπλασιασμός του associativity ή του μεγέθους προκαλεί αύξηση του κύκλου κατά 5% ή 10% αντίστοιχα.

2.1 L1 cache

2.1.1 Παρατηρήσεις-Σχολιασμός Αποτελεσμάτων

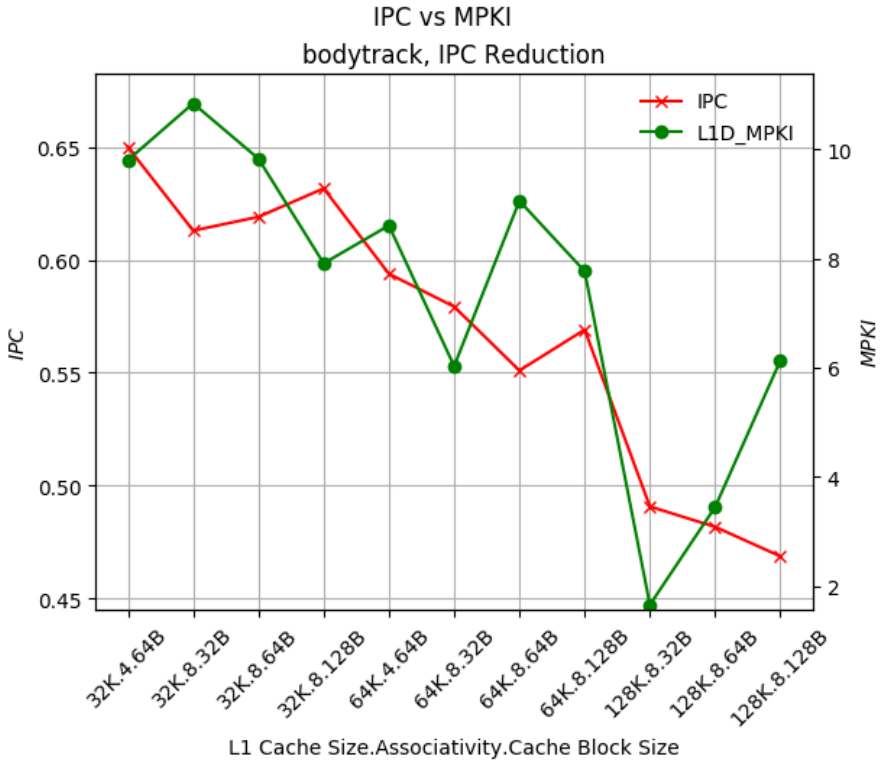
Στα αποτελέσματα που παρουσιάζονται παρακάτων παρατηρούμε πως οι παράγοντες όπου επηρέαζαν την απόδοση για την εκάστοτε προσομοίωση εξακολουθούν να υφίστανται,με την προσθήκη όμως του IPC Reduction.Από τα γραφήματα που παρουσιάζονται γίνεται ξεκάθαρο πως η μείωση του IPC στις περισσότερες περιπτώσεις υπερνικά τα οφέλη ενός μεγαλύτερου μεγέθους ή associativity.Παρατηρούμε λοιπόν πως στα περισσότερα benchmarks ο βέλτιστος συνδυασμός είναι αυτός που περιέχει τη μικρότερη cache,το μικρότερο associativity και μεγάλο/επαρκές μέγεθος μπλοκ.Σίγουρα πάντως σε κάθε περίπτωση οι βέλτιστοι συνδυασμοί δεν αποτελούν συνδυασμό μεγάλου μεγέθους cache,associativity και block size όπως ήταν προηγουμένως,αλλά είναι πλέον δυσκολότερο να τους προβλέψουμε εκ των προτέρων.

2.1.2 blackscholes



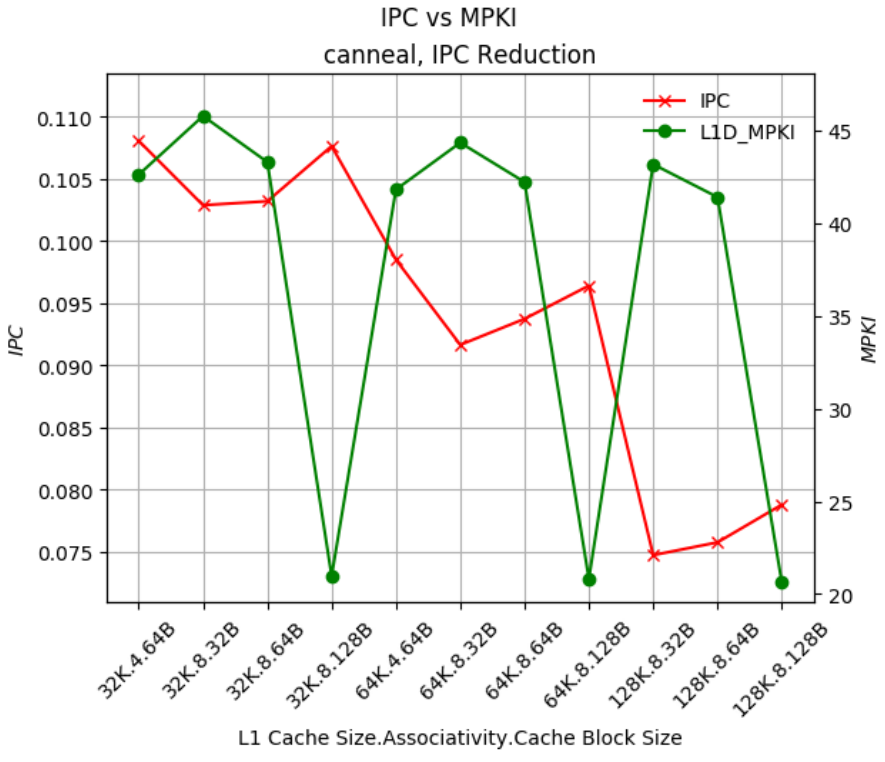
Βέλτιστοι συνδυασμοί : (32,4,64), (32,8,64). Το blackscholes έχει ως σημαντικότερο παράγοντα το μέγεθος της cache και το μέγεθος του μπλοκ.

2.1.3 bodytrack



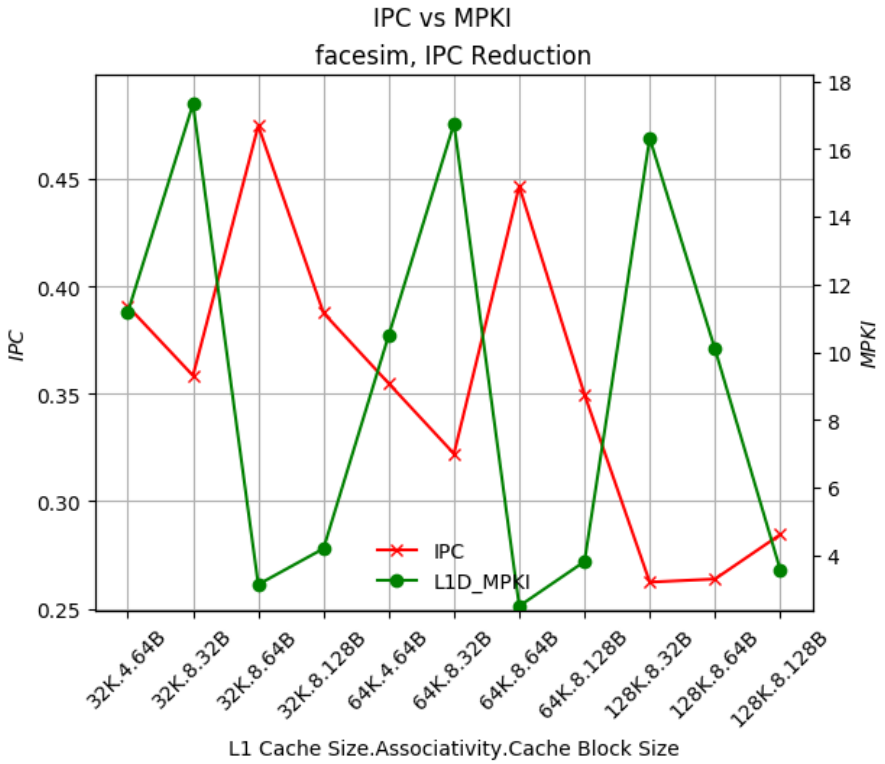
Βέλτιστος συνδυασμός : (32,4,64)
Το bodytrack έχει ως σημαντικότερο παράγοντα το μέγεθος της cache και το μέγεθος του μπλοκ.

