Εθνικο Μετσοβιο Πολυτεχνειο

ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

2η ΑΣΚΗΣΗ

Γεωργίου Δημήτριος (03115106)

<el15106@central.ntua.gr>

Απρίλιος 2019

1 Σκοπός της Άσκησης

- Η συγκεκριμένη εργαστηριακή άσκηση έχει ως σκοπό την μελέτη της επίδρασης διαφορετικών συστημάτων πρόβλεψης εντολών άλματος και την αξιολόγηση τους με αντικειμενικό κριτήριο τον διαθέσιμο χώρο του τσιπ. **Πιο συγκεκριμένα** γίνεται εξέτασης της επίδρασης των βασικών predictors στην απόδοση 12 μετροπρογραμμάτων (**benchmarks**).
- Μετά το πέρας της εξέτασης, θα γίνει επιλογή του καταλληλότερου predictor για παραπάνω 12 μετροπρογράμματα.

2 Εργαλεία

PIN TOOL

- Έγινε χρήση του εργαλείου Pin, ένα εργαλείο ανάλυσης εφαρμογών που αναπτύσεται και συντηρείται από την εταιρεία Intel
- Προσφέρει δυνατότητες για dynamic binary instrumentation, ή εναλλακτικά επιτρέπει την δυναμική εισαγωγή και τροποποίηση κώδικα της εφαρμογής κατά την διάρκεια εκτέλεσης των αυτών των μετροπρογραμμάτων. Απώτερος σκοπός είναι η συλλογή πληροφοριών που αφορούν την εκτέλεση, συγκεκριμένα **αρχιθμός εντολών**, **αριθμός misses και hits στην cache**
- Σημειώνεται ότι στα πλαίσια της παρούσας εργασίας έγινε χρήση της **PIN 97554 Ubuntu Linux 16.04** έκδοσης του PIN με **πυρήνα 4.4**

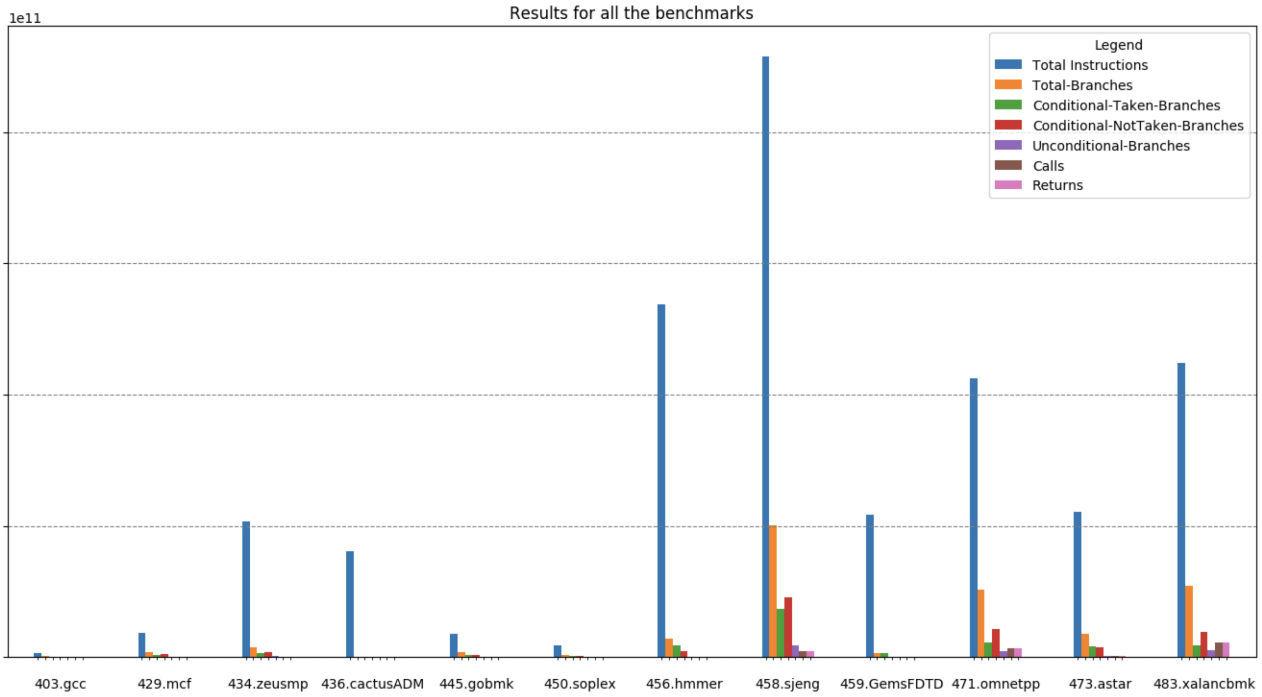
Benchmarks

- Τα 12 μετροπρογράμματα (**benchmarks**) που χρησιμοποιήθηκαν ήταν **SPEC_CPU2006**
- Στα πλαίσια των προσομοιώσεων μας χρησιμοποιήθηκαν 12 από αυτά τα benchmarks. Συγκεκριμένα:
  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

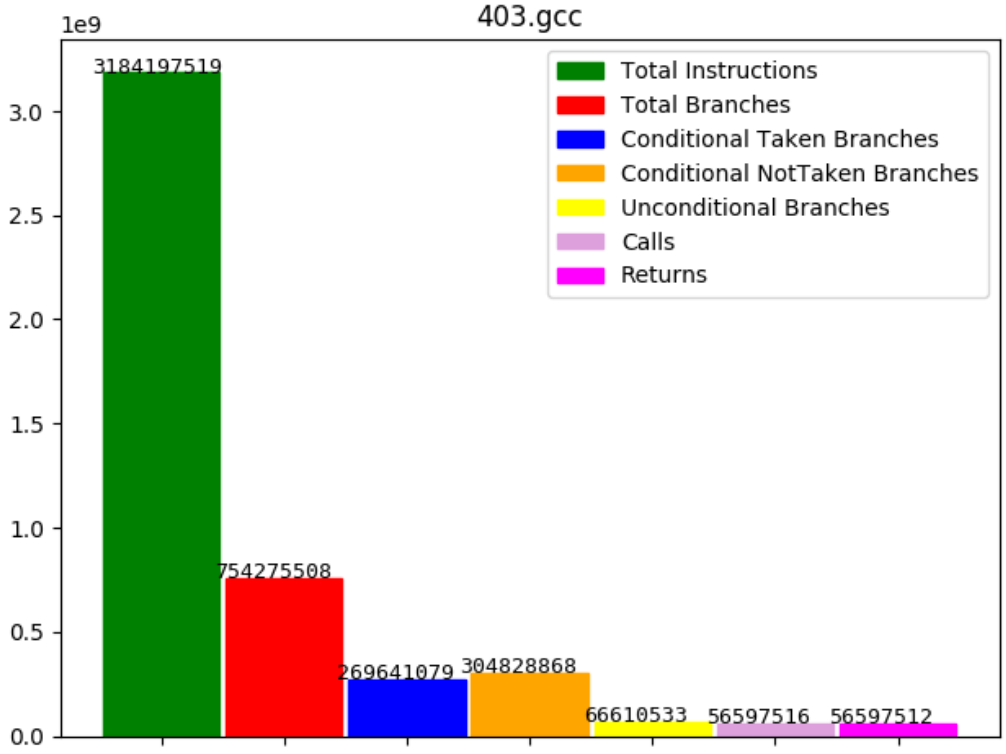
3 Πειραματική Αξιολόγηση

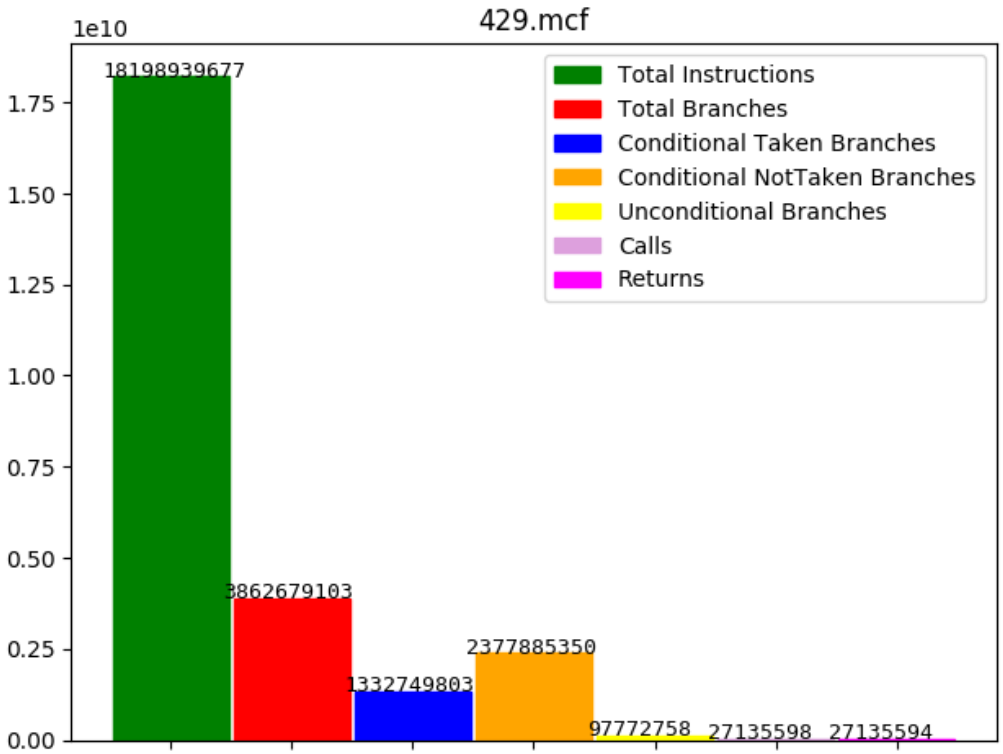
3.1 Μελέτη εντολών άλματος

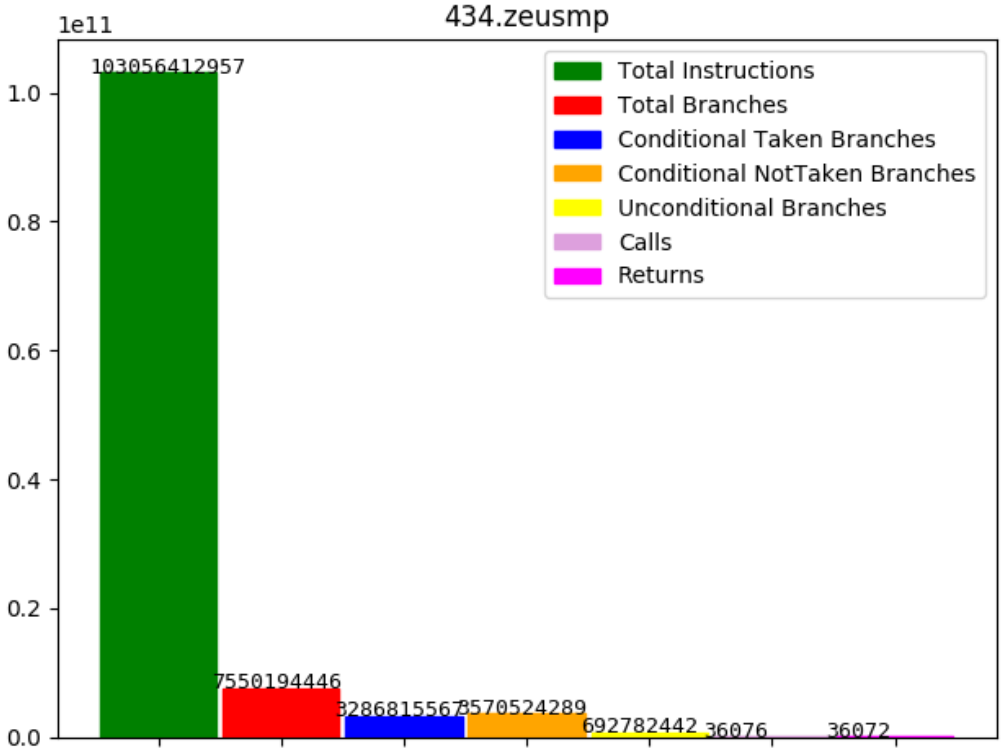
- Στο **Πρώτο Μέρος** της πειραματικής αξιολόγησης για την παρούσα άσκηση, εξετάζουμε τις εντολές άλματος που εκτελούνται για κάθε ένα από τα 12 μετροπρογράμματα. Εν συνεχεία, παρουσιάζουμε ένα διάγραμμα με το πλήθος και τα είδη των εντολών άλματος.
- **Παρατηρούμε**:
  1. Η πλειοψηφία των εντολών είναι conditional-taken και conditional-notTaken
  2. Μια γενικότερη εκτίμηση που προκύπτει από το διάγραμμα είναι ότι οι εντολές άλματος αποτελούν κατά μέσο όρο το **15%** των συνολικών εντολών που αποτελούν το μετροπρόγραμμα
  3. Έχουμε να κάνουμε με μετροπρογράμματα που αποτελούνται γενικότερα από **δισεκατομμύρια** εντολές και μάλιστα, παρατηρούμε ένα range της τάξης από **3 δισ. έως 400 δισ. εντολές**

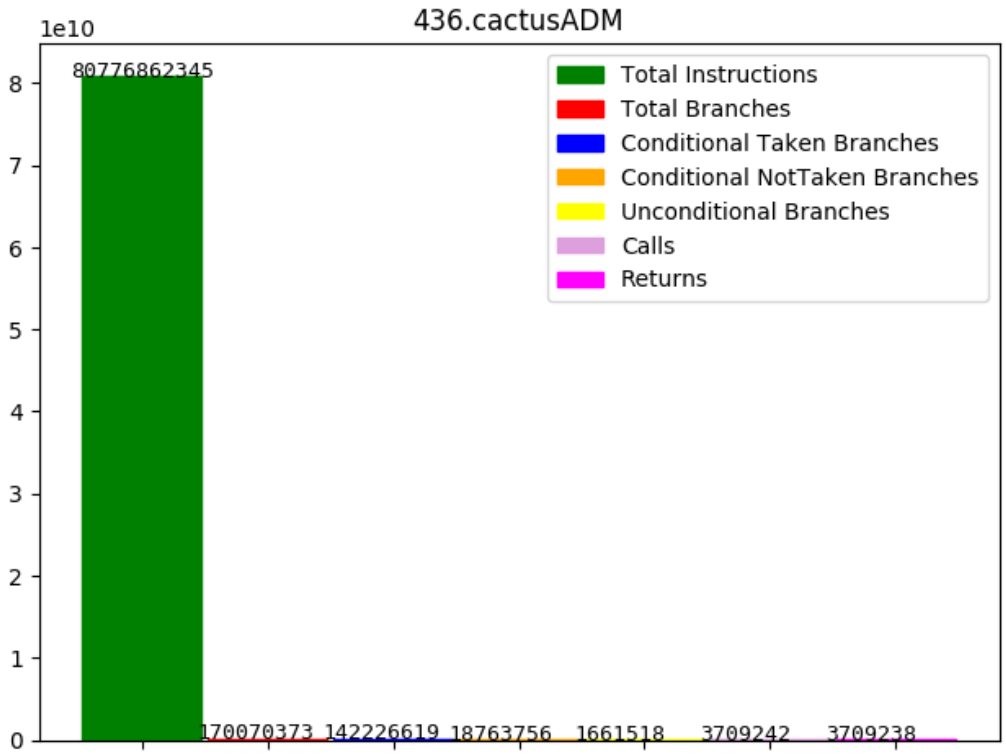


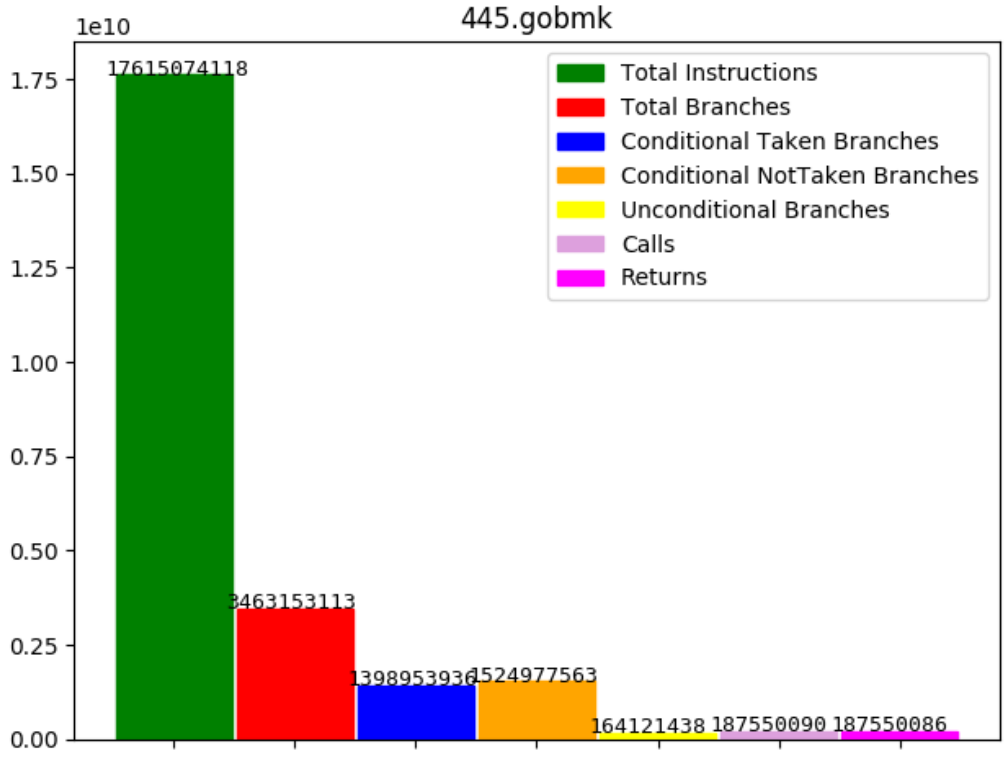
Παρακάτω παραθέτουμε ξεχωριστά για κάθε benchmark το αντίστοιχο διάγραμμα για τις εντολές άλματος, με μεγαλύτερη λεπτομέρεια, για καλύτερη επισκόπηση και εξαγωγή συμπεσμάτων

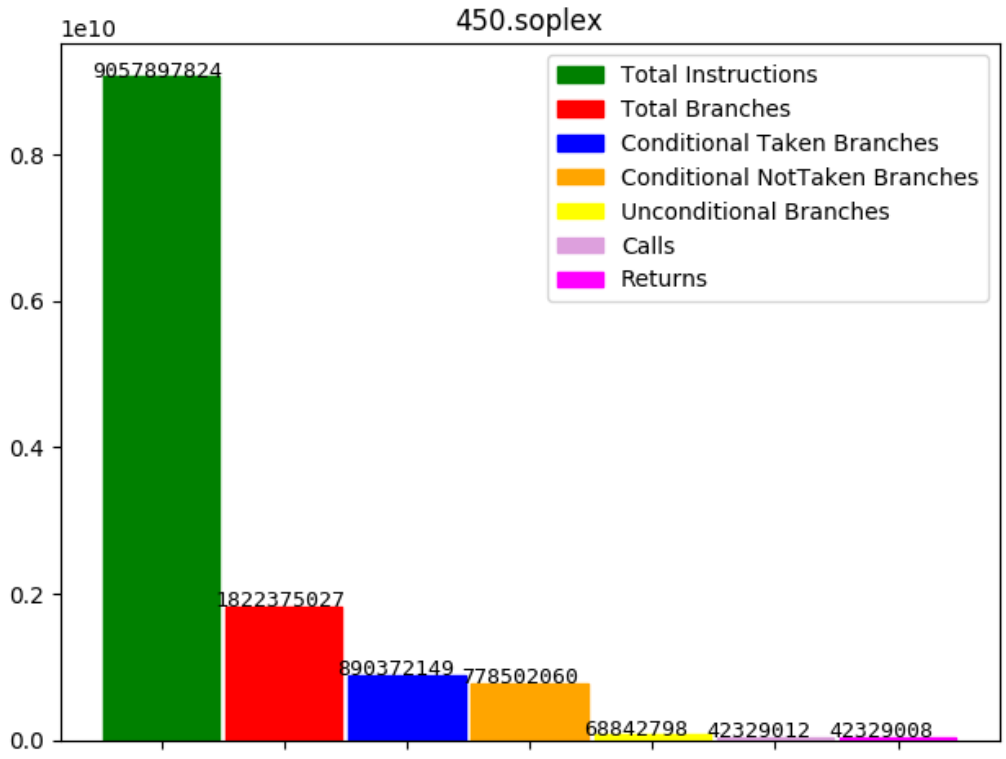


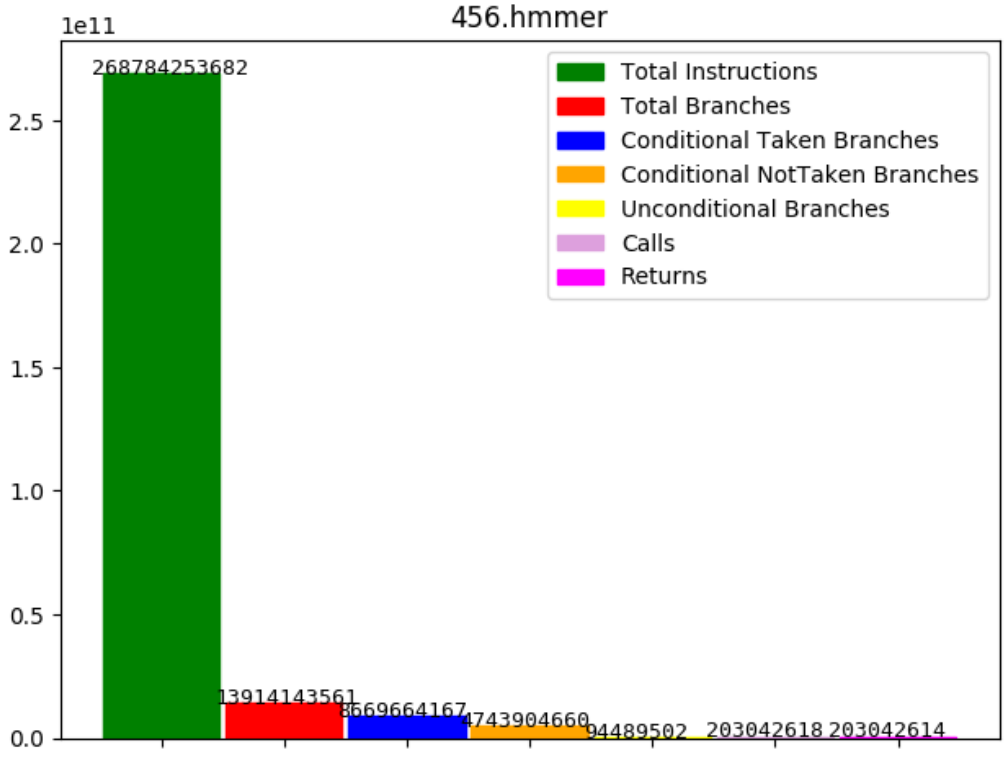

429.mcf
1e10
18198939677
3862679103
1332749803
2377885350
97772758
27135598
27135594
Total Instructions
Total Branches
Conditional Taken Branches
Conditional NotTaken Branches
Unconditional Branches
Calls
Returns
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75

