ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

2η ΑΣΚΗΣΗ

Ακαδημαϊκό Έτος 2019-2020

Εξάμηνο 8ο

Χρόνης Σάκος

03116168

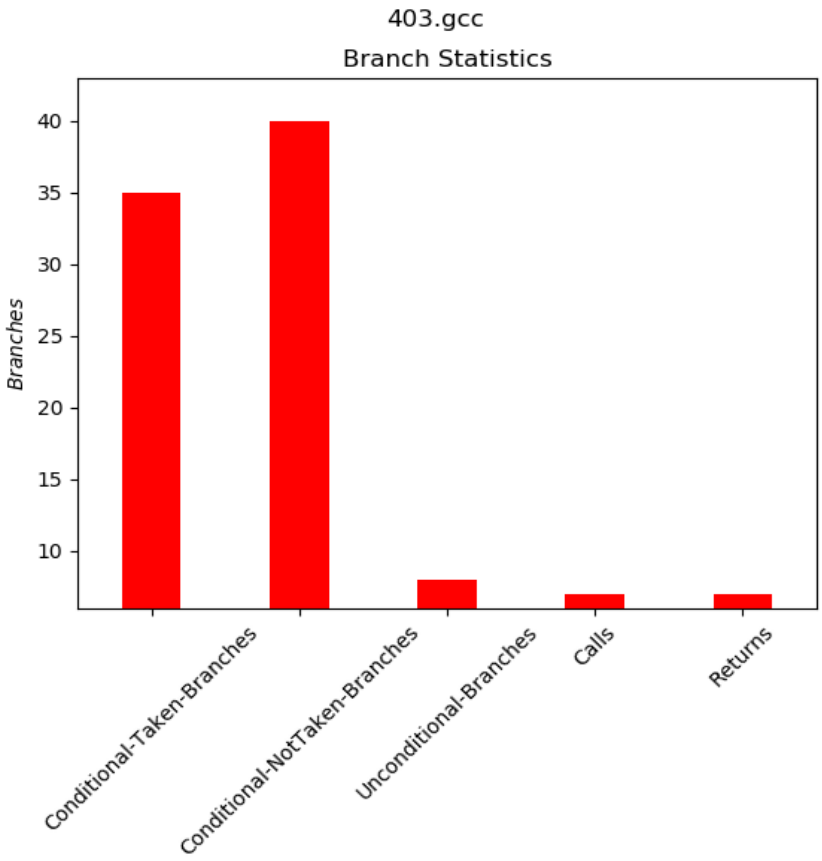
4. Πειραματική Αξιολόγηση

Στα πλαίσια της εργασίας αυτής, θα διερευνηθεί η επίδραση διαφορετικών συστημάτων πρόβλεψης εντολών άλματος καθώς και η αξιολόγησή τους με δεδομένο το διαθέσιμο χώρο πάνω στο τσιπ.

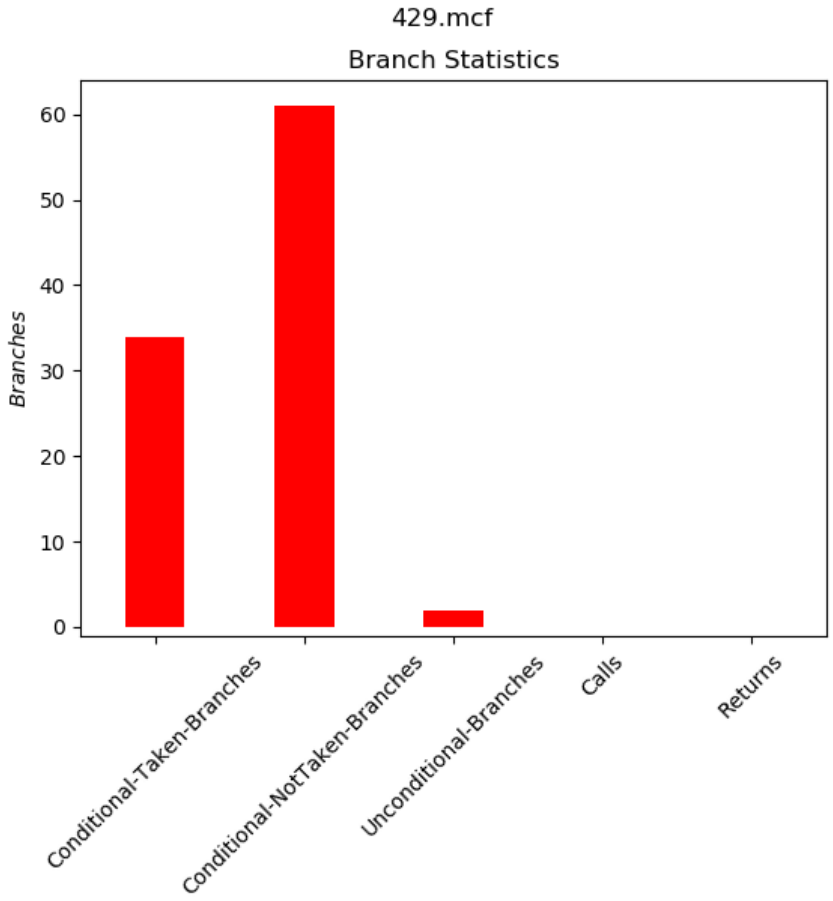
4.1 Μελέτη εντολών άλματος

Στόχος είναι η συλλογή στατιστικών για τις εντολές άλματος που εκτελούνται από τα benchmarks και η παρουσίασή τους σε κοινό διάγραμμα για κάθε ένα από αυτά.