2.1.4 canneal



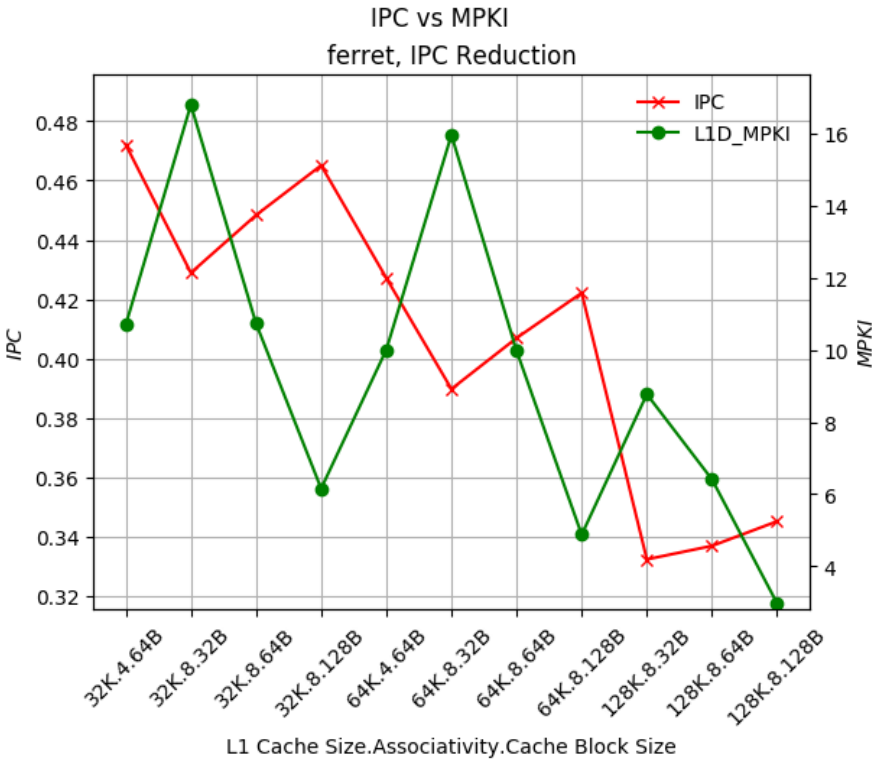
Βέλτιστοι συνδυασμοί : (32,4,64), (32,8,64)

2.1.5 facesim



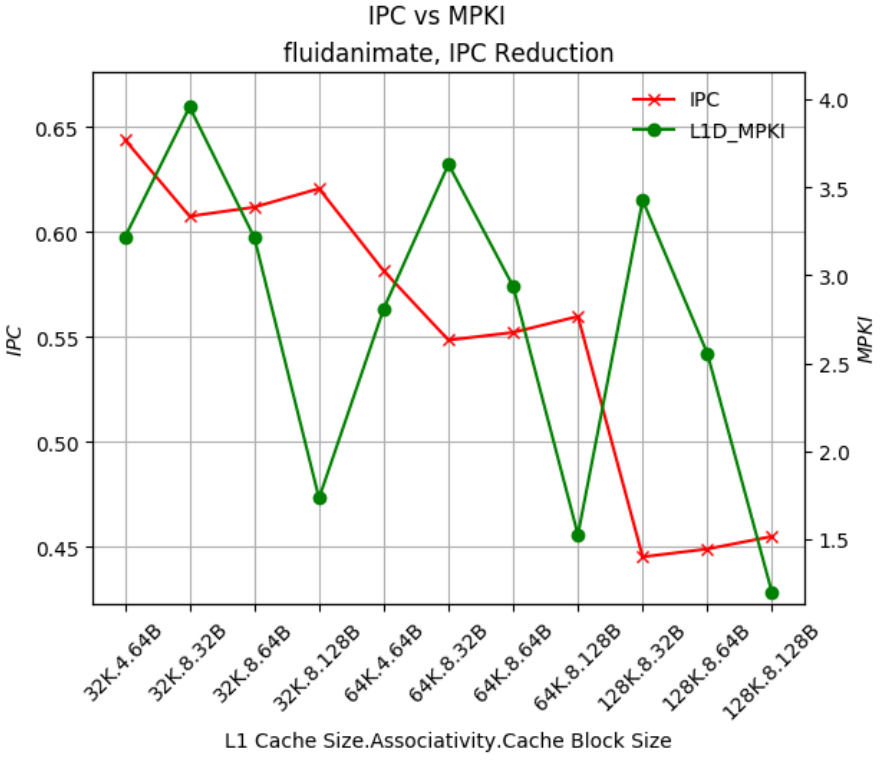
Βέλτιστος συνδυασμός : (32,8,64) .Σημαντικότερος παράγοντας αποτελεί το μέγεθος του μπλοκ.

2.1.6 ferret



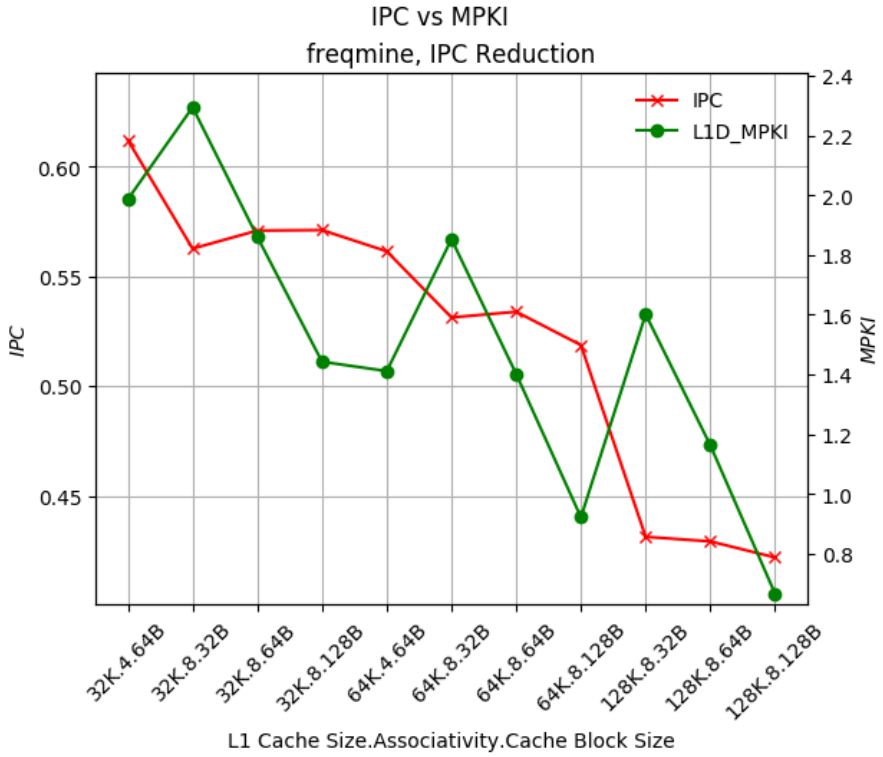
Βέλτιστος συνδυασμός : (32,4,64)