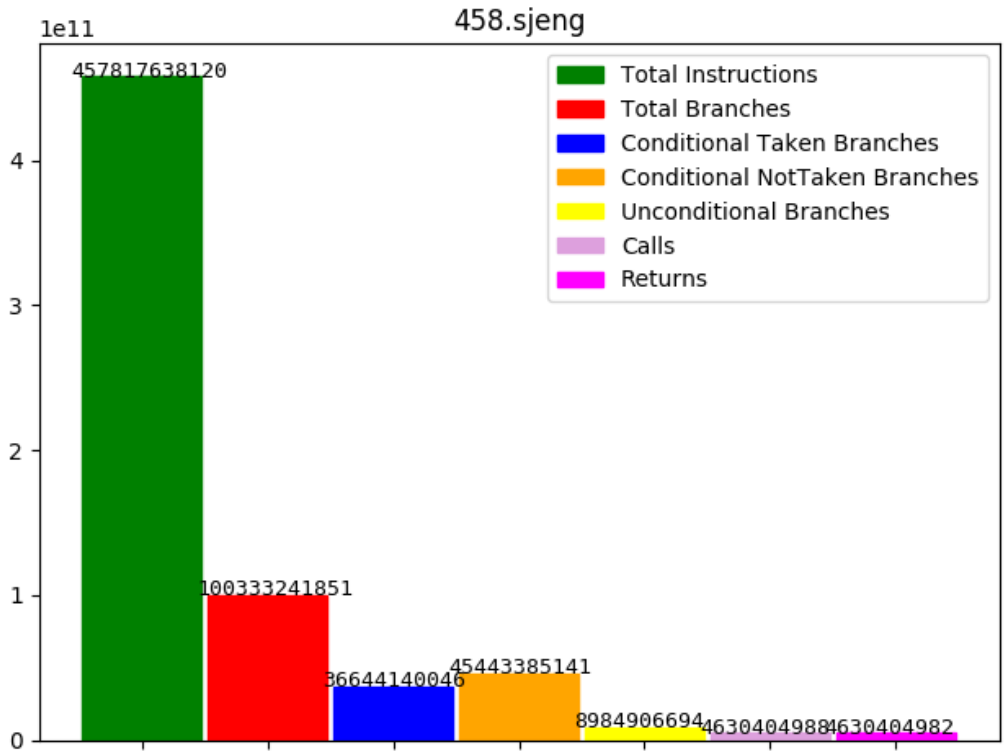

434.zeusmp
1e11
103056412957
7550194446
3286815567
3570524289
692782442
36076
36072
Total Instructions
Total Branches
Conditional Taken Branches
Conditional NotTaken Branches
Unconditional Branches
Calls
Returns
0.0
0.2
0.4
0.6
0.8
1.0

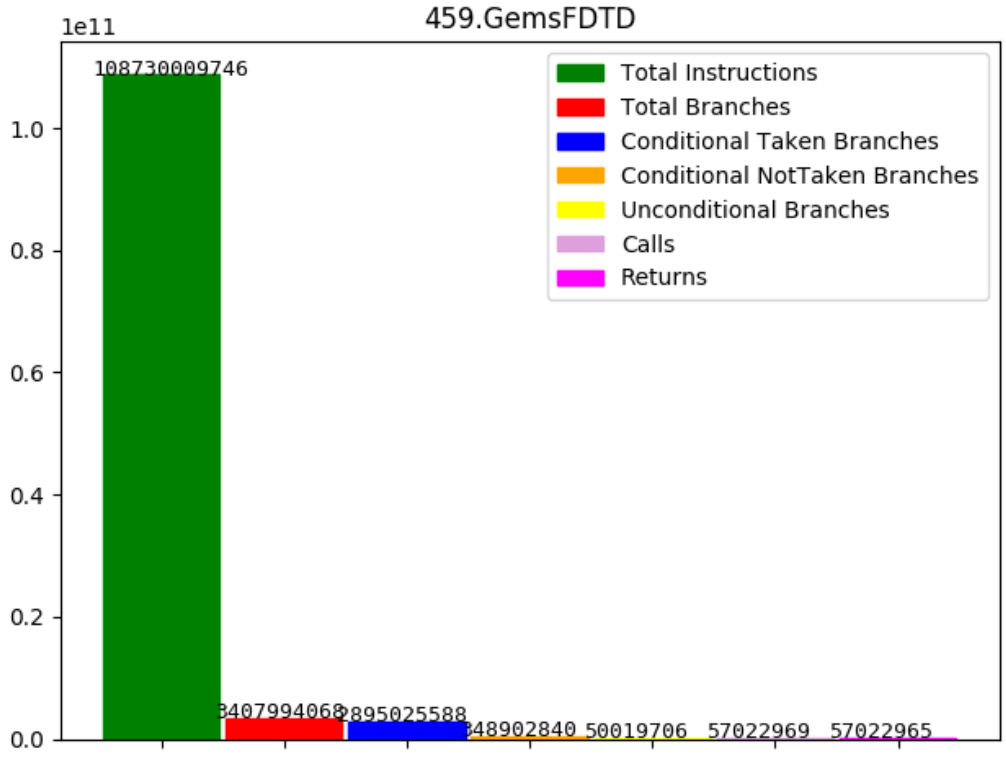

436.cactusADM
1e10
80776862345
170070373
142226619
18763756
1661518
3709242
3709238
Total Instructions
Total Branches
Conditional Taken Branches
Conditional NotTaken Branches
Unconditional Branches
Calls
Returns

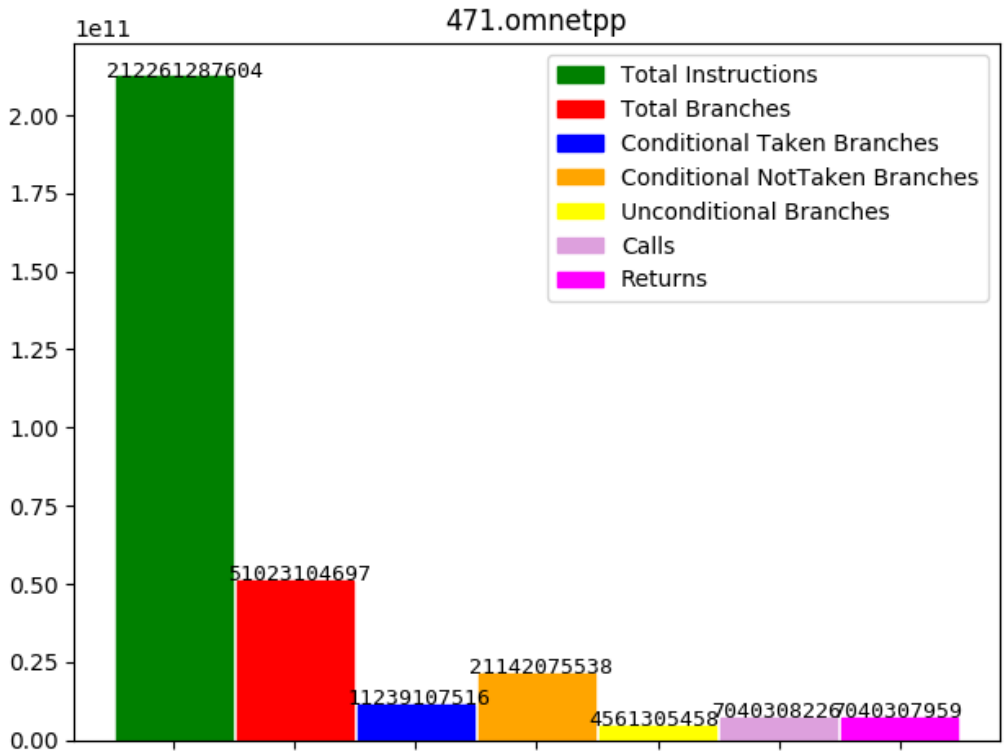

445.gobmk
1e10
17615074118
3463153113
1398953936
1524977563
164121438
187550090
187550086
Total Instructions
Total Branches
Conditional Taken Branches
Conditional NotTaken Branches
Unconditional Branches
Calls
Returns

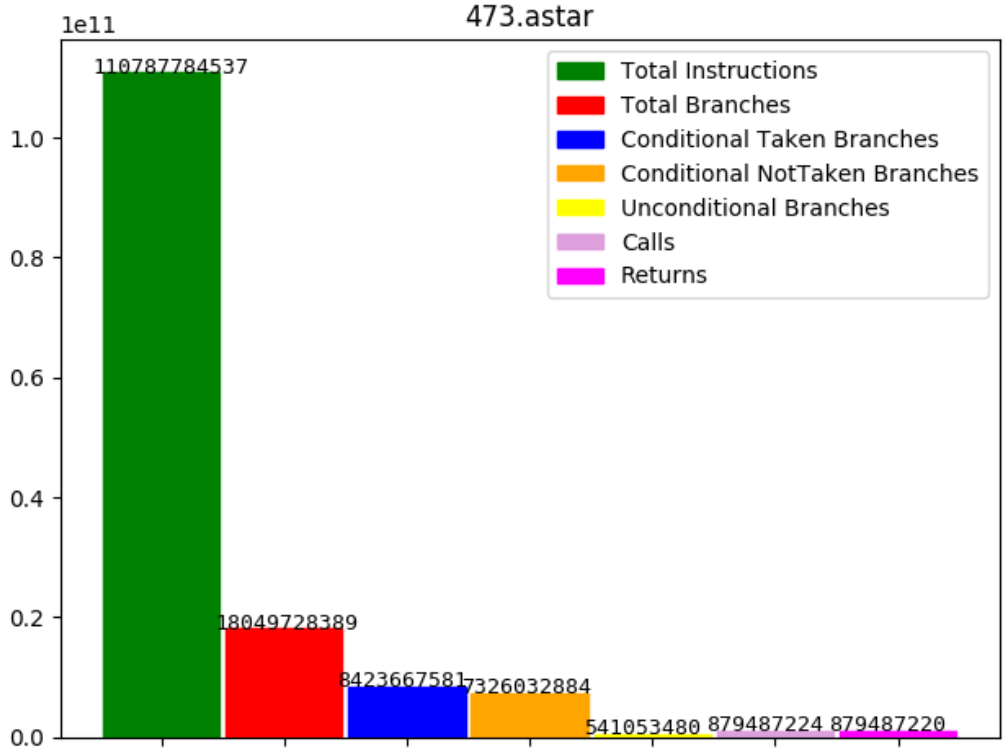

450.soplex
1e10
9057897824
1822375027
890372149
778502060
68842798
42329012
42329008
0.0
0.2
0.4
0.6
0.8
Total Instructions
Total Branches
Conditional Taken Branches
Conditional NotTaken Branches
Unconditional Branches
Calls
Returns

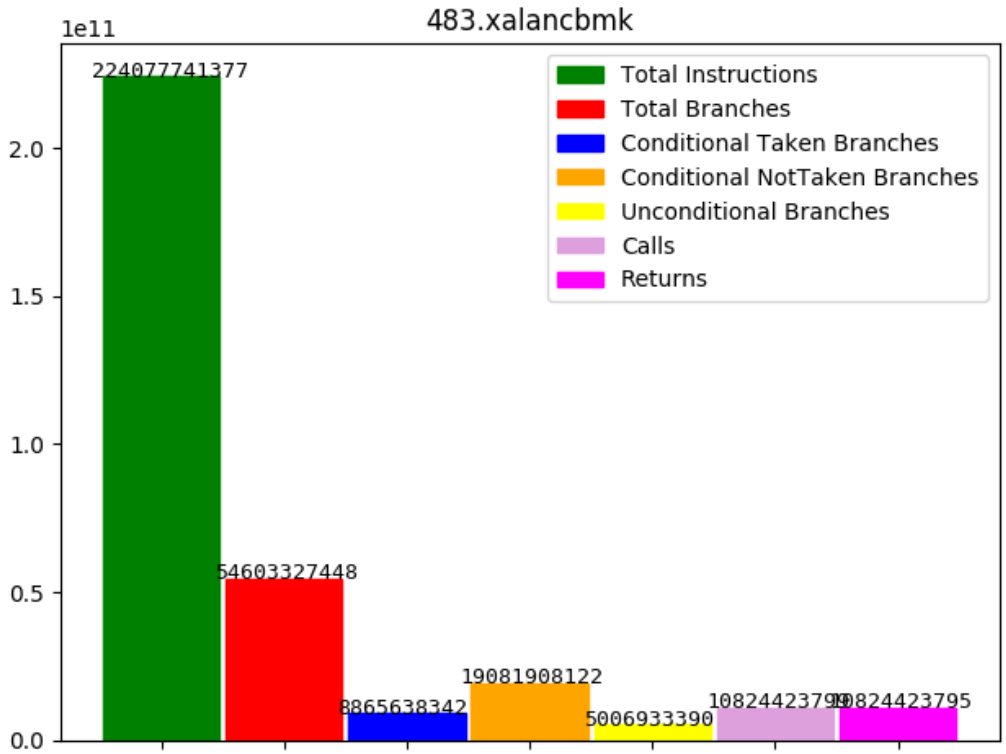

456.hmmer
1e11
268784253682
13914143561
8669664167
4743904660
94489502
203042618
203042614
0.0
0.5
1.0
1.5
2.0
2.5
Total Instructions
Total Branches
Conditional Taken Branches
Conditional NotTaken Branches
Unconditional Branches
Calls
Returns


458.sjeng
1e11
457817638120
100333241851
36644140046
45443385141
8984906694
4630404988
4630404982
Total Instructions
Total Branches
Conditional Taken Branches
Conditional NotTaken Branches
Unconditional Branches
Calls
Returns
0
1
2
3
4


459.GemsFDTD
1e11
108730009746
3407994068
2895025588
348902840
50019706
57022969
57022965
Total Instructions
Total Branches
Conditional Taken Branches
Conditional NotTaken Branches
Unconditional Branches
Calls
Returns
0.0
0.2
0.4
0.6
0.8
1.0


471.omnetpp
1e11
212261287604
51023104697
11239107516
21142075538
4561305458
7040308226
7040307959
Total Instructions
Total Branches
Conditional Taken Branches
Conditional NotTaken Branches
Unconditional Branches
Calls
Returns
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00


473.astar
1e11
110787784537
18049728389
8423667581
7326032884
541053480
879487224
879487220
Total Instructions
Total Branches
Conditional Taken Branches
Conditional NotTaken Branches
Unconditional Branches
Calls
Returns
1.0
0.8
0.6
0.4
0.2
0.0



- **Συμπεράσματα**:

  1. Πολύ λίγα είναι τα unconditional branches, τα calls και returns όπως αναμέναμε.
  2. Γενικότερα το μεγαλύτερο ποσοστό εντολών δεν είναι τα branches, κατά μέσο όρο αποτελούν το 15% των συνολικών εντολών του μετροπρογράμματος. Εντολές **load** και **store** σημειώνουν υψηλά ποσοστά! Πάντως συγκεκριμένα για τα benchmarks μας, το *hmmer* έχει υψηλά ποσοστά εντολών μνήμης, ενώ τα *gcc, mcfm xalancbmk, sjeng* έχουν υψηλότερα ποσοστά εντολών άλματος.
  3. Παρ' όλα αυτά, τα υψηλά αυτά ποσοστά **δεν αποτελούν** ένδειξη για παρουσία περισσότερων **bottlenecks**. Υπάρχουν benchmarks που σημειώνουν υψηλά ποσοστά και στις δύο κατηγορίες εντολών και εντούτοις διατηρούν **υψηλό IPC**

3.2 Μελέτη N-bit predictors

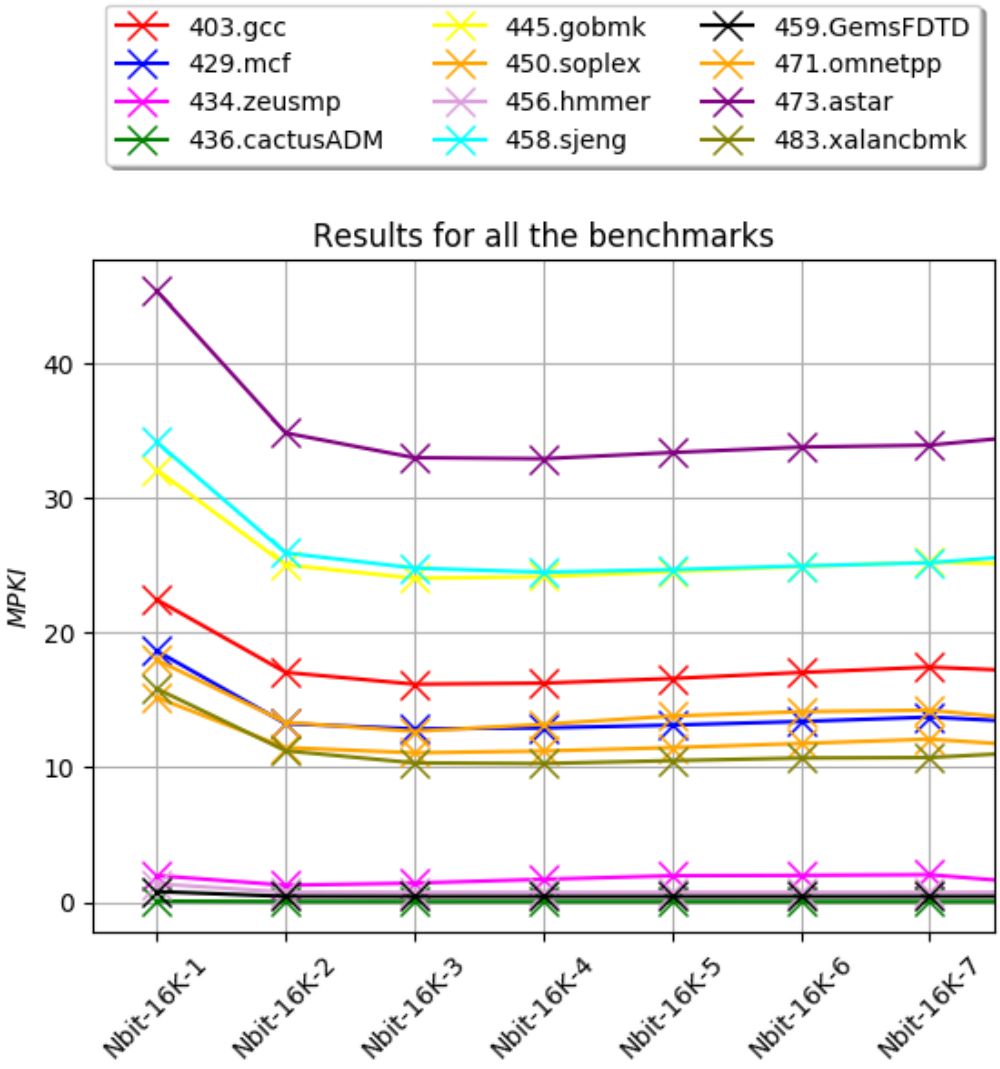
- Στο **Δεύτερο Μέρος** της πειραματικής αξιολόγησης για την παρούσα άσκηση, εξετάζουμε την απόδοση των **n-bits predictors** κάνοντας χρήση της υλποίησης που μας δόθηκε *cslab_branch.cpp*.

- **Συγκεκριμένα** το Δεύτερο Μέρος αποτελείται από δύο επιμέρους στάδια.

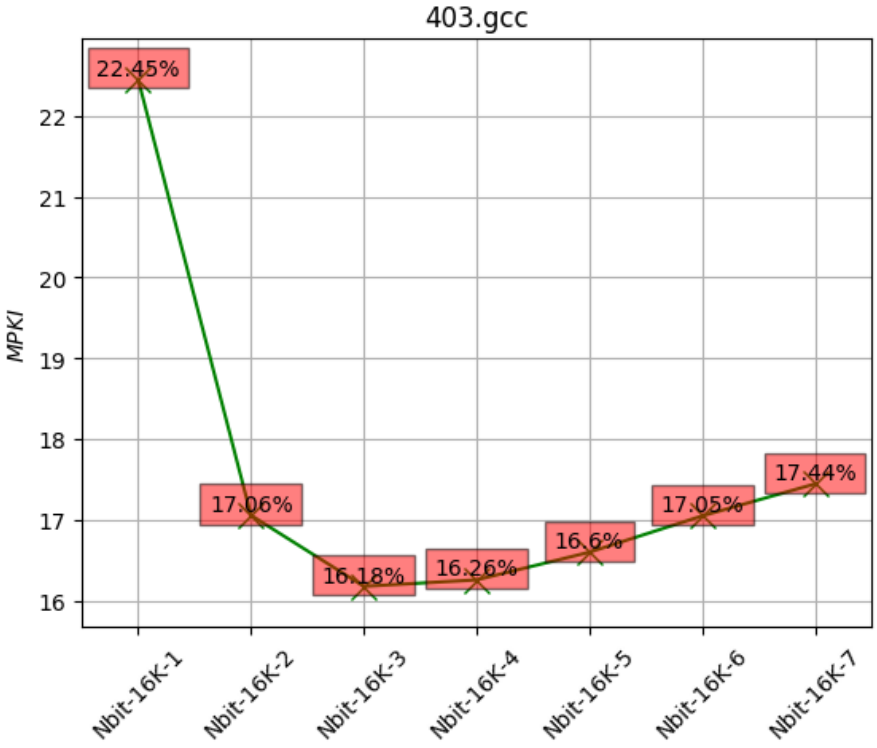
  – Στο **πρώτο στάδιο** διατηρούμε σταθερό τον αριθμό των BHT entries και ίσο με 16K, και γίνεται προσομοιώση των n-bits predictors για $n = 1, ...7$ κάνοντας χρήση των direction MPKI (Mispredictions Per Thousands Instructions)

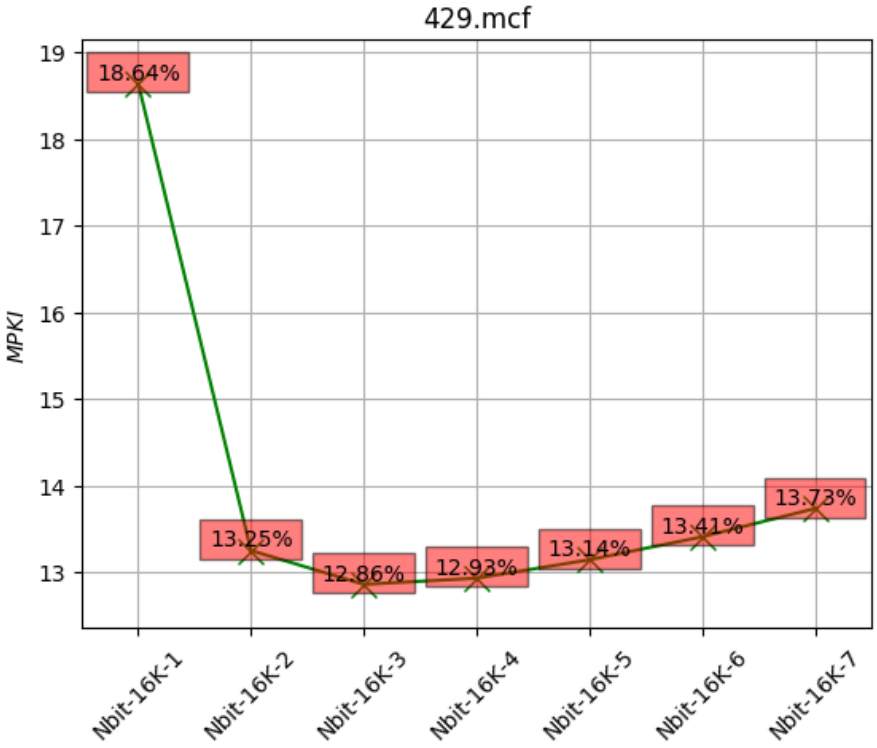
  – Στο **δεύτερο στάδιο** διατηρούμε σταθερό το hardware και ίσο με 32K bits, και γίνεται εκτέλεση των προσομοιώσεων των n-bits predictors για τα 12 benchmarks, για για $n = 1, 2, 4$