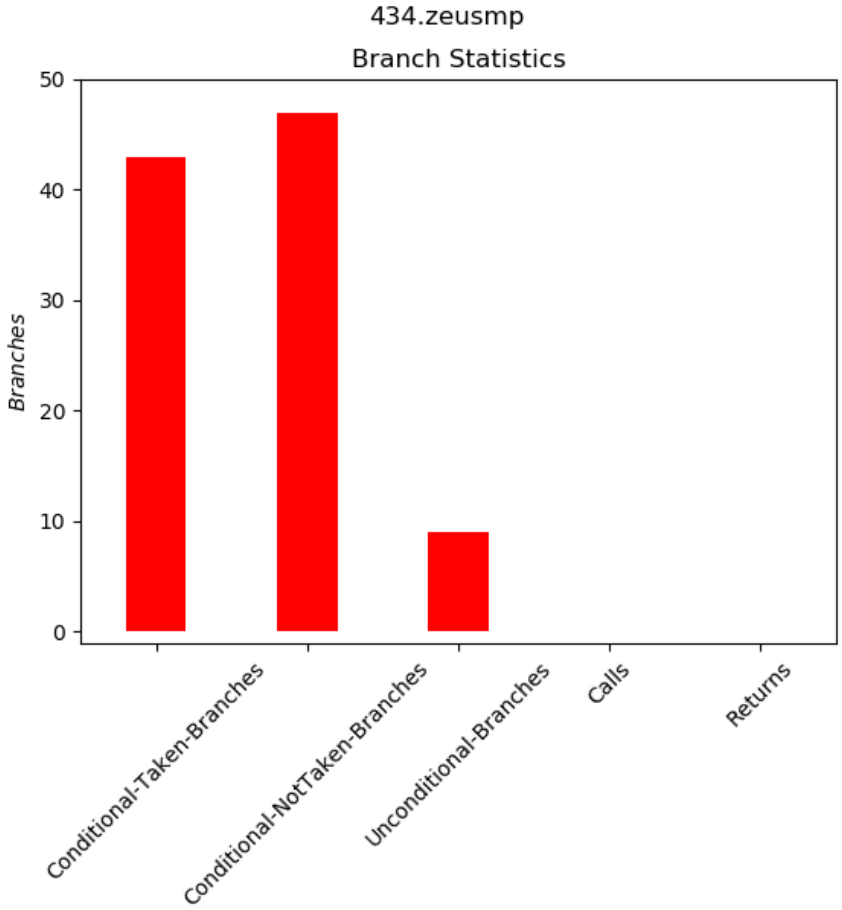
1) 403.gcc



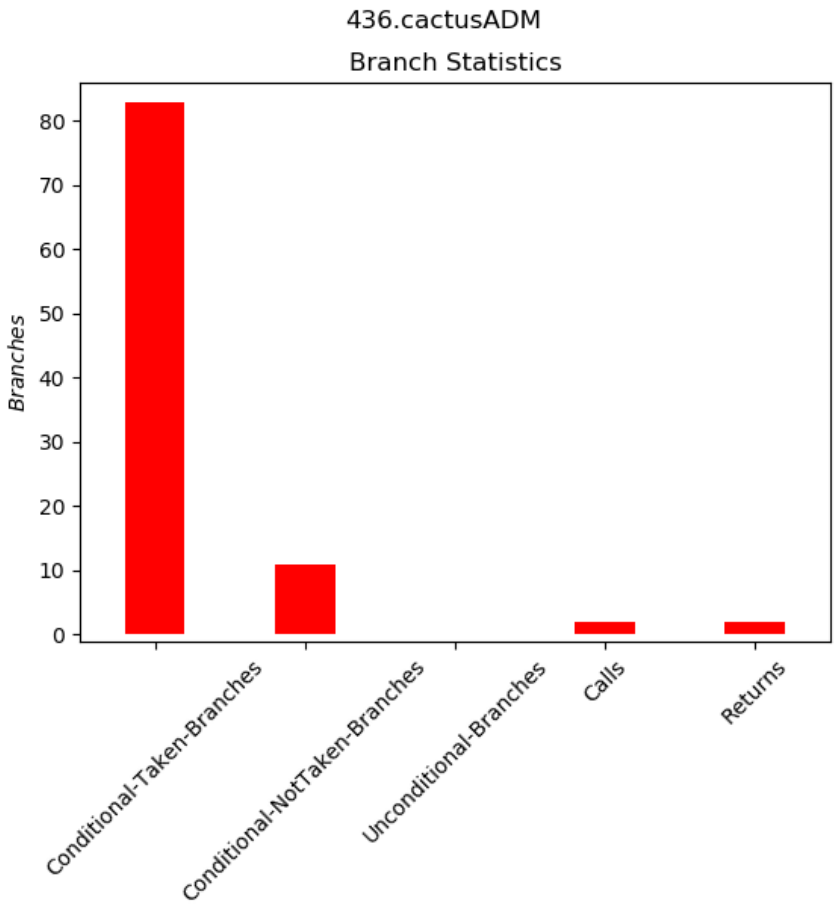
2) 429.mcf



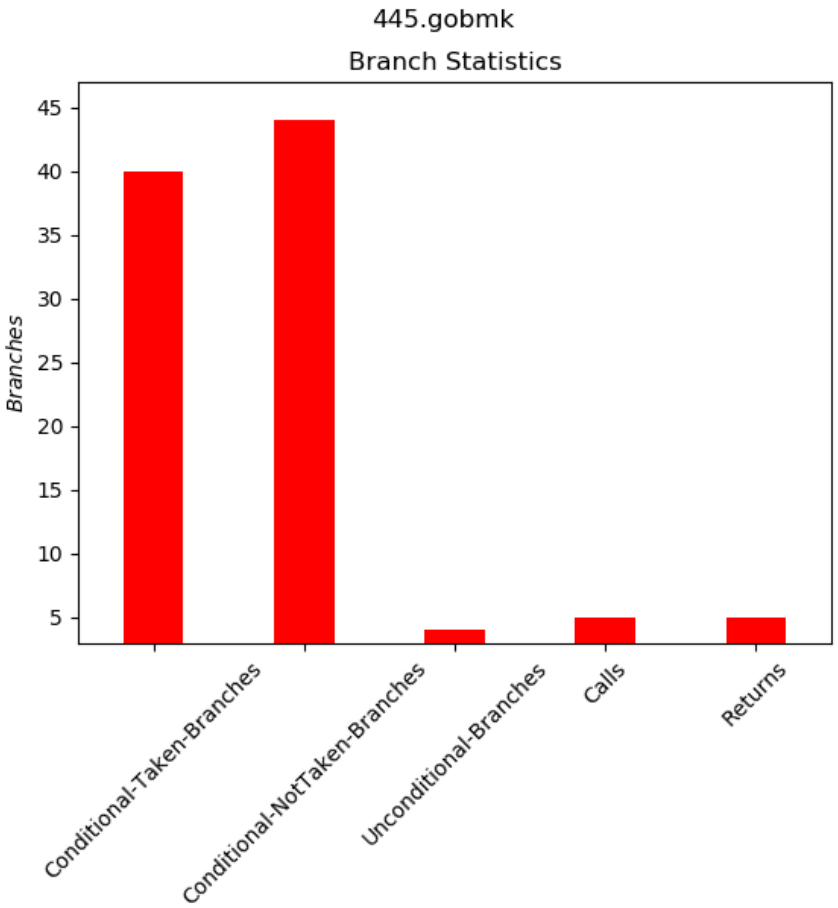
3) 434.zeusmp



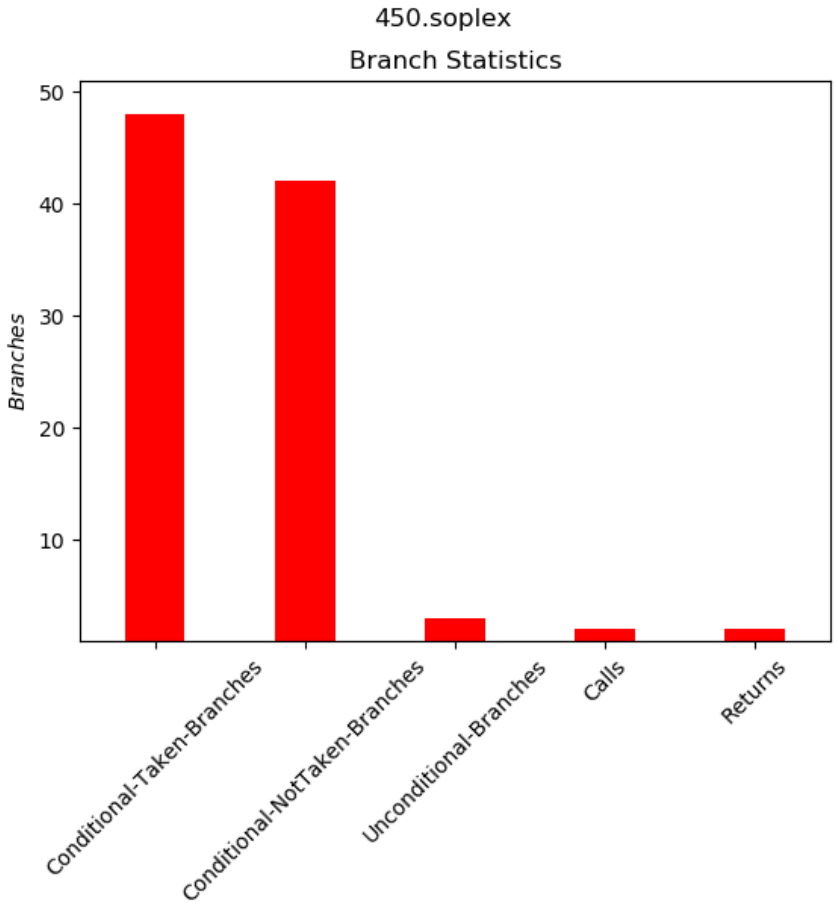
4) 436.cactusADM



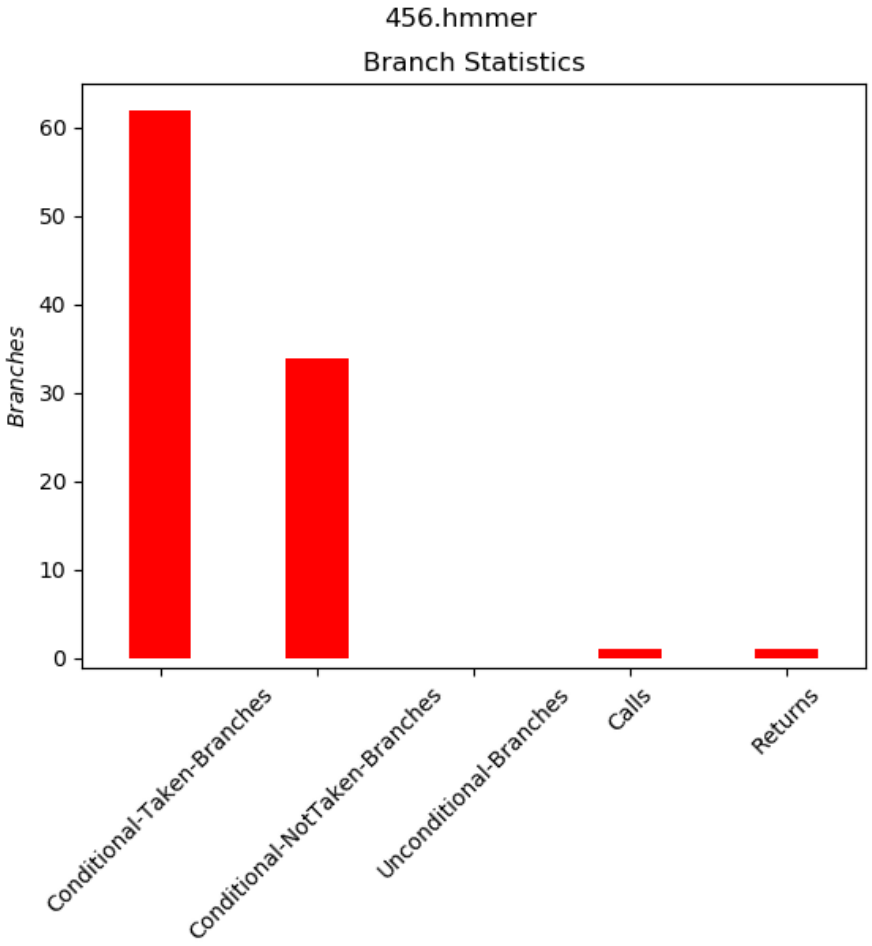
5) 445.gobmk



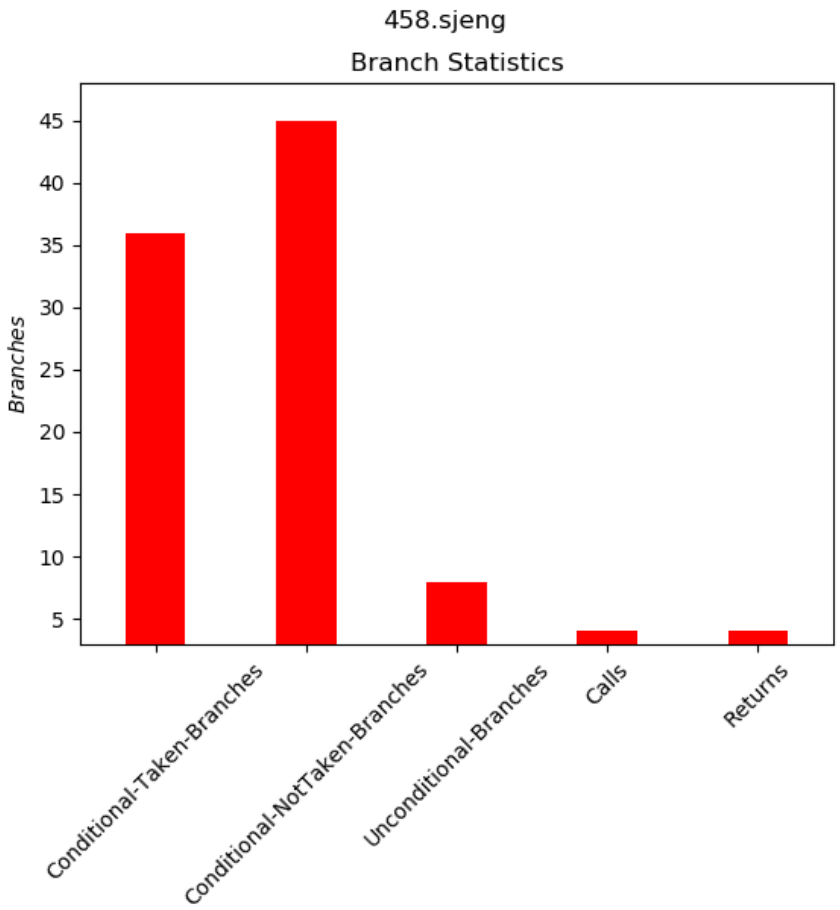
6) 450.soplex



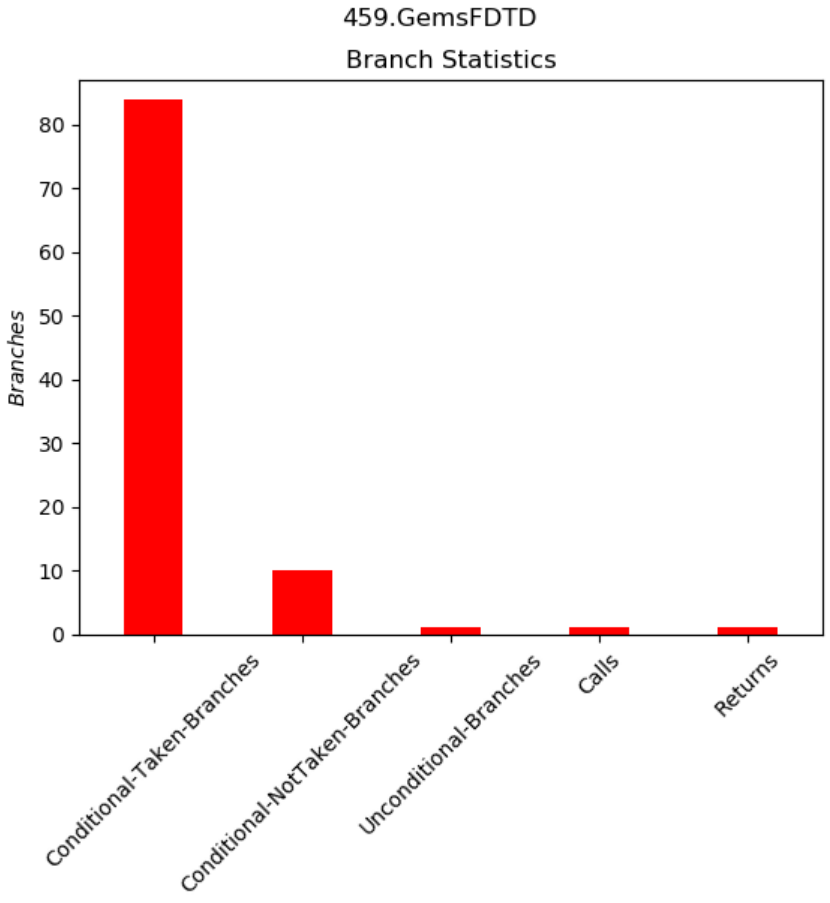
7) 456.hmmer



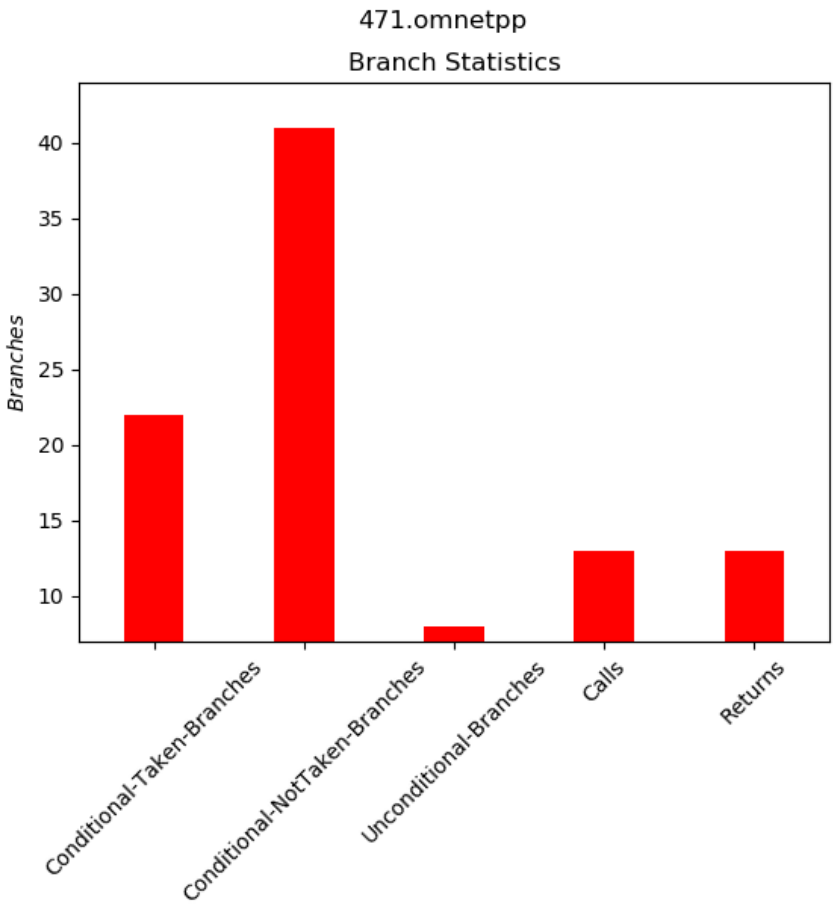
8) 458.sjeng



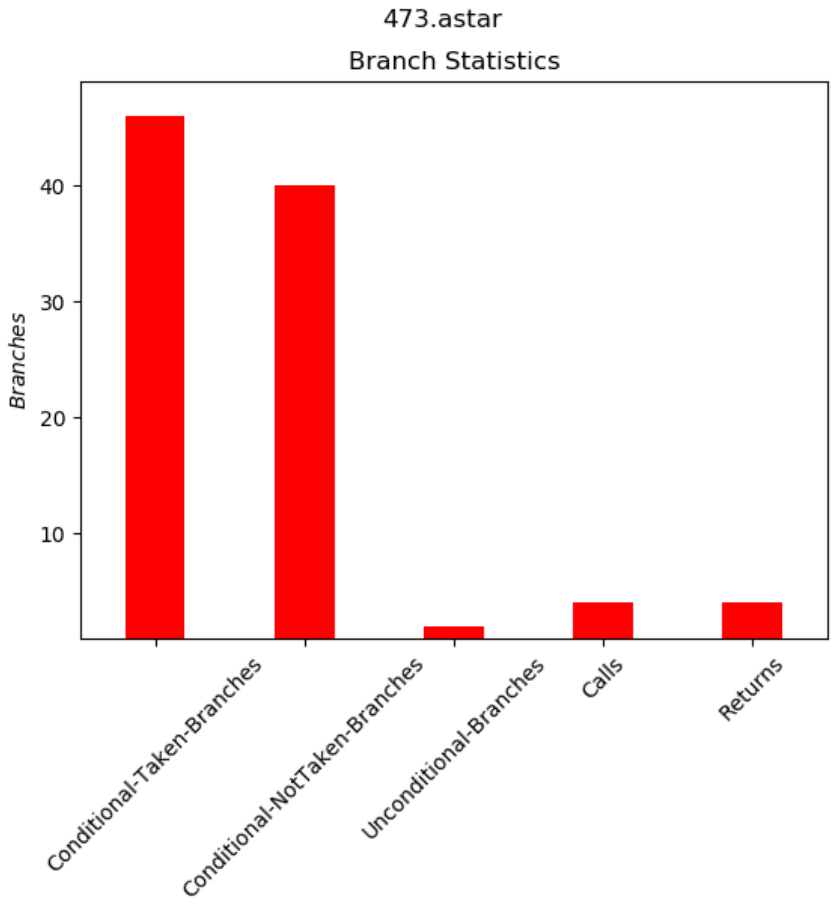
9) 459.GemsFDTD



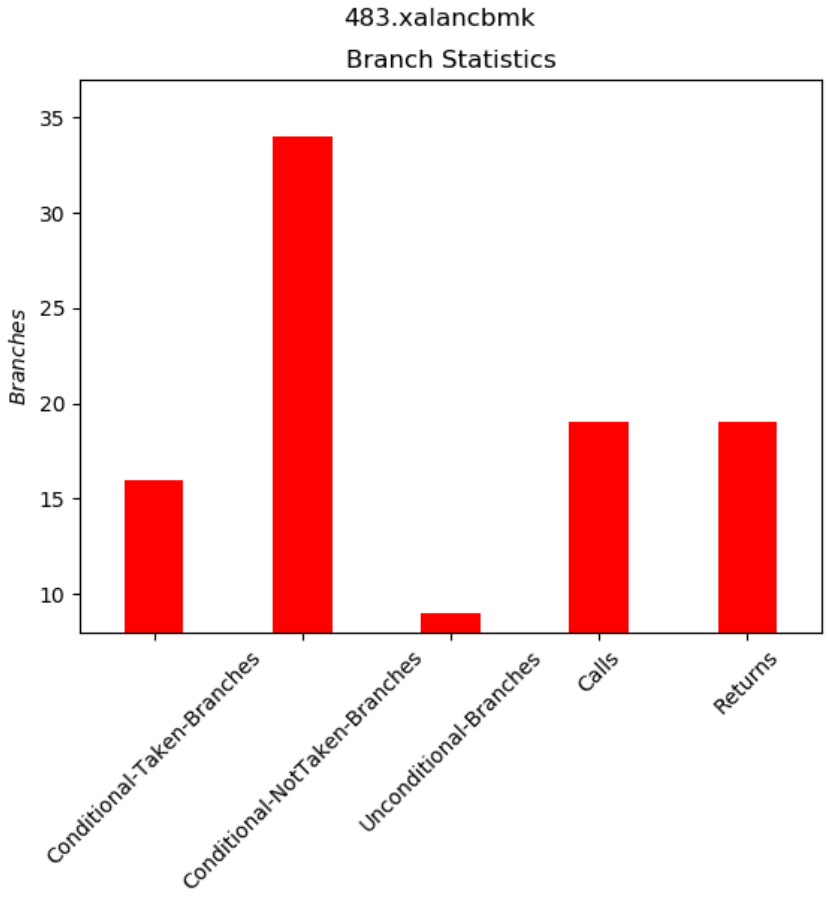
10) 471.omnetpp



11) 473.astar



12) 483.xalancbmk



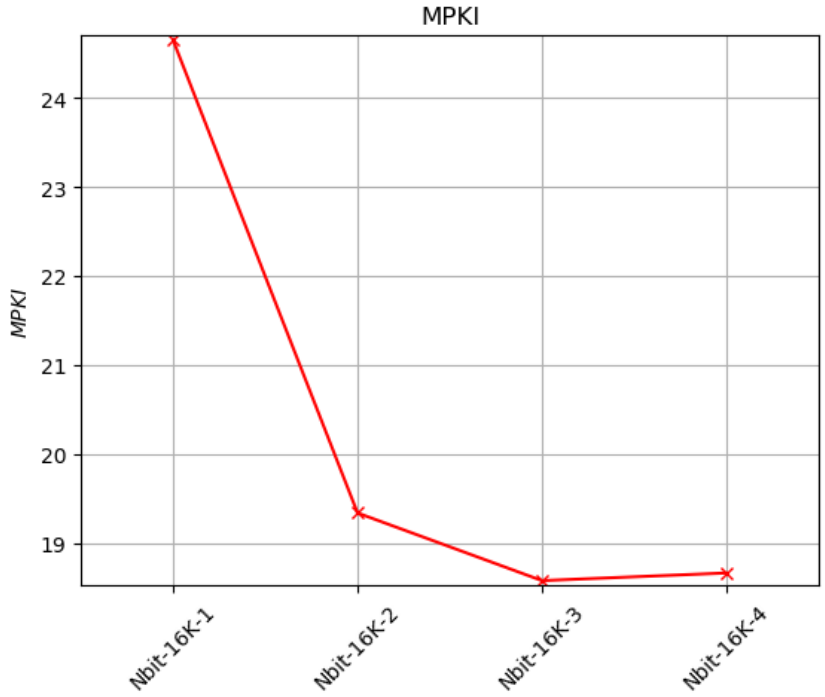
Από τα παραπάνω διαγράμματα διαπιστώνουμε ότι τα Conditional-Taken και τα NotTaken Branches καταλαμβάνουν το μεγαλύτερο ποσοστό διακλαδώσεων σε κάθε benchmark. Ο ρόλος των Unconditional-Branches και των Calls και Returns είναι περιορισμένος, ενώ ακόμα παρατηρούμε ότι τα Calls και Returns έχουν πάντοτε ίδιο ποσοστό (ή σχεδόν ίδιο).

4.2 Μελέτη των N-bit predictors

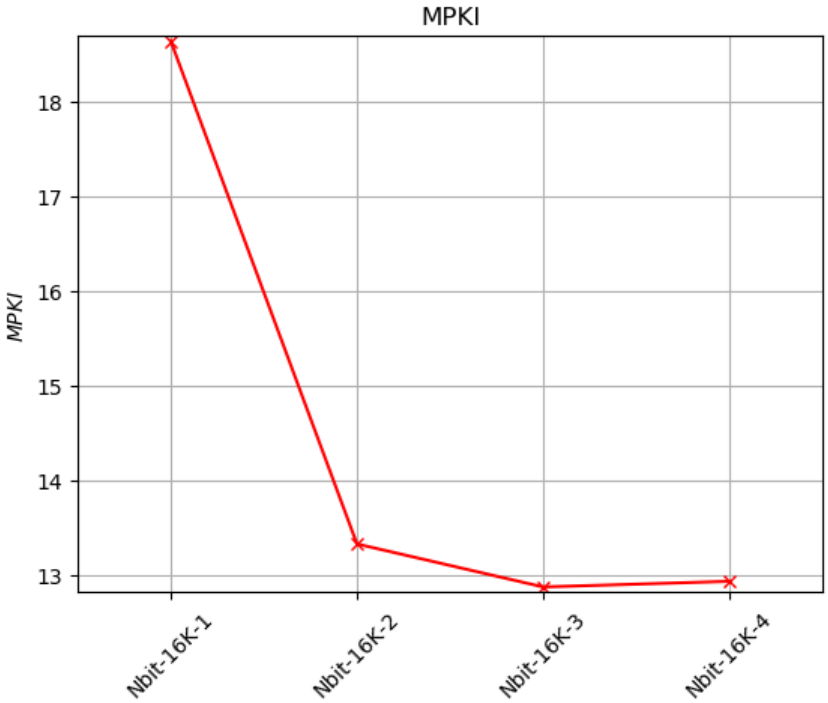
Θα μελετήσουμε την απόδοση των n-bits predictors.

i) Διατηρώντας σταθερό τον αριθμό των BHT entries και ίσο με 16K, προσομοιώνουμε τους n-bit predictors, για N=1, 2, 3, 4.