2.1.7 fluidanimate



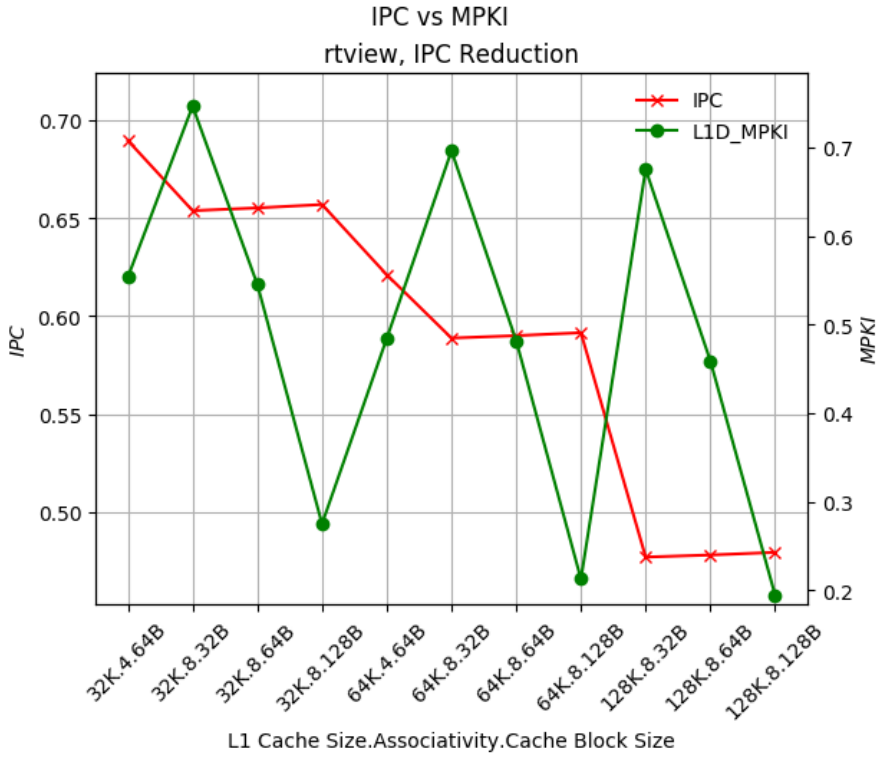
Βέλτιστος συνδυασμός : (32,4,64)

2.1.8 freqmine



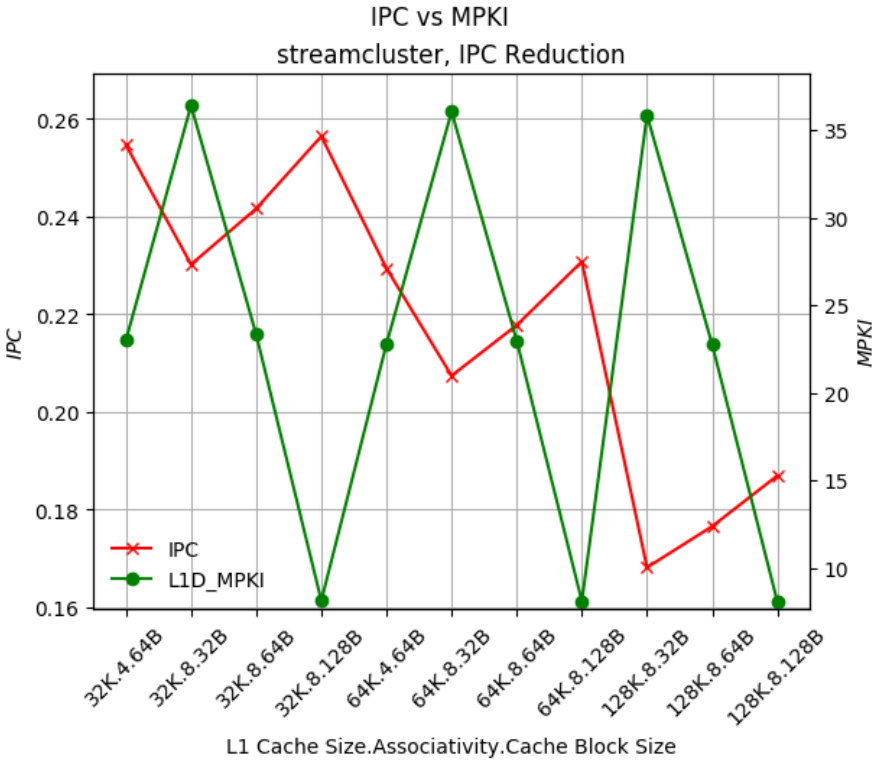
Βέλτιστος συνδυασμός : (32,4,64)

2.1.9 rtview



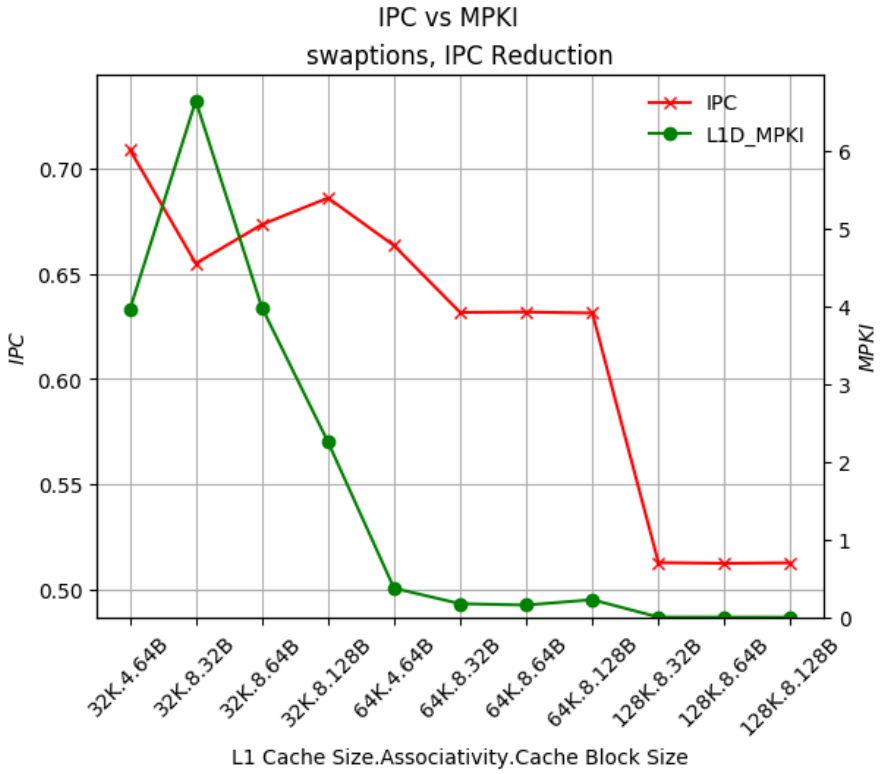
Βέλτιστος συνδυασμός : (32,4,64)

2.1.10 streamcluster



Βέλτιστος συνδυασμός : (32,8,128)
Το streamcluster έχει ισχυρό spatial locality και ένα μεγαλύτερο μέγεθος μπλοκ μειώνει τα compulsory misses.