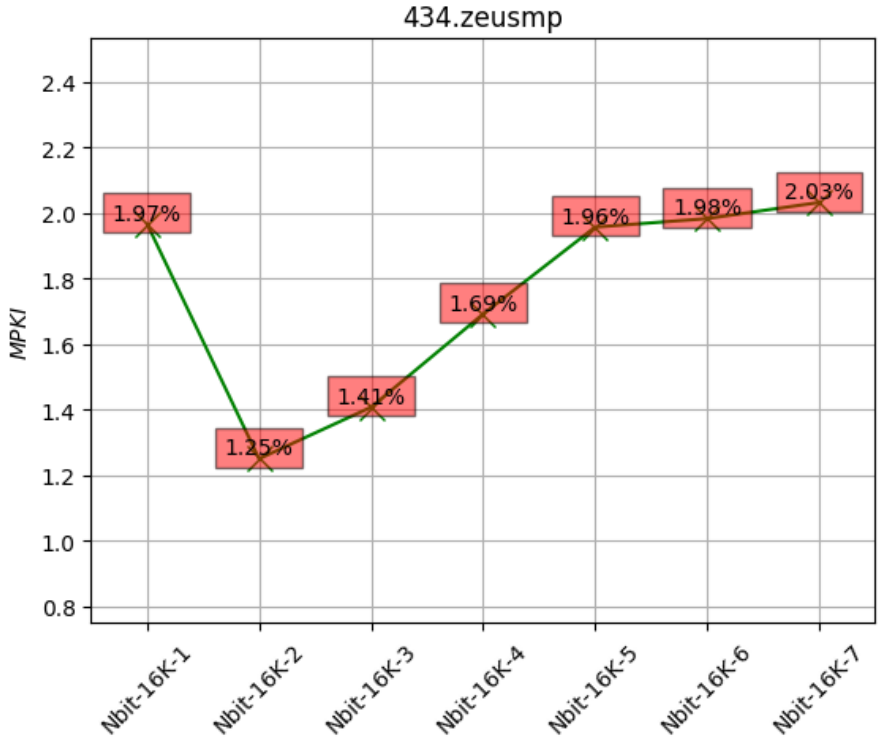
Αποτελέσματα των n-bits predictors για ***BHT entries = 16K, n = {1,...,7 }*** *για 12 benchmarks ΣΥΝΟΛΙΚΑ*

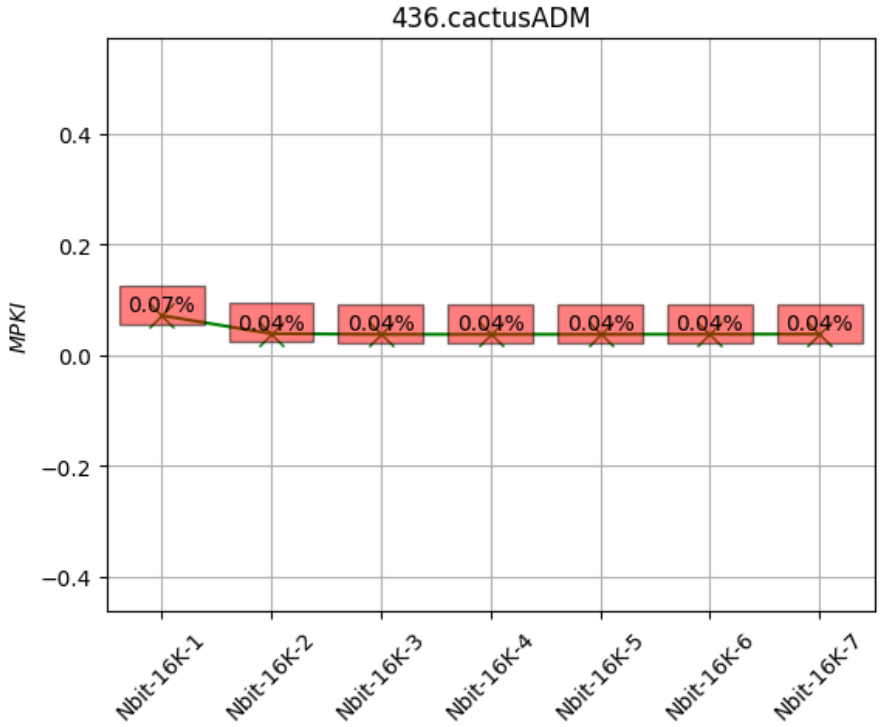


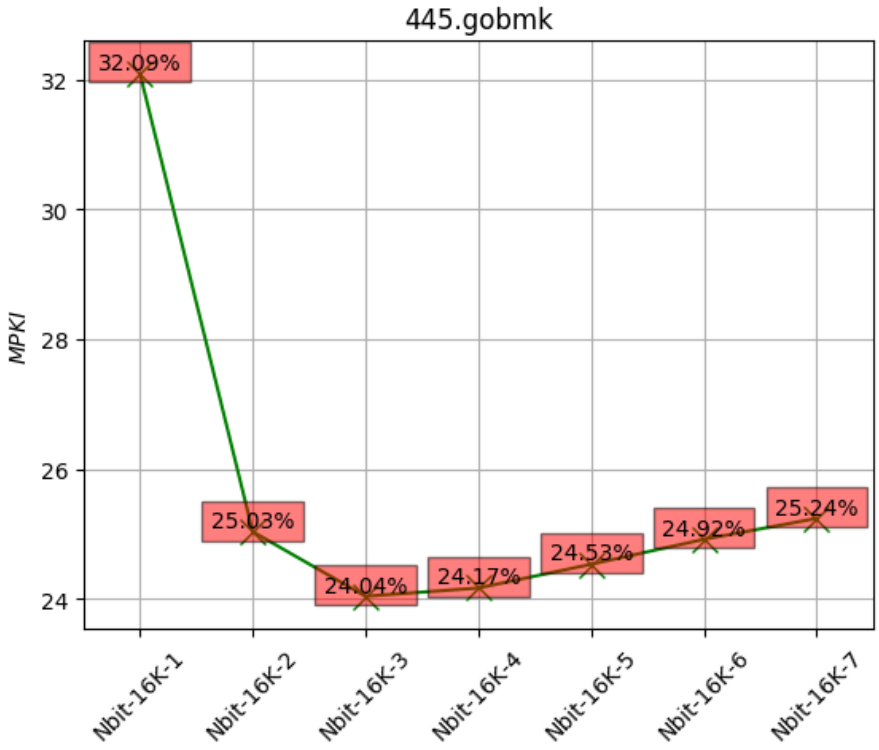
Παρακάτω παραθέτουμε ξεχωριστά για κάθε benchmark το αντίστοιχο διάγραμμα για τους n-bit predictors για $\boldsymbol{BHT\ entries = 16K,\ n = \{1,\ldots,7\}}$ *με μεγαλύτερη λεπτομέρεια, για καλύτερη επισκόπηση και εξαγωγή συμπεσμάτων*

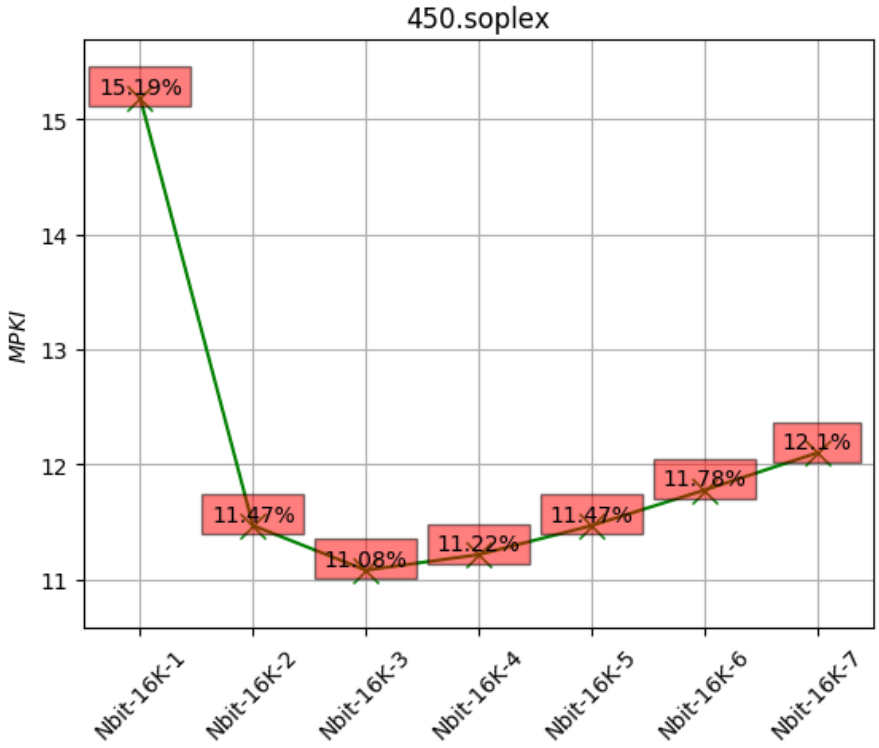

403.gcc
MPKI
22
21
20
19
18
17
16
22.45%
17.06%
16.18%
16.26%
16.6%
17.05%
17.44%
Nbit-16K-1
Nbit-16K-2
Nbit-16K-3
Nbit-16K-4
Nbit-16K-5
Nbit-16K-6
Nbit-16K-7

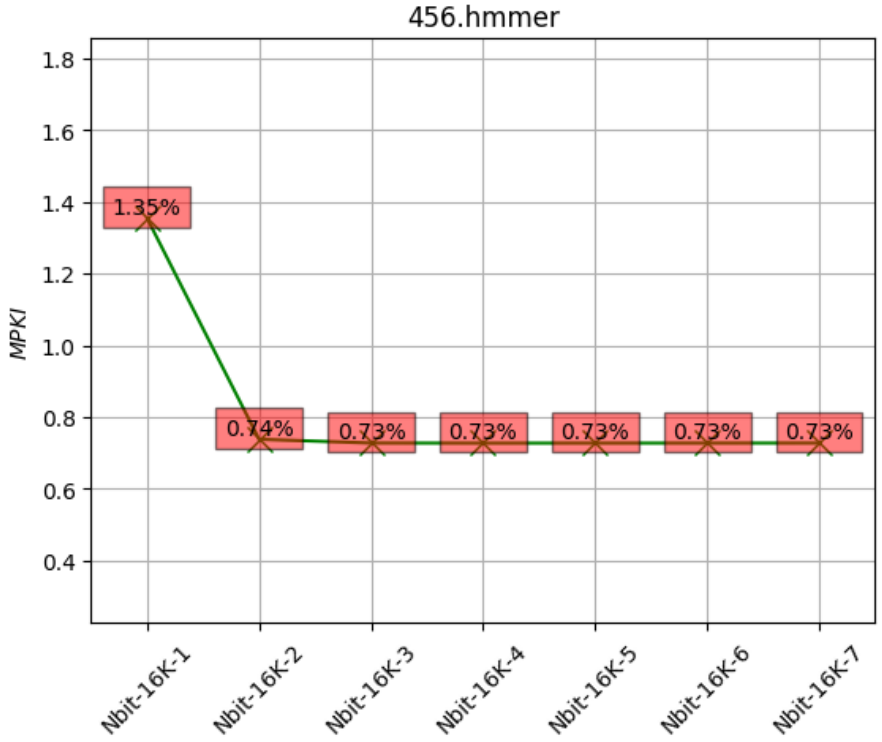

429.mcf
MPKI
19
18
17
16
15
14
13
18.64%
13.25%
12.86%
12.93%
13.14%
13.41%
13.73%
Nbit-16K-1
Nbit-16K-2
Nbit-16K-3
Nbit-16K-4
Nbit-16K-5
Nbit-16K-6
Nbit-16K-7

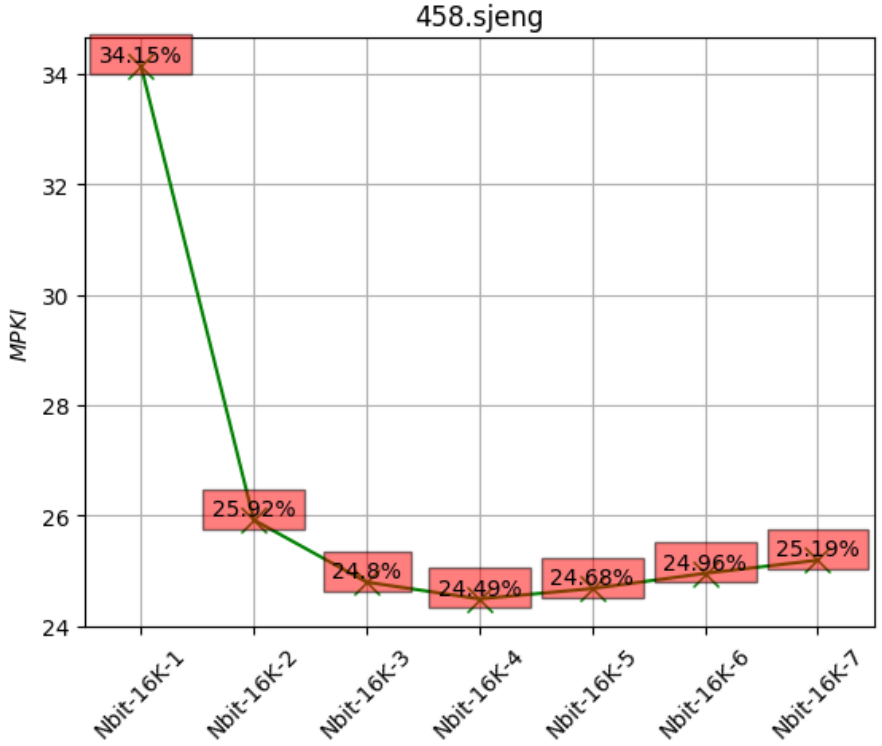

434.zeusmp
MPKI
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
1.97%
1.25%
1.41%
1.69%
1.96%
1.98%
2.03%
Nbit-16K-1
Nbit-16K-2
Nbit-16K-3
Nbit-16K-4
Nbit-16K-5
Nbit-16K-6
Nbit-16K-7

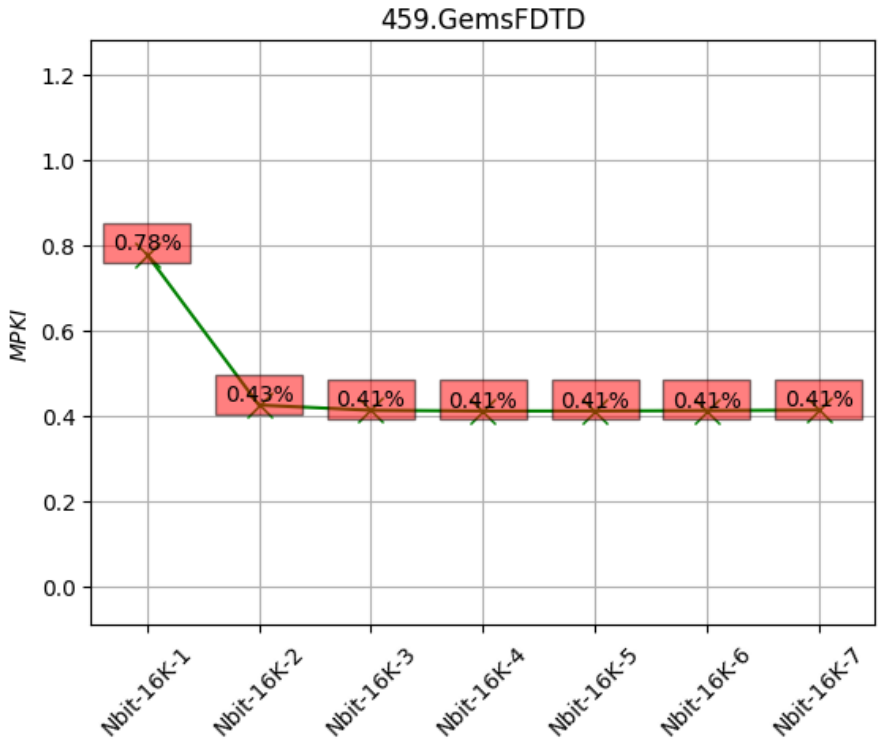

436.cactusADM
MPKI
0.4
0.2
0.0
−0.2
−0.4
0.07%
0.04%
0.04%
0.04%
0.04%
0.04%
0.04%
Nbit-16K-1
Nbit-16K-2
Nbit-16K-3
Nbit-16K-4
Nbit-16K-5
Nbit-16K-6
Nbit-16K-7

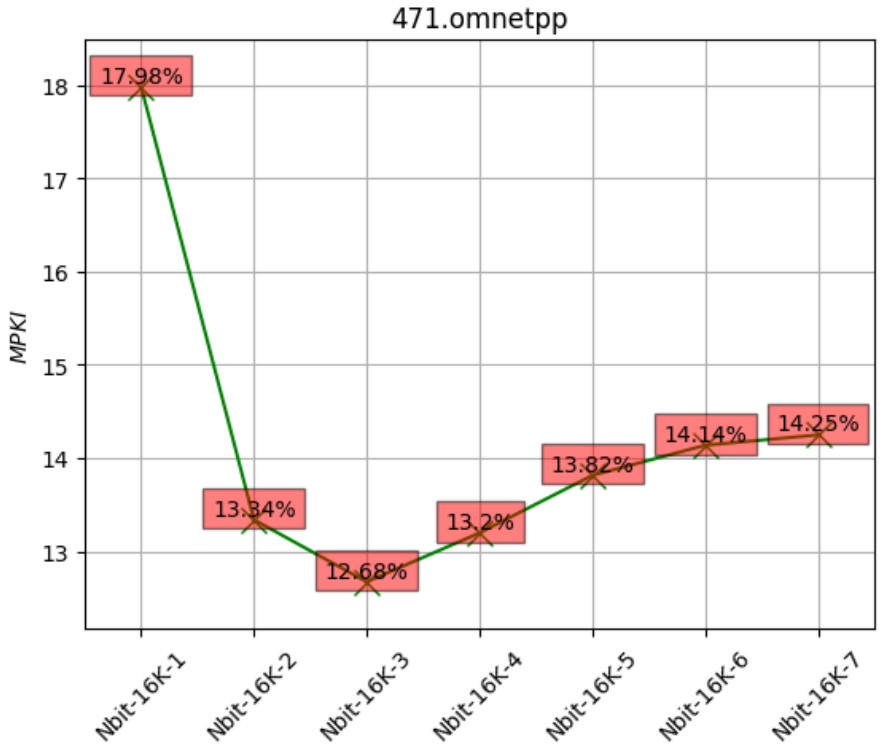

445.gobmk
MPKI
32
30
28
26
24
32.09%
25.03%
24.04%
24.17%
24.53%
24.92%
25.24%
Nbit-16K-1
Nbit-16K-2
Nbit-16K-3
Nbit-16K-4
Nbit-16K-5
Nbit-16K-6
Nbit-16K-7

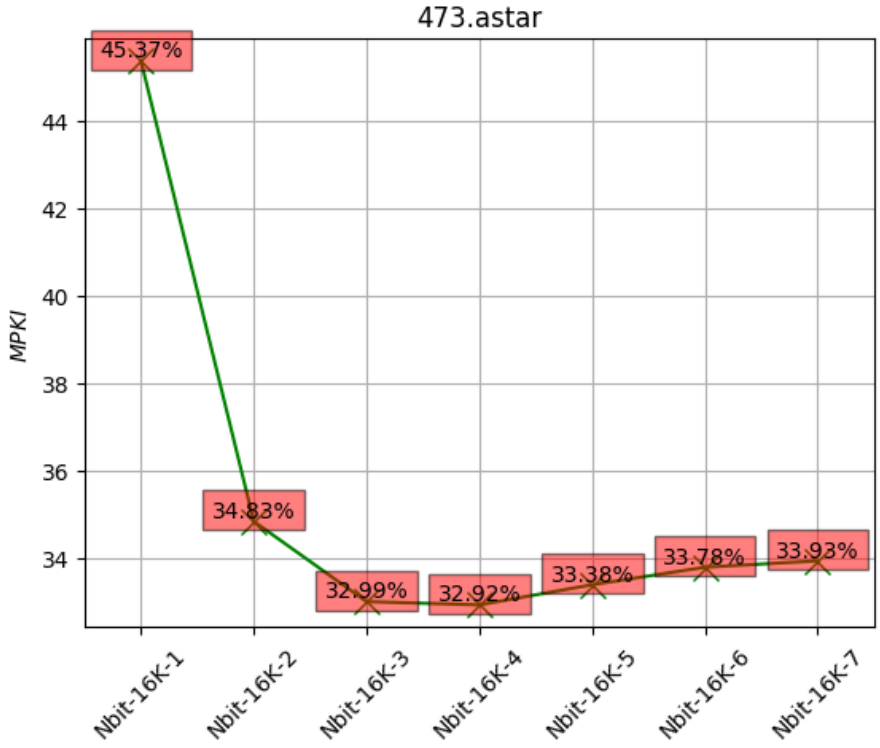

450.soplex
MPKI
15
14
13
12
11
15.19%
11.47%
11.08%
11.22%
11.47%
11.78%
12.1%
Nbit-16K-1
Nbit-16K-2
Nbit-16K-3
Nbit-16K-4
Nbit-16K-5
Nbit-16K-6
Nbit-16K-7