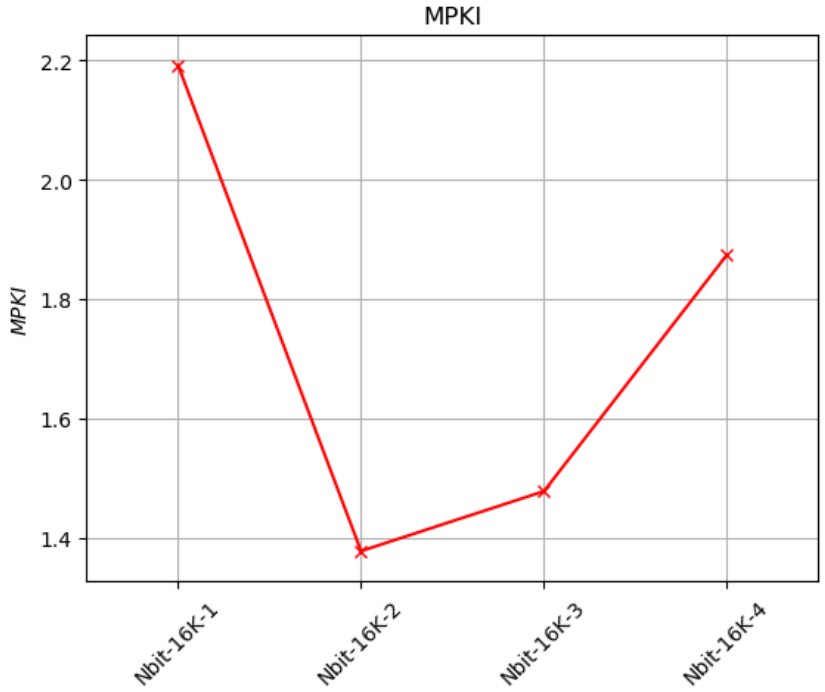
1) 403.gcc



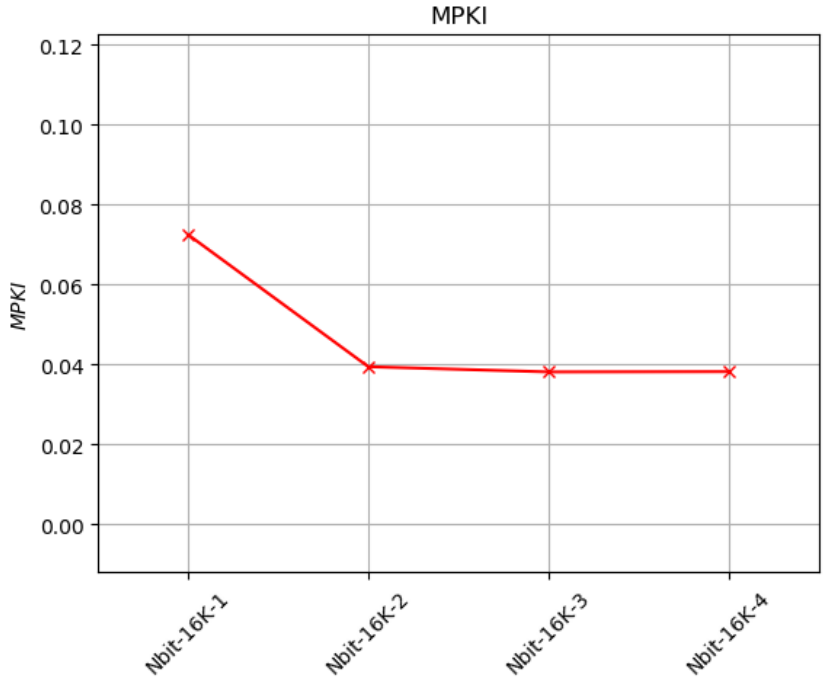
2) 429.mcf



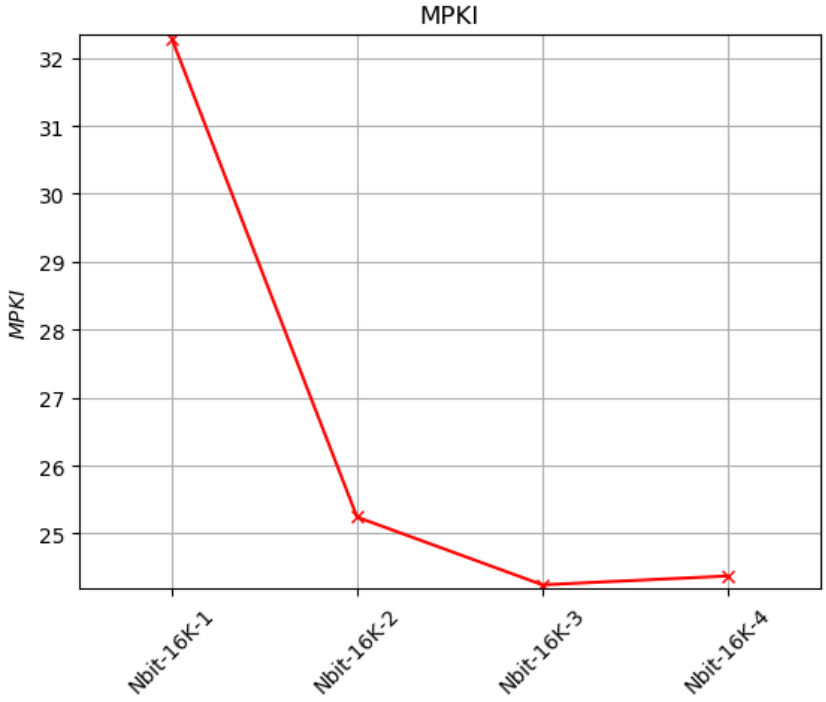
3) 434.zeusmp



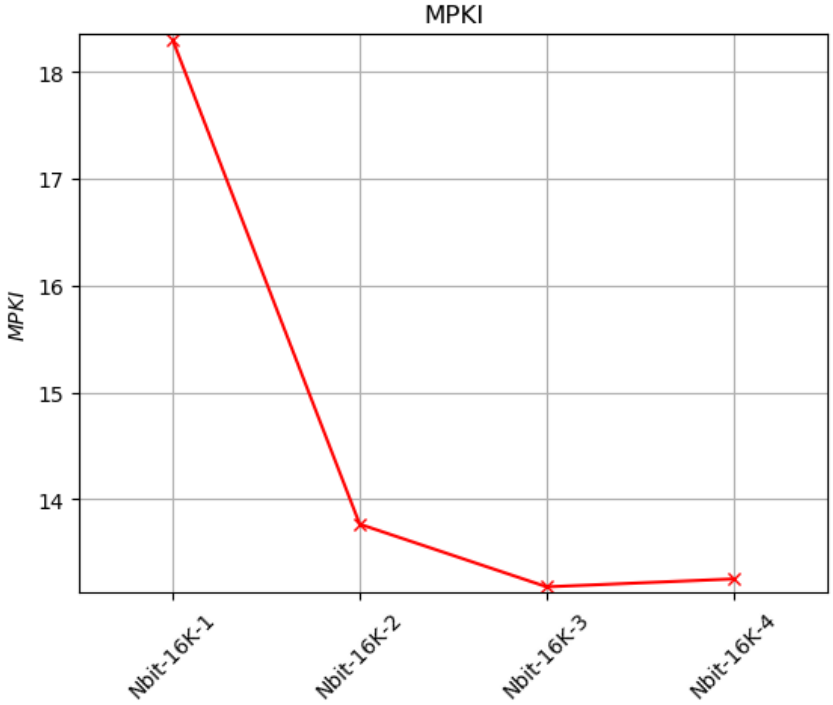
4) 436.cactusADM



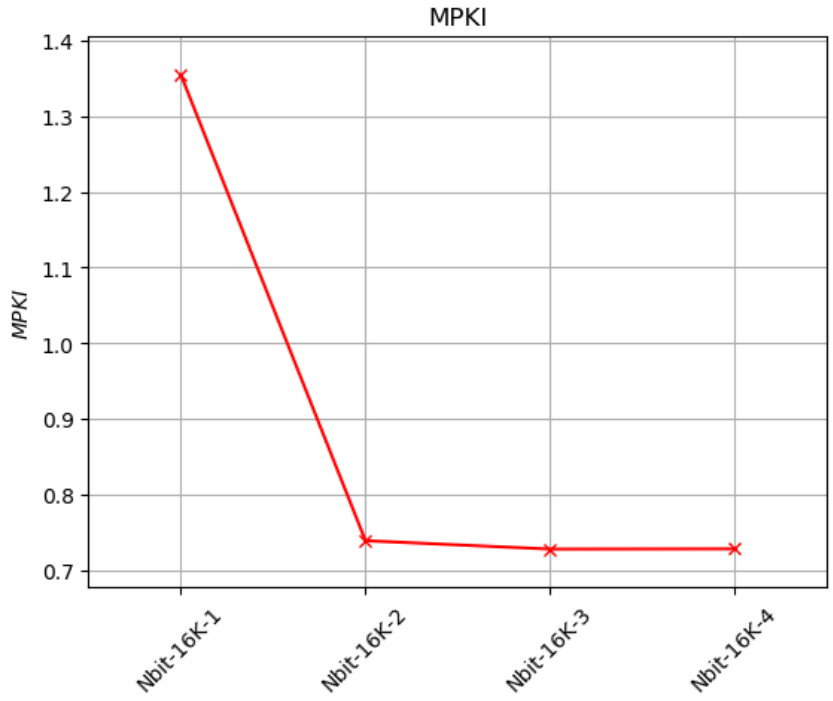
5) 445.gobmk



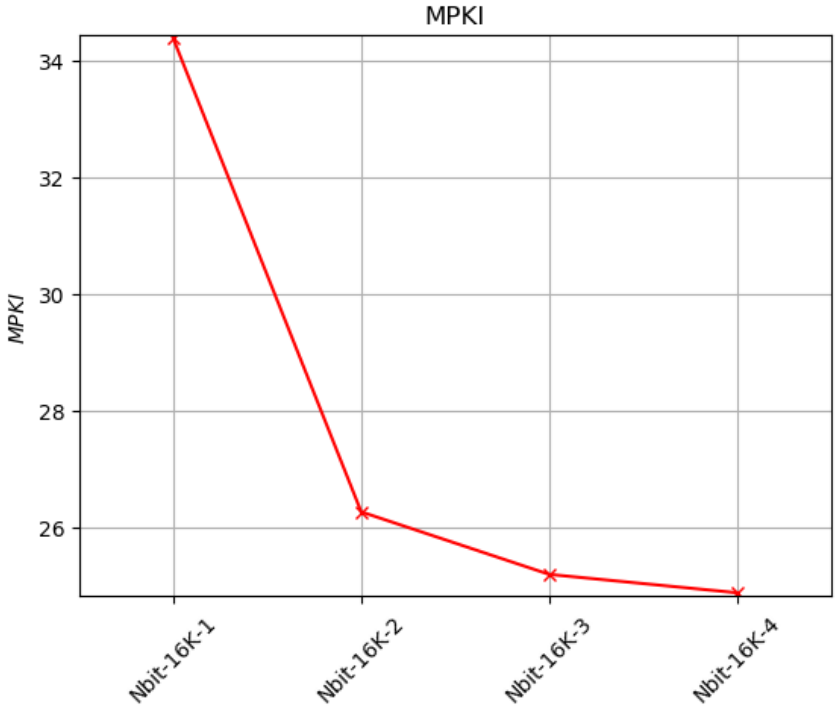
6) 450.soplex



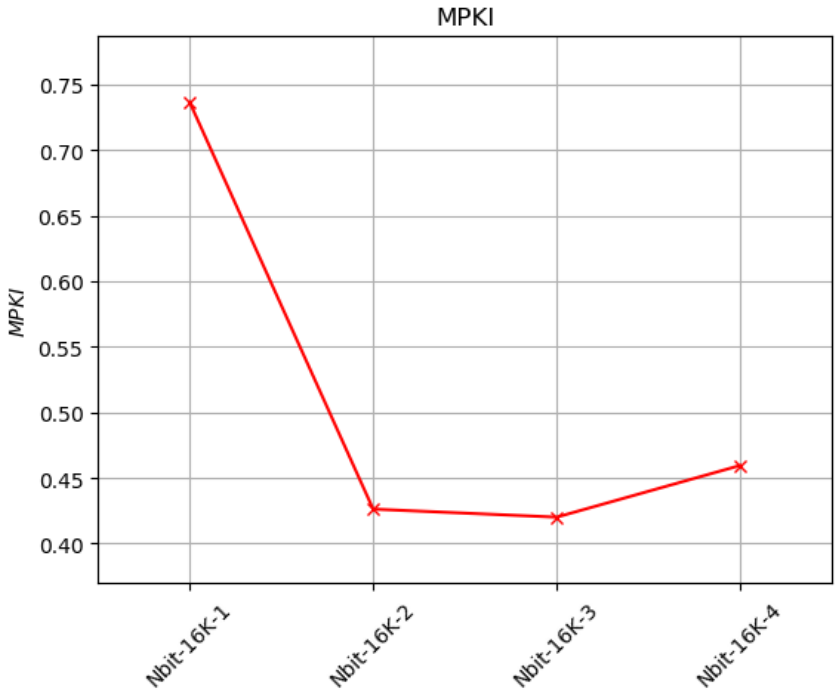
7) 456.hmmer



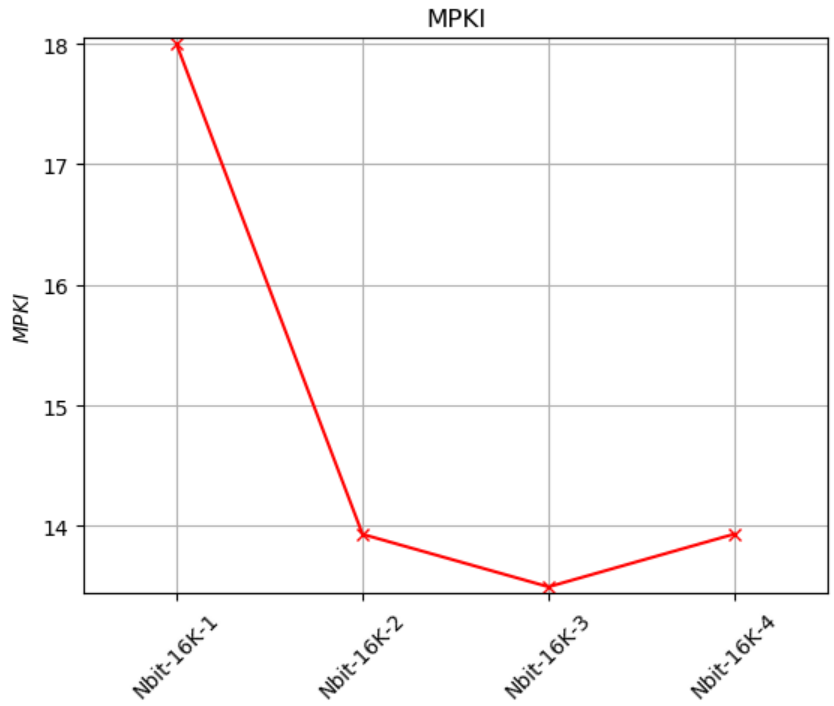
8) 458.sjeng



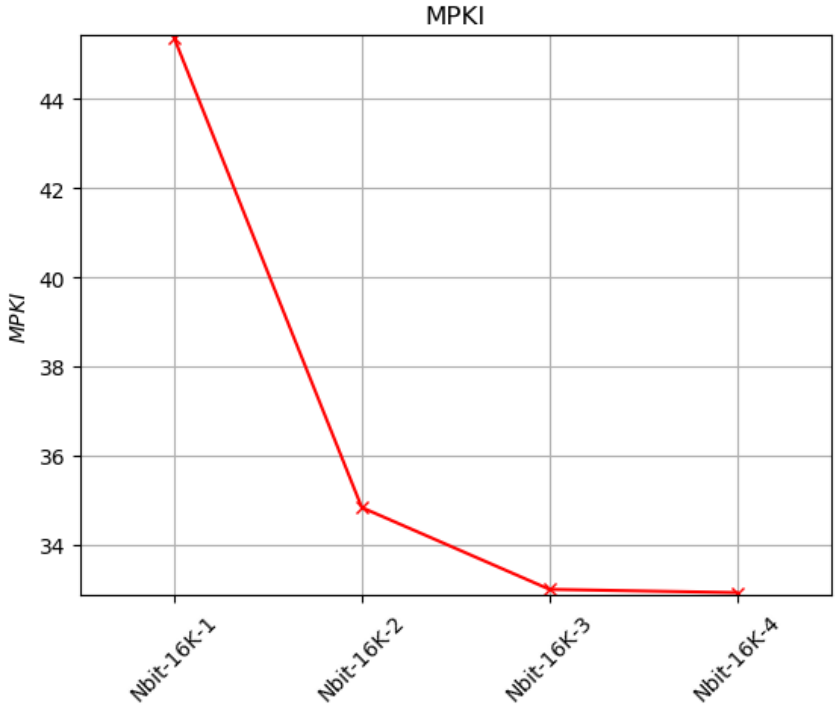
9) 459.GemsFDTD



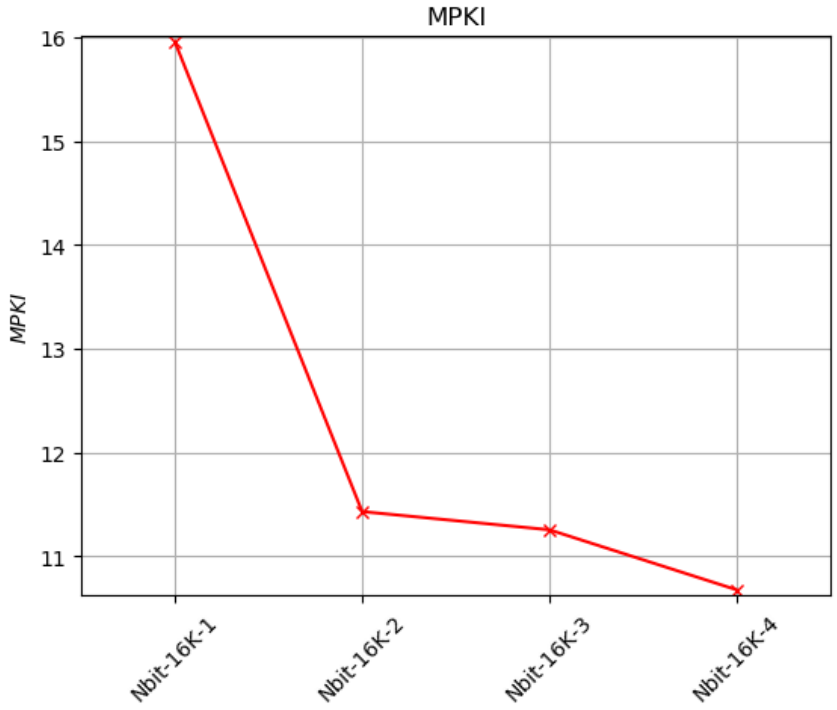
10) 471.omnetpp



11) 473.astar



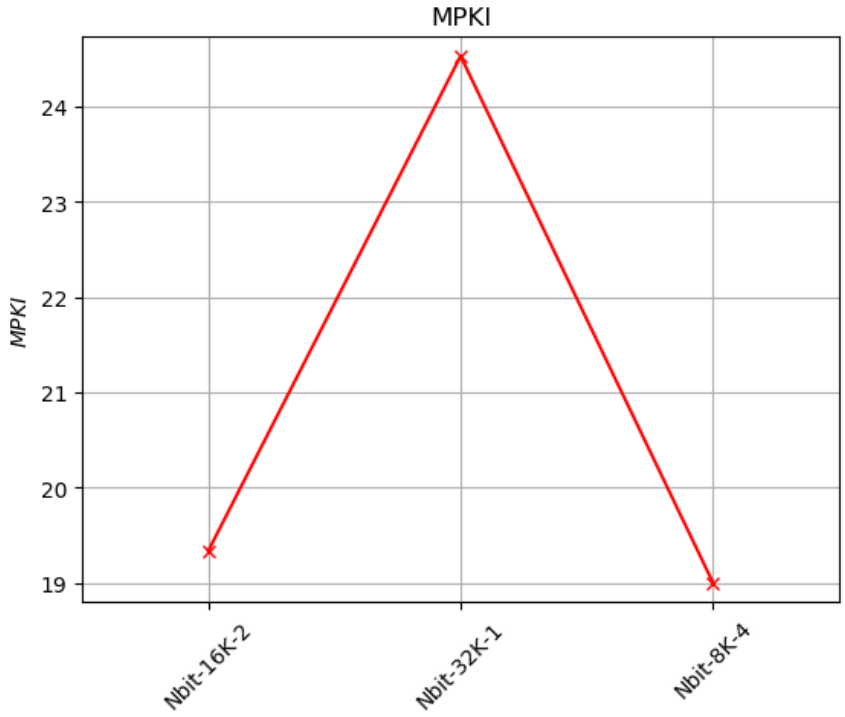
12) 483.xalancbmk



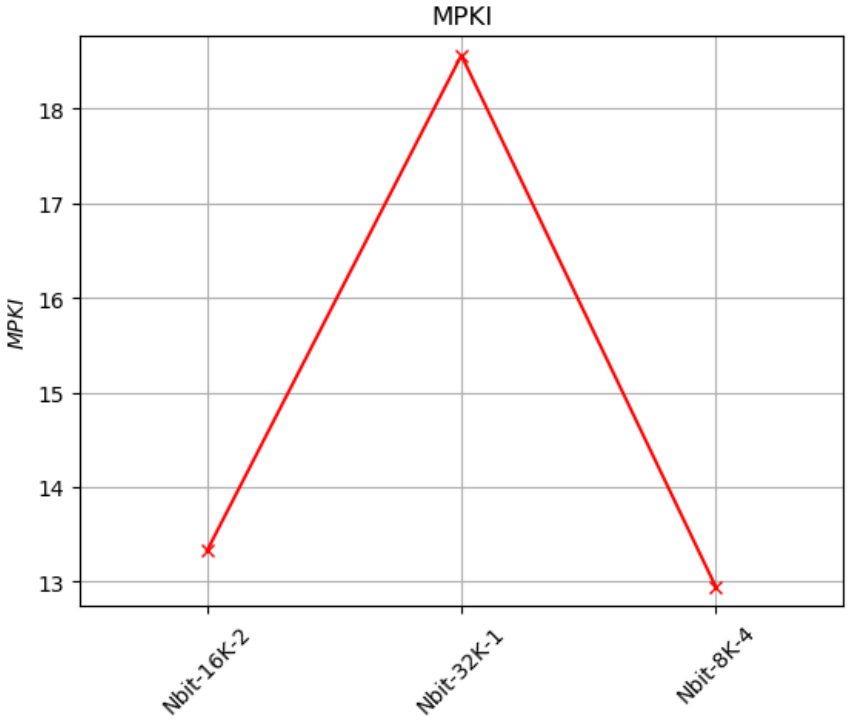
Γενικά, παρατηρούμε ότι η μετάβαση του predictor από το 1 στα 2 bit προκαλεί δραστική βελτίωση της απόδοσης, μειώνοντας σημαντικά τα mpki. Όμως, από εκεί και πέρα οι αυξομειώσεις είναι μηδαμινες στην πλειοψηφία των benchmarks. Έτσι, εκτιμούμε ότι ένα μέγεθος από 2 έως 4 bits έχει μεγαλύτερη σταθερότητα στην απόδοση.

ii) Διατηρώντας τώρα σταθερό το hardware και ίσο με 32K bits, εκτελούμε ξανά τις προσομοιώσεις για τα 12 benchmarks, θέτοντας N=1, 2, 4 και τον κατάλληλο αριθμό entries.