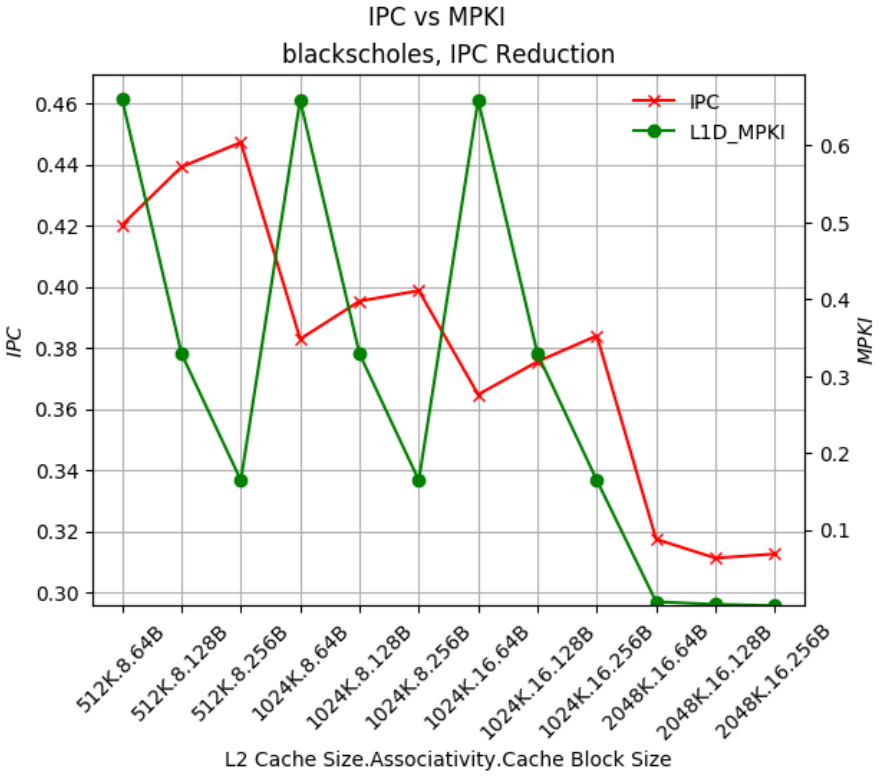
2.1.11 swaptions



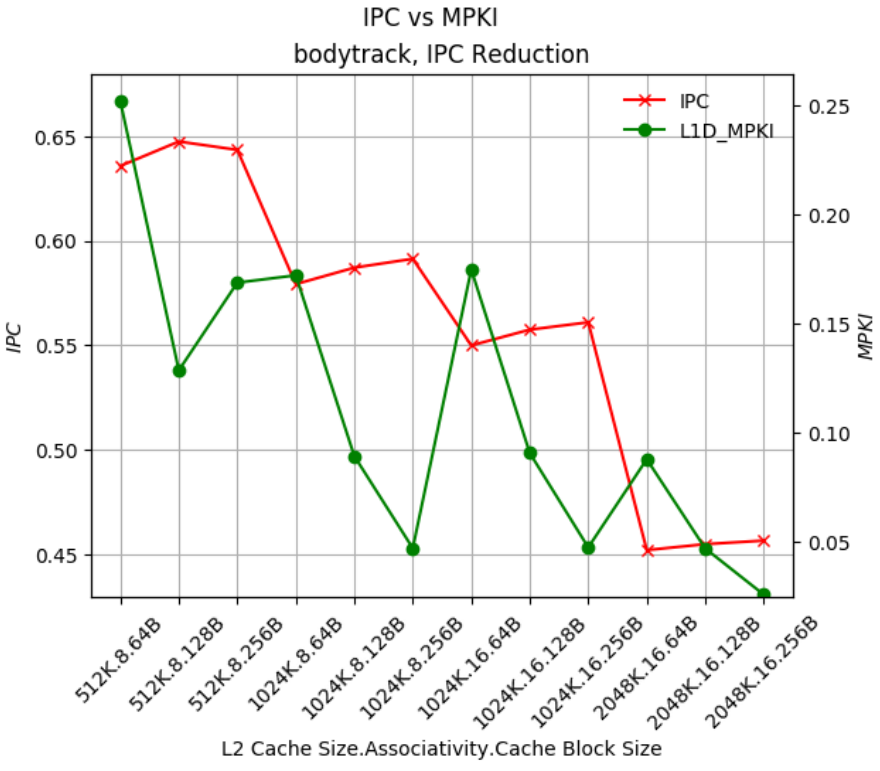
Βέλτιστος συνδυασμός : (32,4,64)