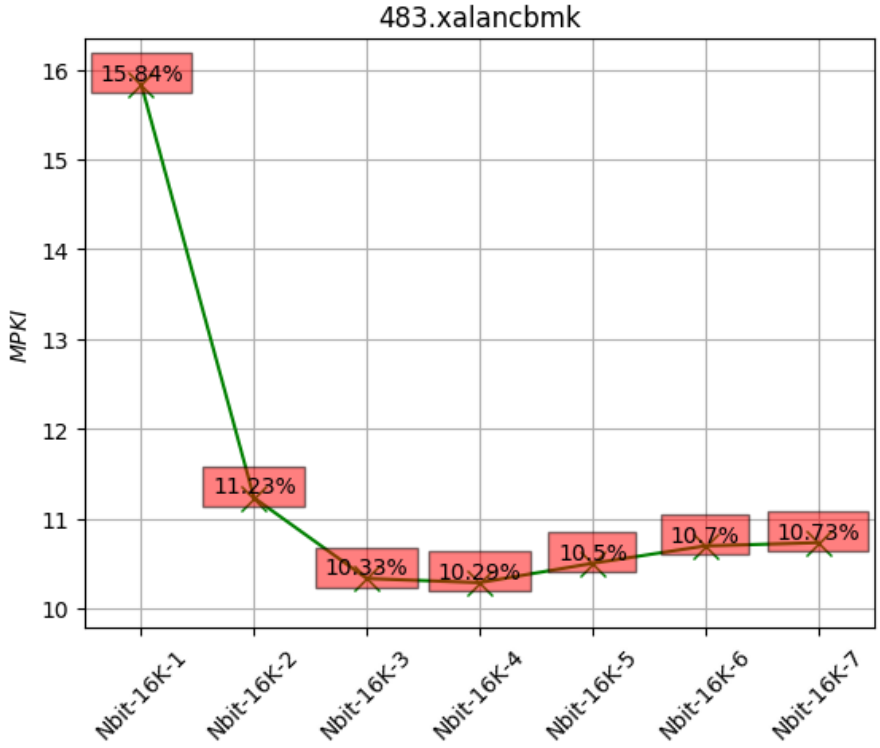

456.hmmer
MPKI
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
1.35%
0.74%
0.73%
0.73%
0.73%
0.73%
0.73%
Nbit-16K-1
Nbit-16K-2
Nbit-16K-3
Nbit-16K-4
Nbit-16K-5
Nbit-16K-6
Nbit-16K-7


458.sjeng
MPKI
34
32
30
28
26
24
34.15%
25.92%
24.8%
24.49%
24.68%
24.96%
25.19%
Nbit-16K-1
Nbit-16K-2
Nbit-16K-3
Nbit-16K-4
Nbit-16K-5
Nbit-16K-6
Nbit-16K-7


459.GemsFDTD
MPKI
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.78%
0.43%
0.41%
0.41%
0.41%
0.41%
0.41%
Nbit-16K-1
Nbit-16K-2
Nbit-16K-3
Nbit-16K-4
Nbit-16K-5
Nbit-16K-6
Nbit-16K-7


471.omnetpp
MPKI
18
17
16
15
14
13
17.98%
13.34%
12.68%
13.2%
13.82%
14.14%
14.25%
Nbit-16K-1
Nbit-16K-2
Nbit-16K-3
Nbit-16K-4
Nbit-16K-5
Nbit-16K-6
Nbit-16K-7





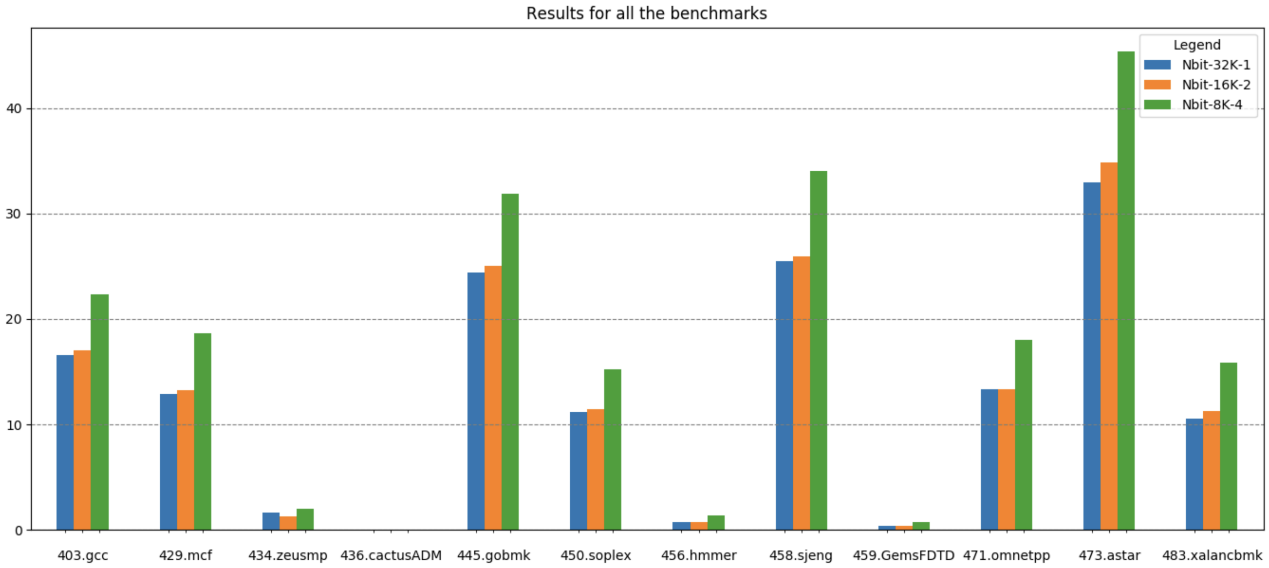
- **Συμπεράσματα**:

  1. Για bit n = 1 bit, παρατηρούμε ότι σχεδόν όλα τα benchmarks έχουν την χειότερη δυνατή απόδοση, καθώς το MPKI αγγίζει μεγαλύτερες τιμές για n = 1 συγκριτικά με οποιοδήποτε άλλο αριθμό bits. Αναλυτικά, η μετάβαση σε n = 2 bit οδηγεί σε **μεγάλη βελτίωση της απόδοσης**, ενώ σε n = 3, ... ,7 bits δεν παρατηρούμε σημαντική βελτίωση αντιθέτως η **απόδοση διατηρείται σταθερή** (δηλαδή και ο δείκτης MPKI) **ή και χειροτερεύει** για τα περισσότερα benchmarks (αρκετά μικρή χειροτέρευση)
  2. Τα χαμηλότερα ποσοστά mispredictions σημειώνονται για τα benchmarks (**Κάτω από 2%**):
     - 434.zeusmp
     - 436.cactusADM

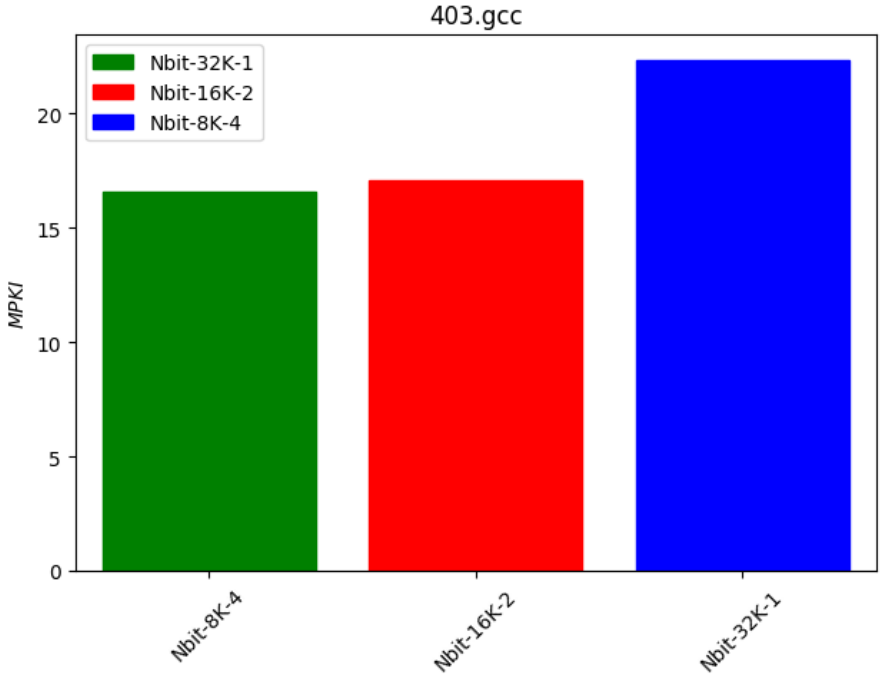
– 456.hmmer
– 459.GemsFDTD

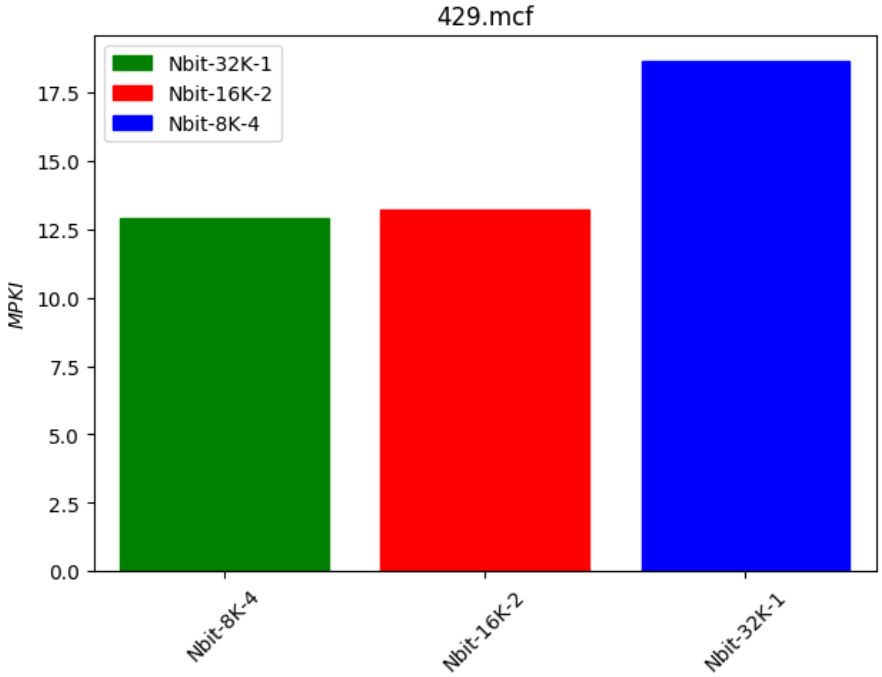
Μάλιστα, η τάξη MPKI που σημειώνουν αυτά τα 4 benchmarks είναι αρκετά μικρότερη συγκριτικά με τα υπόλοιπα 6 benchmarks.

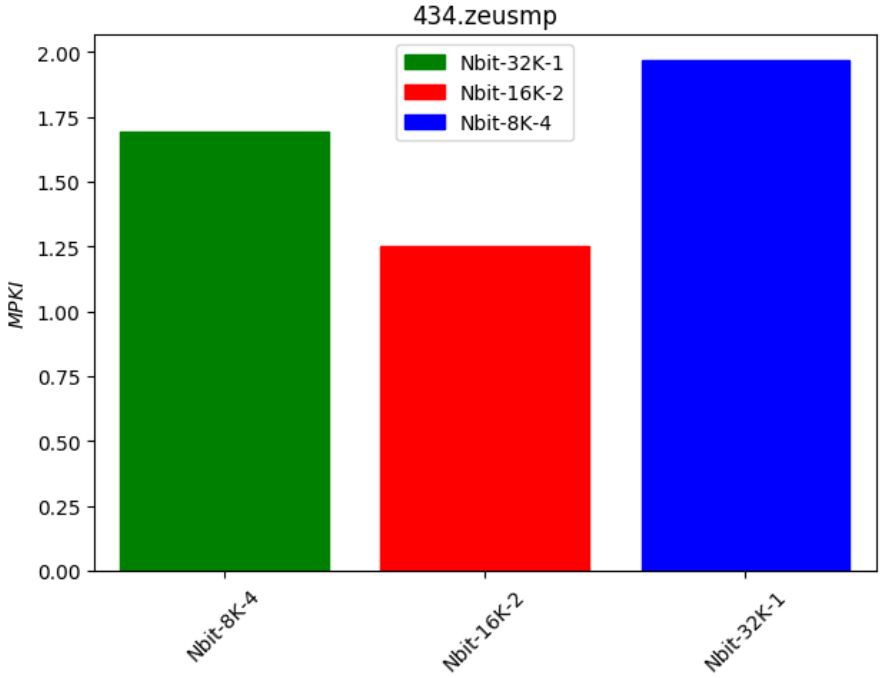
Αποτελέσματα των n-bits predictors για ***Hardware = 32K bits, n = {1,2,4}*** *για 12 benchmarks ΣΥΝΟΛΙΚΑ*

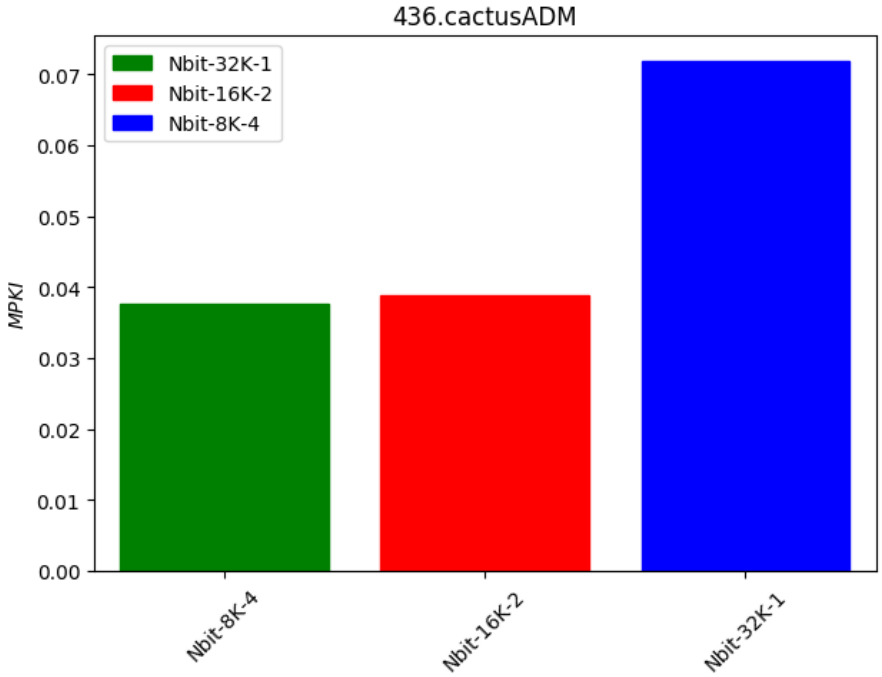


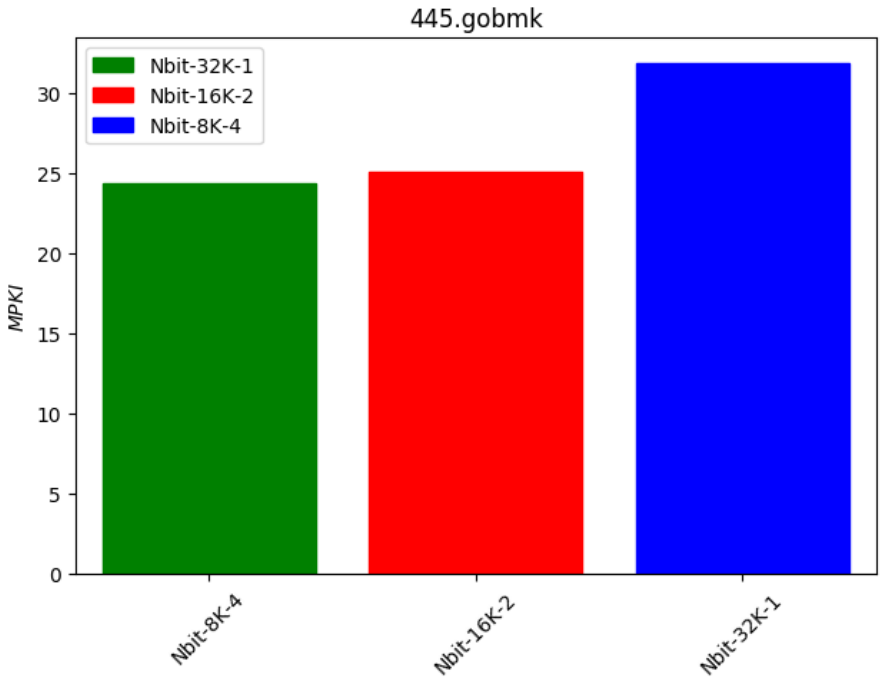
Παρακάτω παραθέτουμε ξεχωριστά για κάθε benchmark το αντίστοιχο διάγραμμα για τους n-bit predictors για ***Hardware = 32K bits, n = {1,2,4}*** *με μεγαλύτερη λεπτομέρεια, για καλύτερη επισκόπηση και εξαγωγή συμπερασμάτων*

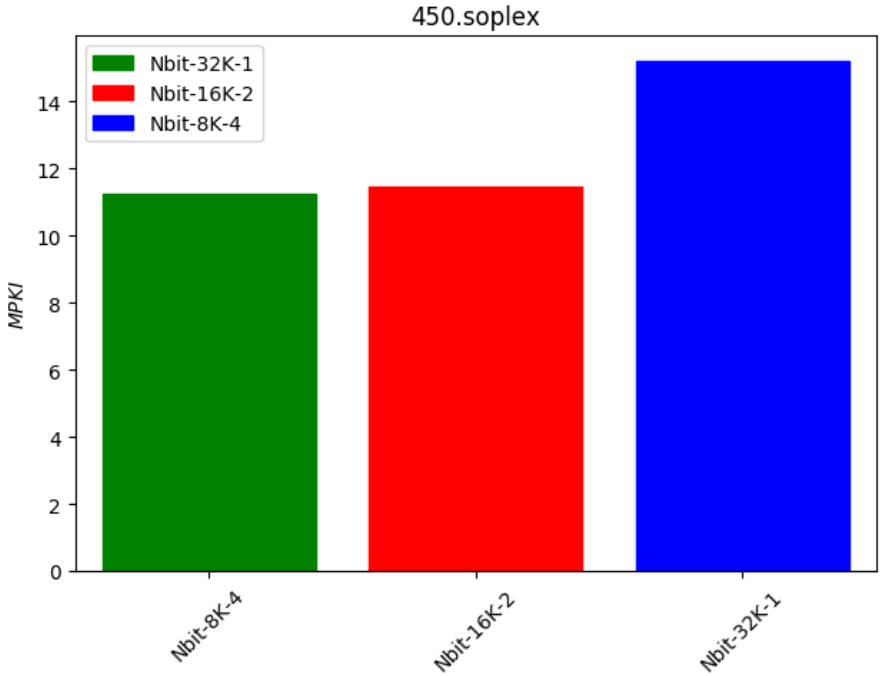


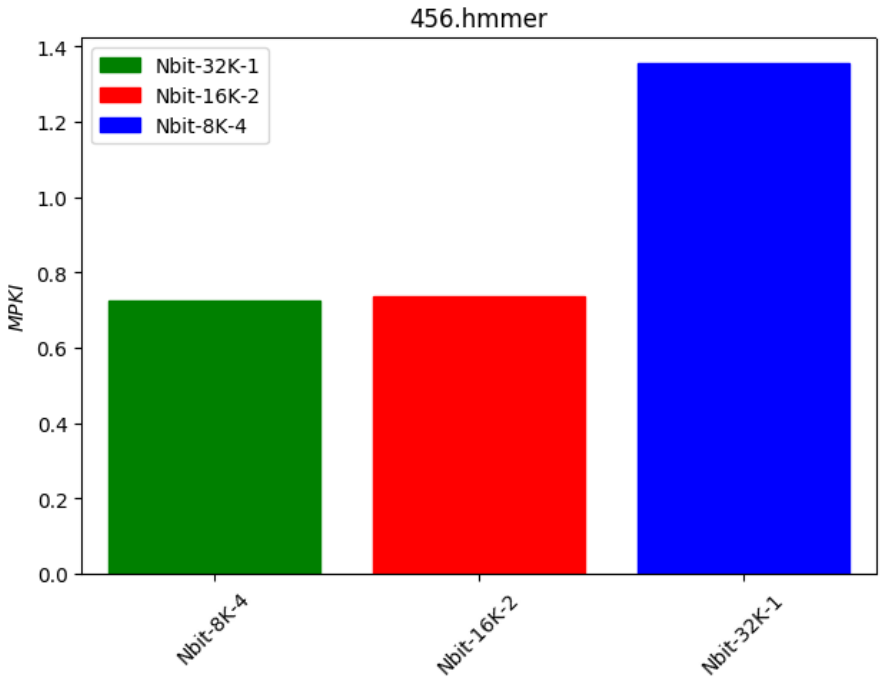

429.mcf
Nbit-32K-1
Nbit-16K-2
Nbit-8K-4
MPKI
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Nbit-8K-4
Nbit-16K-2
Nbit-32K-1

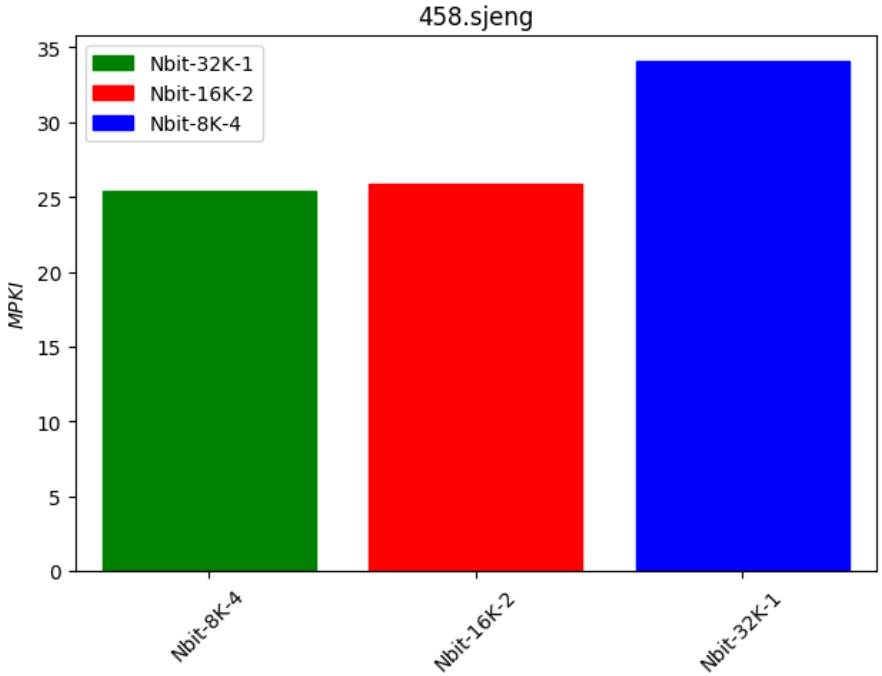

434.zeusmp
Nbit-32K-1
Nbit-16K-2
Nbit-8K-4
MPKI
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Nbit-8K-4
Nbit-16K-2
Nbit-32K-1