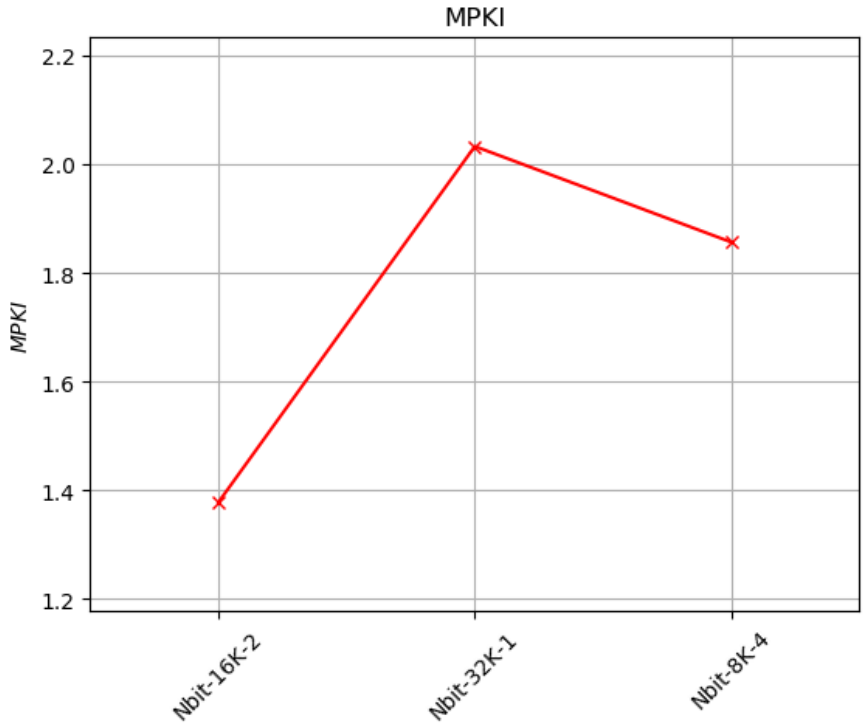
1) <u>403.gcc</u>



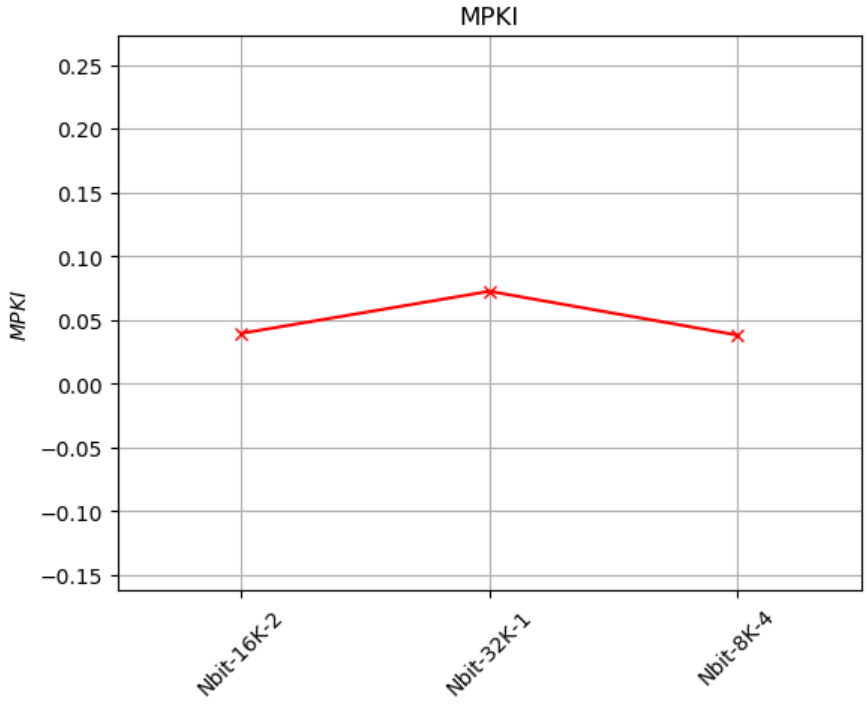
2) <u>429.mcf</u>



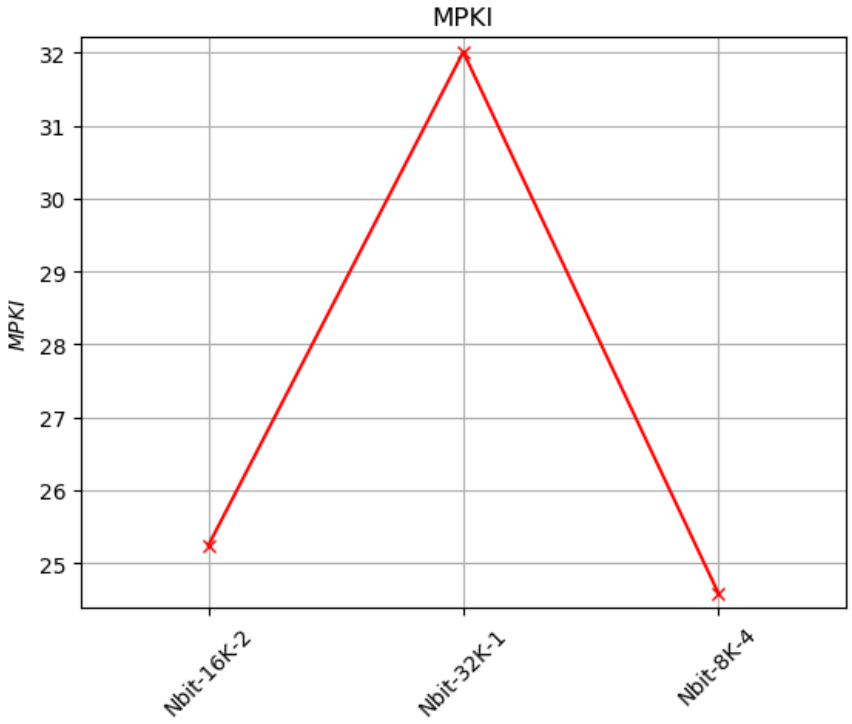
3) 434.zeusmp



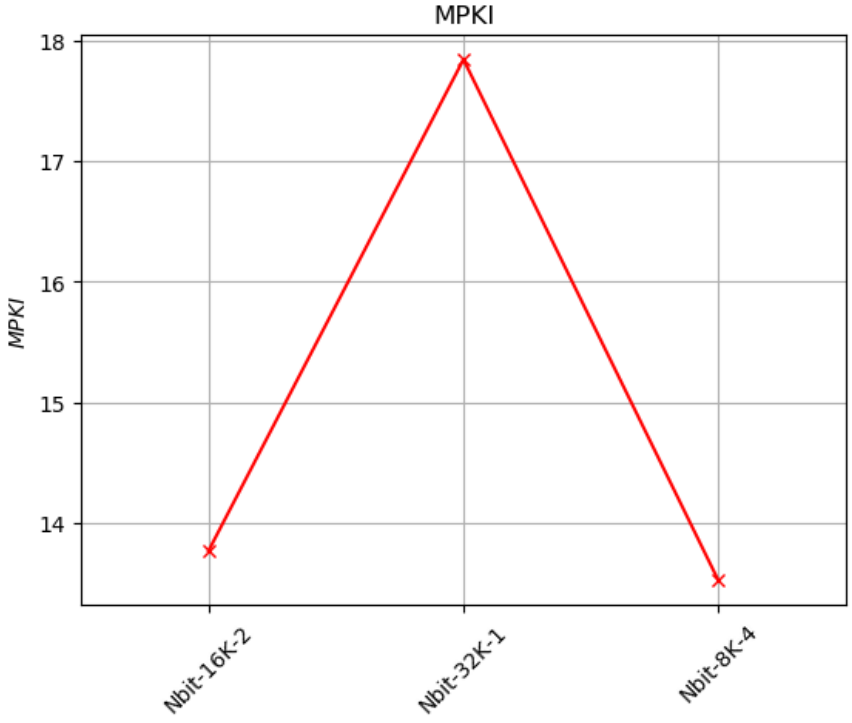
4) 436.cactusADM



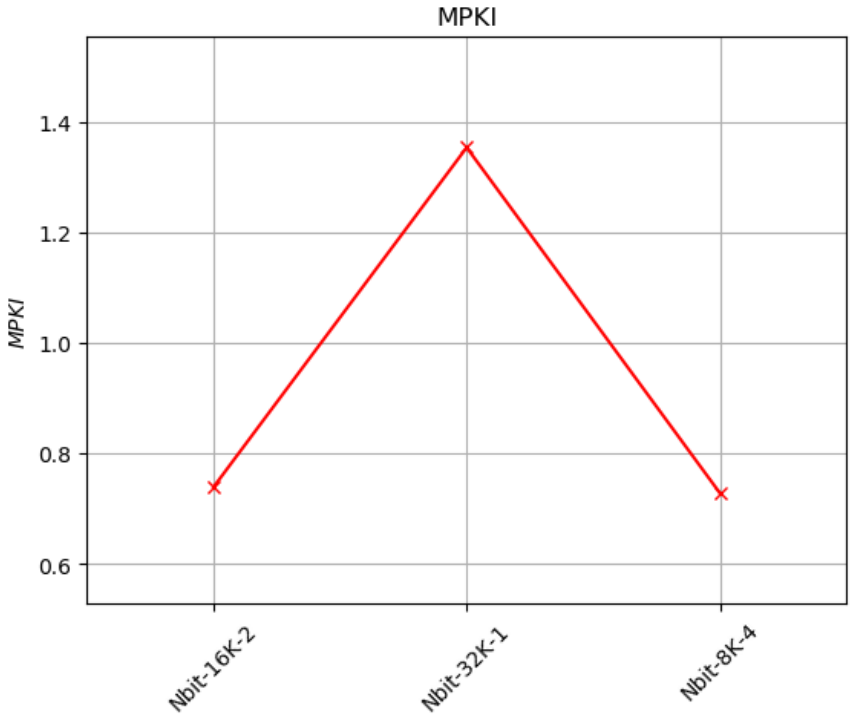
5) 445.gobmk



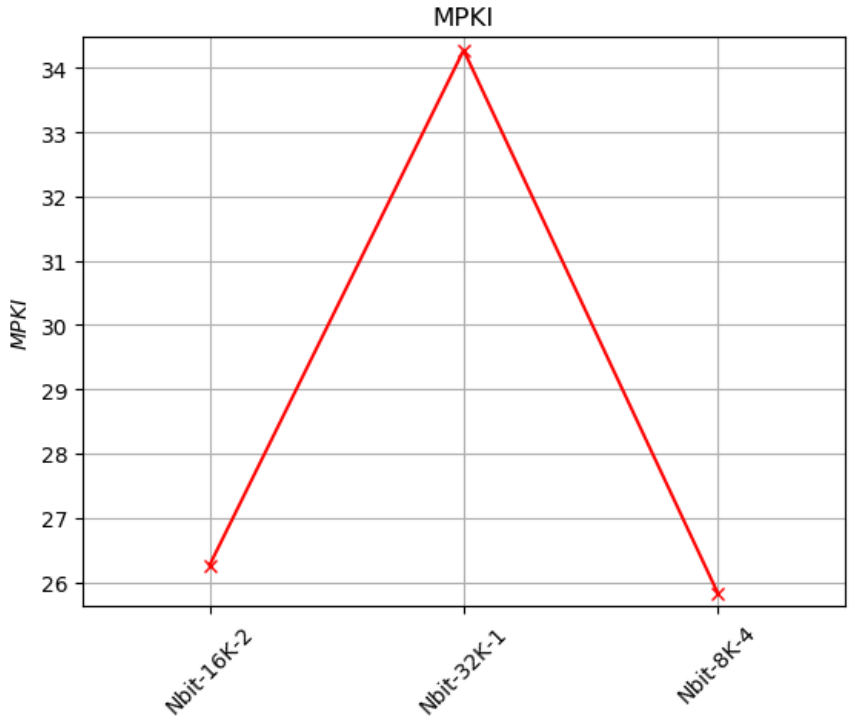
6) 450.soplex



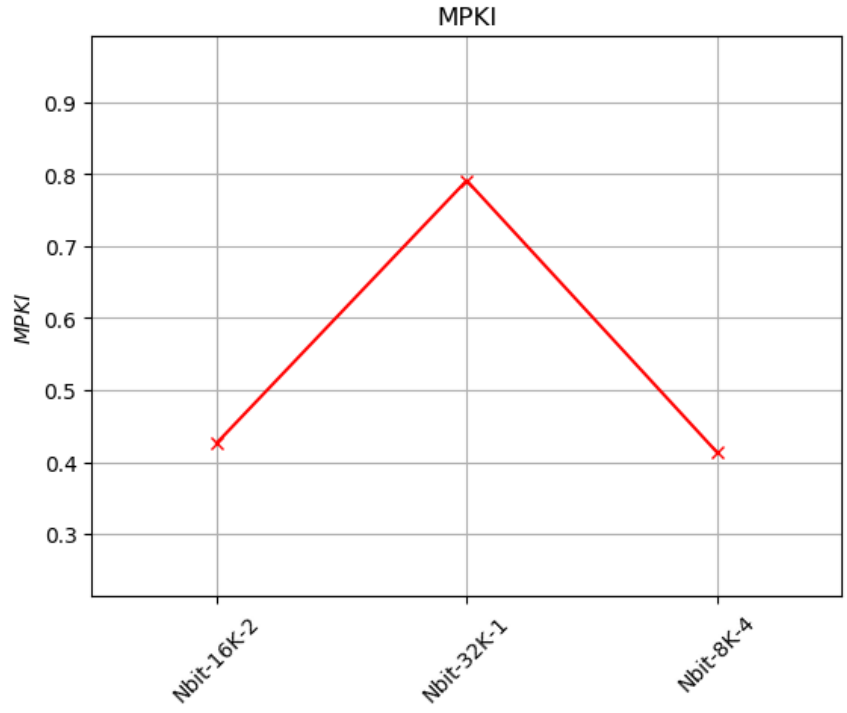
7) 456.hmmer



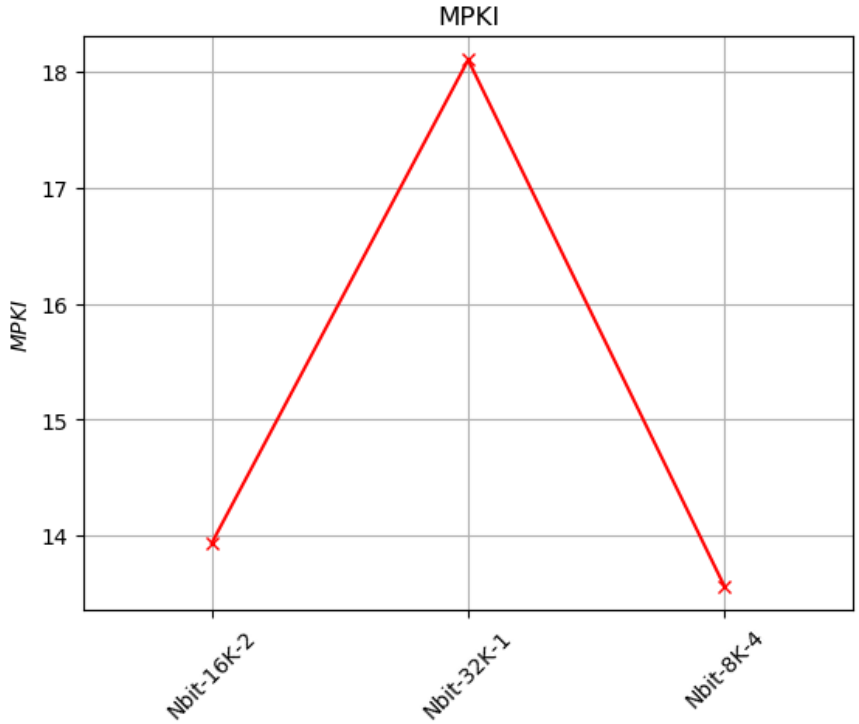
8) 458.sjeng



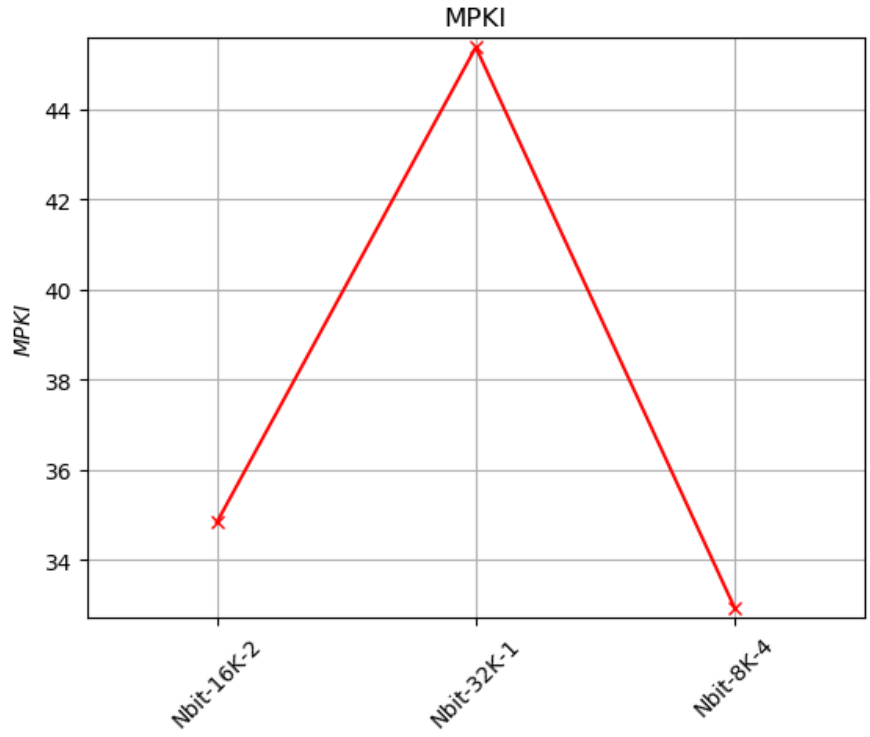
9) 459.GemsFDTD



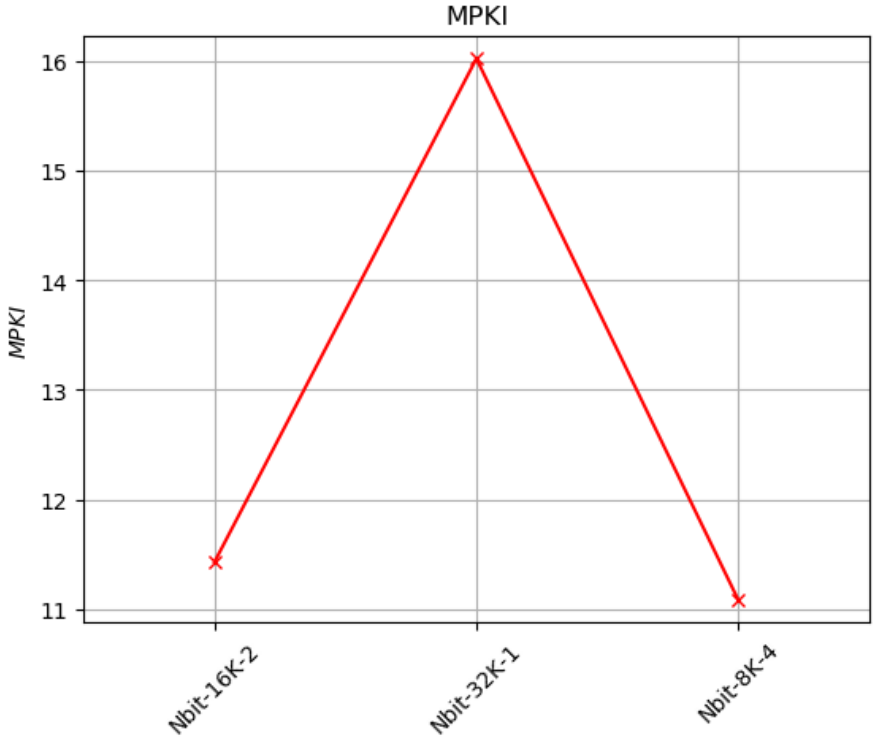
10) 471.omnetpp



11) 473.astar



12) 483.xalancbmk



Όπως και στο προηγούμενο ερώτημα, υπάρχει τεράστια διαφορά στην μείωση των mpki όταν αυξάνουμε τα bits από 1 σε 2. Παρόλα αυτά, από τα 2 στα 4 bits, η απόδοση αυξάνεται ελάχιστα, καθώς μειώνοντας τα entries μειώνουμε το μέγιστο δυνατό σύνολο εντολών που μπορούν να χωρέσουν στο προβλέπτη. Έτσι, ο συνδυασμός 8K-4 είναι ελάχιστα πιο αποδοτικός από τον 16K-2.

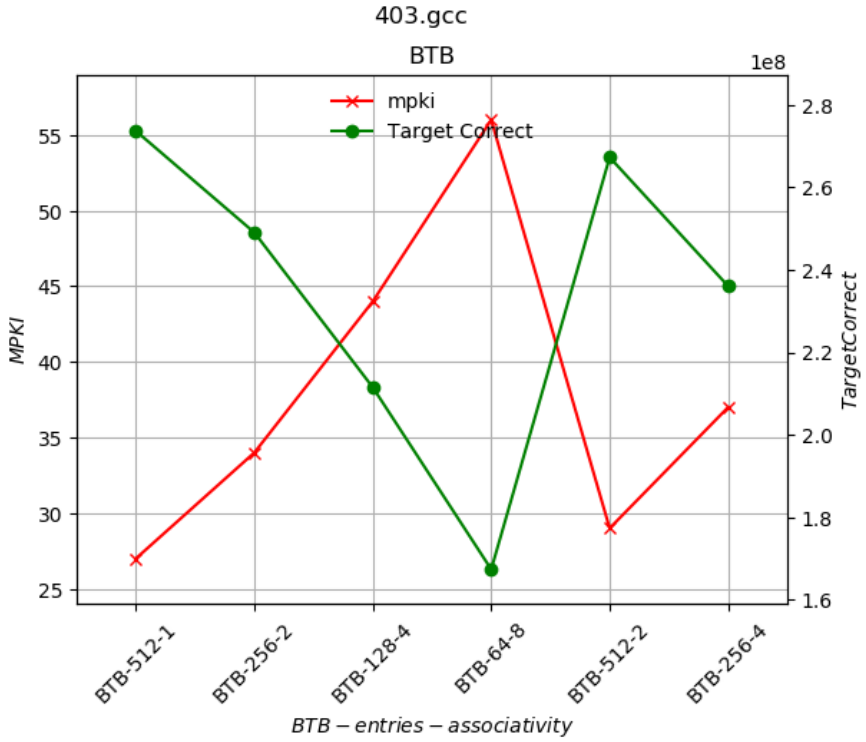
*** Λόγω έλλειψης χρόνου επανάληψης των προσομοιώσεων, απουσιάζει η υλοποίηση του εναλλακτικού FSM predictor ως 2β στα 2 υποερωτήματα.

4.3 Μελέτη του BTB

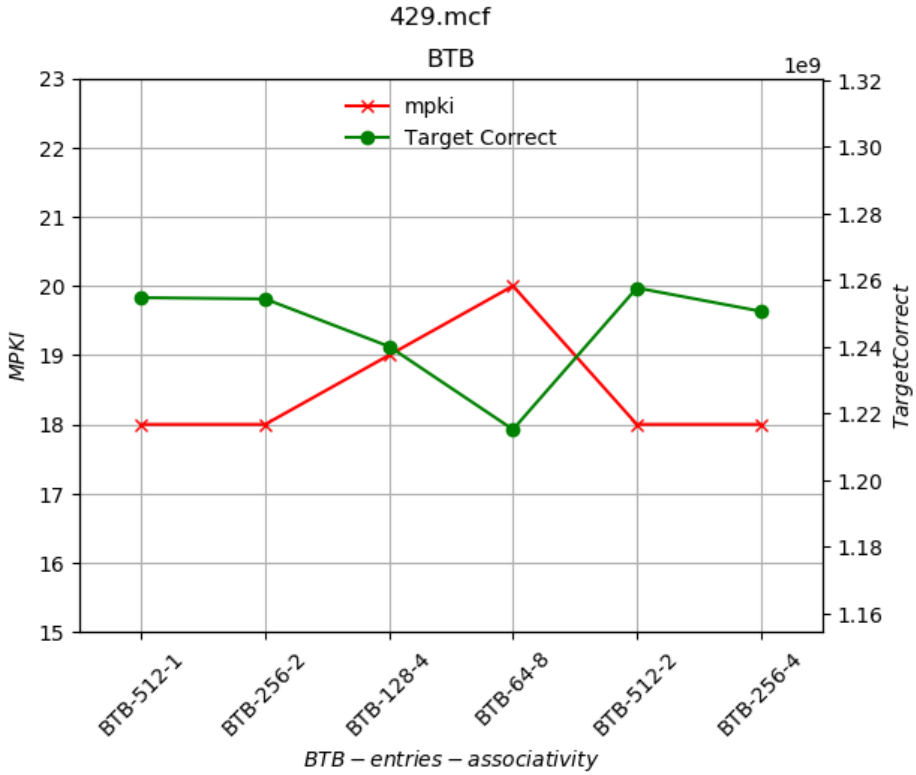
Θα υλοποιήσουμε έναν BTB και θα μελετήσουμε την ακρίβεια πρόβλεψής του για τις ακόλουθες περιπτώσεις:

btb entries	btb associativity
512	1, 2
256	2, 4
128	4
64	8

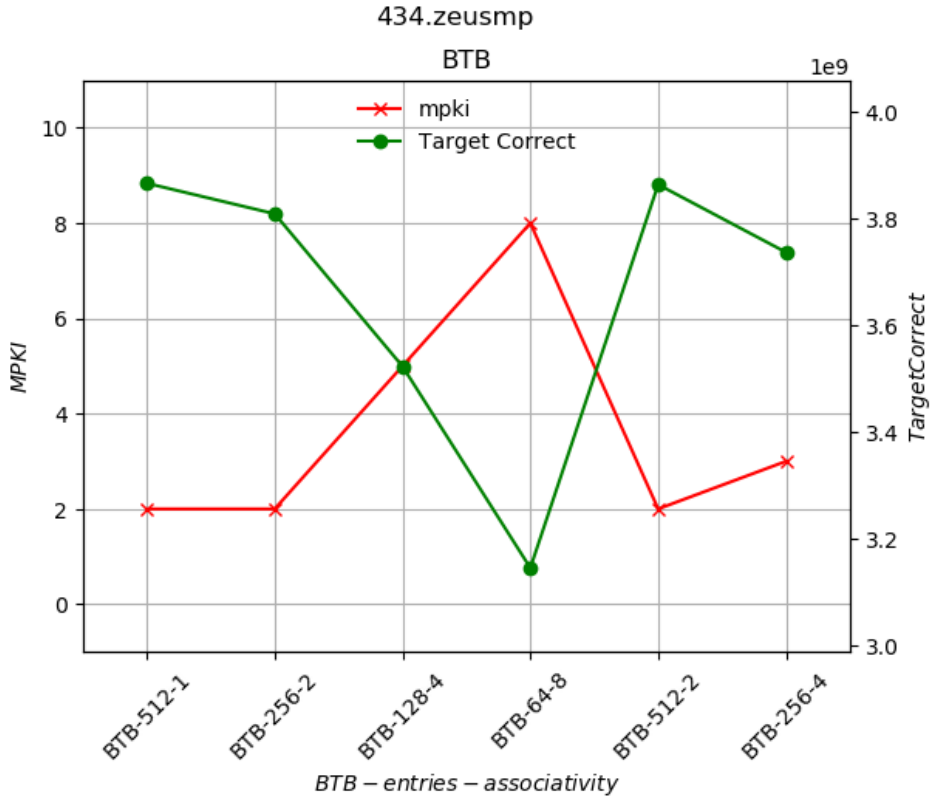
1) 403.gcc



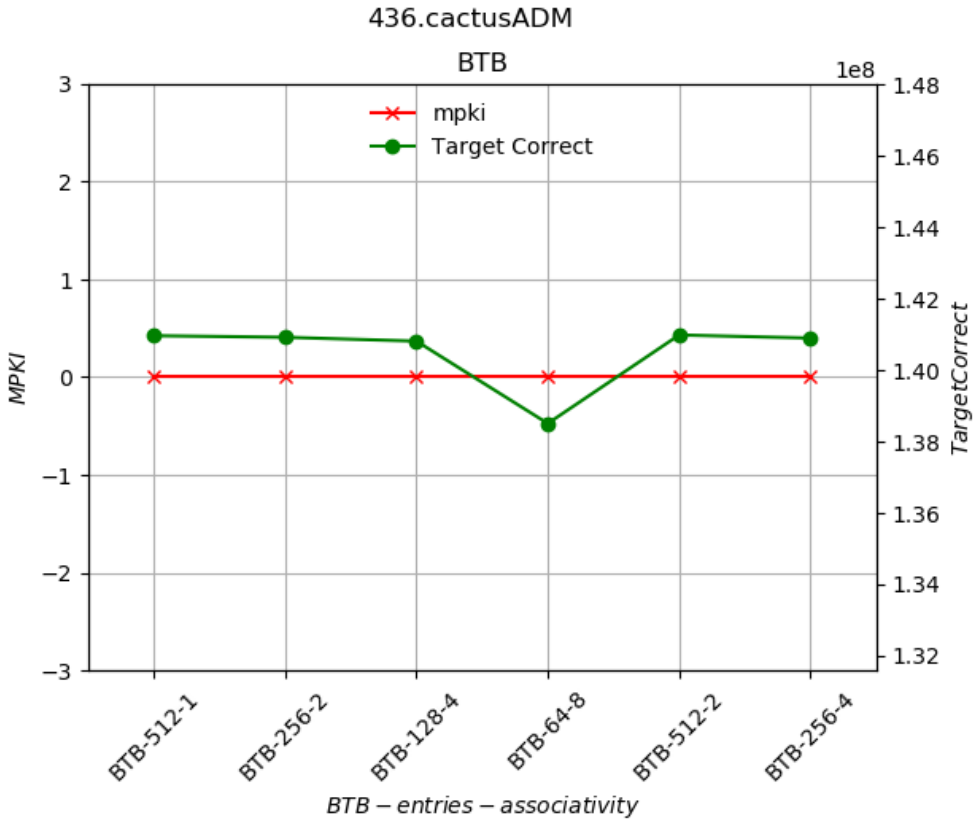
2) 429.mcf



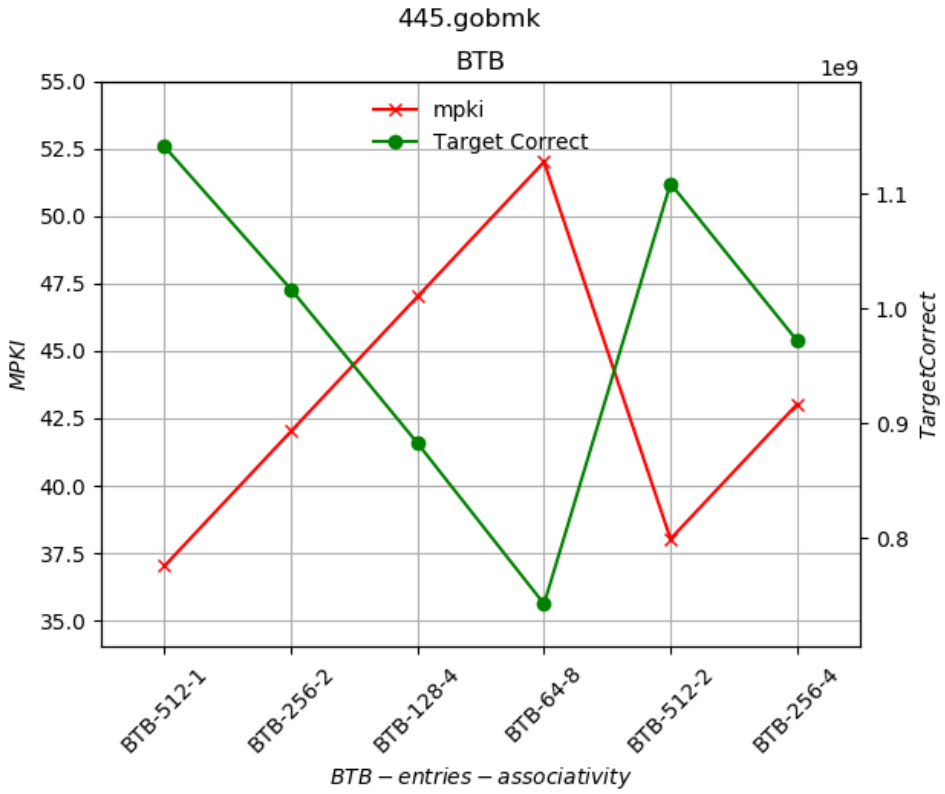
3) 434.zeusmp



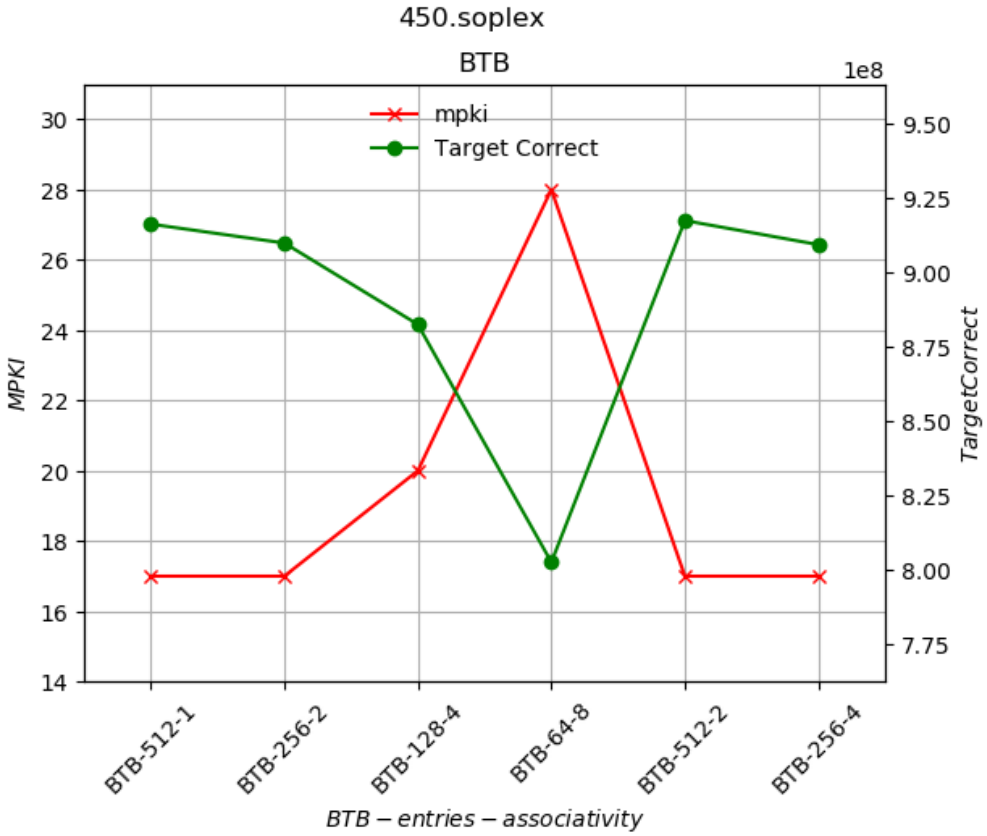
4) 436.cactusADM



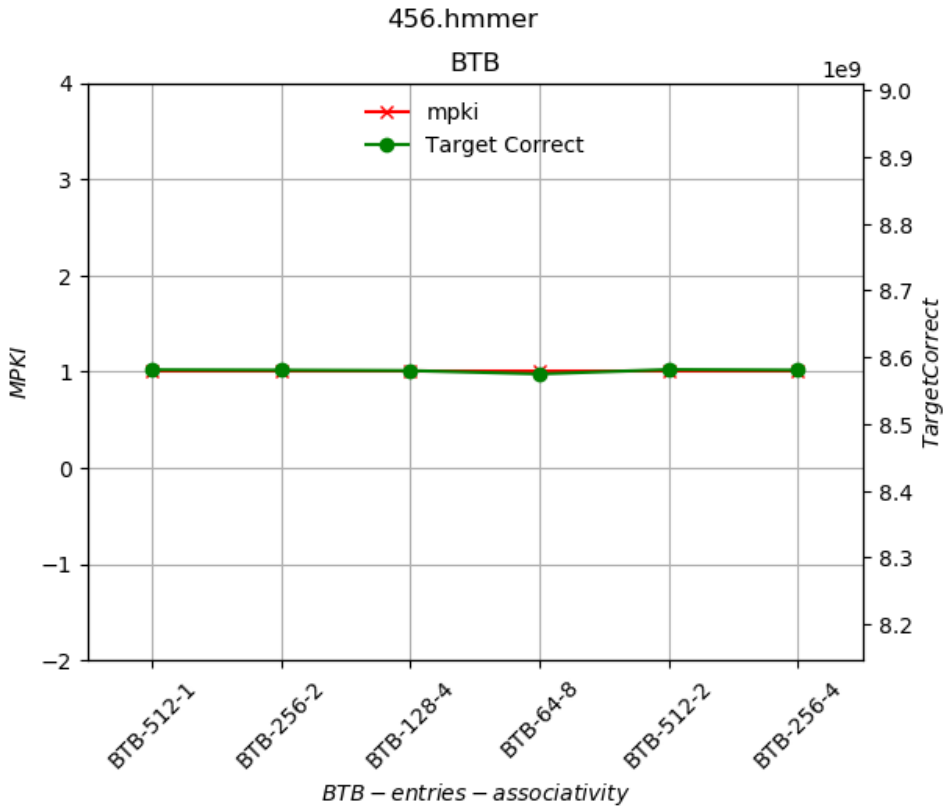
5) 445.gobmk



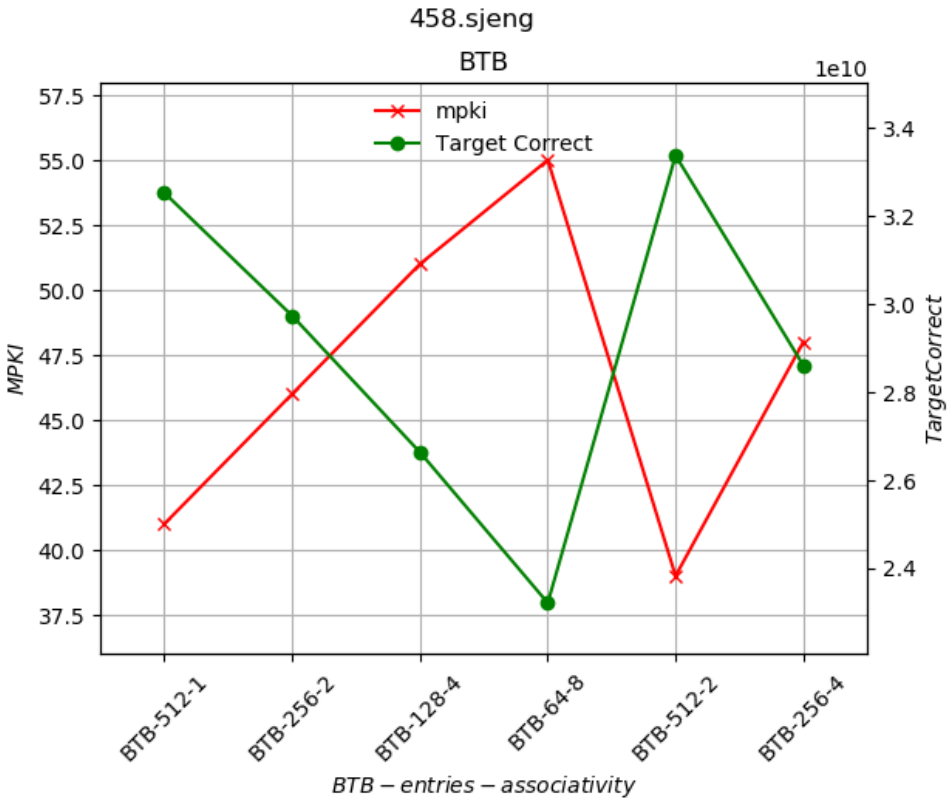
6) 450.soplex



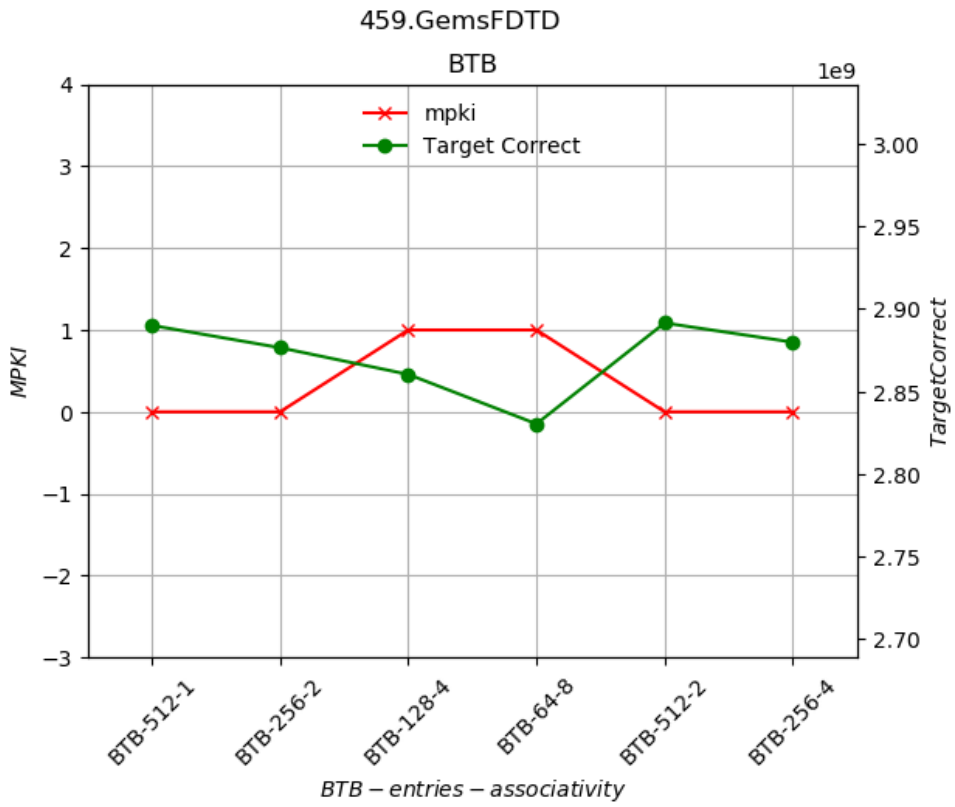
7) 456.hmmer



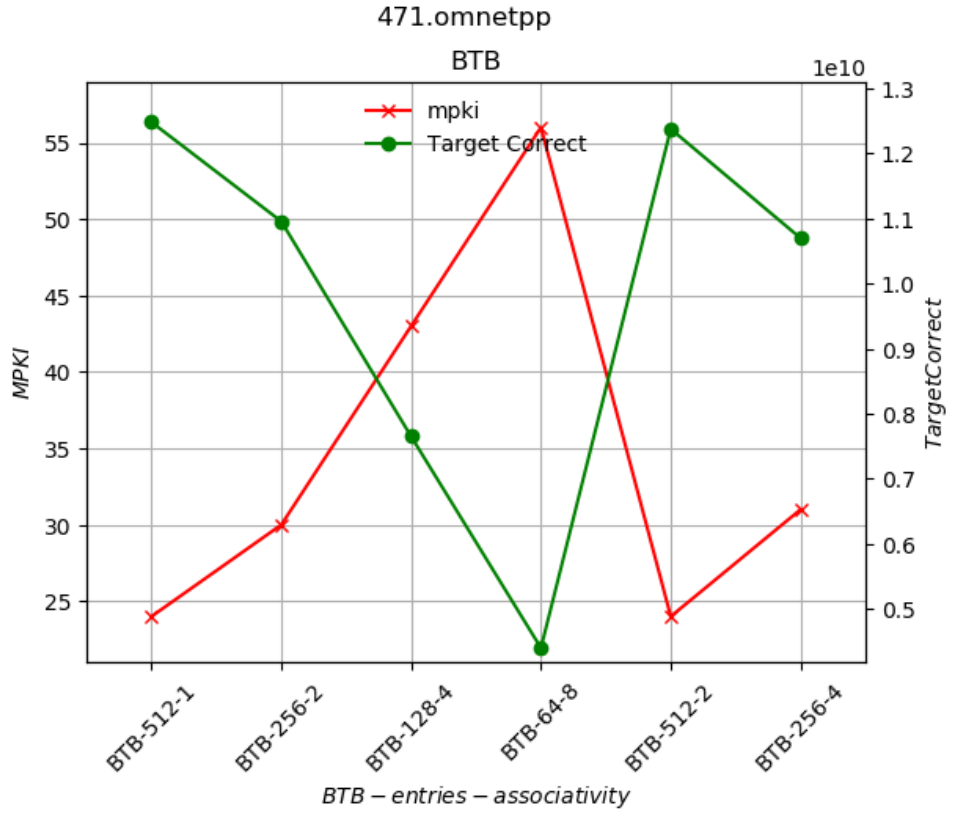
8) 458.sjeng



9) 459.GemsFDTD

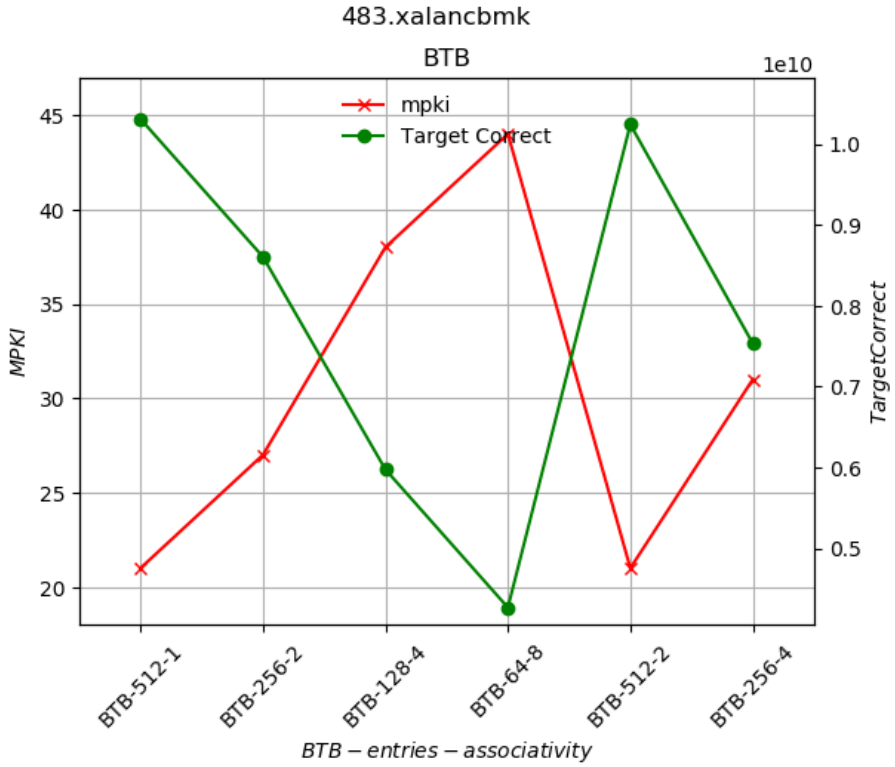


10) 471.omnetpp



11) 473.astar

12) 483.xalancbmk



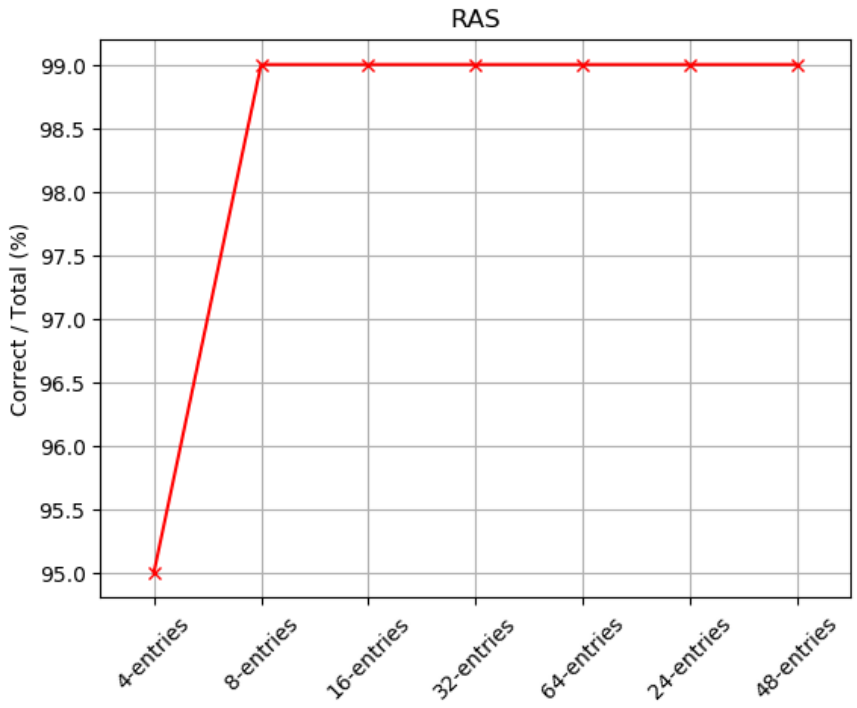
Τα αποτελέσματα στα 12 benchmarks ήταν αναμενόμενα. Όσο αυξάναμε το associativity και μειώναμε τα entries, τόσο περισσότερα mpki είχαμε. Μπορεί η αύξηση του associativity να βοηθά όταν έχουμε conflicts, αλλά ο περιορισμένος χώρος (λιγότερα entries) είχε αρνητική επίδραση. Παρόλα αυτά, παρατηρήσαμε ότι κρατώντας τον αριθμό των entries σταθερό (πχ. 512) είχαμε σχεδόν την ίδια απόδοση είτε για associativity = 1 είτε 2. Τέλος, είναι εμφανές ότι όσο μειώνεται το associativity, μειώνονται τα TargetCorrect, δηλαδή έχουμε κυρίως target misses.

4.4 Μελέτη του RAS

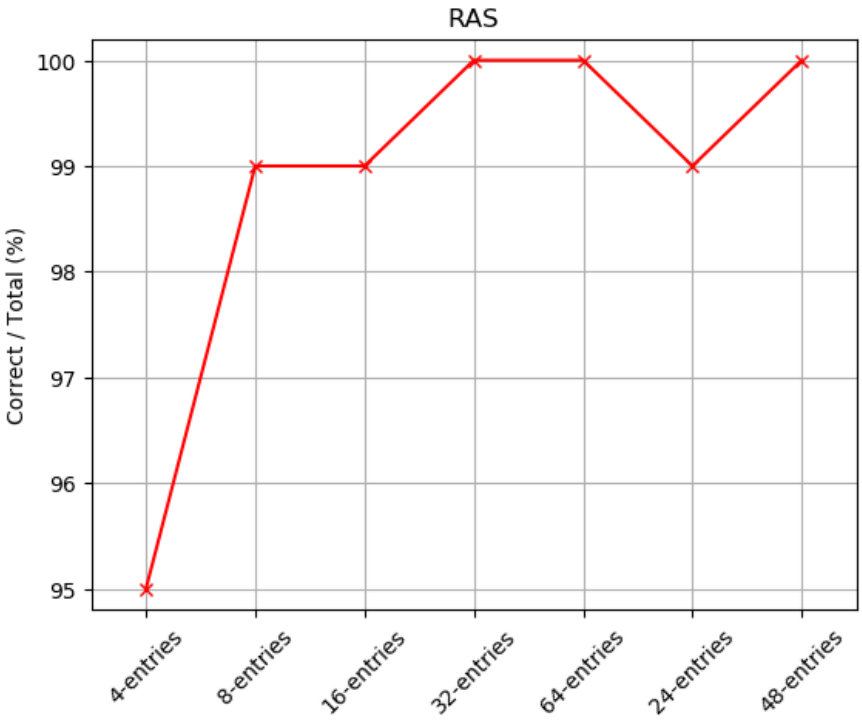
Χρησιμοποιώντας την υλοποίηση της RAS που δίνεται, θα μελετήσουμε το ποσοστό αστοχίας για τις ακόλουθες περιπτώσεις:

Αριθμός εγγραφών στη RAS	
4	32
8	48
16	64
24	

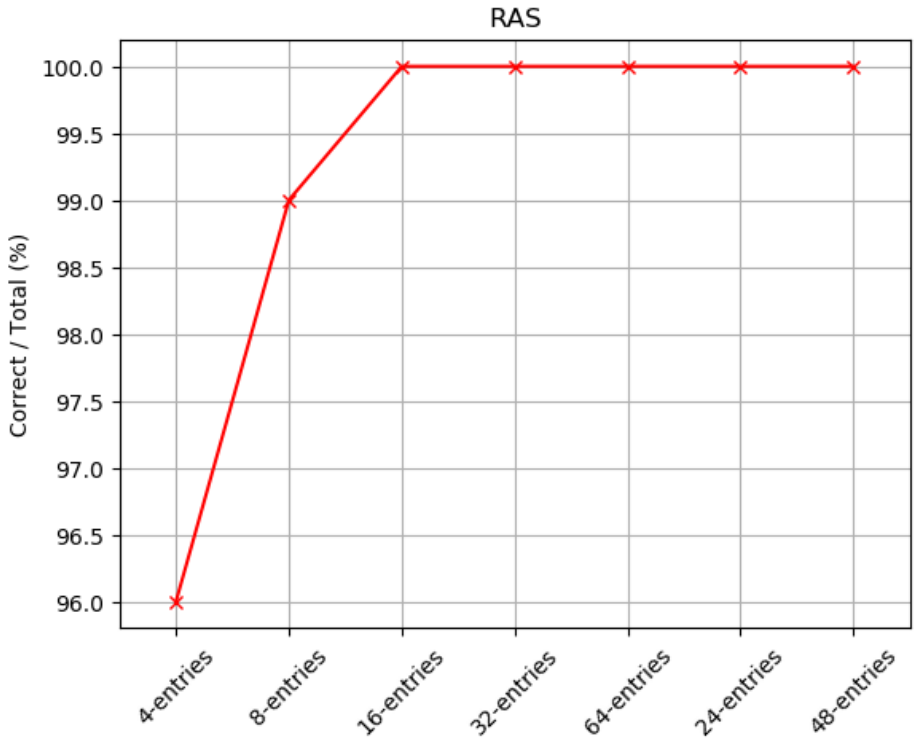
1) 403.gcc



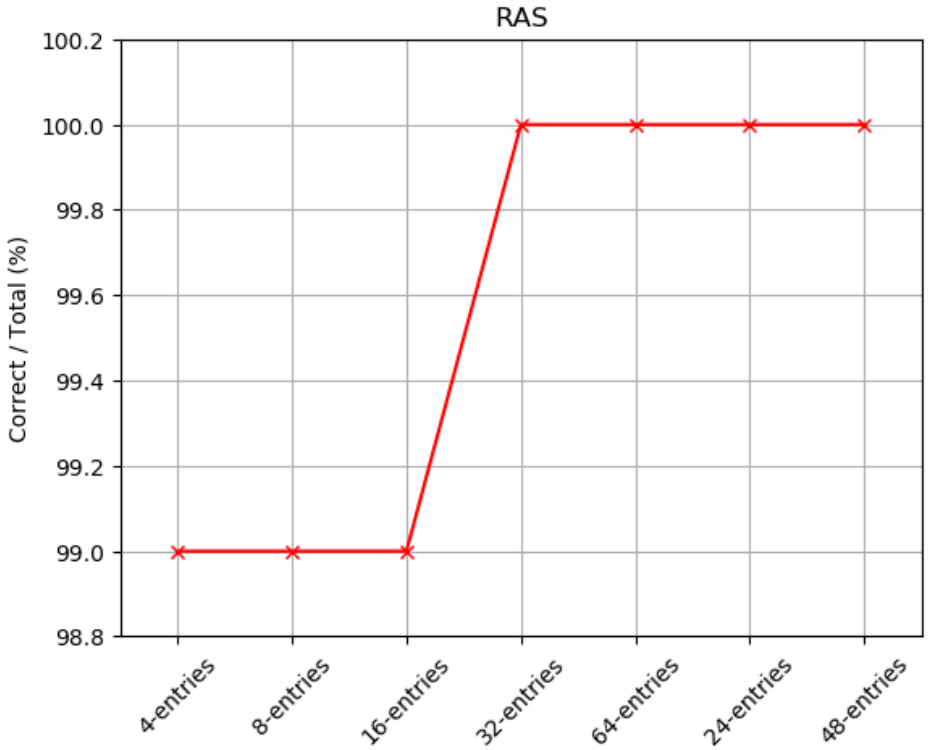
2) 429.mcf



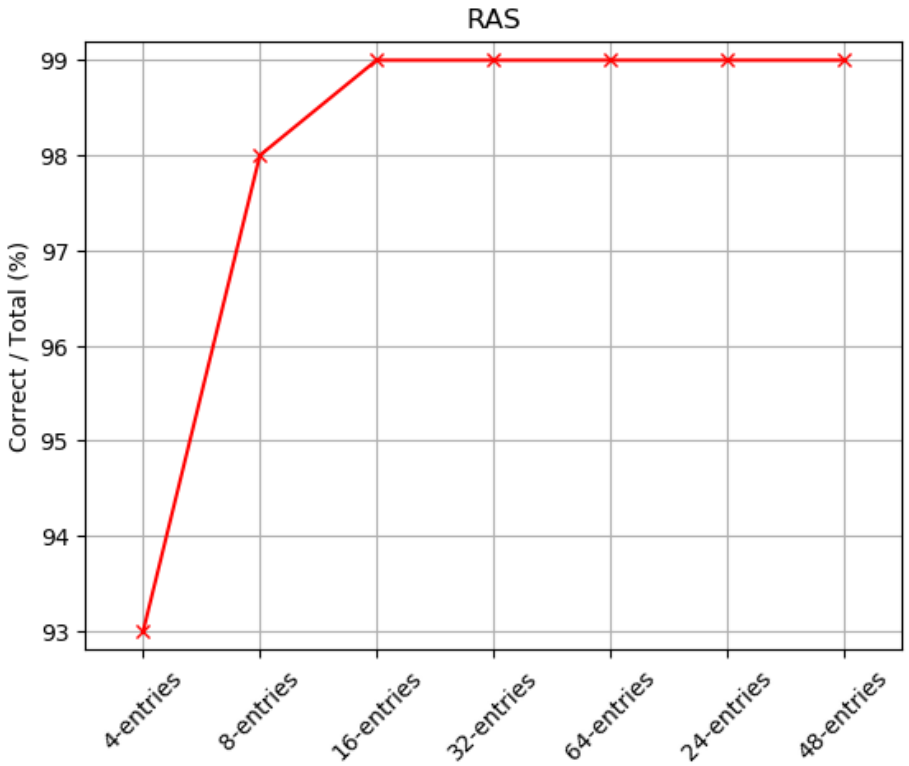
3) 434.zeusmp



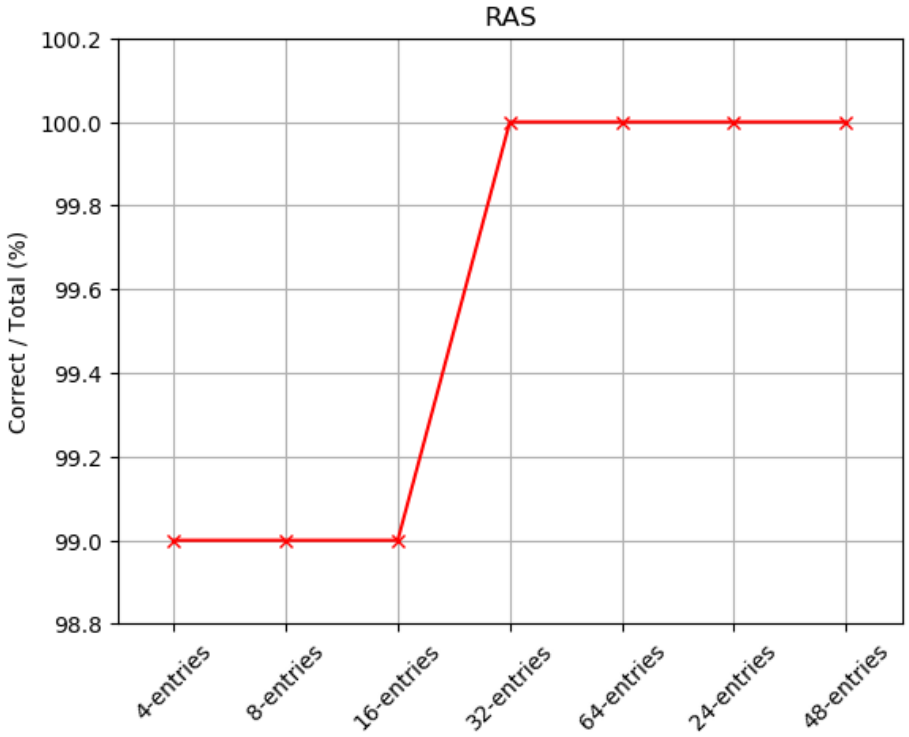
4) 436.cactusADM



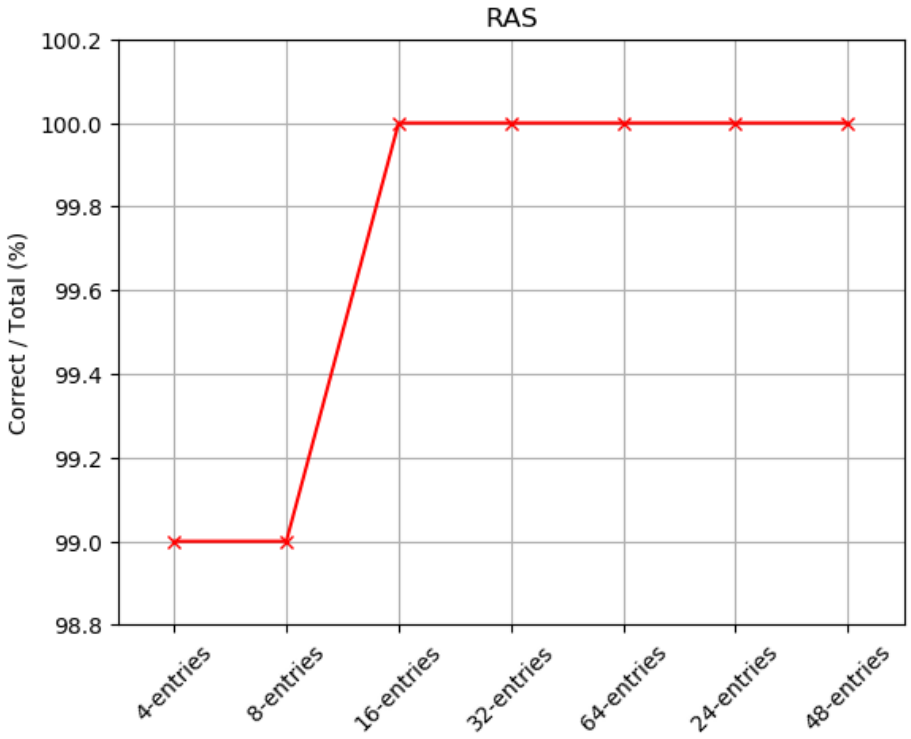
5) 445.gobmk



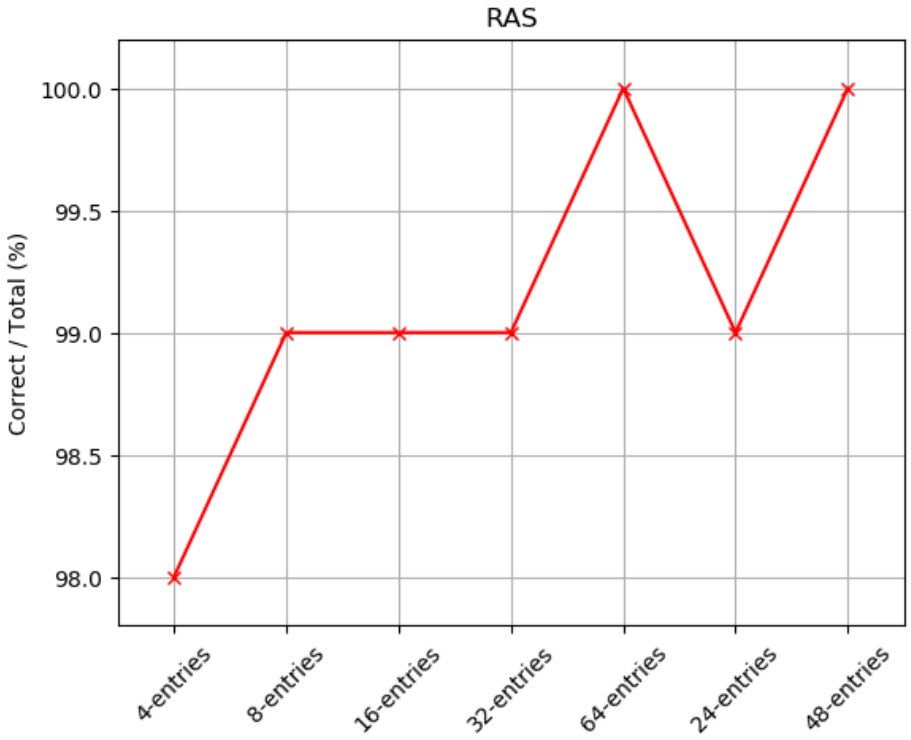
6) 450.soplex



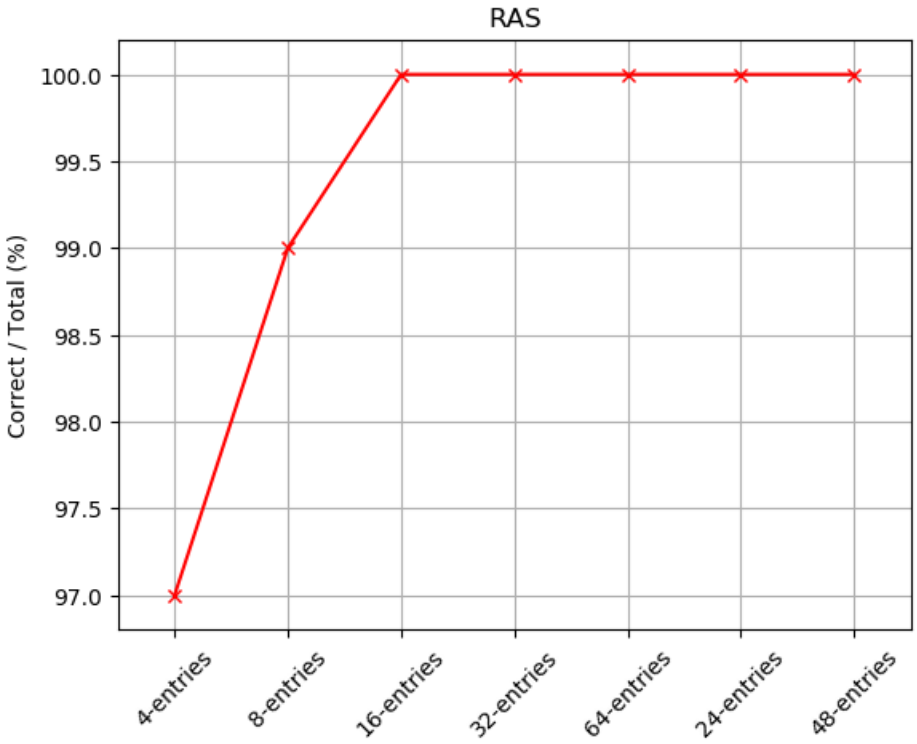
7) 456.hmmer



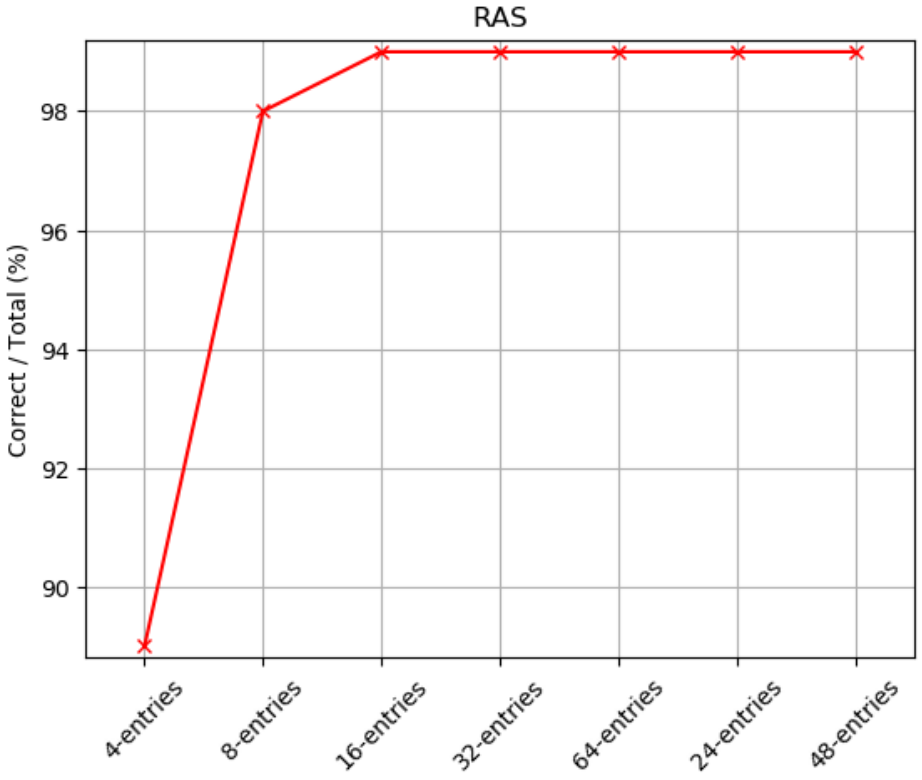
8) 458.sjeng



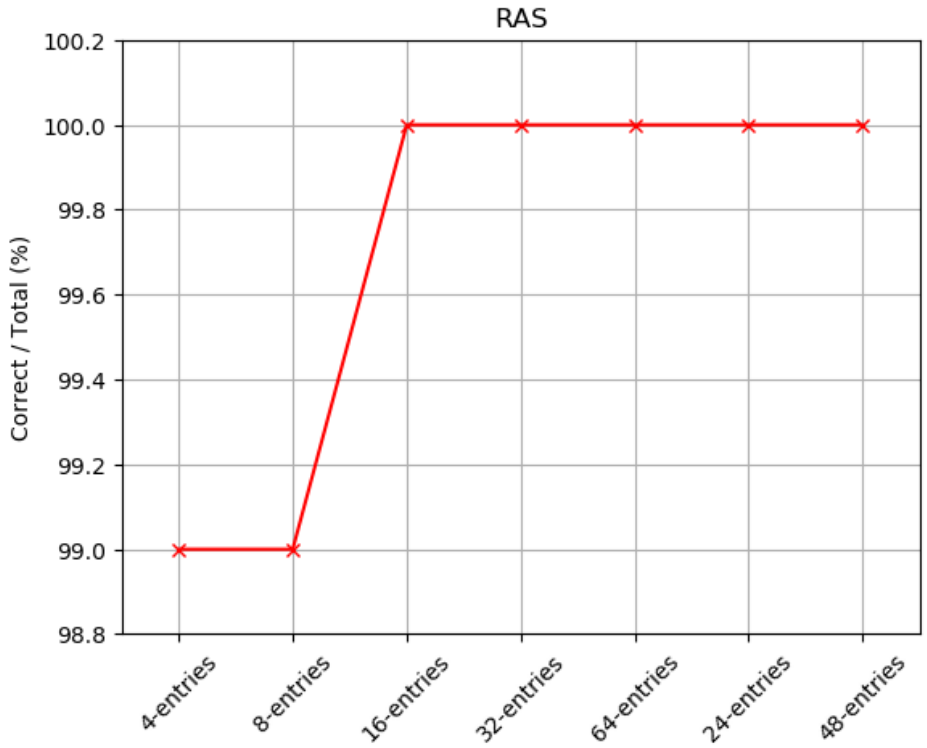
9) 459.GemsFDTD



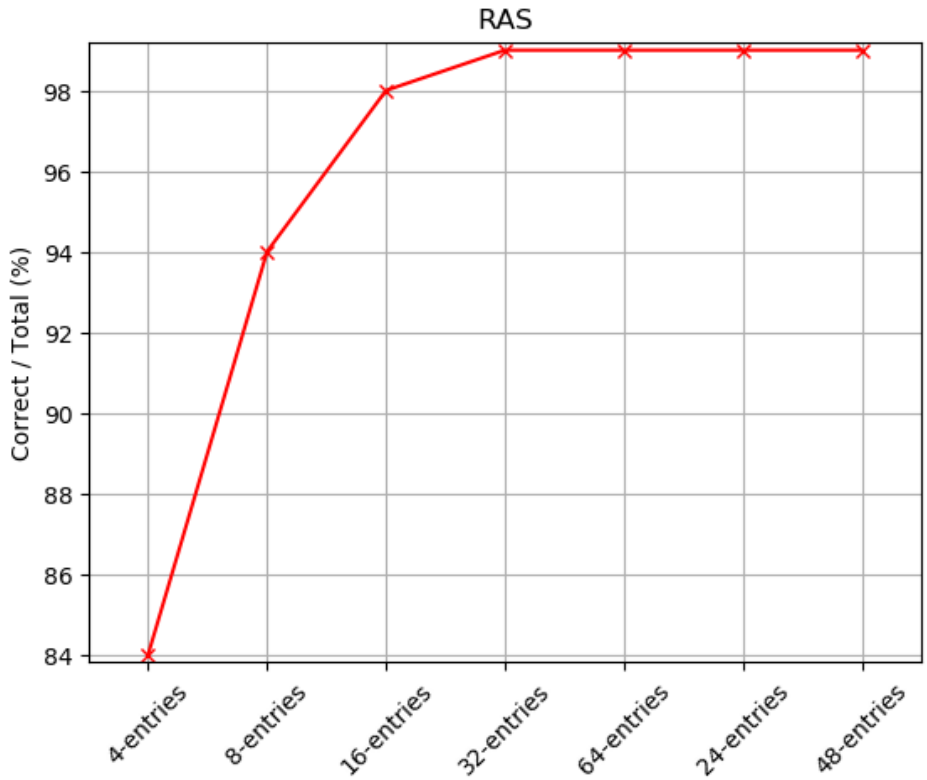
10) 471.omnetpp



11) 473.astar



12) 483.xalancbmk



Τα αποτελέσματα στα παραπάνω benchmarks είναι αναμενόμενα. Όσο αυξάνεται το πλήθος των εγγραφών (entries), βελτιώνεται και το ποσοστό των σωστών προβλέψεων. Στην αρχή (για 4 και 8 entries) συνήθως η βελτίωση ήταν ραγδαία έως ότου προσέγγιζε το 100%. Αυτό γινόταν στην πλειοψηφία των περιπτώσεων από τις 16 εγγραφές και μετά. Δεδομένου ότι το RAS n-entries αποθηκεύει τις τελευταίες n διευθύνσεις από τις οποίες καλέστηκε, είναι λογικό όσο το n αυξάνεται να πλησιάσει το 100% σωστών προβλέψεων, αφού όλο και περισσότερες πιθανές διευθύνσεις κλήσης θα έχουν αποθηκευθεί

4.5 Σύγκριση διαφορετικών predictors

Στο κομμάτι αυτό θα συγκρίνουμε τους παρακάτω predictors:

- Static AlwaysTaken
- Static BTFNT (BackwardTaken-ForwardNotTaken)
- Ο n-bit predictor που επιλέξαμε στο 4.2 (ii)
- Pentium-M predictor (δίνεται ότι το hardware overhead είναι περίπου 30K)
- Local-History two-level predictors (βλ. διαφάνειες μαθήματος) με τα εξής χαρακτηριστικά :
    - PHT entries = 8192
    - PHT n-bit counter length = 2
    - BHT entries = X
    - BHT entry length = Z

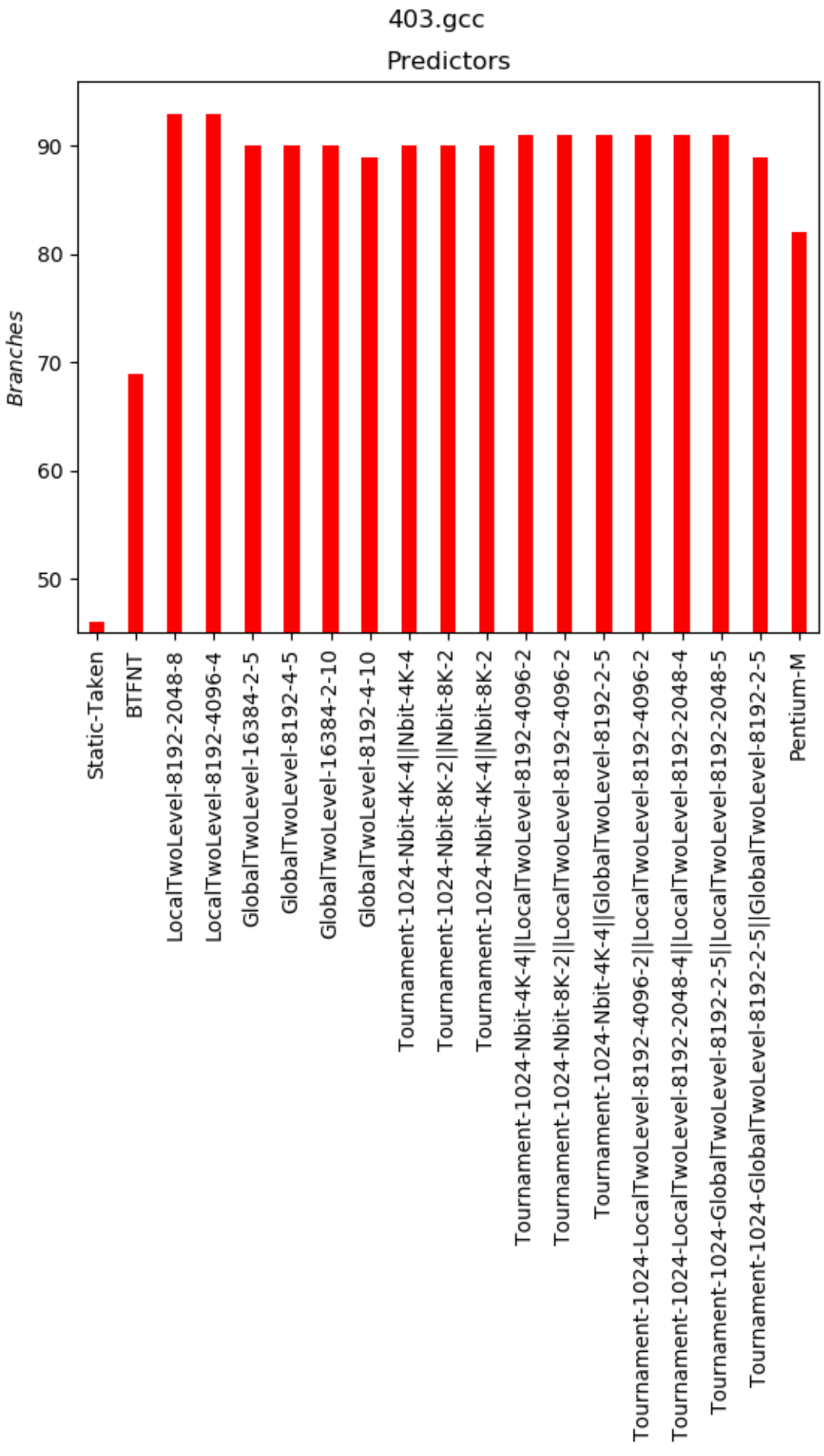
    Υπολογίζουμε το Z ώστε το απαιτούμενο hardware να είναι σταθερό και ίσο με 32K, όταν X=2048 και X=4096

- Global History two-level predictors με τα εξής χαρακτηριστικά:
    - PHT entries = Z
    - PHT n-bit counter length = X
    - BHR length = 5, 10

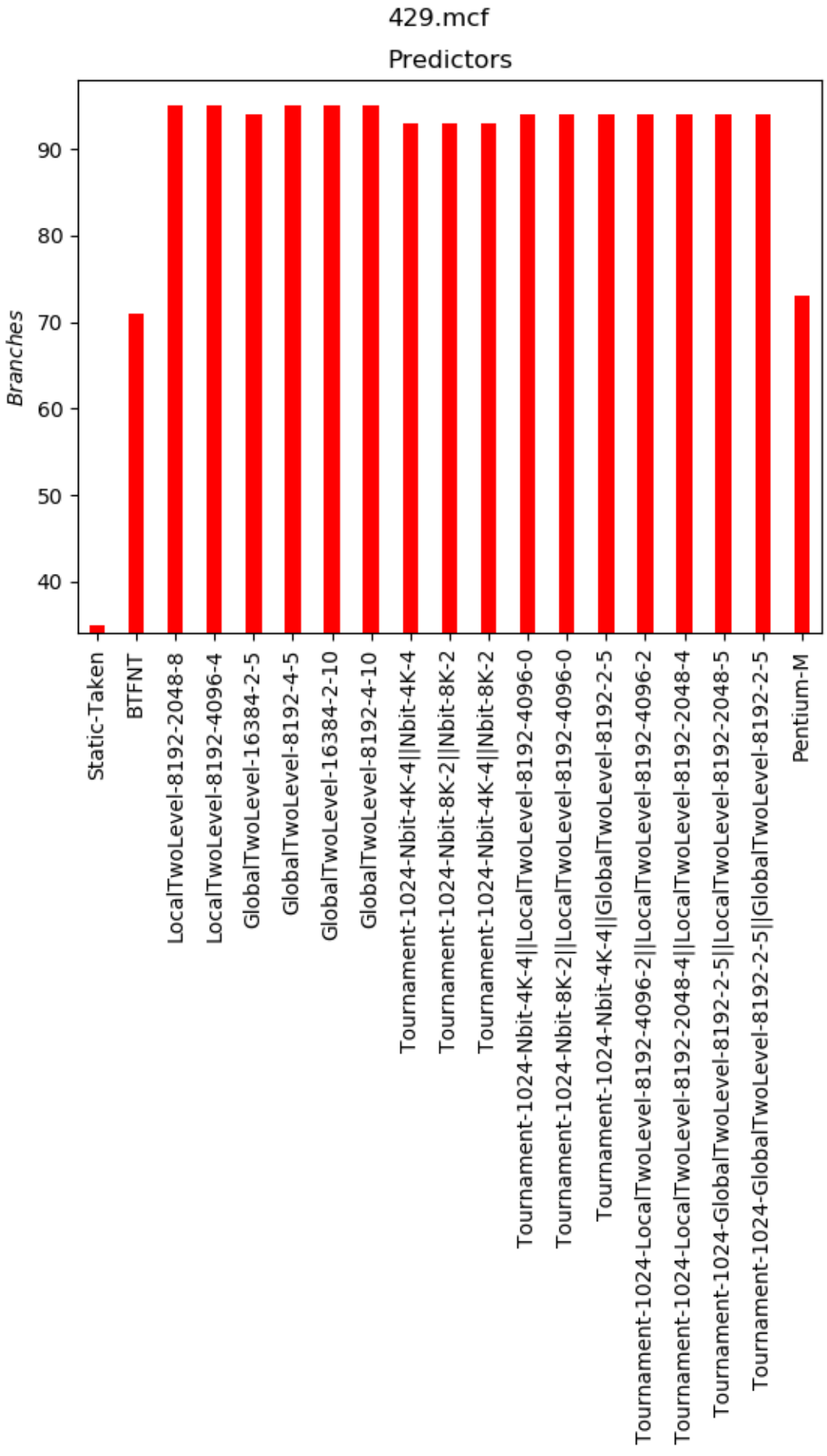
    Υπολογίζουμε το Z ώστε το απαιτούμενο hardware να είναι σταθερό και ίσο με 32K όταν X=2 και X = 4. Το κόστος του Branch History Register (5 και 10 bits) θεωρείται αμελητέο.