2.2 L2 cache

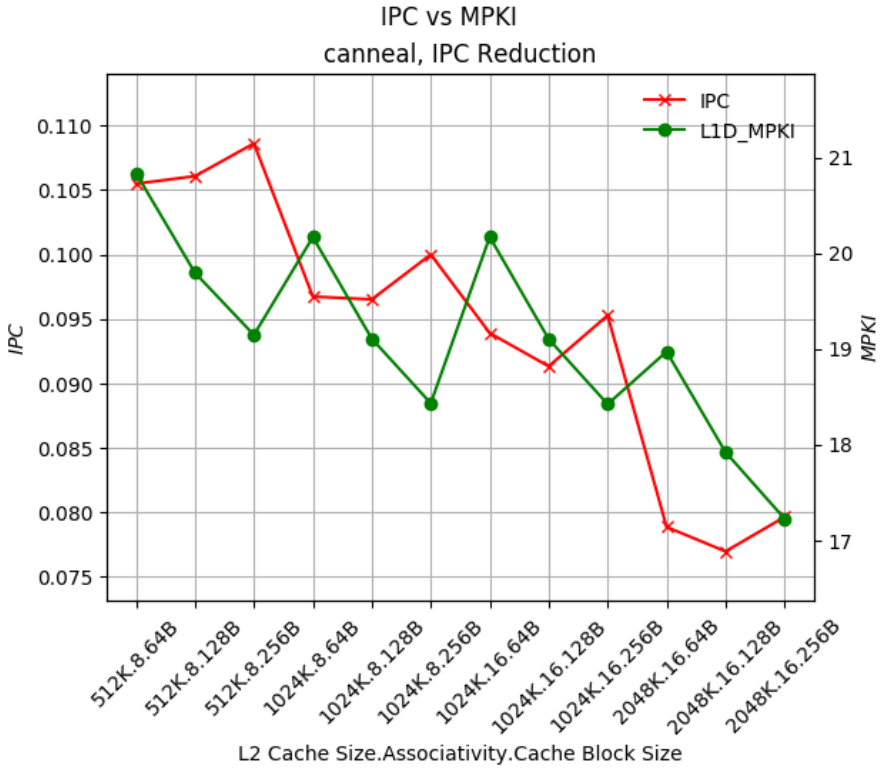
2.2.1 blackscholes



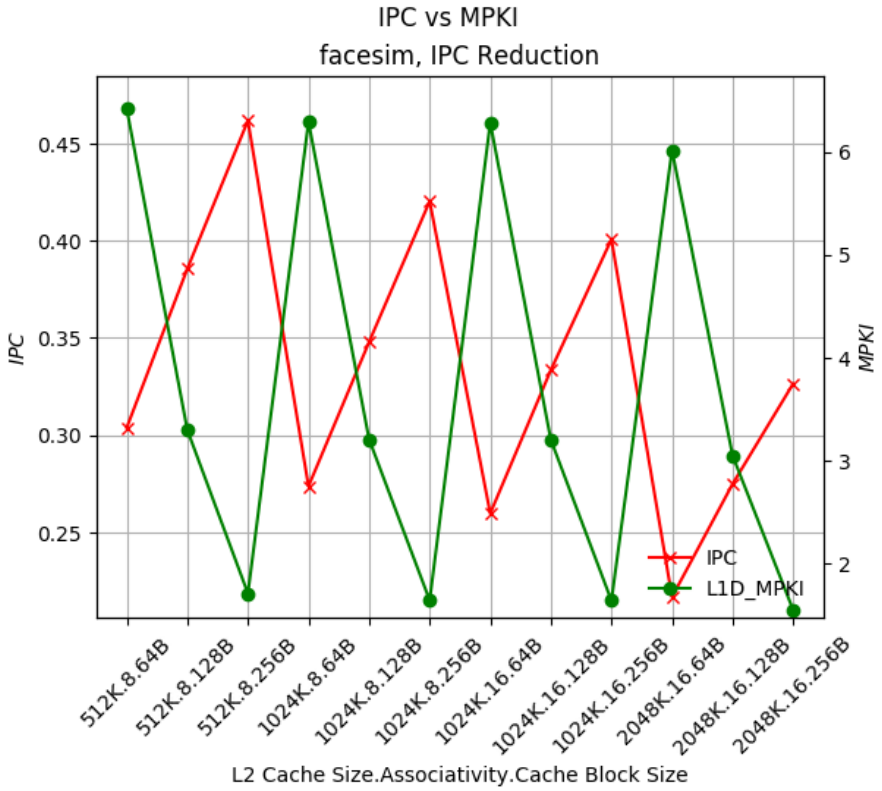
2.2.2 bodytrack



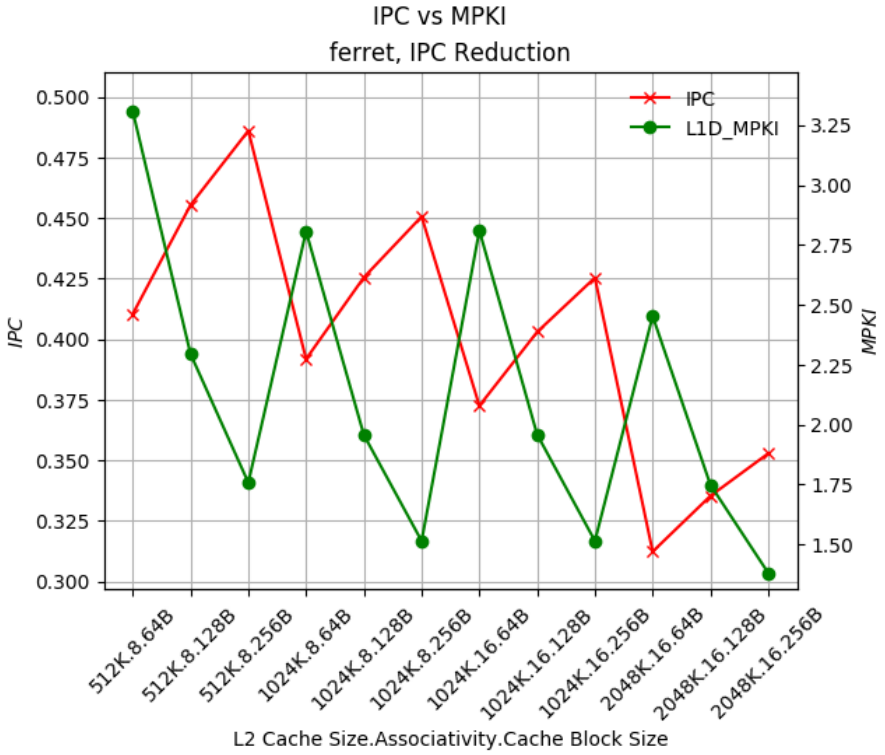
2.2.3 canneal



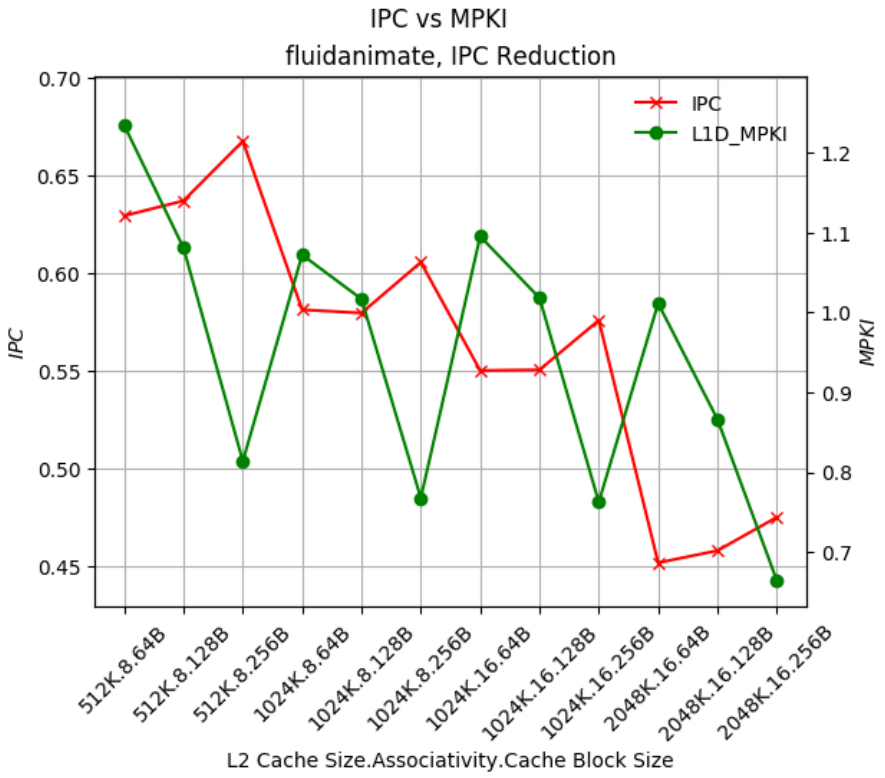
2.2.4 facesim



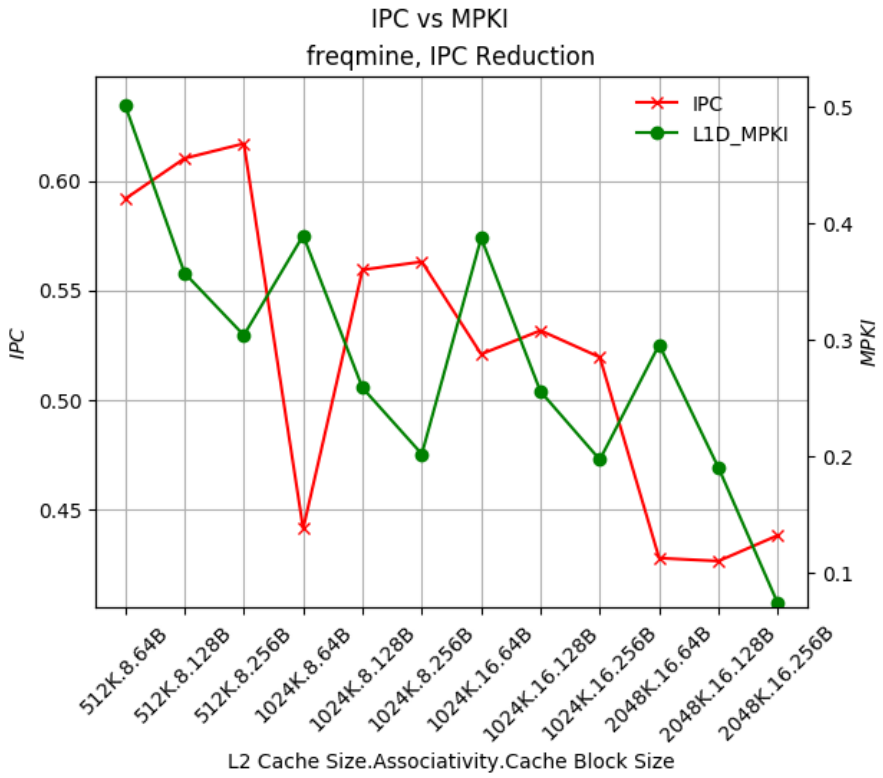