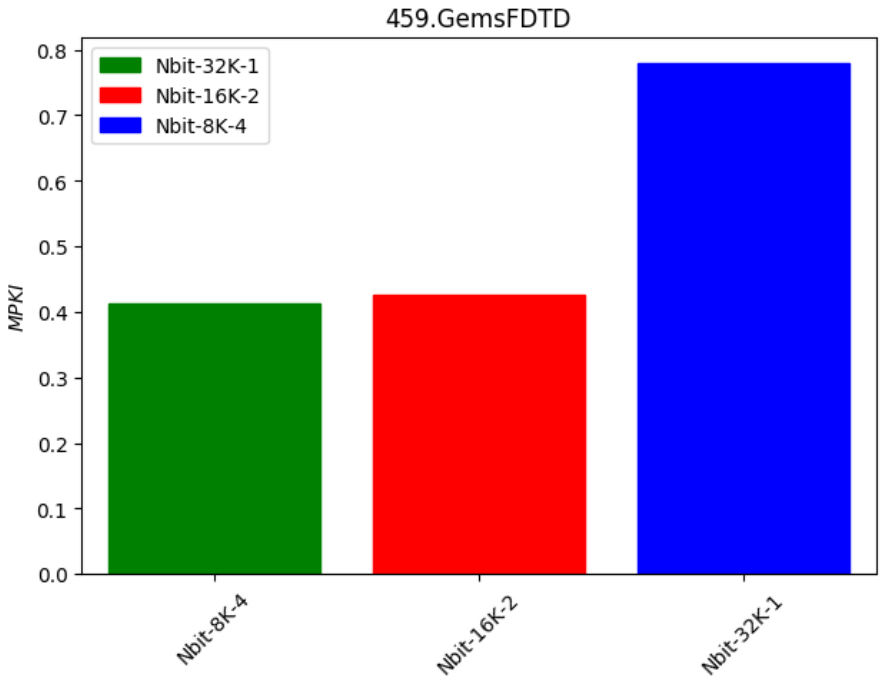

436.cactusADM
Nbit-32K-1
Nbit-16K-2
Nbit-8K-4
MPKI
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Nbit-8K-4
Nbit-16K-2
Nbit-32K-1

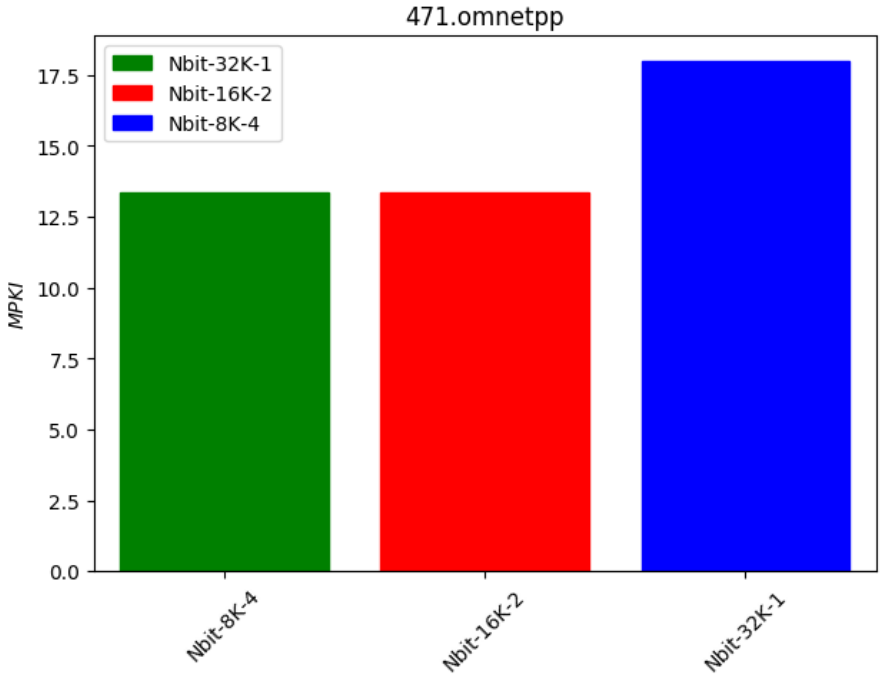

445.gobmk
Nbit-32K-1
Nbit-16K-2
Nbit-8K-4
MPKI
0
5
10
15
20
25
30
Nbit-8K-4
Nbit-16K-2
Nbit-32K-1

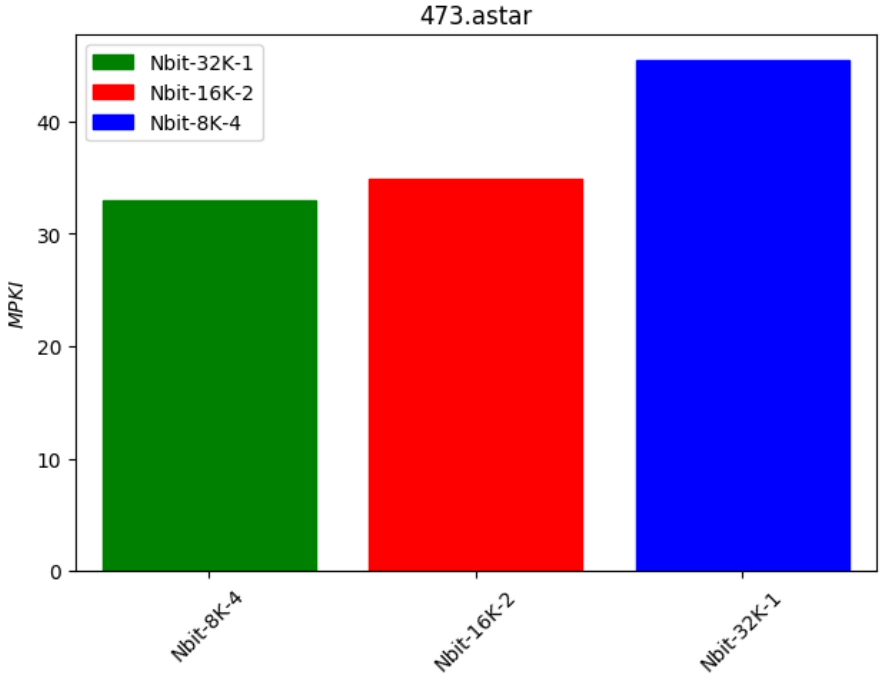

450.soplex
Nbit-32K-1
Nbit-16K-2
Nbit-8K-4
MPKI
0
2
4
6
8
10
12
14
Nbit-8K-4
Nbit-16K-2
Nbit-32K-1

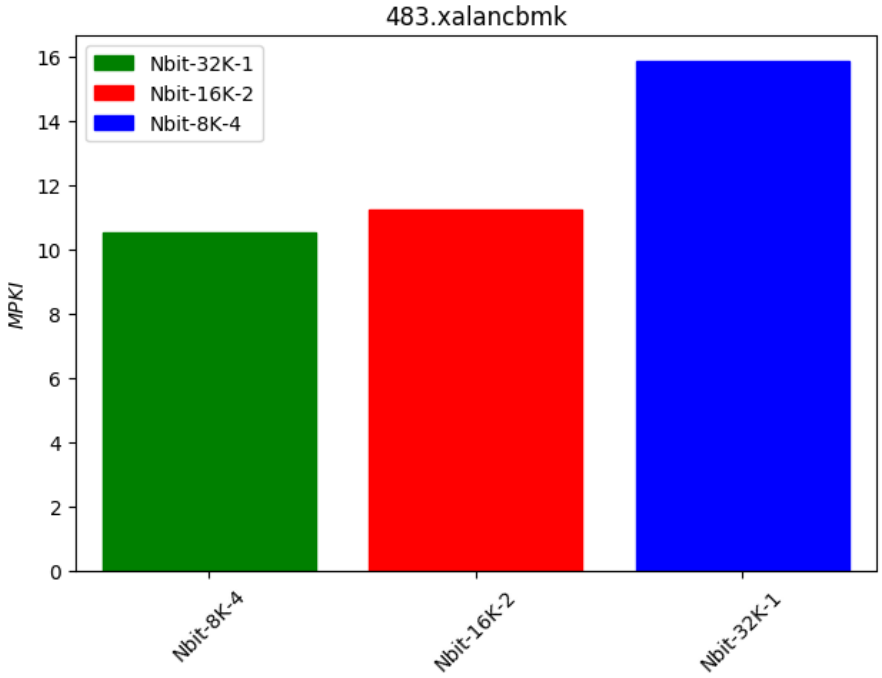

456.hmmer
Nbit-32K-1
Nbit-16K-2
Nbit-8K-4
MPKI
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Nbit-8K-4
Nbit-16K-2
Nbit-32K-1


458.sjeng
Nbit-32K-1
Nbit-16K-2
Nbit-8K-4
MPKI
35
30
25
20
15
10
5
0
Nbit-8K-4
Nbit-16K-2
Nbit-32K-1


459.GemsFDTD
Nbit-32K-1
Nbit-16K-2
Nbit-8K-4
MPKI
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Nbit-8K-4
Nbit-16K-2
Nbit-32K-1


471.omnetpp
Nbit-32K-1
Nbit-16K-2
Nbit-8K-4
MPKI
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Nbit-8K-4
Nbit-16K-2
Nbit-32K-1


473.astar
Nbit-32K-1
Nbit-16K-2
Nbit-8K-4
MPKI
0
10
20
30
40
Nbit-8K-4
Nbit-16K-2
Nbit-32K-1



- **Συμπεράσματα**:

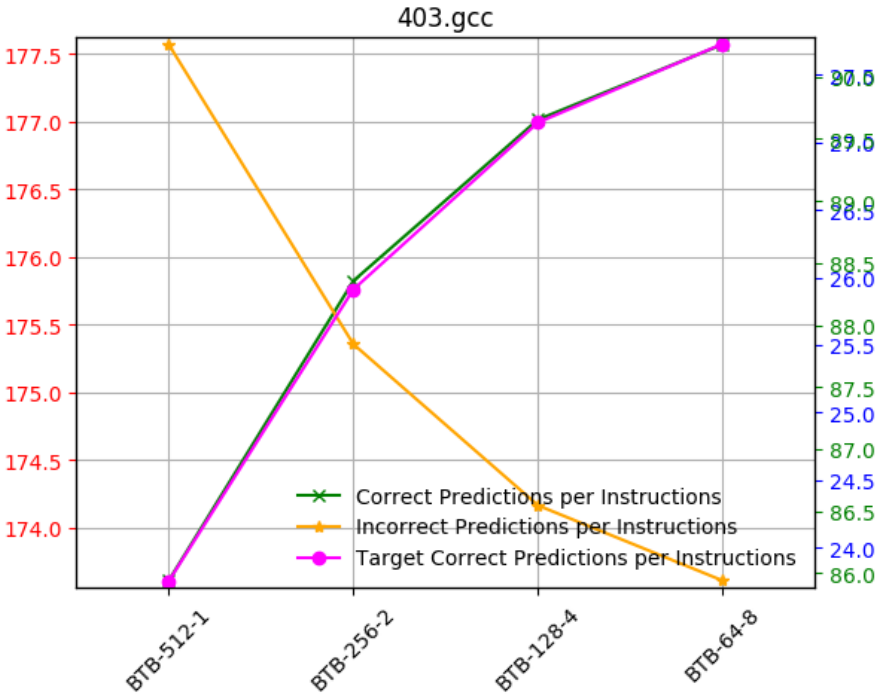
  1. Σημειώνεται πως η αύξηση του bits συνδέεται άρρηκτα και πλήρως με την αύξηση του απαιτούμενου hardware που απαιτεί η εφαρμογή, κάθως ο αριθμός των BHT entries διατηρείται σταθερός.
  2. Όπως και προηγουμένως ο **predictor για n = 1 bit** έχει την χειρότερη απόδοση σε όλα τα benchmarks. Επιπλέον, η απόδοση για **n = 2 & n = 4 bits** κυμαίνεται στα ίδια επίπεδα, αλλά είναι σχετικά καλύτερη από αυτή για **n = 1 bit**.
  3. Βάσει της παραπάνω διερεύνησης και αποτελεσμάτων, επιλέγουμε τον **2-predictor** καθώς κατά μέσο όρο παρουσιάζει σχετικά καλύτερη απόδοση από όλους τους υπόλοιπους! Αξίζει να ανφερθεί πάντως ότι κατά μια γενική ομολογία, στην πραγματικότητα αυτό που χρησιμοποιείται είναι κυρίως ένα κράμα διαφορετικών τύπων predictors, όπως πχ υβριδικό από 2-bit counter και global predictor, το οποίο **βελτιώνει την απόδοση, αλλά οδηγεί σε αυξημένες καθυστερήσεις** (tradeoff)

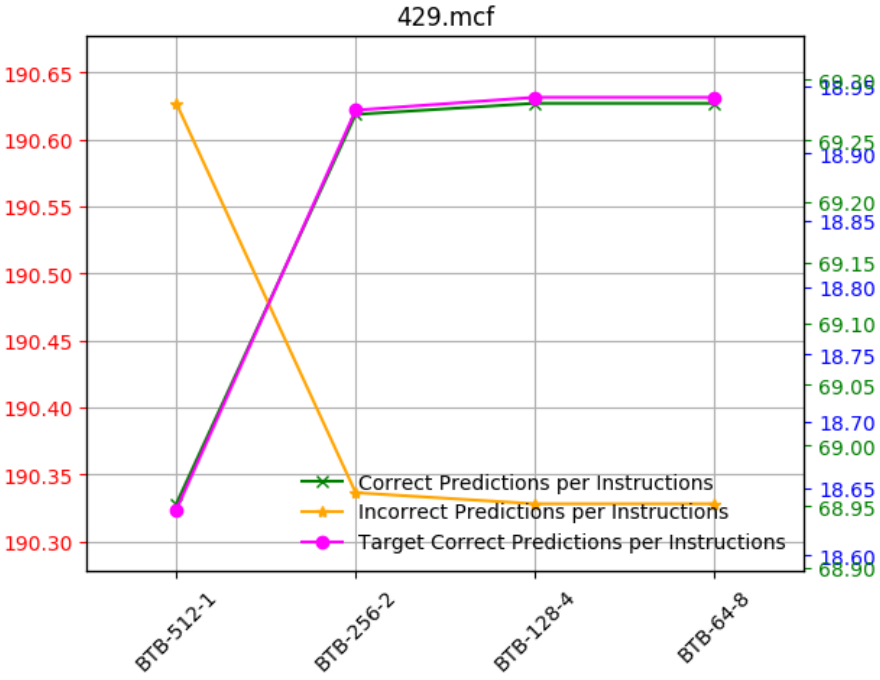
3.3 Μελέτη του BTB

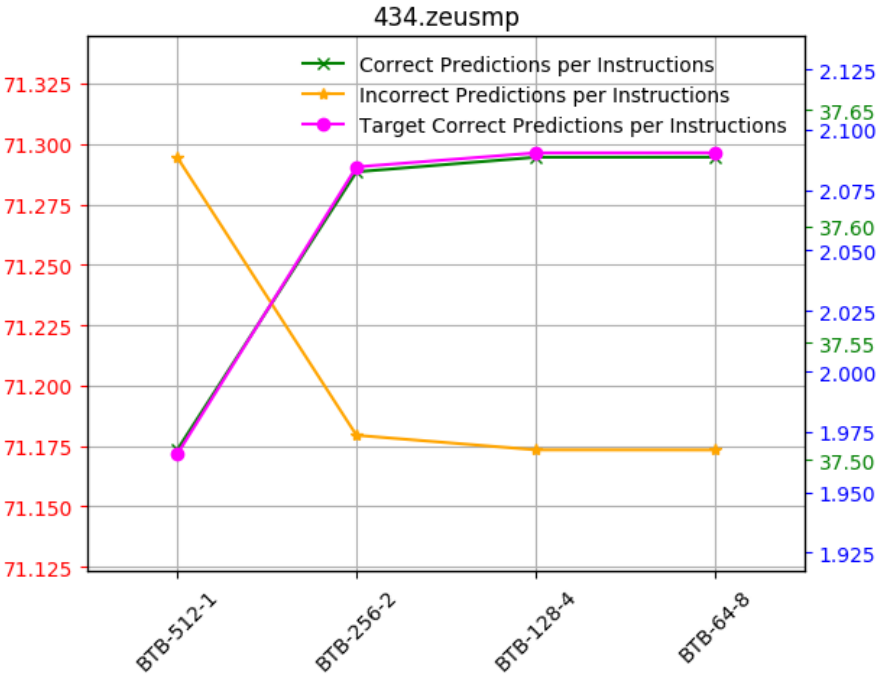
- Στο **Τρίτο Μέρος** της πειραματικής αξιολόγησης για την παρούσα άσκηση, εξετάζουμε την **ακρίβεια πρόβλεψης** κάνοντας χρήση ενός **Branch Target Buffer** (BTB). Συγκεκρίμενα, εκτελούμε προσομοιώσεις για τις παρακάτω περιπτώσεις:

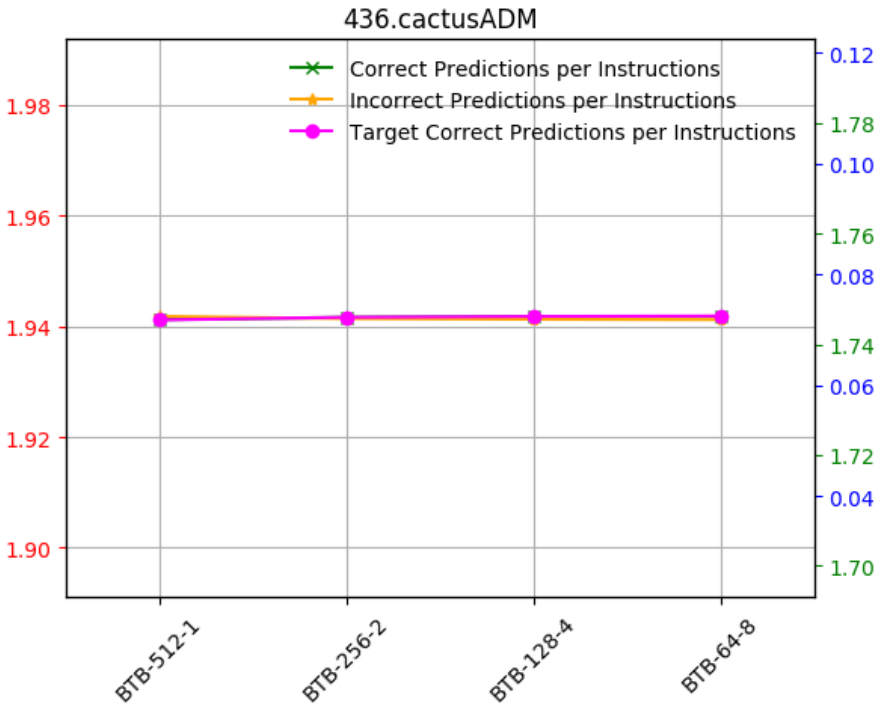
BTB entries	BTB associativity
512	1
256	2
128	4
64	8

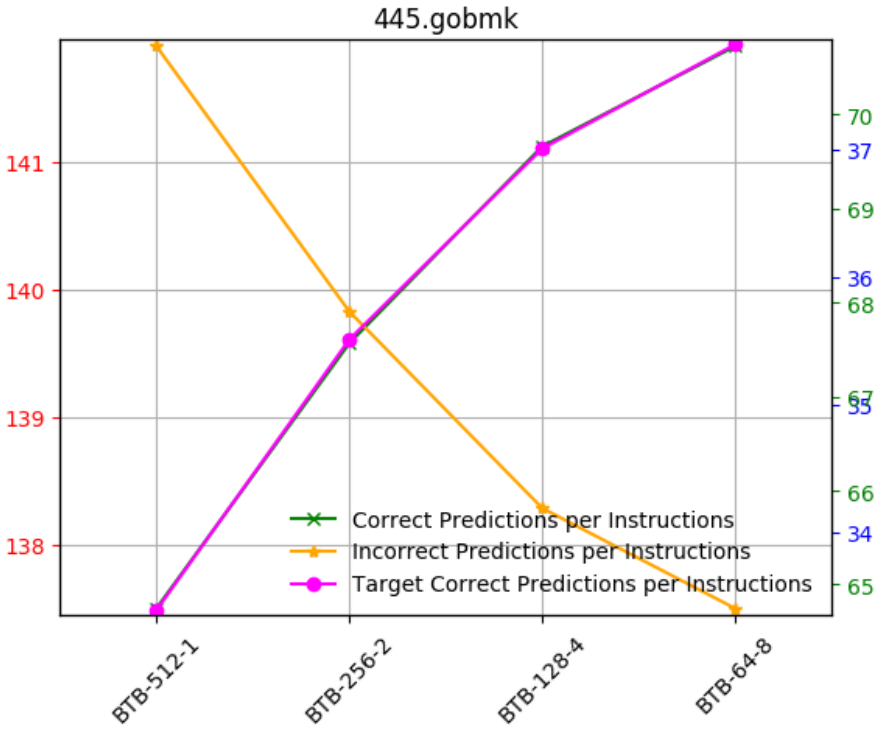
*Παρακάτω παραθέτουμε ξεχωριστά για κάθε benchmark το αντίστοιχο διάγραμμα για **[BTB entries, BTB associativity] =[512-1,256-2,128-4,64-8]** με μεγαλύτερη λεπτομέρεια, για καλύτερη επισκόπηση και εξαγωγή συμπεσμάτων*

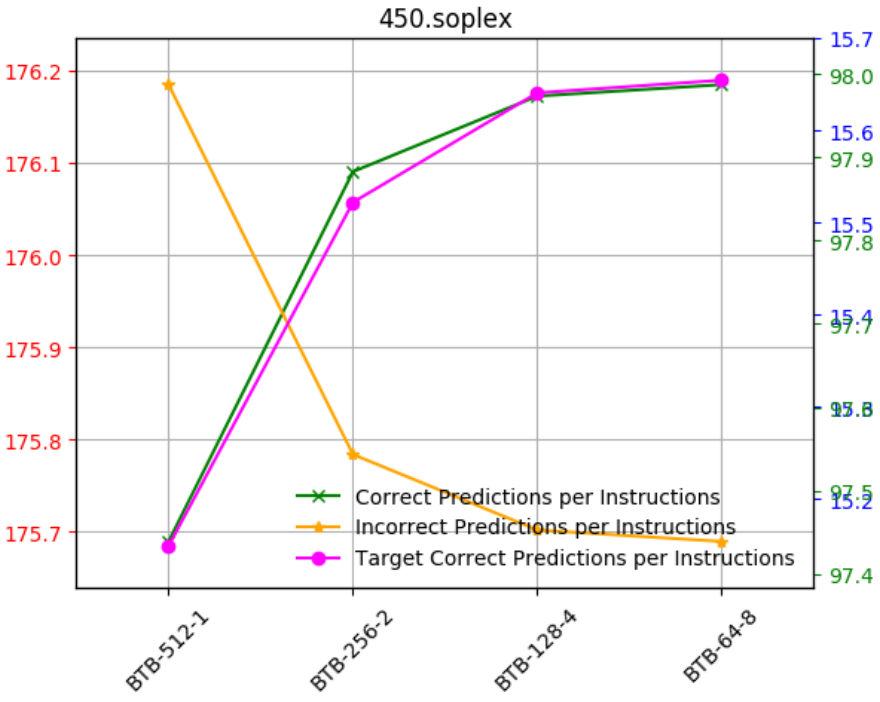

403.gcc
177.5
177.0
176.5
176.0
175.5
175.0
174.5
174.0
89.0
88.5
88.0
87.5
87.0
86.5
86.0
26.5
26.0
25.5
25.0
24.5
24.0
Correct Predictions per Instructions
Incorrect Predictions per Instructions
Target Correct Predictions per Instructions
BTB-512-1
BTB-256-2
BTB-128-4
BTB-64-8