- Alpha 21264 predictor (hardware overhead 29K)
- Tournament Hybrid predictors με τα εξής χαρακτηριστικά:
    - Ο meta-predictor M είναι ένας 2-bit predictor με 1024 ή 2048 entries (το overhead του μπορούμε να το αγνοήσουμε στην ανάλυση μας)
    - Οι P0, P1 μπορούν να είναι n-bit, local-history ή global-history predictors
    - Οι P0, P1 έχουν overhead 16K ο καθένας

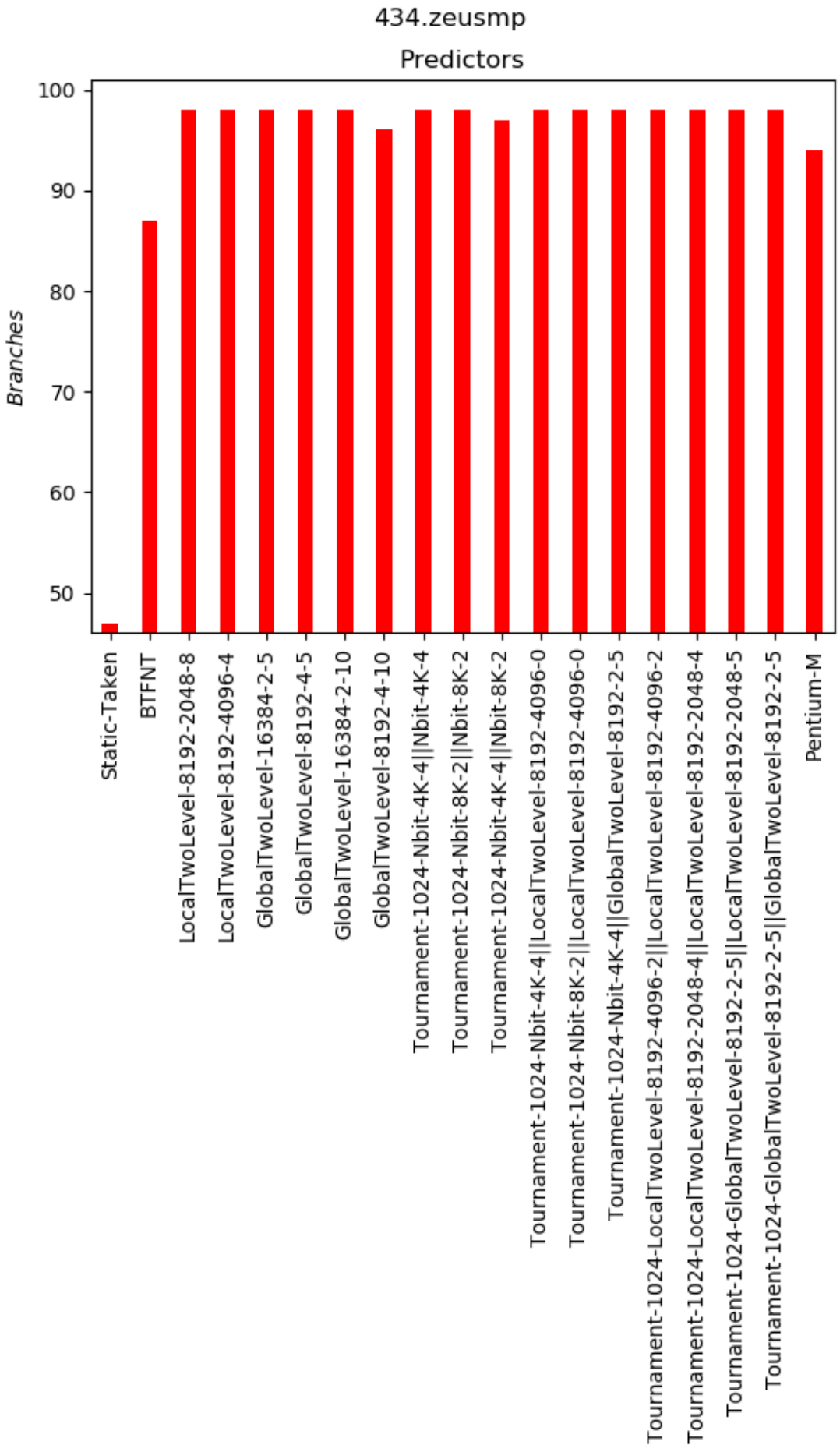
1) **403.gcc**


403.gcc
Predictors
Branches
90
80
70
60
50
Static-Taken
BTFNT
LocalTwoLevel-8192-2048-8
LocalTwoLevel-8192-4096-4
GlobalTwoLevel-16384-2-5
GlobalTwoLevel-8192-4-5
GlobalTwoLevel-16384-2-10
GlobalTwoLevel-8192-4-10
Tournament-1024-Nbit-4K-4||Nbit-4K-4
Tournament-1024-Nbit-8K-2||Nbit-8K-2
Tournament-1024-Nbit-4K-4||Nbit-8K-2
Tournament-1024-Nbit-4K-4||LocalTwoLevel-8192-4096-2
Tournament-1024-Nbit-8K-2||LocalTwoLevel-8192-4096-2
Tournament-1024-Nbit-4K-4||GlobalTwoLevel-8192-2-5
Tournament-1024-LocalTwoLevel-8192-4096-2||LocalTwoLevel-8192-4096-2
Tournament-1024-LocalTwoLevel-8192-2048-4||LocalTwoLevel-8192-2048-4
Tournament-1024-GlobalTwoLevel-8192-2-5||LocalTwoLevel-8192-2048-5
Tournament-1024-GlobalTwoLevel-8192-2-5||GlobalTwoLevel-8192-2-5
Pentium-M

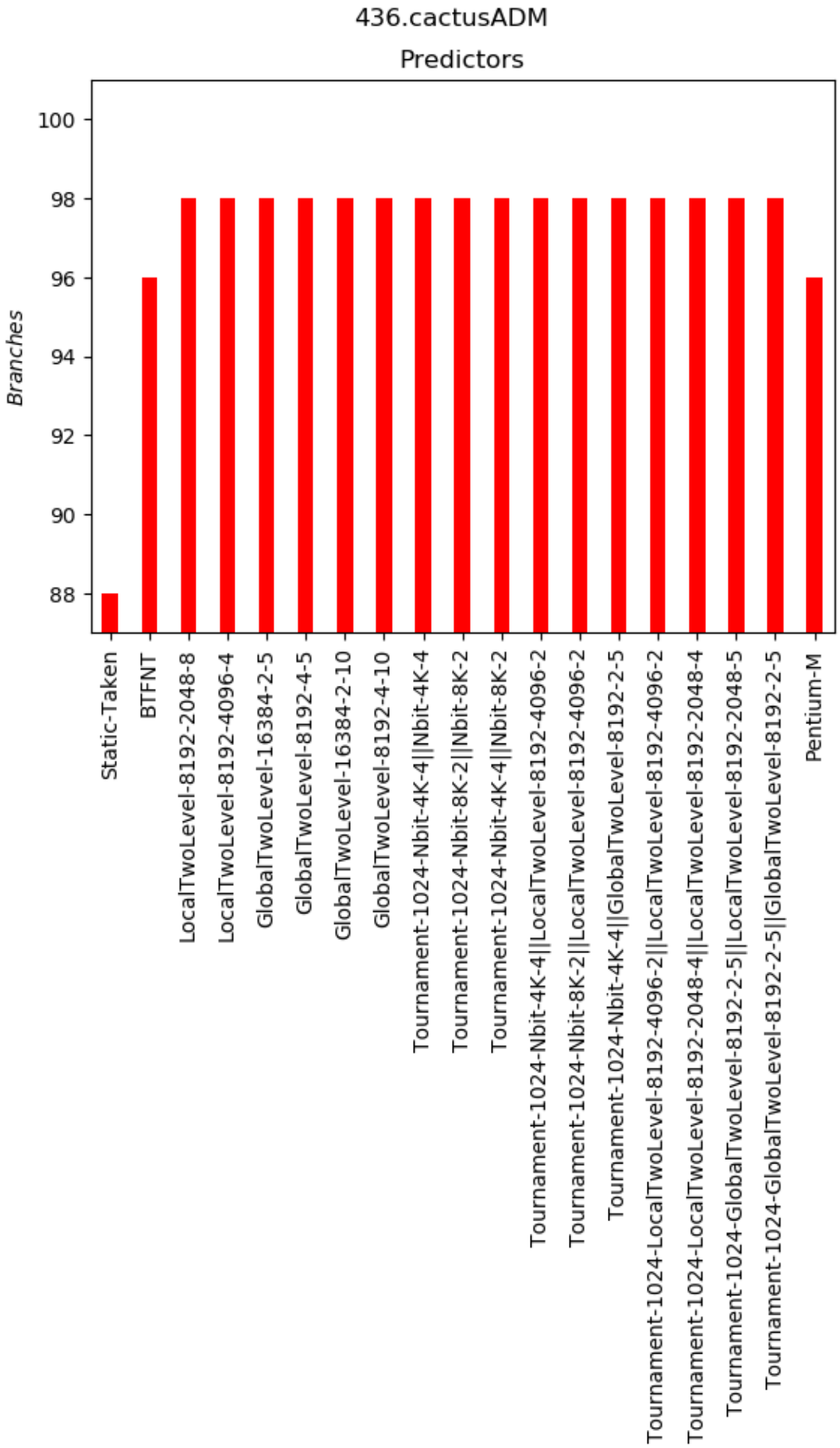
2) <u>429.mcf</u>


429.mcf
Predictors
Branches
90
80
70
60
50
40
Static-Taken
BTFNT
LocalTwoLevel-8192-2048-8
LocalTwoLevel-8192-4096-4
GlobalTwoLevel-16384-2-5
GlobalTwoLevel-8192-4-5
GlobalTwoLevel-16384-2-10
GlobalTwoLevel-8192-4-10
Tournament-1024-Nbit-4K-4||Nbit-4K-4
Tournament-1024-Nbit-8K-2||Nbit-8K-2
Tournament-1024-Nbit-4K-4||Nbit-8K-2
Tournament-1024-Nbit-4K-4||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-8K-2||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-4K-4||GlobalTwoLevel-8192-2-5
Tournament-1024-LocalTwoLevel-8192-4096-2||LocalTwoLevel-8192-4096-2
Tournament-1024-LocalTwoLevel-8192-2048-4||LocalTwoLevel-8192-2048-4
Tournament-1024-GlobalTwoLevel-8192-2-5||LocalTwoLevel-8192-2048-5
Tournament-1024-GlobalTwoLevel-8192-2-5||GlobalTwoLevel-8192-2-5
Pentium-M

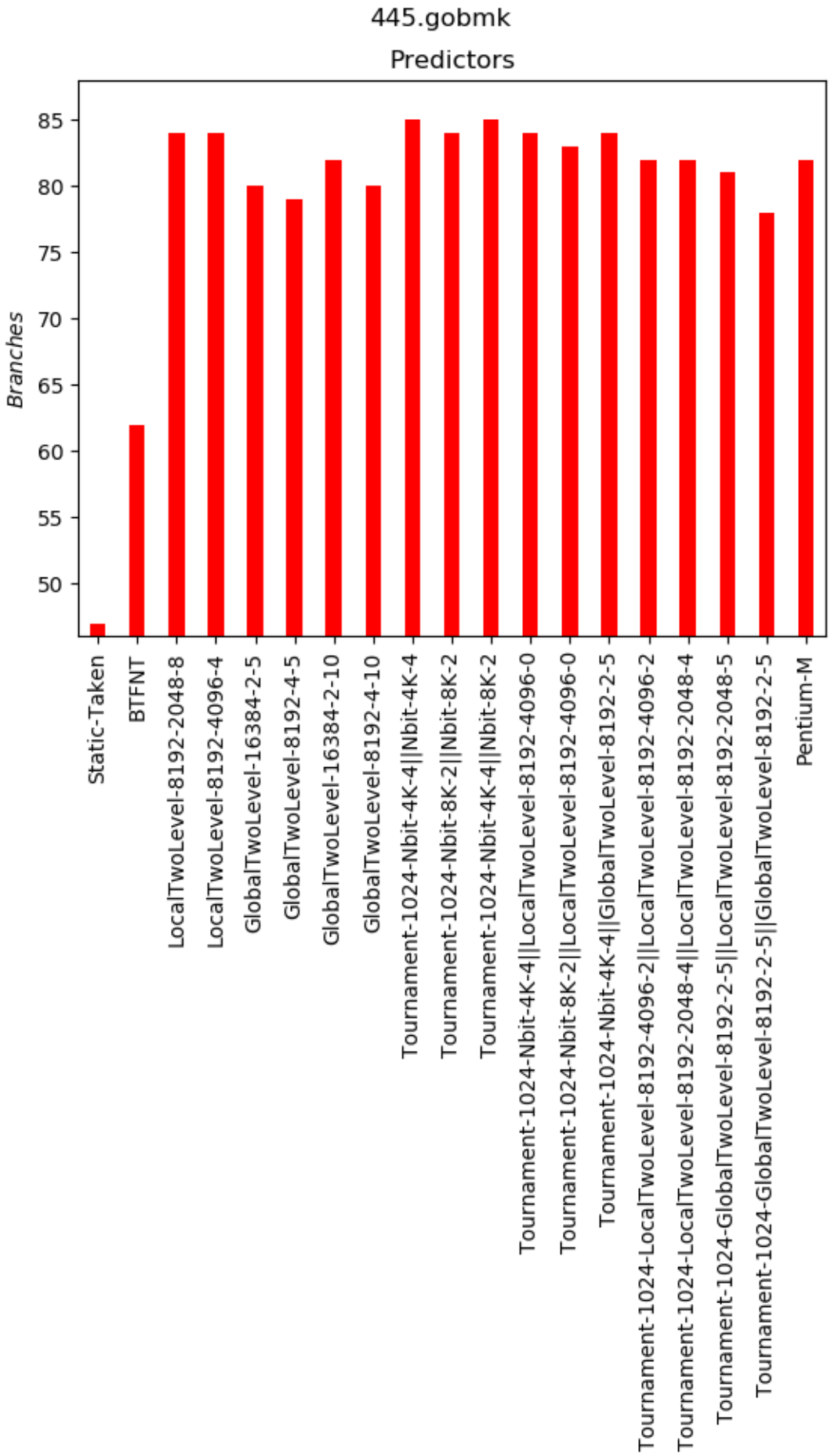
3) 434.zeusmp


434.zeusmp
Predictors
Branches
100
90
80
70
60
50
Static-Taken
BTFNT
LocalTwoLevel-8192-2048-8
LocalTwoLevel-8192-4096-4
GlobalTwoLevel-16384-2-5
GlobalTwoLevel-8192-4-5
GlobalTwoLevel-16384-2-10
GlobalTwoLevel-8192-4-10
Tournament-1024-Nbit-4K-4||Nbit-4K-4
Tournament-1024-Nbit-8K-2||Nbit-8K-2
Tournament-1024-Nbit-4K-4||Nbit-8K-2
Tournament-1024-Nbit-4K-4||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-8K-2||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-4K-4||GlobalTwoLevel-8192-2-5
Tournament-1024-LocalTwoLevel-8192-4096-2||LocalTwoLevel-8192-4096-2
Tournament-1024-LocalTwoLevel-8192-2048-4||LocalTwoLevel-8192-2048-4
Tournament-1024-GlobalTwoLevel-8192-2-5||LocalTwoLevel-8192-2048-5
Tournament-1024-GlobalTwoLevel-8192-2-5||GlobalTwoLevel-8192-2-5
Pentium-M

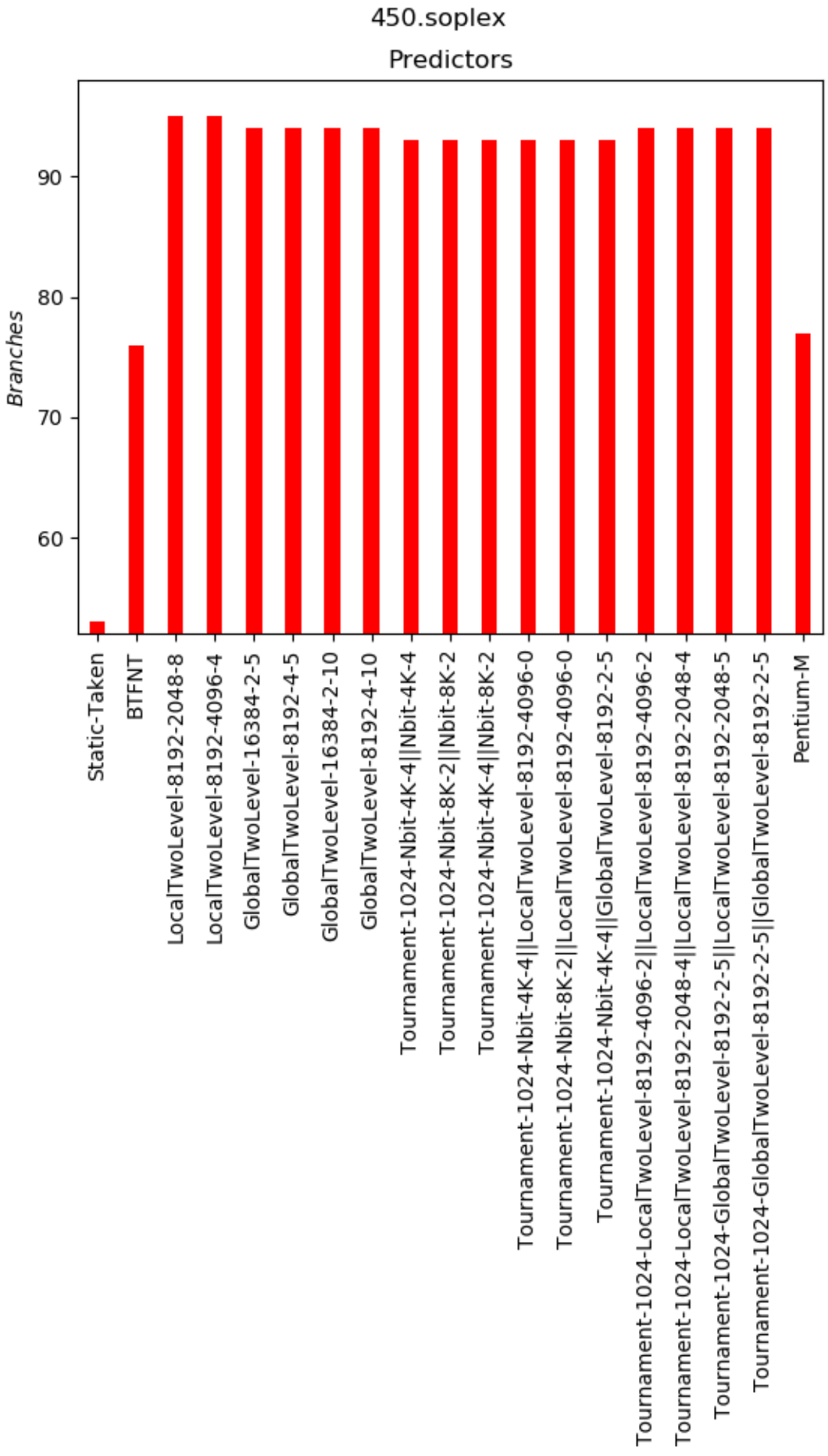
4) 436.cactusADM



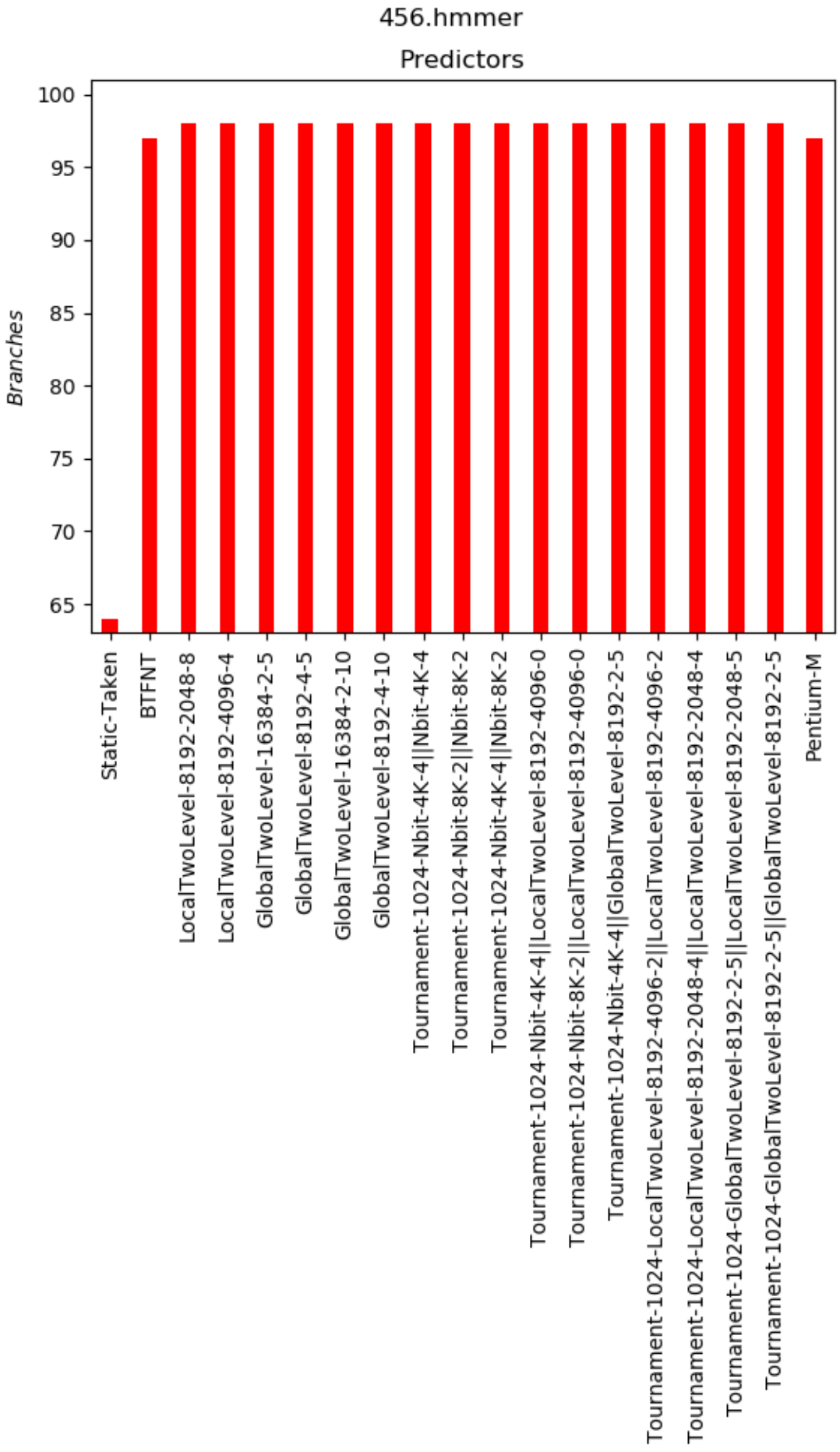
5) 445.gobmk


445.gobmk
Predictors
Branches
85
80
75
70
65
60
55
50
Static-Taken
BTFNT
LocalTwoLevel-8192-2048-8
LocalTwoLevel-8192-4096-4
GlobalTwoLevel-16384-2-5
GlobalTwoLevel-8192-4-5
GlobalTwoLevel-16384-2-10
GlobalTwoLevel-8192-4-10
Tournament-1024-Nbit-4K-4||Nbit-4K-4
Tournament-1024-Nbit-8K-2||Nbit-8K-2
Tournament-1024-Nbit-4K-4||Nbit-8K-2
Tournament-1024-Nbit-4K-4||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-8K-2||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-4K-4||GlobalTwoLevel-8192-2-5
Tournament-1024-LocalTwoLevel-8192-4096-2||LocalTwoLevel-8192-4096-2
Tournament-1024-LocalTwoLevel-8192-2048-4||LocalTwoLevel-8192-2048-4
Tournament-1024-GlobalTwoLevel-8192-2-5||LocalTwoLevel-8192-2048-5
Tournament-1024-GlobalTwoLevel-8192-2-5||GlobalTwoLevel-8192-2-5
Pentium-M