2.2.5 ferret



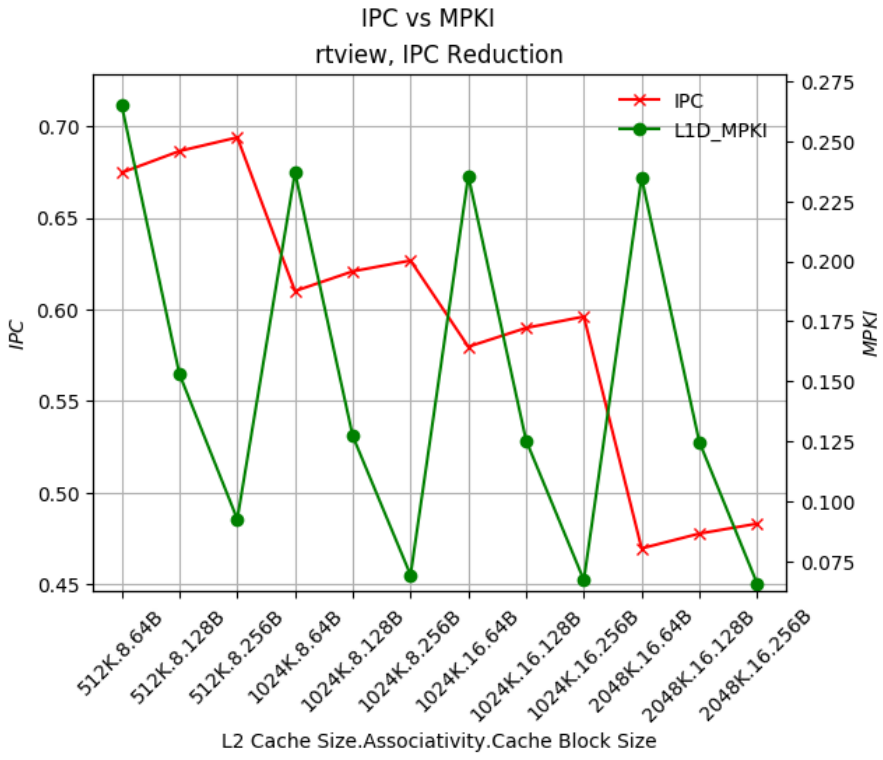
2.2.6 fluidanimate



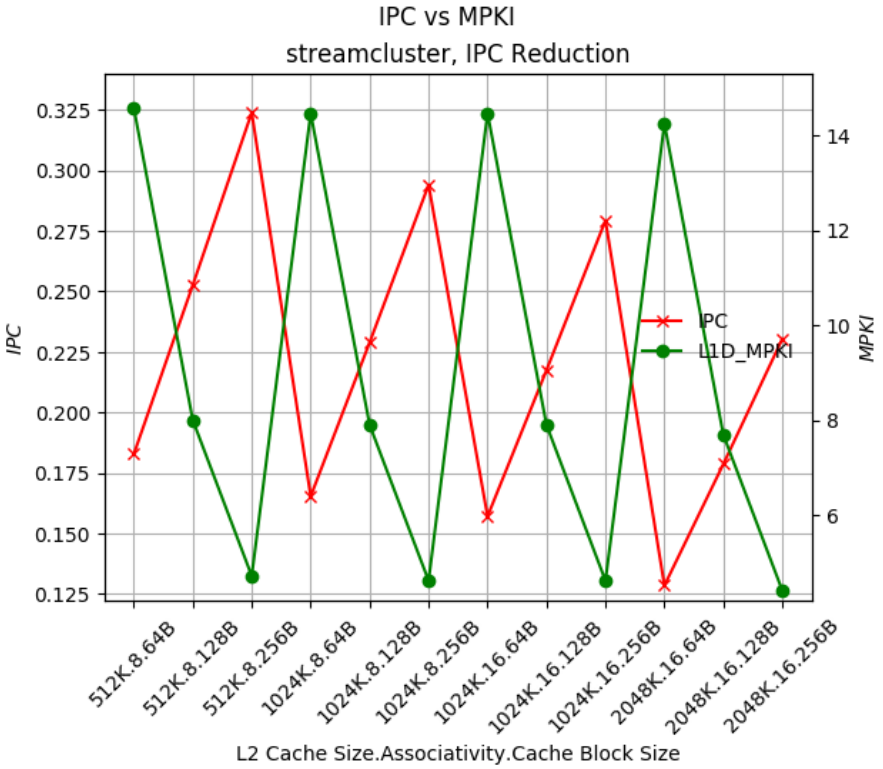
2.2.7 freqmine



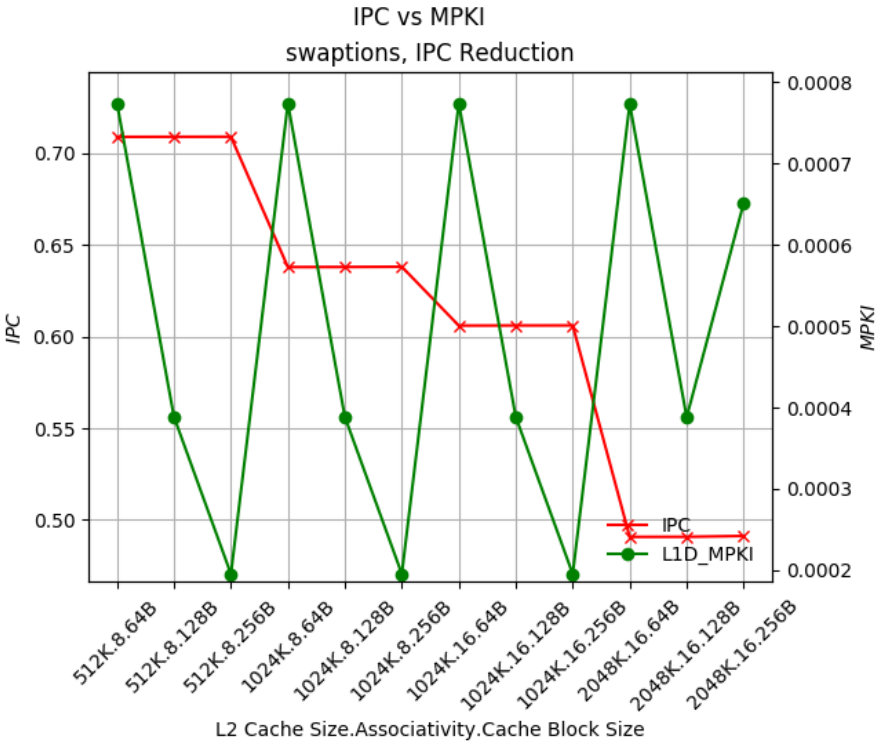
2.2.8 rtview



2.2.9 streamcluster

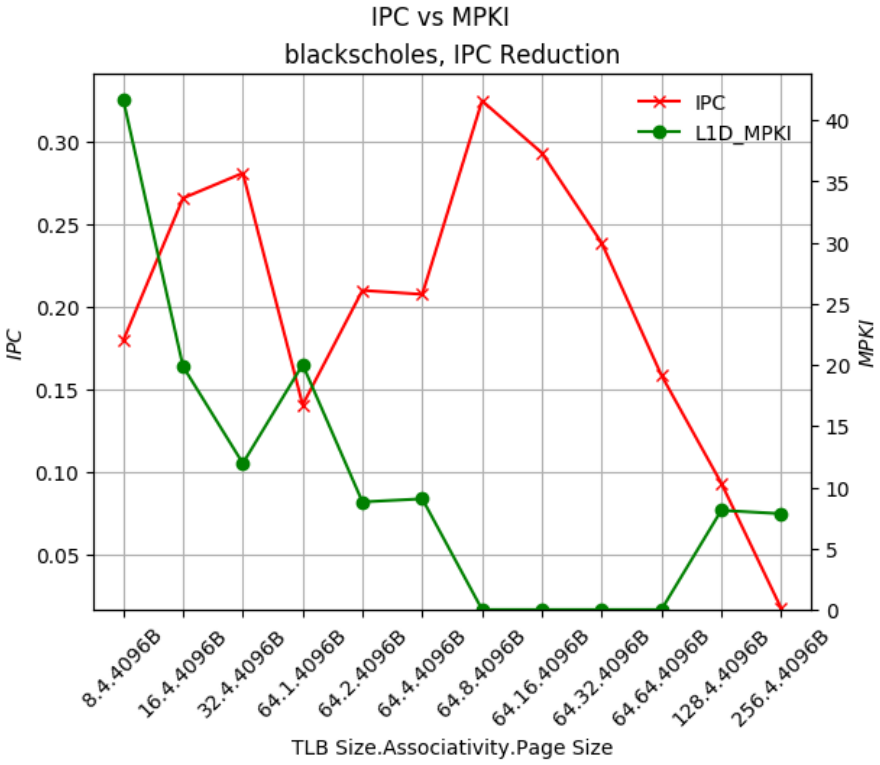