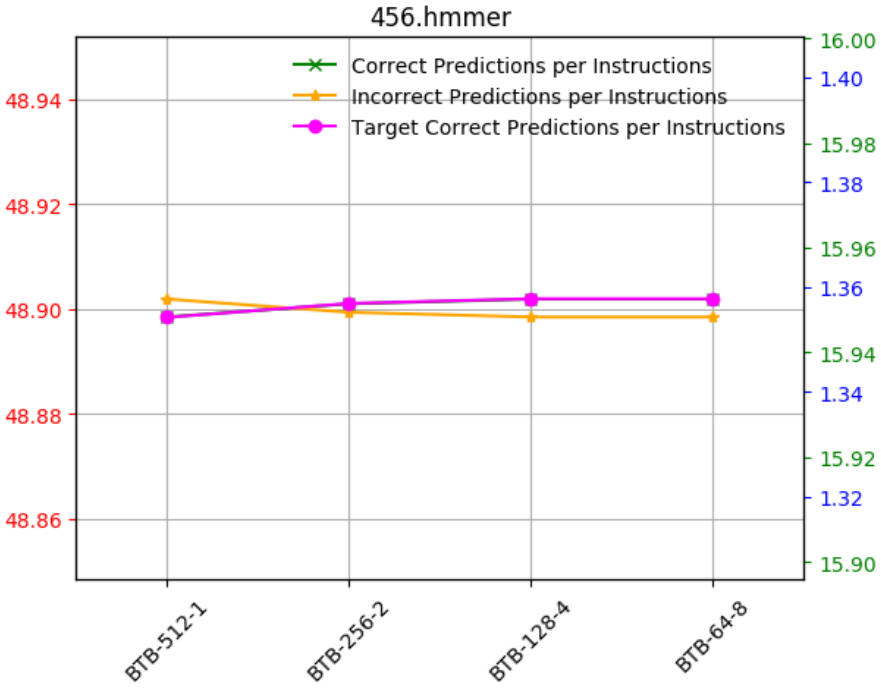

429.mcf
190.65
190.60
190.55
190.50
190.45
190.40
190.35
190.30
69.25
69.20
69.15
69.10
69.05
69.00
68.95
68.90
18.90
18.85
18.80
18.75
18.70
18.65
18.60
Correct Predictions per Instructions
Incorrect Predictions per Instructions
Target Correct Predictions per Instructions
BTB-512-1
BTB-256-2
BTB-128-4
BTB-64-8

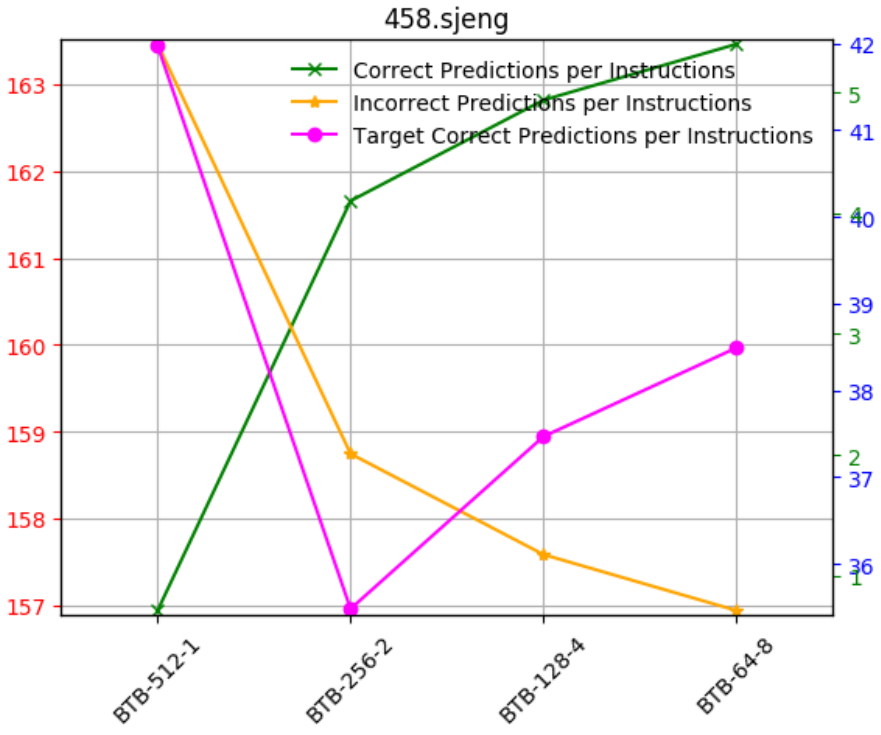

434.zeusmp
Correct Predictions per Instructions
Incorrect Predictions per Instructions
Target Correct Predictions per Instructions
71.325
71.300
71.275
71.250
71.225
71.200
71.175
71.150
71.125
2.125
37.65
2.100
2.075
37.60
2.050
2.025
37.55
2.000
1.975
37.50
1.950
1.925
BTB-512-1
BTB-256-2
BTB-128-4
BTB-64-8

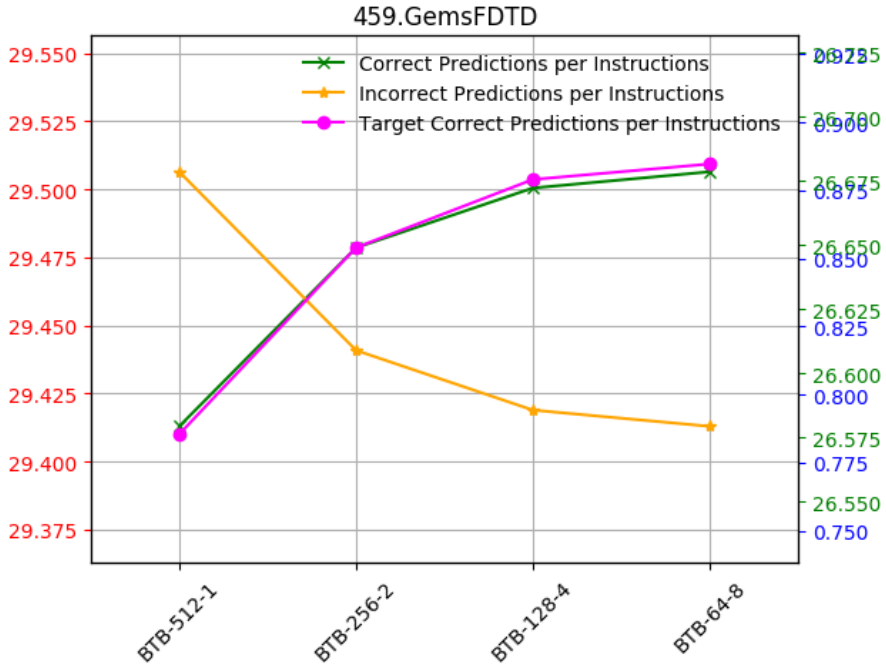

436.cactusADM
Correct Predictions per Instructions
Incorrect Predictions per Instructions
Target Correct Predictions per Instructions
1.98
1.96
1.94
1.92
1.90
0.12
1.78
0.10
1.76
0.08
1.74
0.06
1.72
0.04
1.70
BTB-512-1
BTB-256-2
BTB-128-4
BTB-64-8

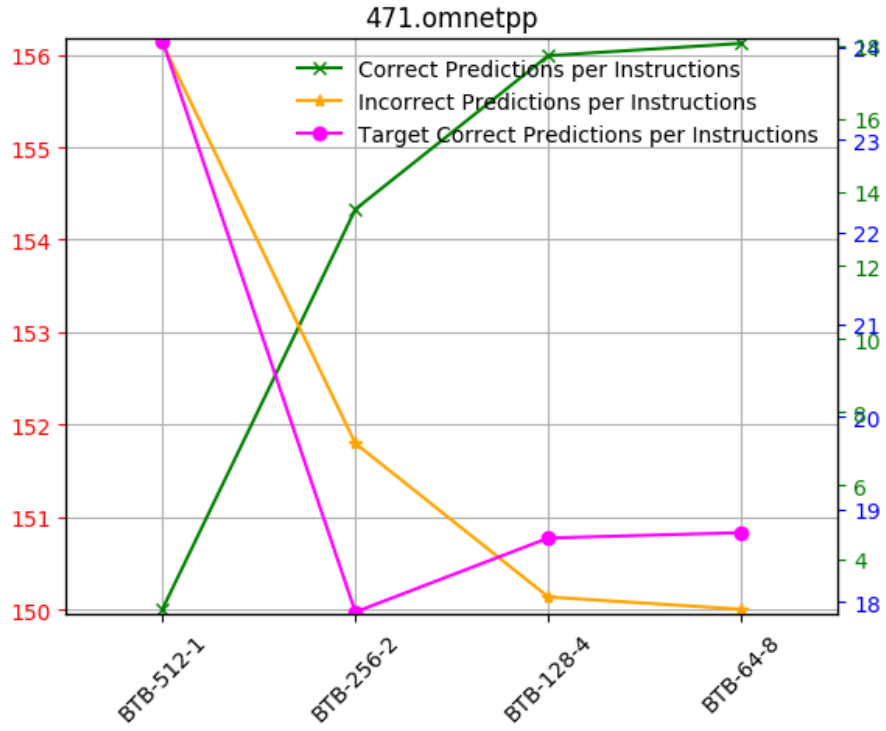

445.gobmk
141
140
139
138
70
37
69
36
68
67
35
66
34
65
Correct Predictions per Instructions
Incorrect Predictions per Instructions
Target Correct Predictions per Instructions
BTB-512-1
BTB-256-2
BTB-128-4
BTB-64-8

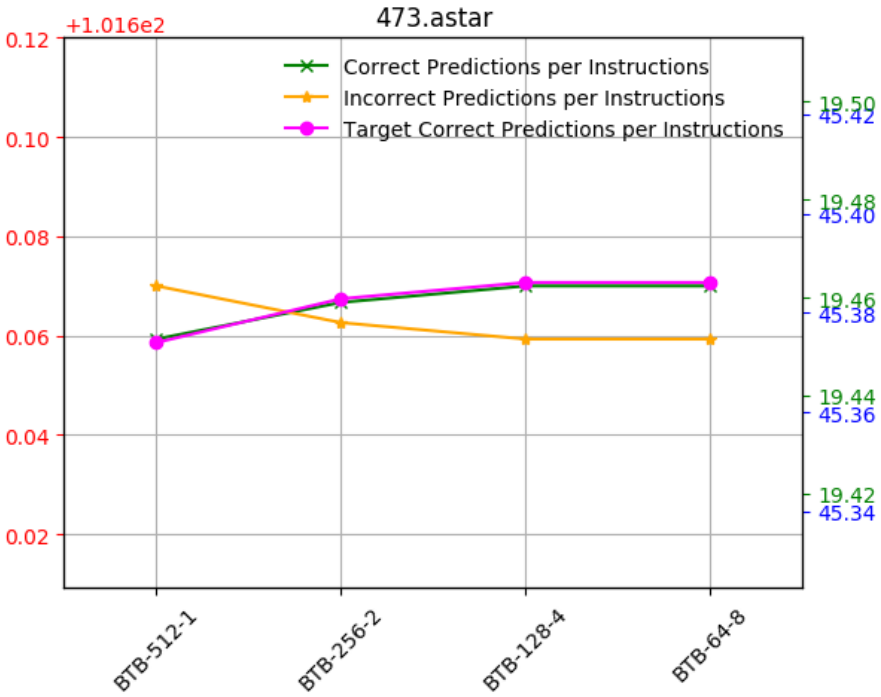

450.soplex
176.2
176.1
176.0
175.9
175.8
175.7
15.7
98.0
15.6
97.9
15.5
97.8
15.4
97.7
97.5
15.2
97.4
Correct Predictions per Instructions
Incorrect Predictions per Instructions
Target Correct Predictions per Instructions
BTB-512-1
BTB-256-2
BTB-128-4
BTB-64-8

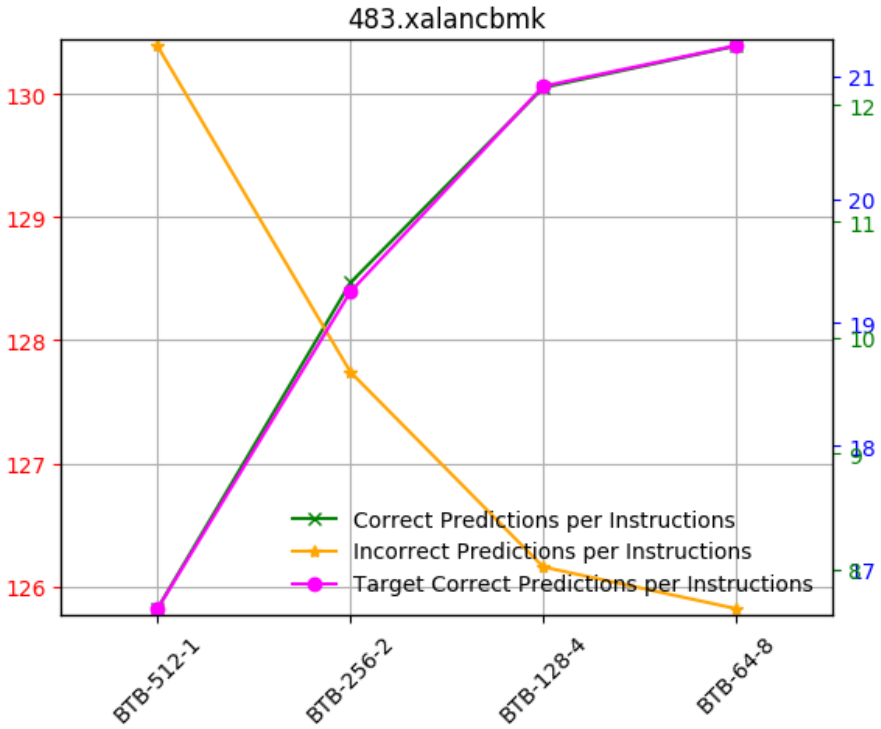

456.hmmer
Correct Predictions per Instructions
Incorrect Predictions per Instructions
Target Correct Predictions per Instructions
48.94
48.92
48.90
48.88
48.86
16.00
1.40
15.98
1.38
15.96
1.36
15.94
1.34
15.92
1.32
15.90
BTB-512-1
BTB-256-2
BTB-128-4
BTB-64-8


458.sjeng
Correct Predictions per Instructions
Incorrect Predictions per Instructions
Target Correct Predictions per Instructions
163
162
161
160
159
158
157
42
5
41
4
40
39
3
38
2
37
36
1
BTB-512-1
BTB-256-2
BTB-128-4
BTB-64-8


459.GemsFDTD
Correct Predictions per Instructions
Incorrect Predictions per Instructions
Target Correct Predictions per Instructions
29.550
29.525
29.500
29.475
29.450
29.425
29.400
29.375
26.725
26.700
26.675
26.650
26.625
26.600
26.575
26.550
0.900
0.875
0.850
0.825
0.800
0.775
0.750
BTB-512-1
BTB-256-2
BTB-128-4
BTB-64-8


471.omnetpp
Correct Predictions per Instructions
Incorrect Predictions per Instructions
Target Correct Predictions per Instructions
156
155
154
153
152
151
150
24
16
23
14
22
12
21
10
20
6
19
4
18
BTB-512-1
BTB-256-2
BTB-128-4
BTB-64-8





- **Συμπεράσματα**:

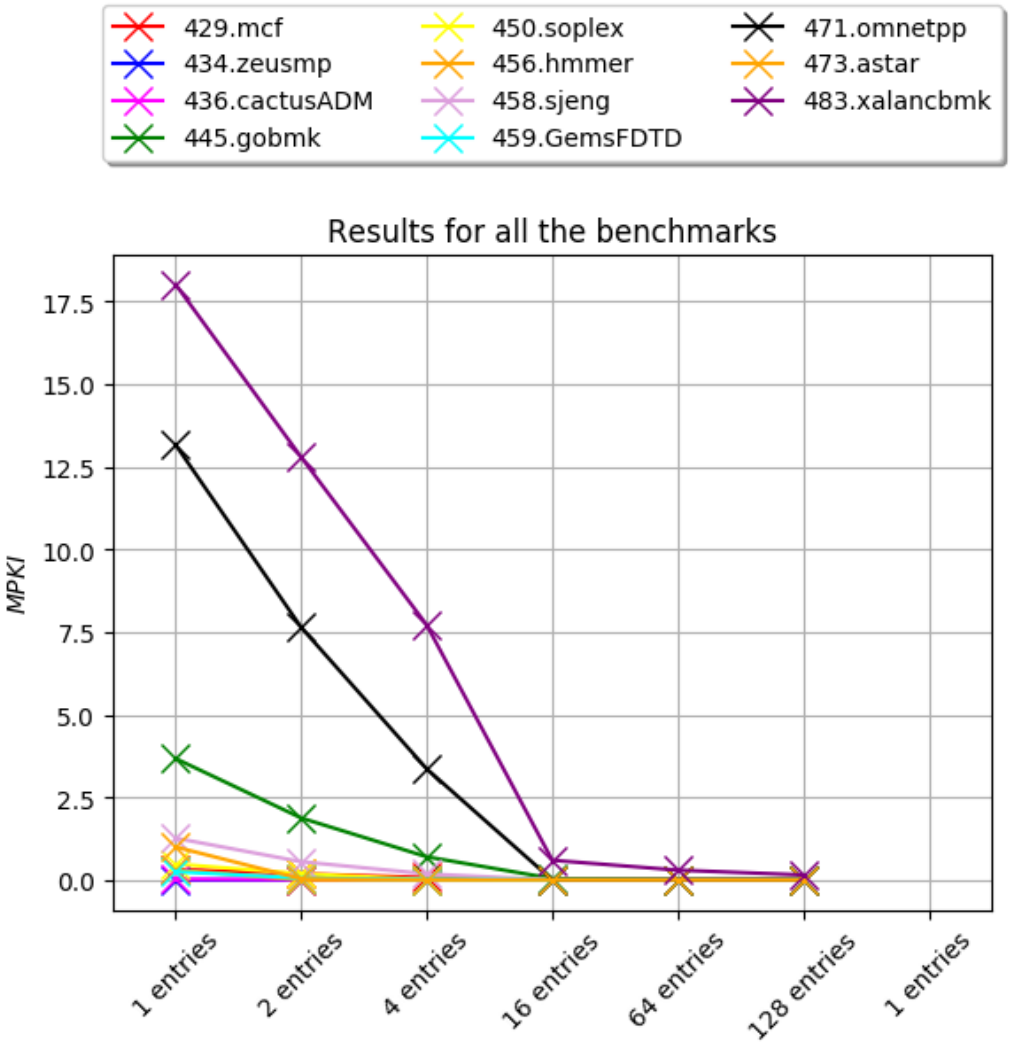
  1. Αρχικά να σημειωθεί πως τα BTB entries έχουν **μεγαλύτερο υπολογιστικό κόστος** από BHT entries , αλλά κάνουν redirect fetches σε πολύ νωρίτερο στάδιο στο pipeline και μπορούν να **επιταχύνουν τα indirect branches**. Δεν ξεχνάμε όμως ότι τα BHT entries διατηρούν περισσότερα entries και έχουν μεγαλύτερη ακρίβεια.
  2. Παρ'όλα αυτά η συγκεκρίμενη μέθοδος θεωρείται αρκετά **αποτελεσματική**. Προφανώς βέβαια υπάρχουν σφάλματα λανθασμένου στόχου διακλάδωσης, τα οποία οφείλονται στην μη αποθήκευση τους στον πίνακα παρά την σωστή απόφαση για την πορεία της διακλάδωσης.
  3. **Οι predictors φτάνουν παρόμοια επίπεδα απόδοσης κατά μέσο όρο για οποιονδήποτε απο τους 4 παραπάνω συνδυασμούς entries και associativity.**
  4. Τέλος, αξίζει να σημειωθεί ο *τρόπος αντικατάστασης για υψηλό associativity*: Η κάθε εντολή θα πρέπε να αντικατασταθεί αυτόματα από κάποια άλλη στο BTB πίνακα με **ίδια διεύθυνση**. Αντιθέτως, απαιτούνται τόσες εντολές branch (με ίδια θέση στον πίνακα BTB) = associativity

3.4 Μελέτη του RAS

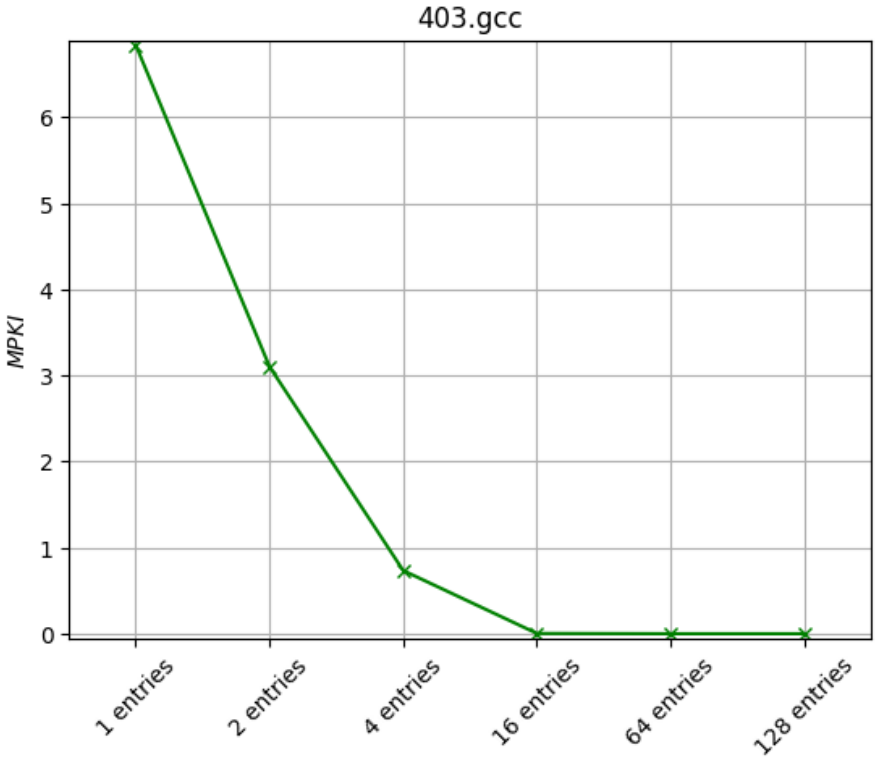
- Στο **Τέταρτο Μέρος** της πειραματικής αξιολόγησης για την παρούσα άσκηση, εξετάζουμε το ποσοστό αστοχίας που προκύπτει κάνοντας χρήση διαφορετικού αριθμού εγγράφων RAS.
- Για την υλοποίηση της RAS, η οποία είναι πολύ απλή και αποτελεί μέθοδο για την **πρόβλεψη εντολών return**, χρησιμοποιούμε στο **"ras.h"** (με μικρές προσαρμογές) μία στοίβα (FILO) η οποία λειτουργεί ως εξής: *Σε κάθε κλήση προσθέτουμε διευθύνσεις και σε κάθε return αφαιρούμε.*
- Συγκεκριμένα, εκτελούμε προσομοιώσεις για τις παρακάτω περιπτώσεις:

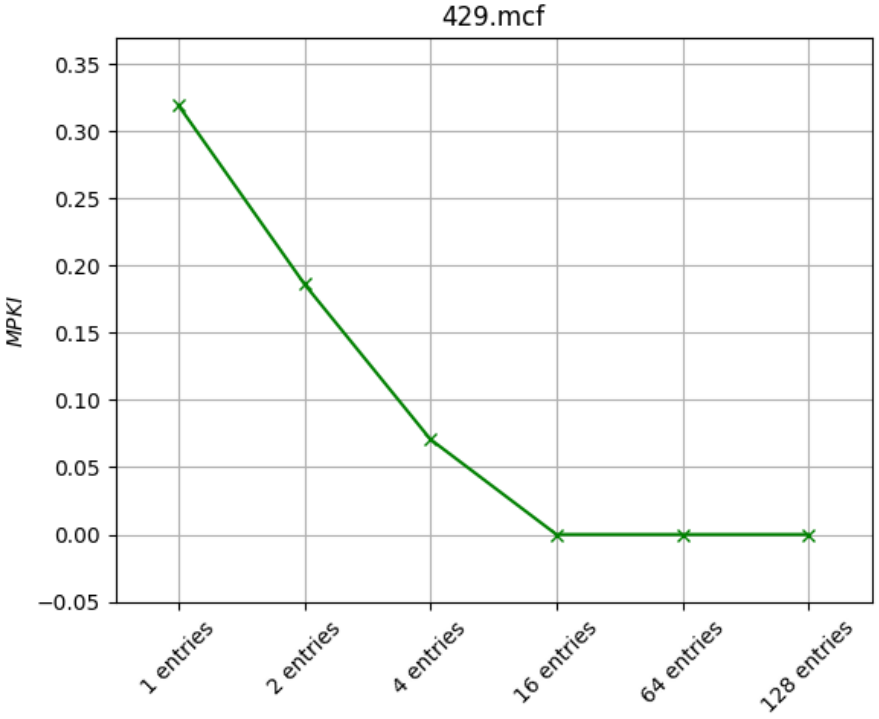
Εγγραφές στην RAS
1
2
4
8
16
64
128

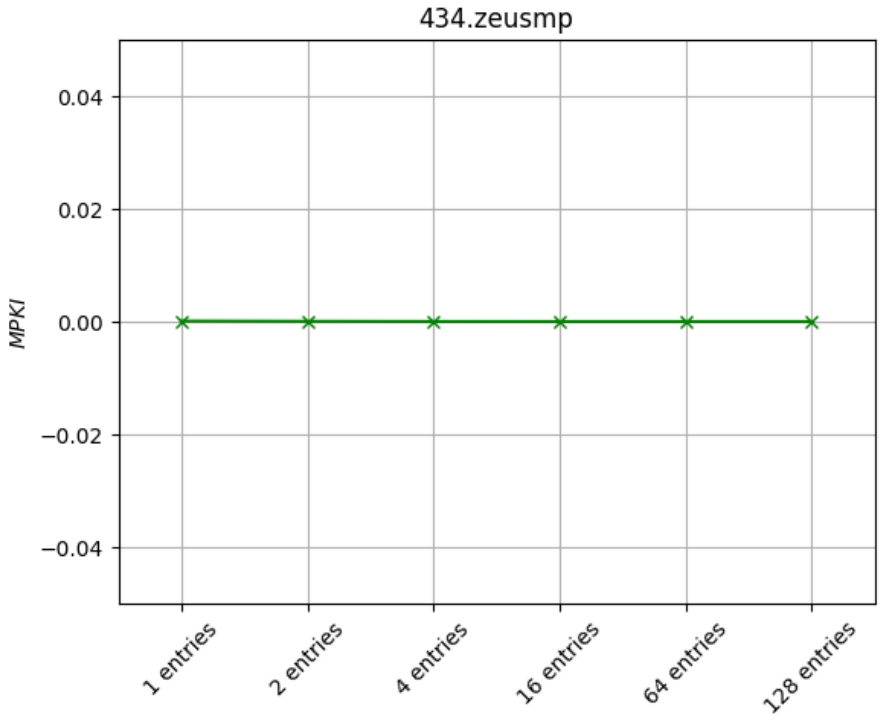
Αποτελέσματα των n-bits predictors για ***#RAS = {1,2,4,8,16,64,128 }*** *για 12 benchmarks ΣΥΝΟΛΙΚΑ*

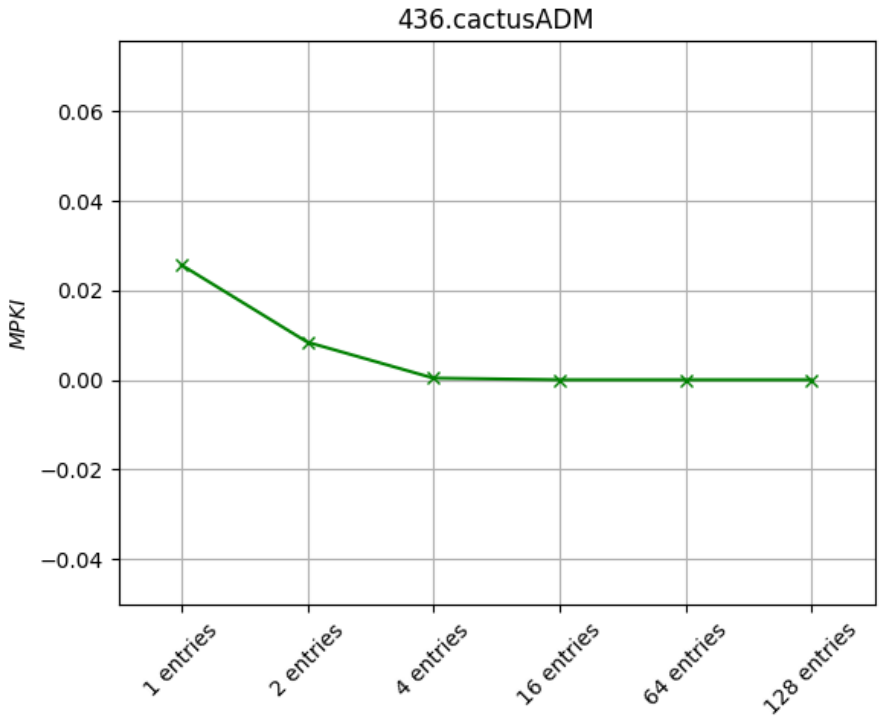


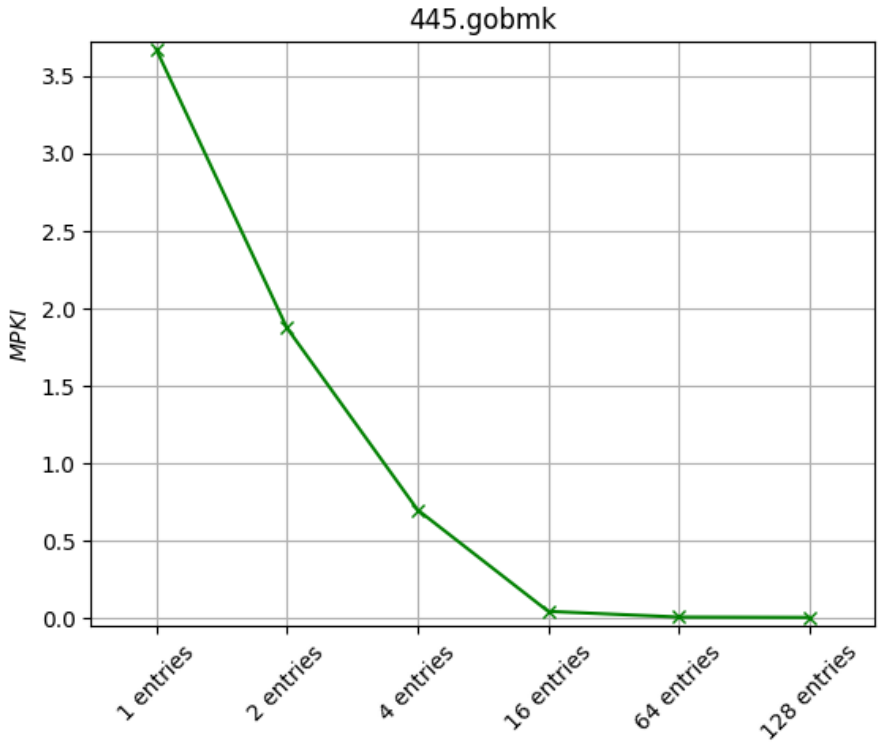
Παρακάτω παραθέτουμε ξεχωριστά για κάθε benchmark το αντίστοιχο διάγραμμα για ***#RAS = {1,2,4,8,16,64,128 }*** *με μεγαλύτερη λεπτομέρεια, για καλύτερη επισκόπηση και εξαγωγή συμπεσμάτων*

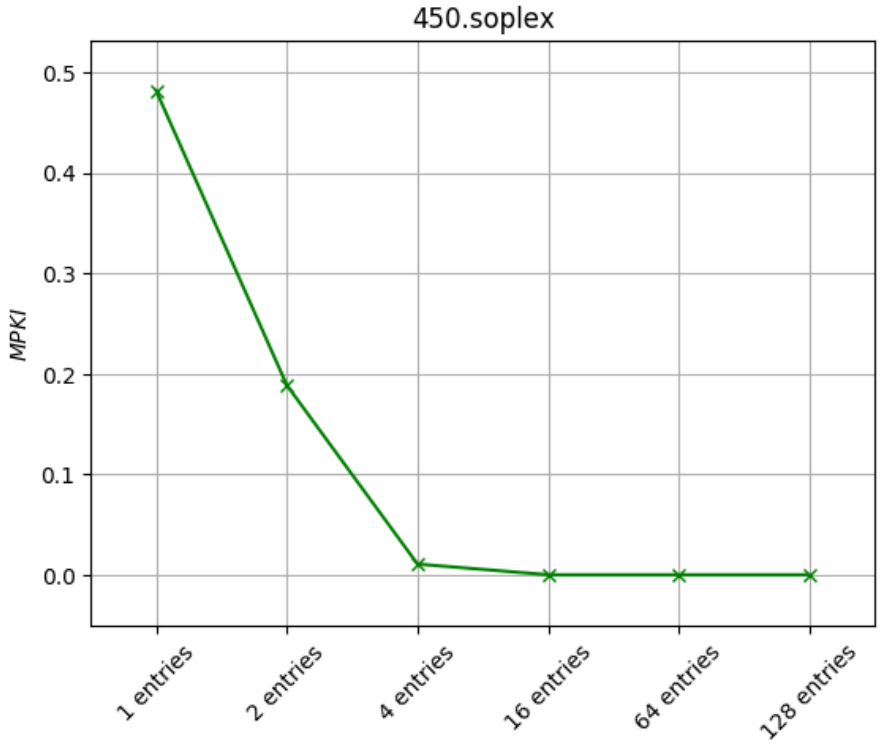

403.gcc
MPKI
0
1
2
3
4
5
6
1 entries
2 entries
4 entries
16 entries
64 entries
128 entries

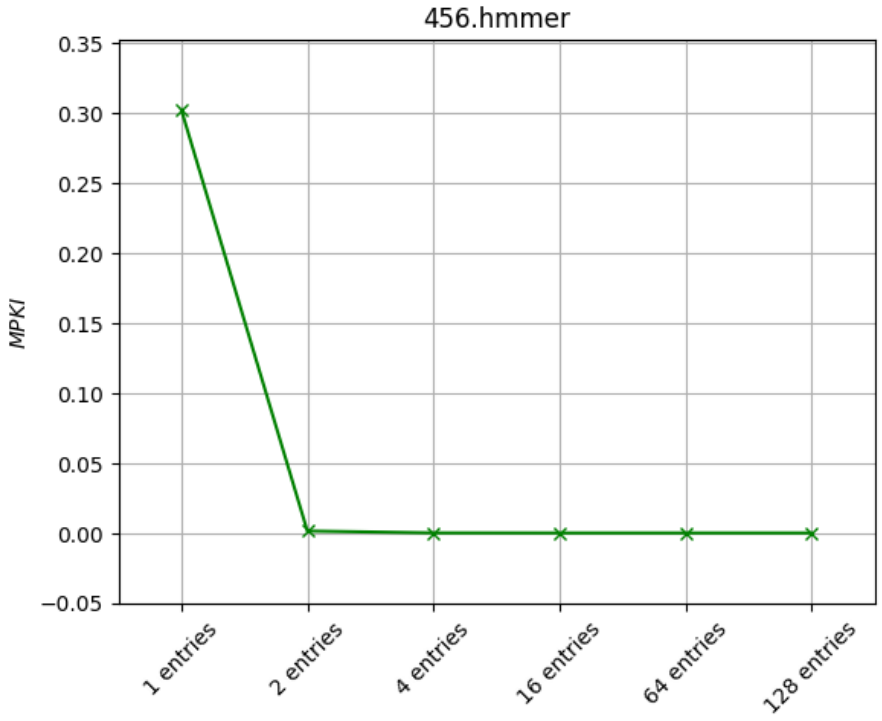

429.mcf
MPKI
−0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
1 entries
2 entries
4 entries
16 entries
64 entries
128 entries

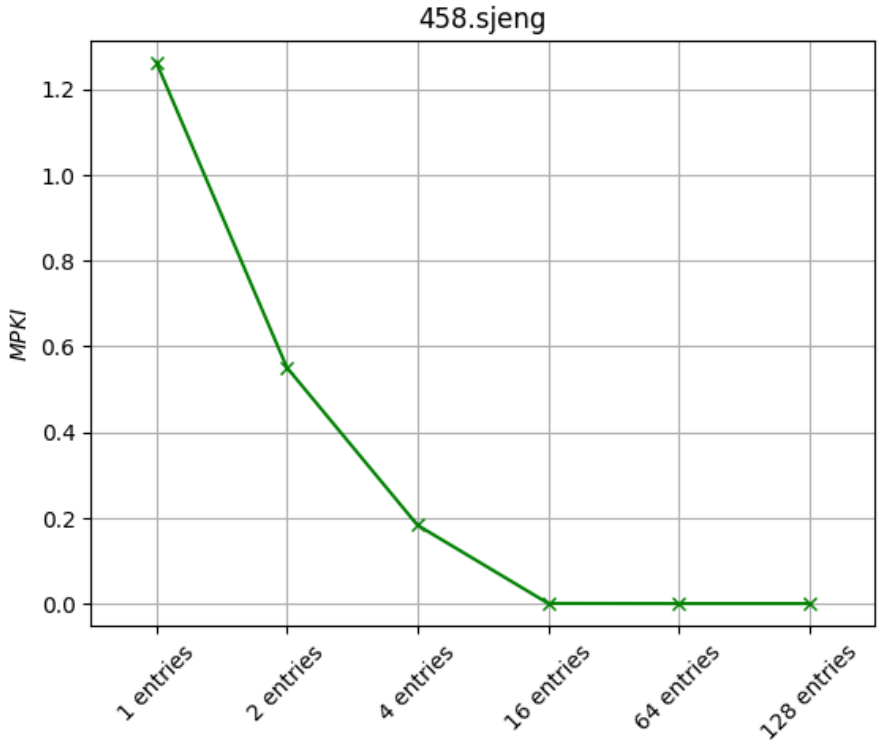

434.zeusmp
MPKI
0.04
0.02
0.00
−0.02
−0.04
1 entries
2 entries
4 entries
16 entries
64 entries
128 entries

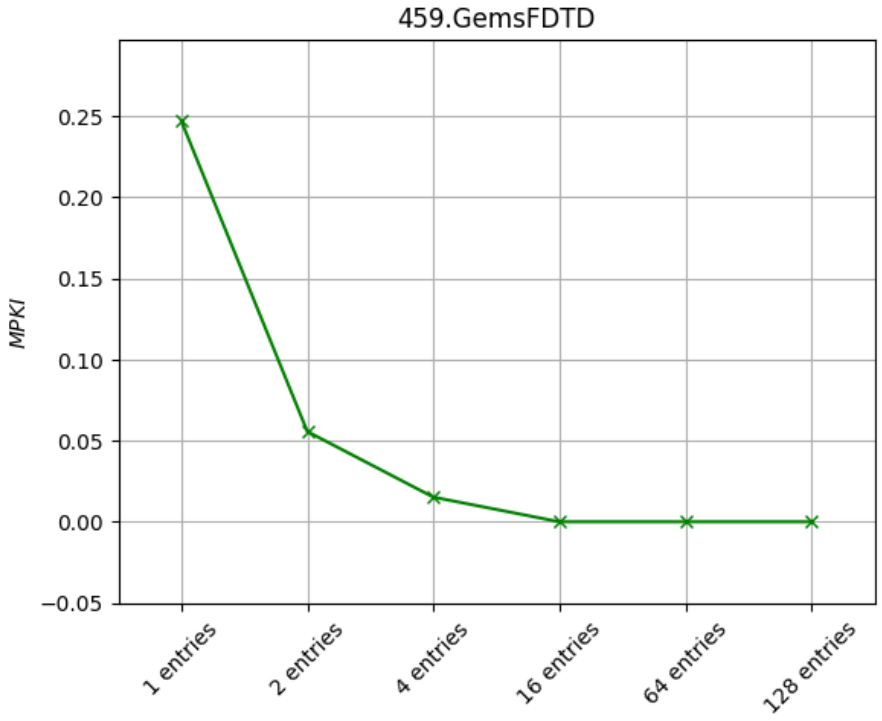

436.cactusADM
MPKI
0.06
0.04
0.02
0.00
−0.02
−0.04
1 entries
2 entries
4 entries
16 entries
64 entries
128 entries

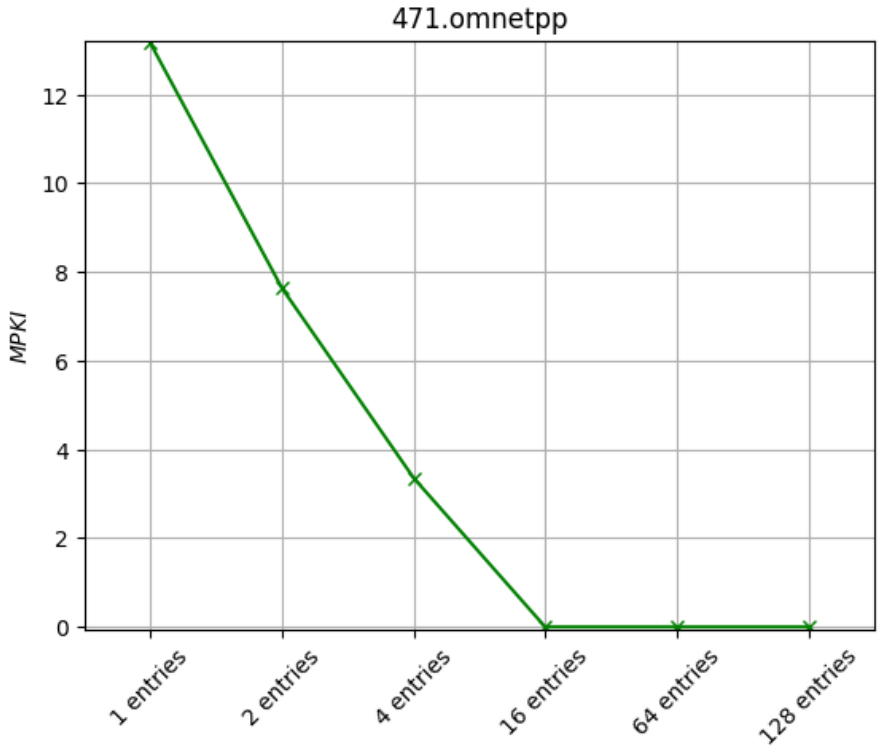

445.gobmk
MPKI
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1 entries
2 entries
4 entries
16 entries
64 entries
128 entries

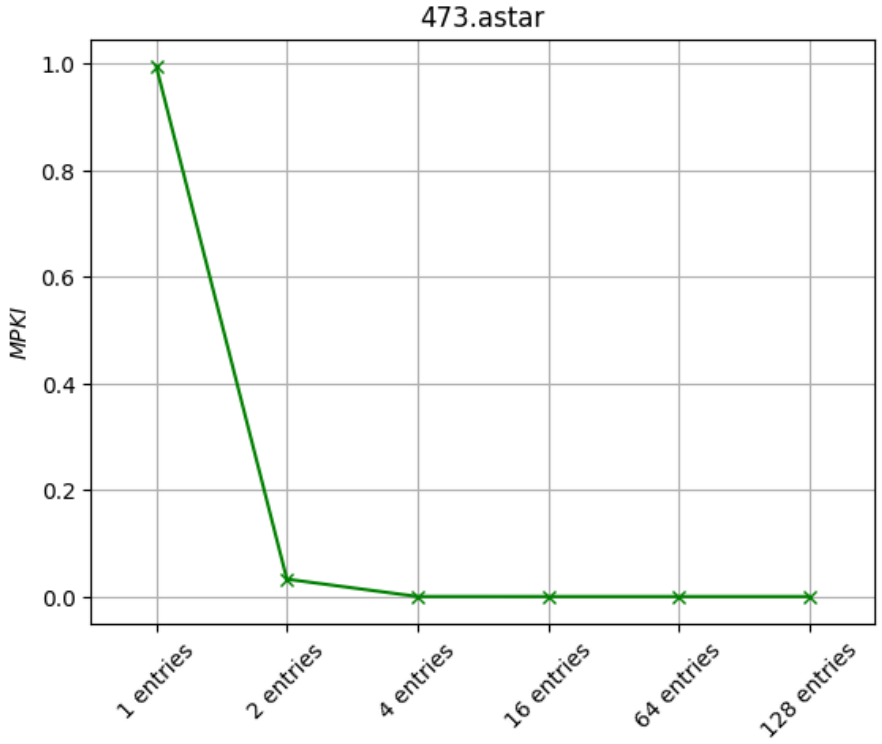

450.soplex
MPKI
0.5
0.4
0.3
0.2
0.1
0.0
1 entries
2 entries
4 entries
16 entries
64 entries
128 entries

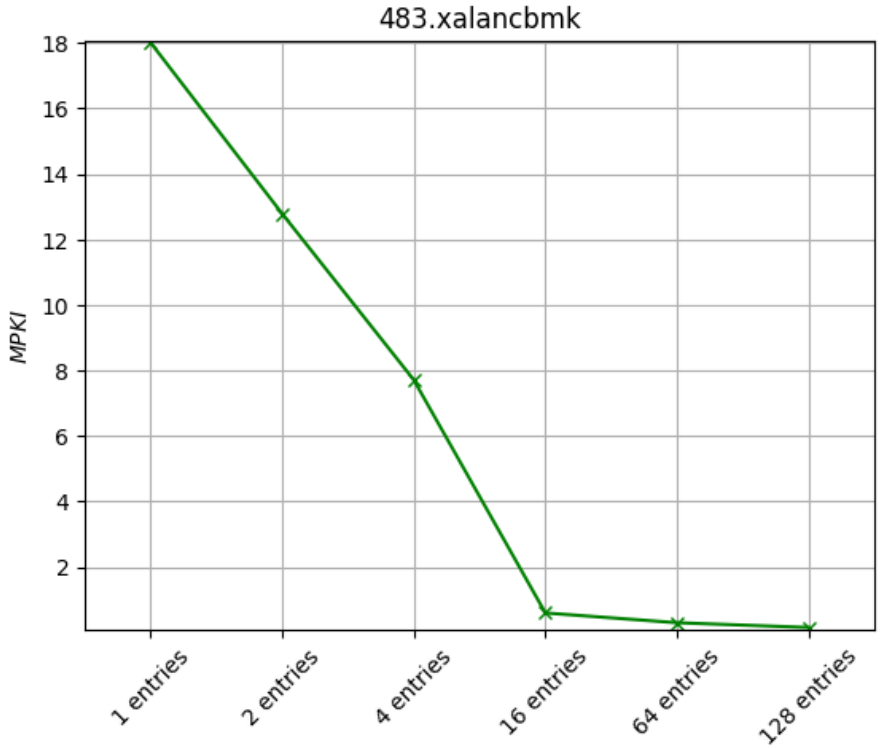

456.hmmer
MPKI
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
1 entries
2 entries
4 entries
16 entries
64 entries
128 entries