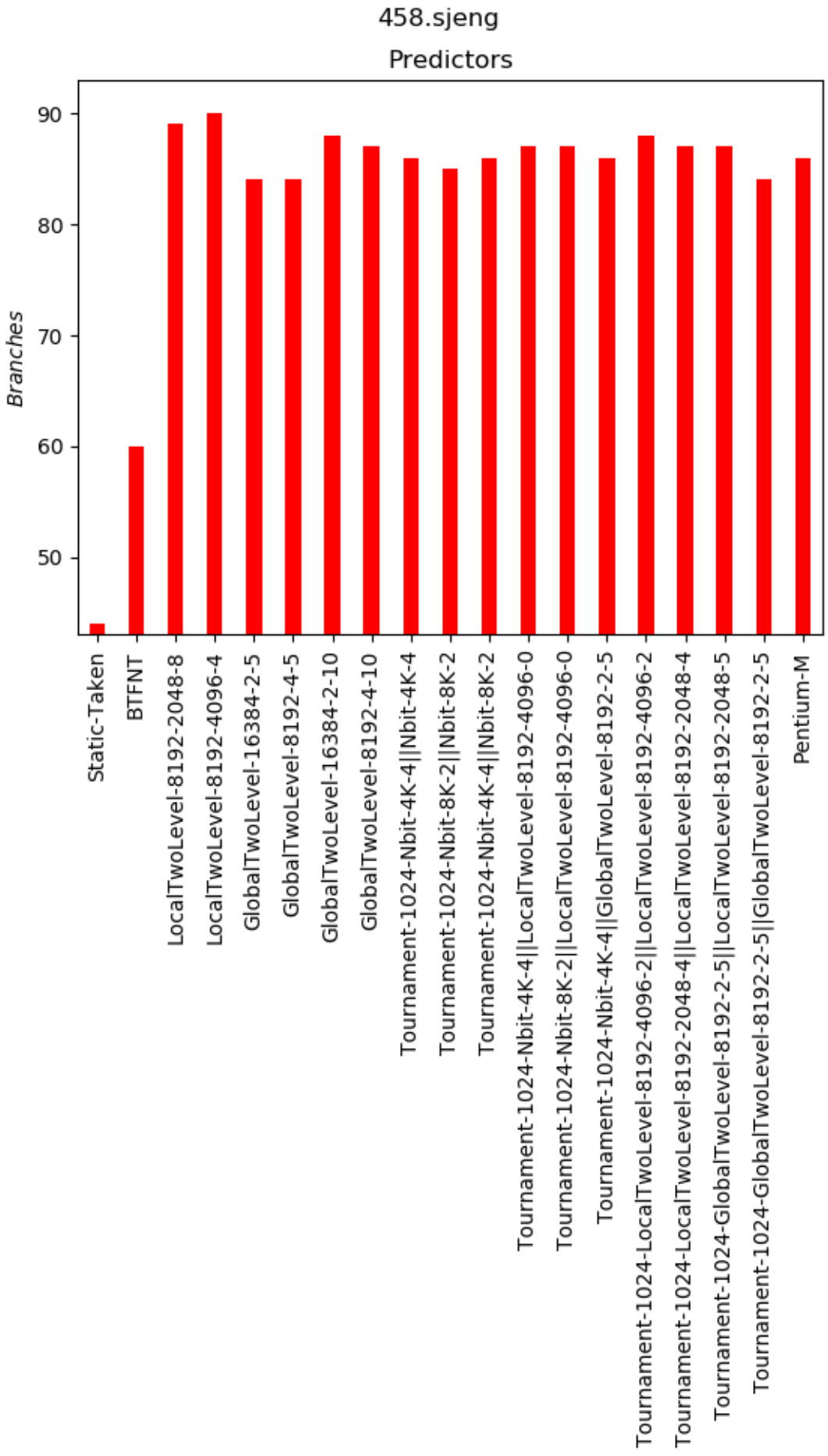
6) 450.soplex


450.soplex
Predictors
Branches
90
80
70
60
Static-Taken
BTFNT
LocalTwoLevel-8192-2048-8
LocalTwoLevel-8192-4096-4
GlobalTwoLevel-16384-2-5
GlobalTwoLevel-8192-4-5
GlobalTwoLevel-16384-2-10
GlobalTwoLevel-8192-4-10
Tournament-1024-Nbit-4K-4||Nbit-4K-4
Tournament-1024-Nbit-8K-2||Nbit-8K-2
Tournament-1024-Nbit-4K-4||Nbit-8K-2
Tournament-1024-Nbit-4K-4||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-8K-2||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-4K-4||GlobalTwoLevel-8192-2-5
Tournament-1024-LocalTwoLevel-8192-4096-2||LocalTwoLevel-8192-4096-2
Tournament-1024-LocalTwoLevel-8192-2048-4||LocalTwoLevel-8192-2048-4
Tournament-1024-GlobalTwoLevel-8192-2-5||LocalTwoLevel-8192-2048-5
Tournament-1024-GlobalTwoLevel-8192-2-5||GlobalTwoLevel-8192-2-5
Pentium-M

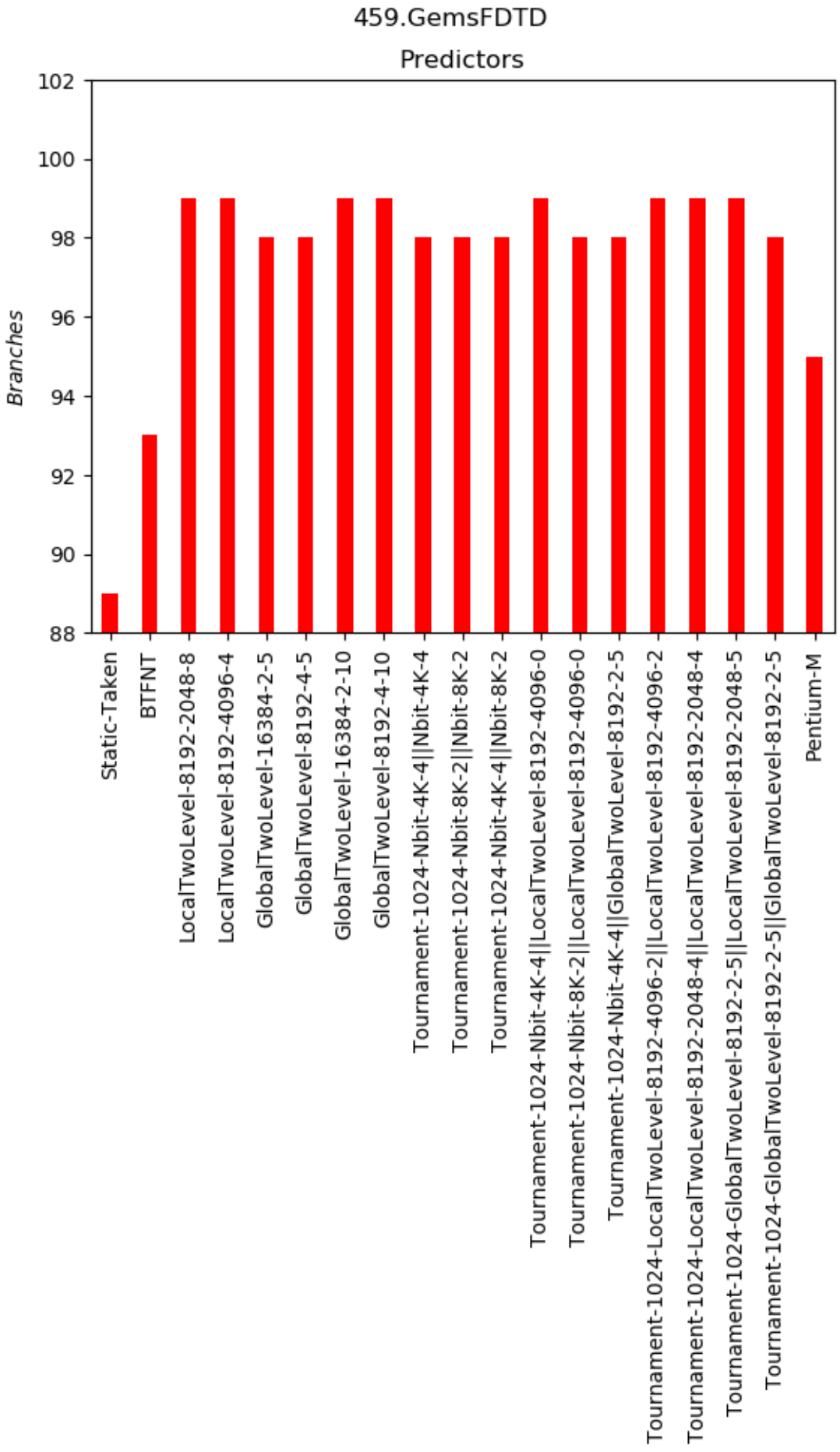
7) 456.hmmer


456.hmmer
Predictors
Branches
100
95
90
85
80
75
70
65
Static-Taken
BTFNT
LocalTwoLevel-8192-2048-8
LocalTwoLevel-8192-4096-4
GlobalTwoLevel-16384-2-5
GlobalTwoLevel-8192-4-5
GlobalTwoLevel-16384-2-10
GlobalTwoLevel-8192-4-10
Tournament-1024-Nbit-4K-4||Nbit-4K-4
Tournament-1024-Nbit-8K-2||Nbit-8K-2
Tournament-1024-Nbit-4K-4||Nbit-8K-2
Tournament-1024-Nbit-4K-4||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-8K-2||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-4K-4||GlobalTwoLevel-8192-2-5
Tournament-1024-LocalTwoLevel-8192-4096-2||LocalTwoLevel-8192-4096-2
Tournament-1024-LocalTwoLevel-8192-2048-4||LocalTwoLevel-8192-2048-4
Tournament-1024-GlobalTwoLevel-8192-2-5||LocalTwoLevel-8192-2048-5
Tournament-1024-GlobalTwoLevel-8192-2-5||GlobalTwoLevel-8192-2-5
Pentium-M

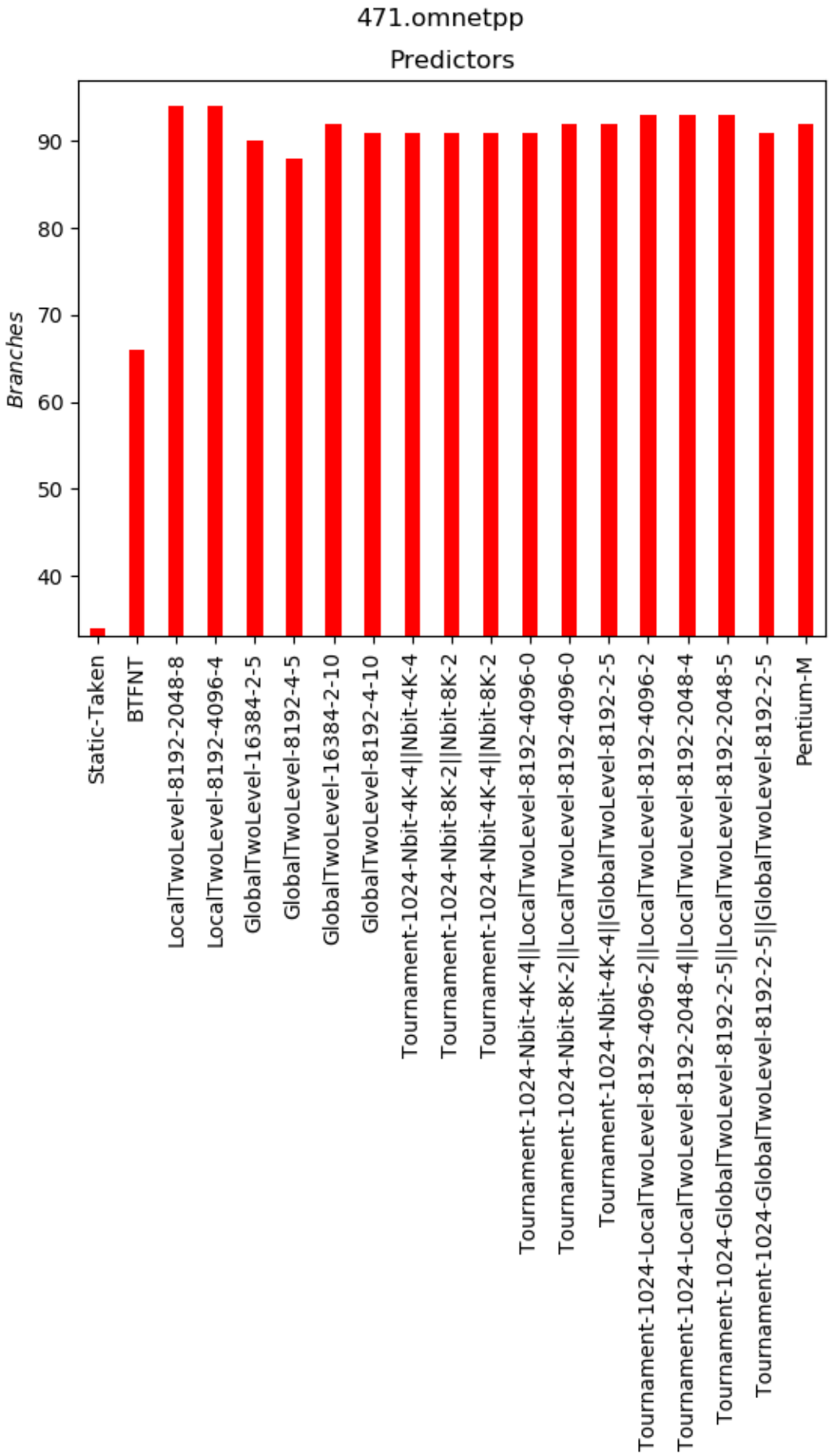
8) <u>458.sjeng</u>



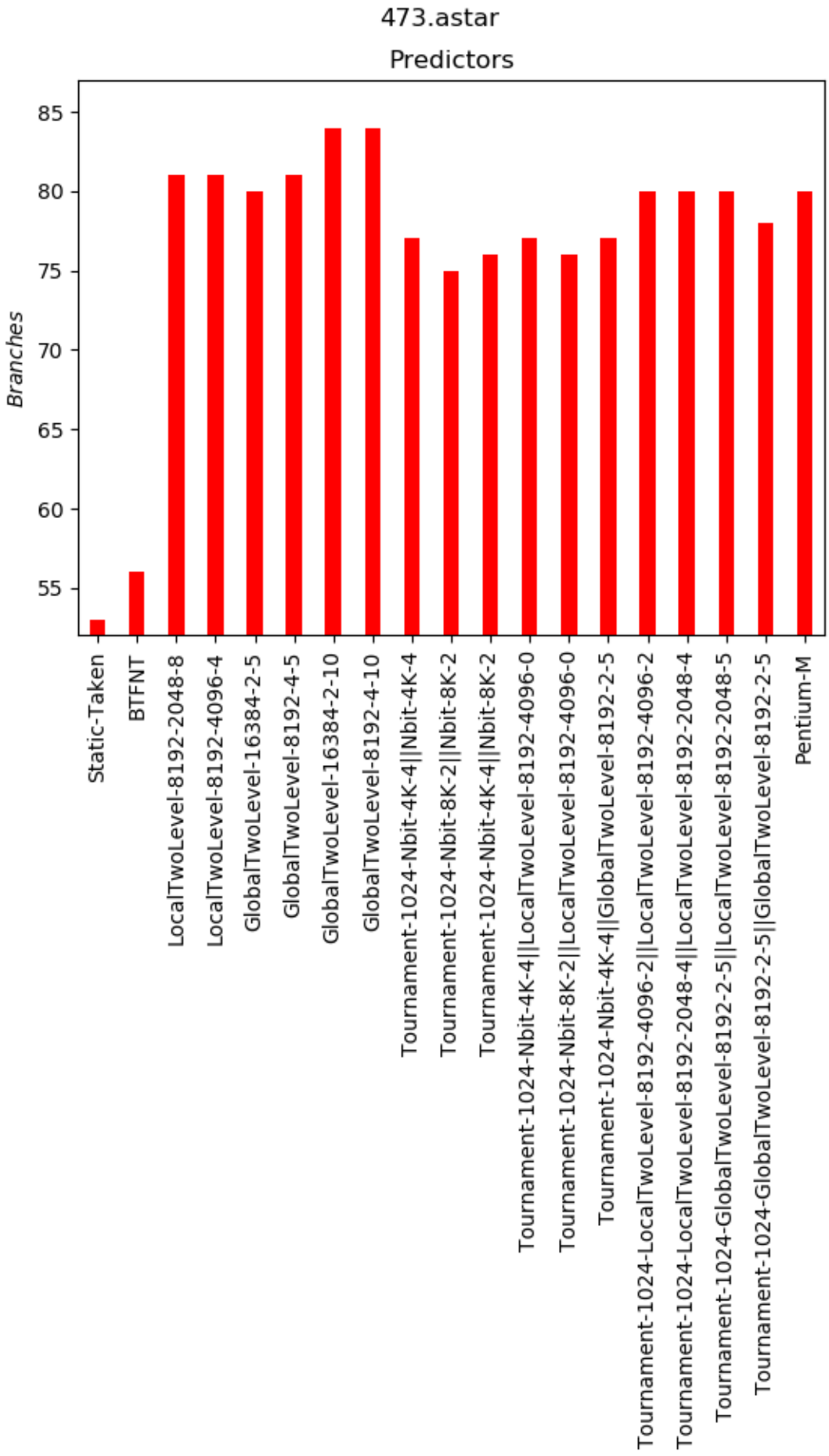
9) 459.GemsFDTD



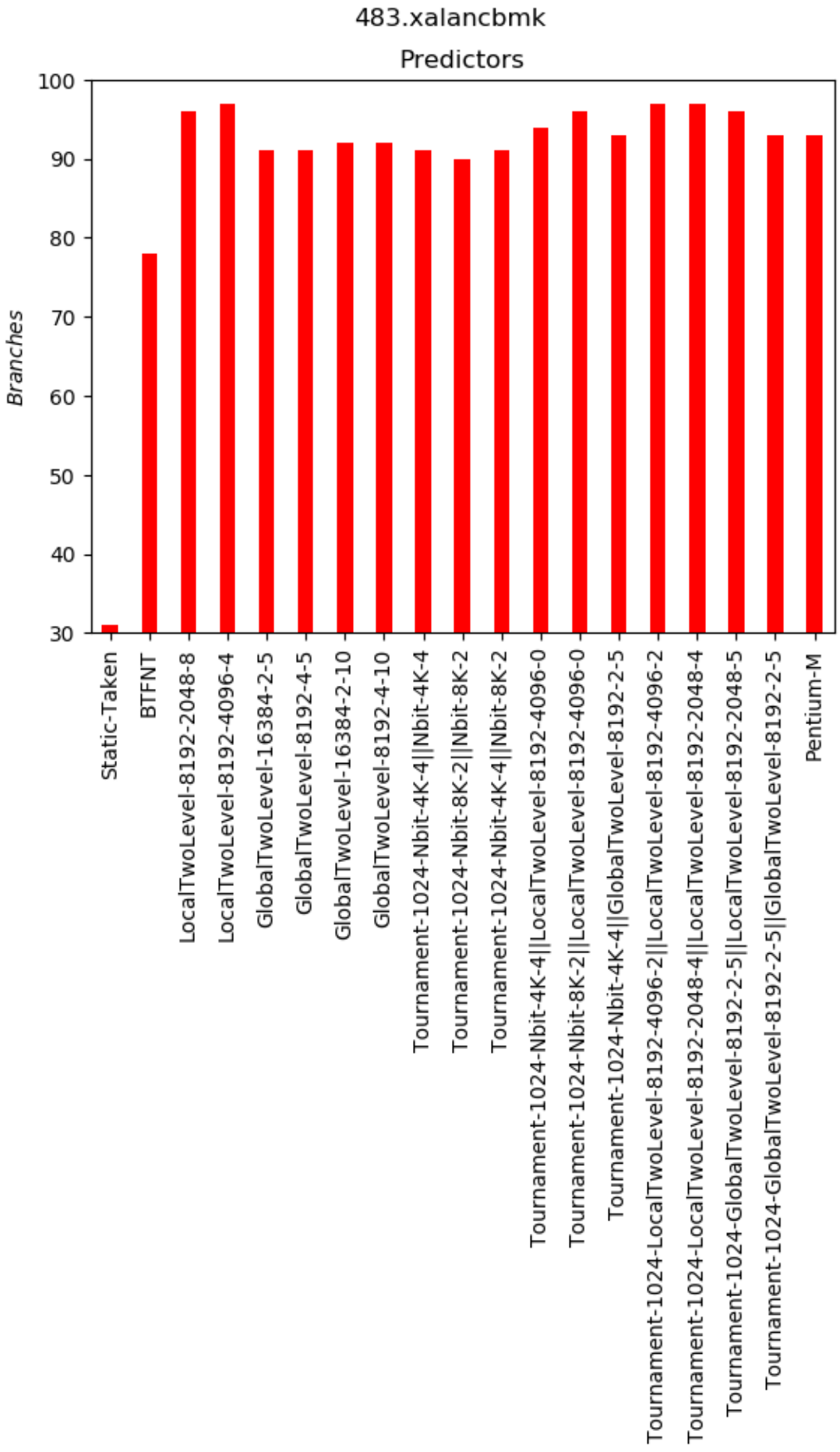
10) **471.omnetpp**



11) **473.astar**


473.astar
Predictors
Branches
85
80
75
70
65
60
55
Static-Taken
BTFNT
LocalTwoLevel-8192-2048-8
LocalTwoLevel-8192-4096-4
GlobalTwoLevel-16384-2-5
GlobalTwoLevel-8192-4-5
GlobalTwoLevel-16384-2-10
GlobalTwoLevel-8192-4-10
Tournament-1024-Nbit-4K-4||Nbit-4K-4
Tournament-1024-Nbit-8K-2||Nbit-8K-2
Tournament-1024-Nbit-4K-4||Nbit-8K-2
Tournament-1024-Nbit-4K-4||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-8K-2||LocalTwoLevel-8192-4096-0
Tournament-1024-Nbit-4K-4||GlobalTwoLevel-8192-2-5
Tournament-1024-LocalTwoLevel-8192-4096-2||LocalTwoLevel-8192-4096-2
Tournament-1024-LocalTwoLevel-8192-2048-4||LocalTwoLevel-8192-2048-4
Tournament-1024-GlobalTwoLevel-8192-2-5||LocalTwoLevel-8192-2048-5
Tournament-1024-GlobalTwoLevel-8192-2-5||GlobalTwoLevel-8192-2-5
Pentium-M

12) **483.xalancbmk**



Υπάρχουν δύο ειδών predictors, οι static και οι dynamic. Οι πρώτοι λειτουργούν ανεξάρτητα από την ροή του προγράμματος, ενώ οι δεύτεροι μπορούν να αλλάξουν τον τρόπο πρόβλεψης κατά την εκτέλεση ενός προγράμματος. Οι predictors στους οποίους θα εστιάσουμε αναφέρονται παρακάτω :

Static predictors :

1. Static AlwaysTaken
2. BTFNT

Οι στατικοί προβλεπτές συνήθως δεν έχουν καλή επίδοση καθώς δεν μπορούν να προσαρμοστούν στο εκάστοτε πρόγραμμα. Ωστόσο είναι αρκετά εύκολοι στην υλοποίηση και το κόστος τους ελάχιστο.

Dynamic predictors :

Εδώ θα διακρίνουμε τις εξής περιπτώσεις :

1. N-bit predictor.
2. Global History Predictor.
3. Local History Predictor.

Είναι εμφανές πως οι παραπάνω τεχνικές έχουν βασικές αδυναμίες για συγκεκριμένα branch, γεγονός που μπορεί να ρίξει πολύ την απόδοση του συστήματος. Θα θέλαμε κάτι καλύτερο από όλα τα παραπάνω. Για τον σκοπό αυτό, έχουμε τους λεγόμενους tournament predictors. Πρακτικά αποτελείται από δύο κυρίως βαθμίδες. Έναν meta predictor στο πρώτο σκέλος, και έναν οποιαδήποτε συνδυασμό δύο από τους παραπάνω προβλεπτές. Επιλέξαμε να προσομοιώσουμε διάφορους συνδυασμούς από predictors για τον tournament ώστε να έχουμε μια ολοκληρωμένη εικόνα.

Βάσει των παραπάνω διαγραμμάτων, είναι ξεκάθαρη η υπεροχή των dynamic predictors έναντι των static, καθώς οι dynamic εξασφαλίζουν ικανοποιητικά ποσοστά απόδοσης, που στην πλειοψηφία των benchmarks ξεπερνούν το 90%. Στο σύνολό τους, όλοι οι dynamic predictors παρουσιάζουν παρόμοια ποσοστά απόδοσης, με ορισμένους να εμφανίζουν μεγαλύτερη σταθερότητα ανά τα benchmarks. Συγκεκριμένα, φαίνεται να έχουν καλύτερες αποδόσεις οι LocalTwoLevel 8192-2048-8 και 8192-4096-4. Όσον αφορά τους tournament predictors, είναι εμφανές πως όλοι οι διαφορετικοί συνδυασμοί έχουν ικανοποιητικά αποτελέσματα, με τον καθένα από αυτούς να υπερτερεί ανάλογα την εφαρμογή. Σίγουρα 2 από τους κορυφαίους συνδυασμούς είναι οι δύο local-history-predictors(είτε 8192-4096-2 είτε 8192-2048-4) καθώς και ένας local με έναν global history predictor.