2.2.10 swaptions

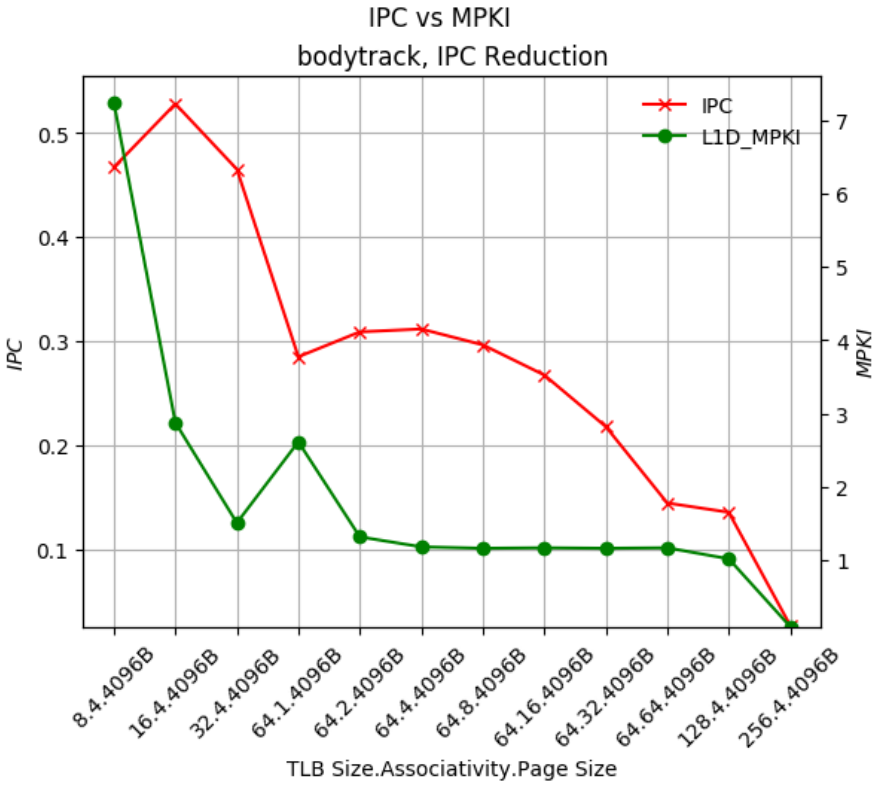


2.3 TLB

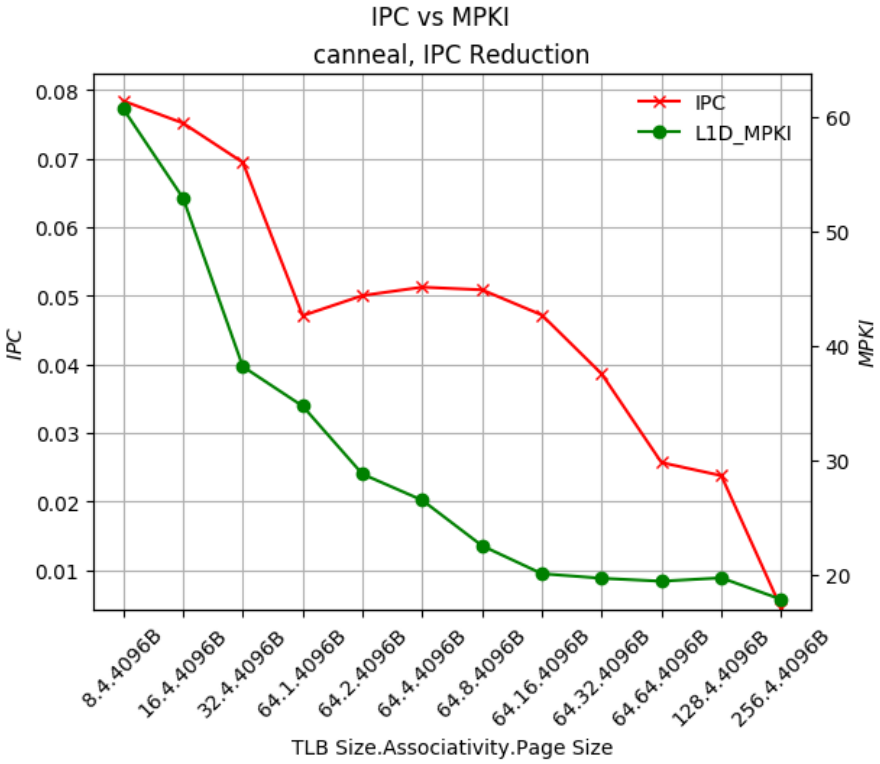
2.3.1 blackscholes



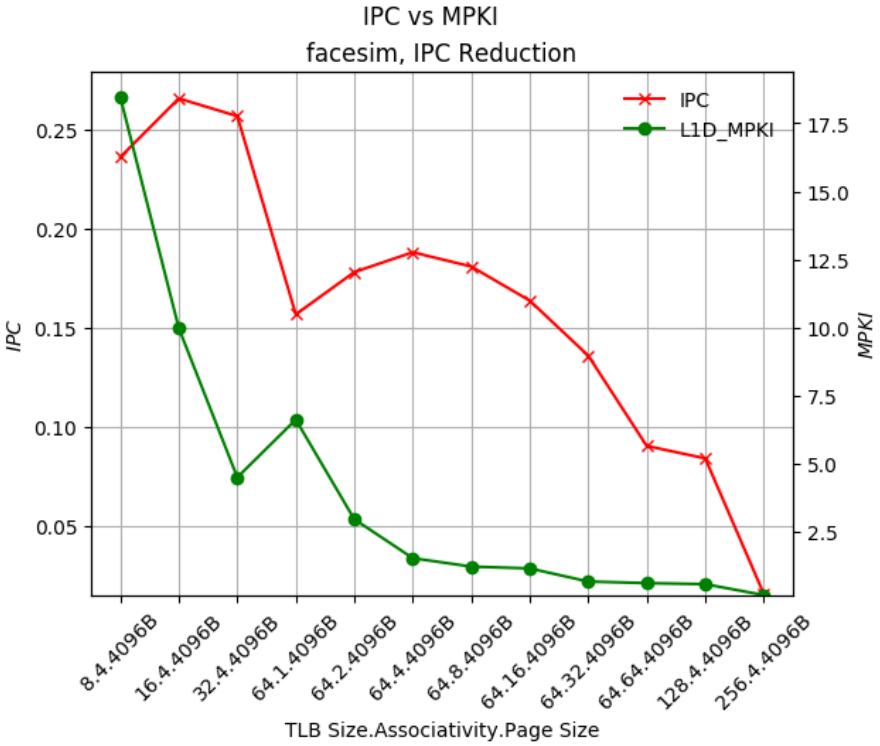
2.3.2 bodytrack



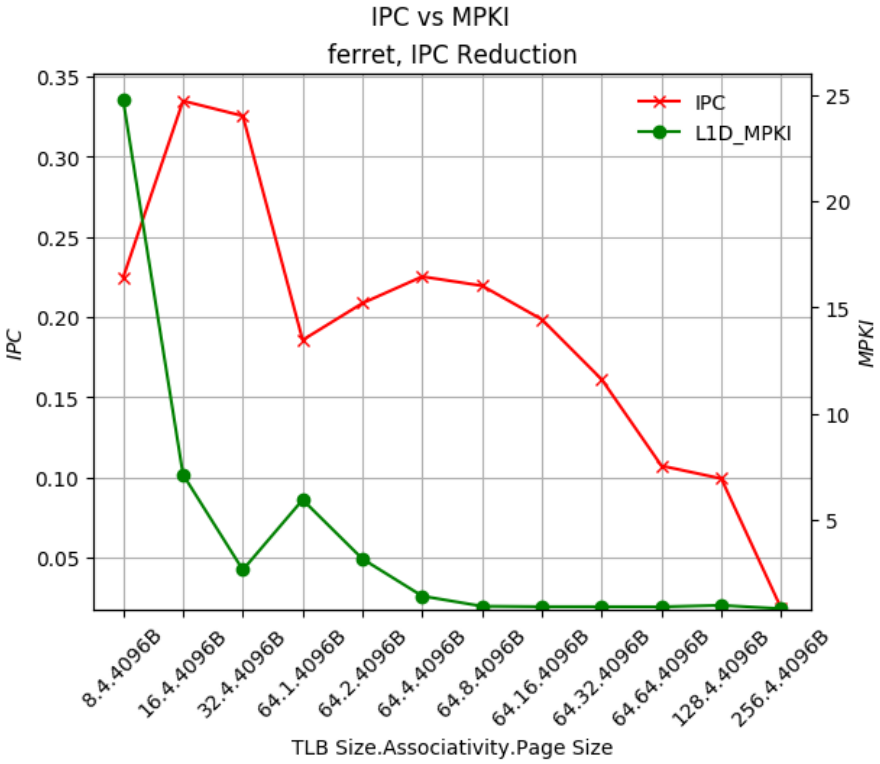
2.3.3 canneal



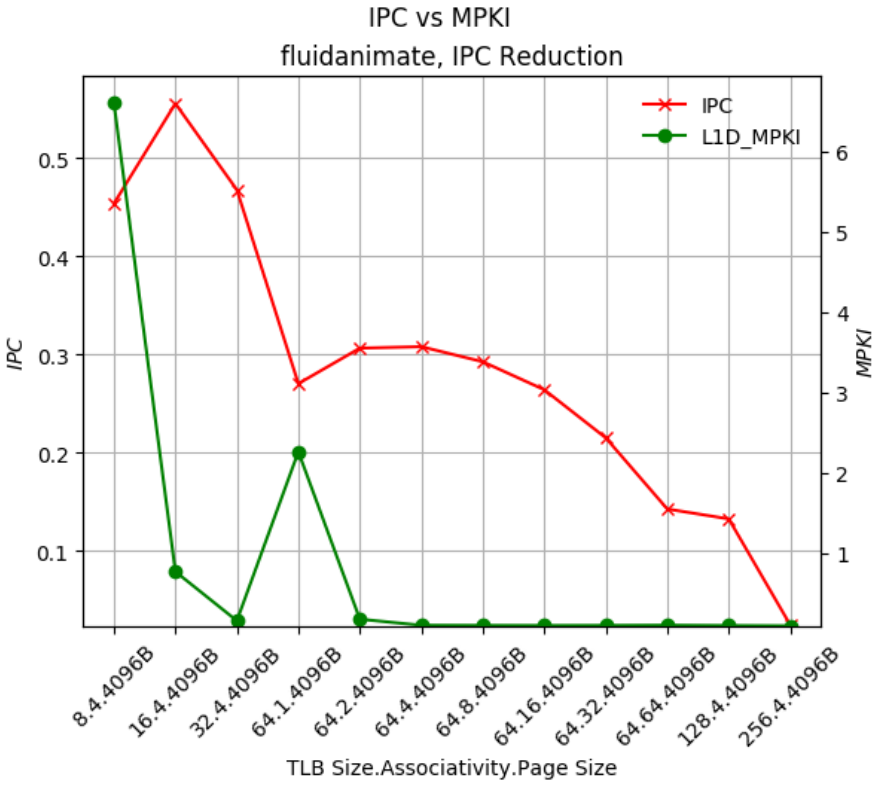
2.3.4 facesim



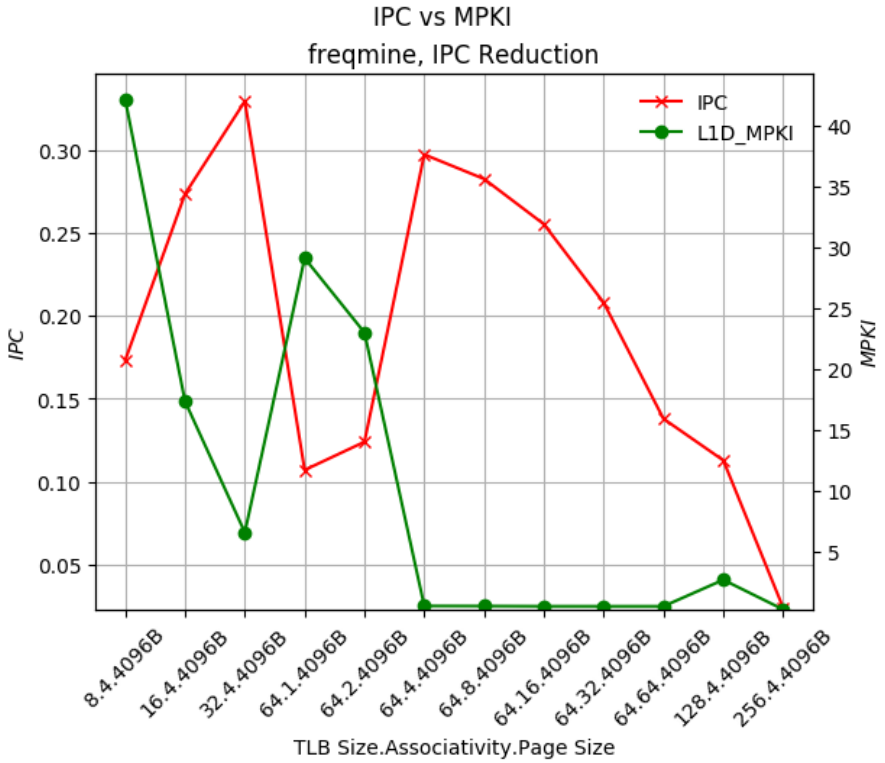
2.3.5 ferret



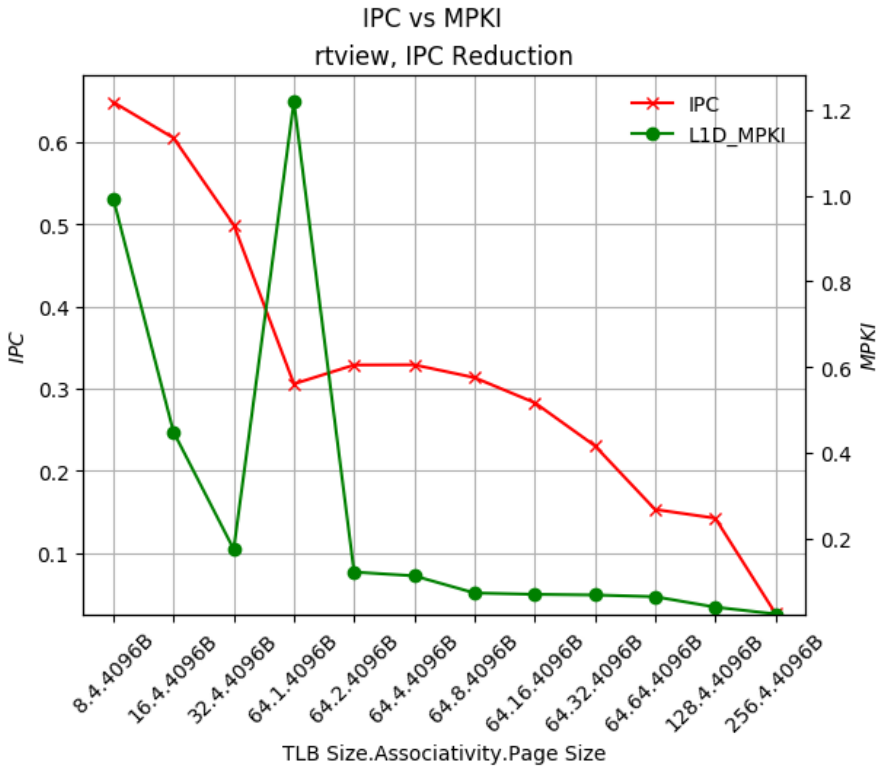
2.3.6 fluidanimate



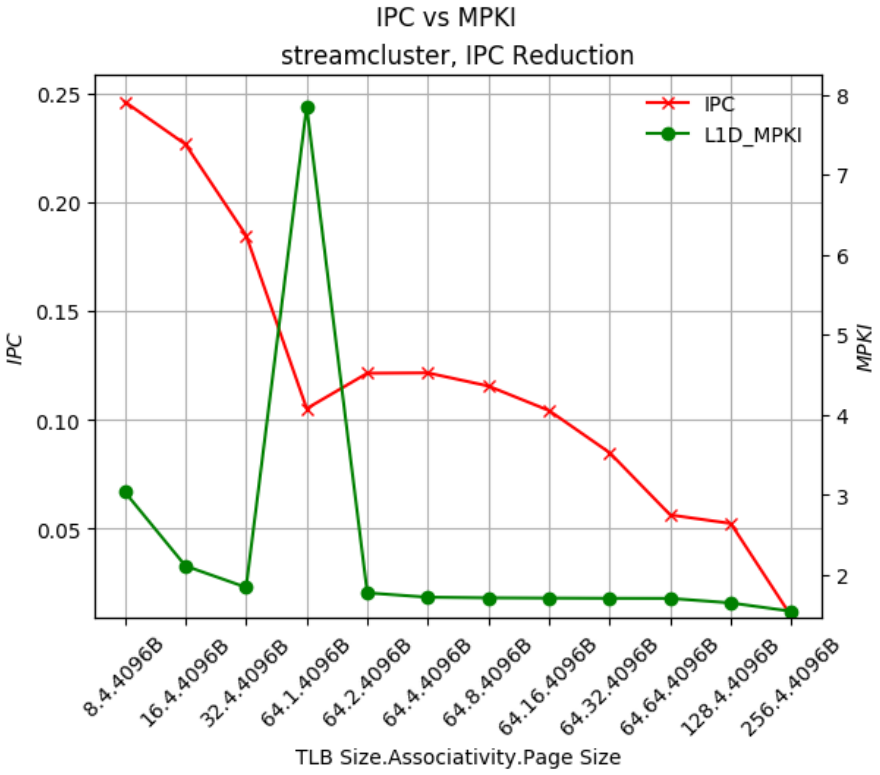
2.3.7 freqmine



2.3.8 rtview



2.3.9 streamcluster



2.3.10 swaptions