458.sjeng
MPKI
1.2
1.0
0.8
0.6
0.4
0.2
0.0
1 entries
2 entries
4 entries
16 entries
64 entries
128 entries


459.GemsFDTD
MPKI
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
1 entries
2 entries
4 entries
16 entries
64 entries
128 entries


471.omnetpp
MPKI
12
10
8
6
4
2
0
1 entries
2 entries
4 entries
16 entries
64 entries
128 entries





- **Συμπεράσματα**:

  1. Κατά μια γενική ομολογία και κάνοντας μια πολύ απλή σύγκριση με τις γραφικές MPKI που αφορούν την (3.2) Ενότητα - Μελέτη n-bit predictors, παρατηρούμε χαμηλότερη επίπεδα κατά μέσο όρο, με τα χειρότερα benchmarks να αγγίζουν 18% σε αντίθεση με προηγουμένως που κάποια ξεπερνούσαν και το 35%.
  2. Όταν $\#RAS = 1$, έχουμε παρ' όλα αυτά πολύ μεγάλο MPKI σχεδόν για όλα τα benchmarks, ενώ οταν αυξήσουμε τον αριθμό αυτόν σε 2 , παρατηρείται **μεγάλη μείωση του δείκτη**, ενώ για περισσότερα από 2 έγγραφα σημειώνεται αρκετά μικρότερων κλιμακίων βελτίωση.
  3. Τονίζουμε ότι οι παραπάνω προσομοιώσεις ήταν καταλυτικές για να κατανοήσουμε ότι *η αύξηση*

του # RAS δεν μπορεί να οδηγήσει σε μείωση της απόσοδης σε καμία των περιπτώσεων, άρα παρατηρούμε *μόνο διατήρηση ή και βελτίωση της απόσοδης*

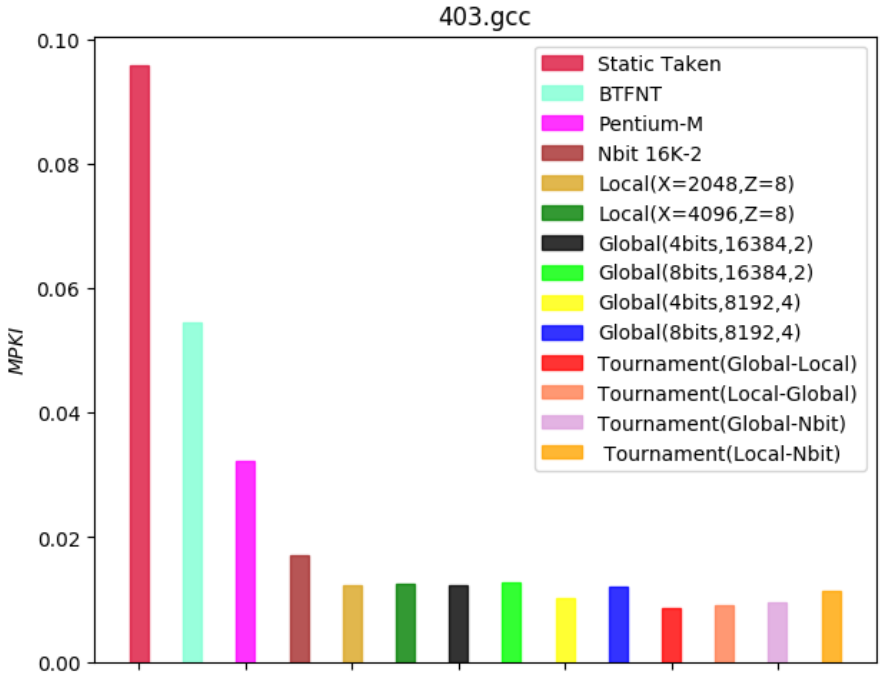
4. **Με αντικειμενικό κριτήριο το κόστος του υλικού, επιλέγουμε υλοποιήση που κάνει χρήση # RAS = 4 ή 8. (εφόσον δεν παρατηρείται και μεγαλη βελτωση με χρήση περισσότερων εγγράφων)**

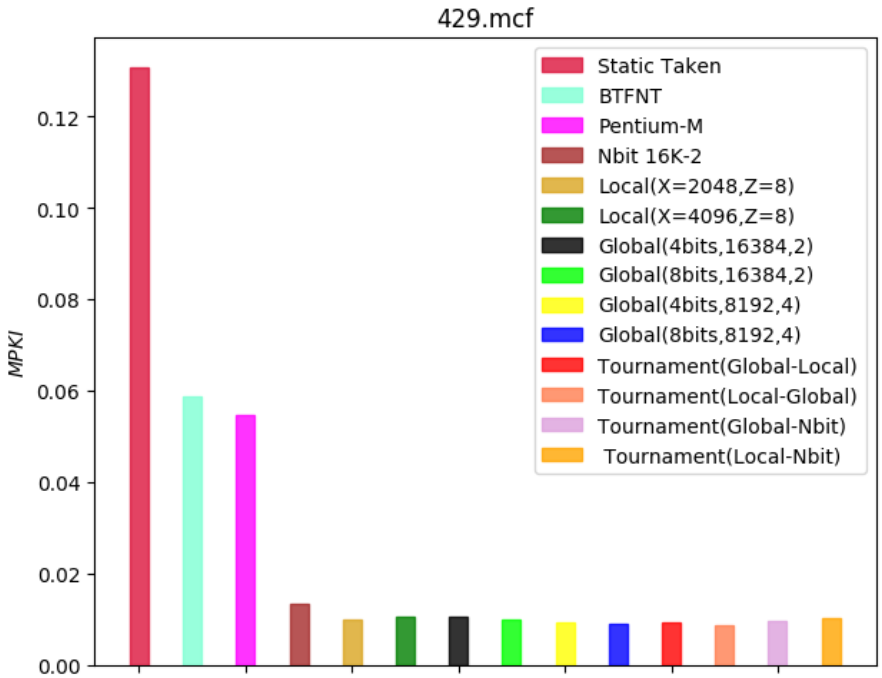
3.5 Σύγκριση διαφορετικών predictors

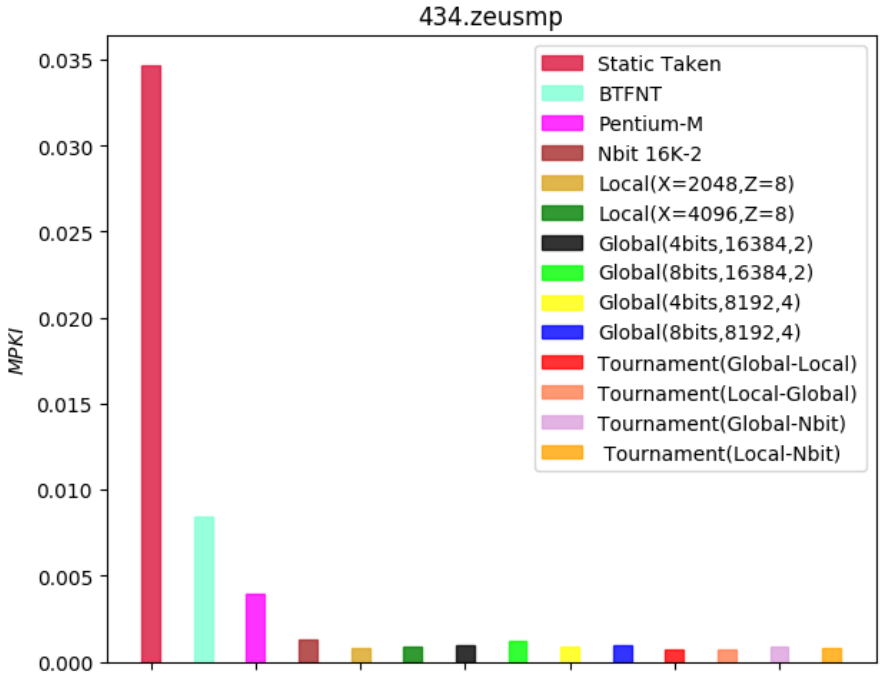
- Στο **Πέμπτο Μέρος** και τελευταίο της πειραματικής αξιολόγησης για την παρούσα άσκηση, εξετάζουμε την απόσοδη διαφορετικών predictors. Συγκεκριμένα

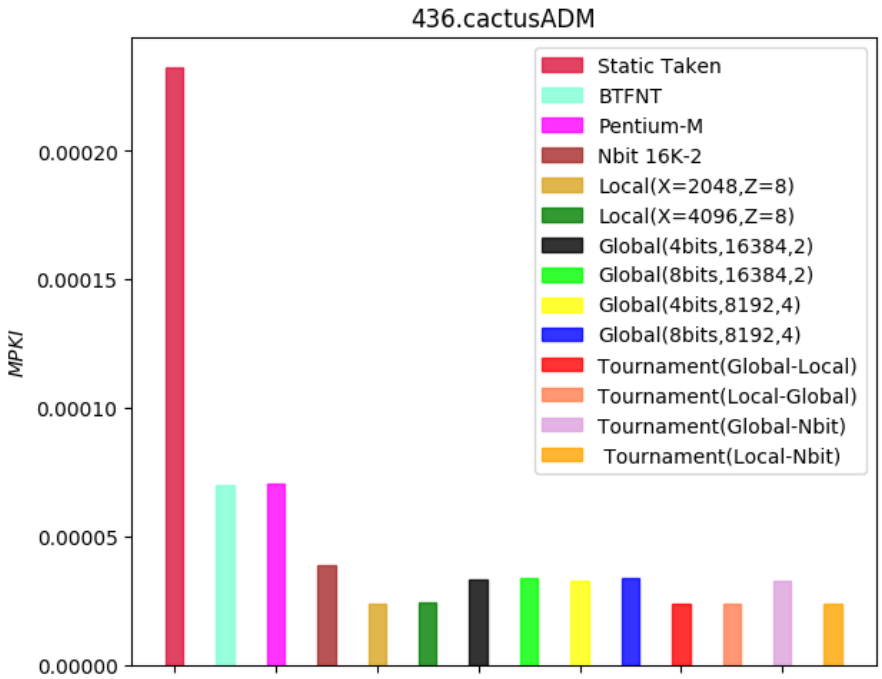
Predictors to compare
Static Taken
Pentium-M
Static BTFNT
Best **N-bit predictor** from part 4.2 → **16K-2**
Local-History 2-level [PHT entries = 8192, PHT n-bit counter length = 2, BHT entries = 2048, BHT entry length = 8]
Local-History 2-level: [PHT entries = 8192, PHT n-bit counter length = 2, BHT entries = 4096, BHT entry length = 4]
Global-History 2-level: [PHT entries = 16384, PHT n-bit counter length = 2, BHR length = 4]
Global-History 2-level: [PHT entries = 16384, PHT n-bit counter length = 2, BHR length = 8]
Global-History 2-level: [PHT entries = 16384, PHT n-bit counter length = 4, BHR length = 4]
Global-History 2-level: [PHT entries = 16384, PHT n-bit counter length = 4, BHR length = 8]
Tournament Hybrid predictor Global(optimal)-Local(optimal)
Tournament Hybrid predictor Local(optimal)-Global(optimal)
Tournament Hybrid predictor Local(optimal)-N-bit(optimal)
Tournament Hybrid predictor GLobal(optimal)-N-bit(optimal)

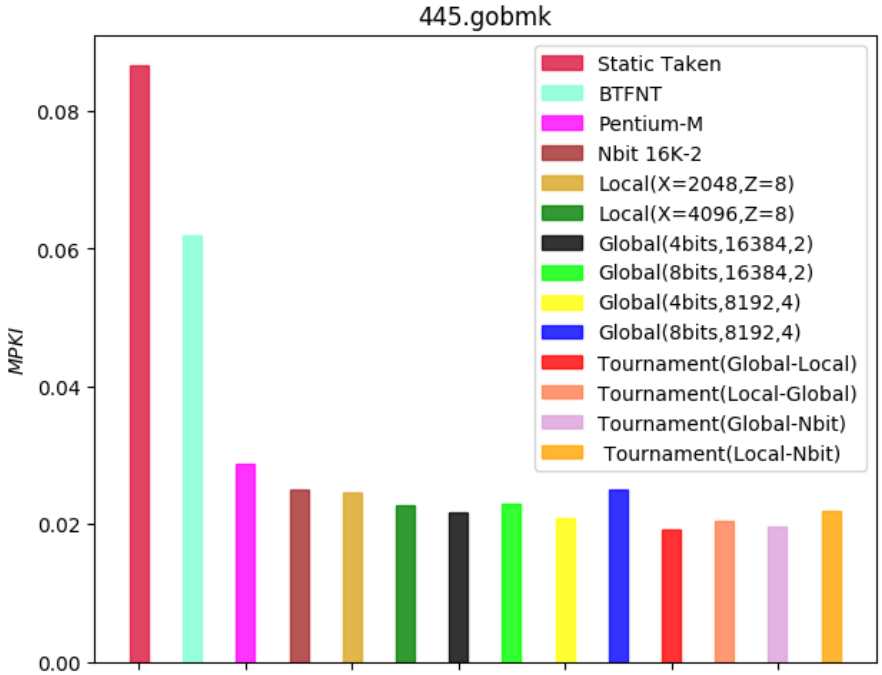
Αποτελέσματα MPKI για όλους τους παραπάνω predictors

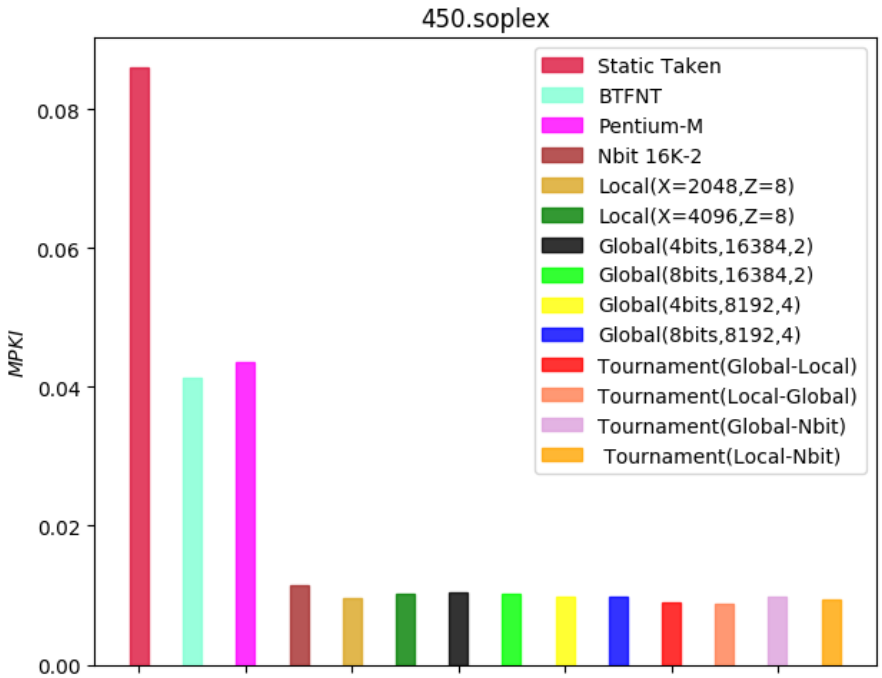

403.gcc
MPKI
0.10
0.08
0.06
0.04
0.02
0.00
Static Taken
BTFNT
Pentium-M
Nbit 16K-2
Local(X=2048,Z=8)
Local(X=4096,Z=8)
Global(4bits,16384,2)
Global(8bits,16384,2)
Global(4bits,8192,4)
Global(8bits,8192,4)
Tournament(Global-Local)
Tournament(Local-Global)
Tournament(Global-Nbit)
Tournament(Local-Nbit)

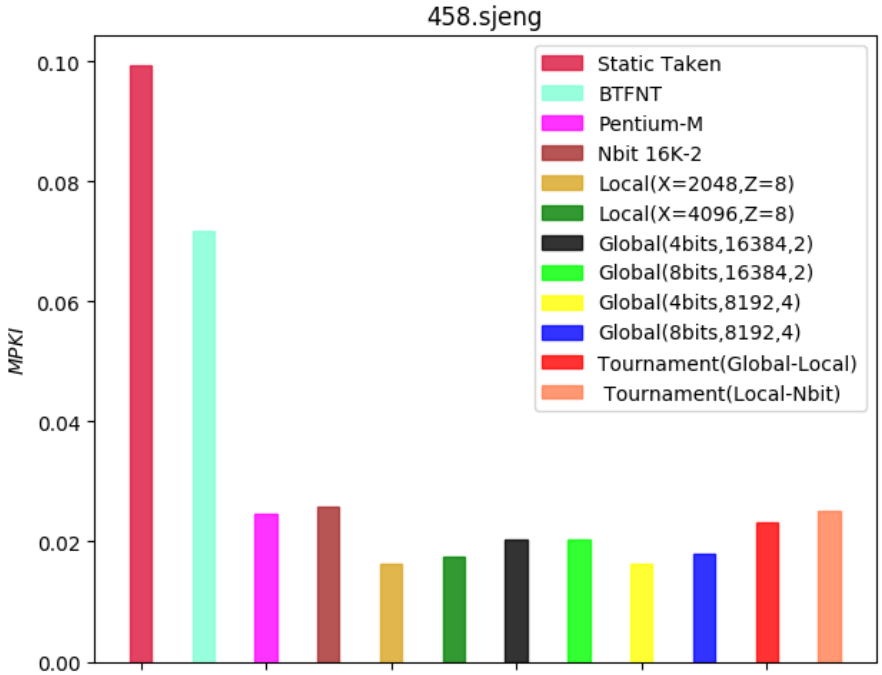

429.mcf
MPKI
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Static Taken
BTFNT
Pentium-M
Nbit 16K-2
Local(X=2048,Z=8)
Local(X=4096,Z=8)
Global(4bits,16384,2)
Global(8bits,16384,2)
Global(4bits,8192,4)
Global(8bits,8192,4)
Tournament(Global-Local)
Tournament(Local-Global)
Tournament(Global-Nbit)
Tournament(Local-Nbit)

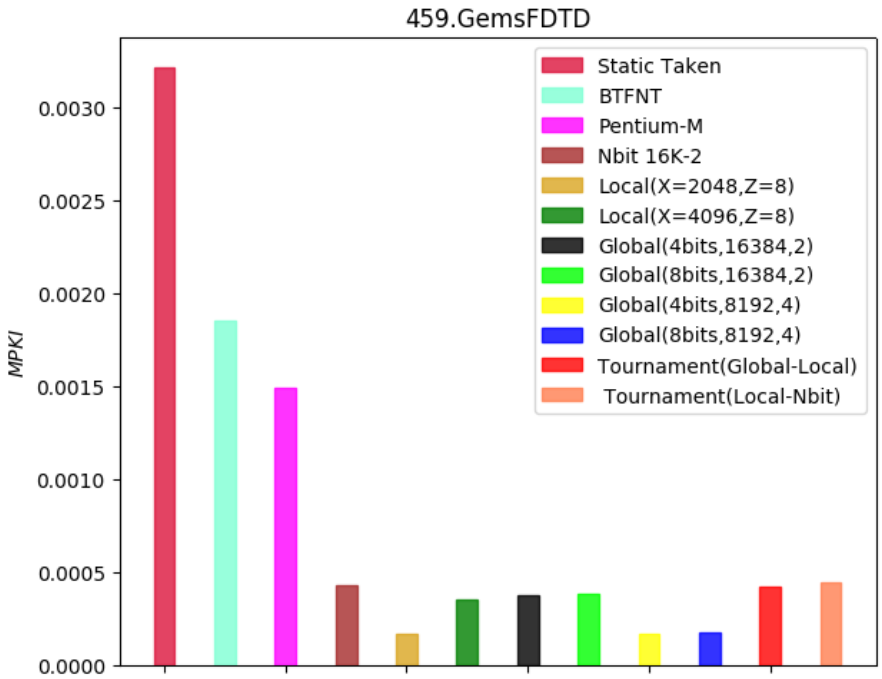

434.zeusmp
MPKI
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Static Taken
BTFNT
Pentium-M
Nbit 16K-2
Local(X=2048,Z=8)
Local(X=4096,Z=8)
Global(4bits,16384,2)
Global(8bits,16384,2)
Global(4bits,8192,4)
Global(8bits,8192,4)
Tournament(Global-Local)
Tournament(Local-Global)
Tournament(Global-Nbit)
Tournament(Local-Nbit)

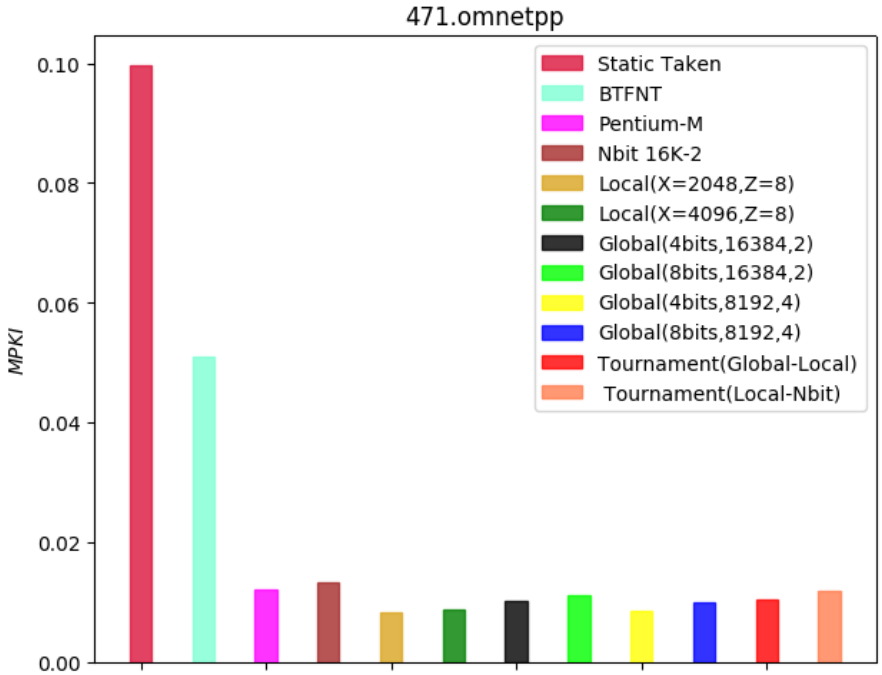

436.cactusADM
MPKI
0.00020
0.00015
0.00010
0.00005
0.00000
Static Taken
BTFNT
Pentium-M
Nbit 16K-2
Local(X=2048,Z=8)
Local(X=4096,Z=8)
Global(4bits,16384,2)
Global(8bits,16384,2)
Global(4bits,8192,4)
Global(8bits,8192,4)
Tournament(Global-Local)
Tournament(Local-Global)
Tournament(Global-Nbit)
Tournament(Local-Nbit)

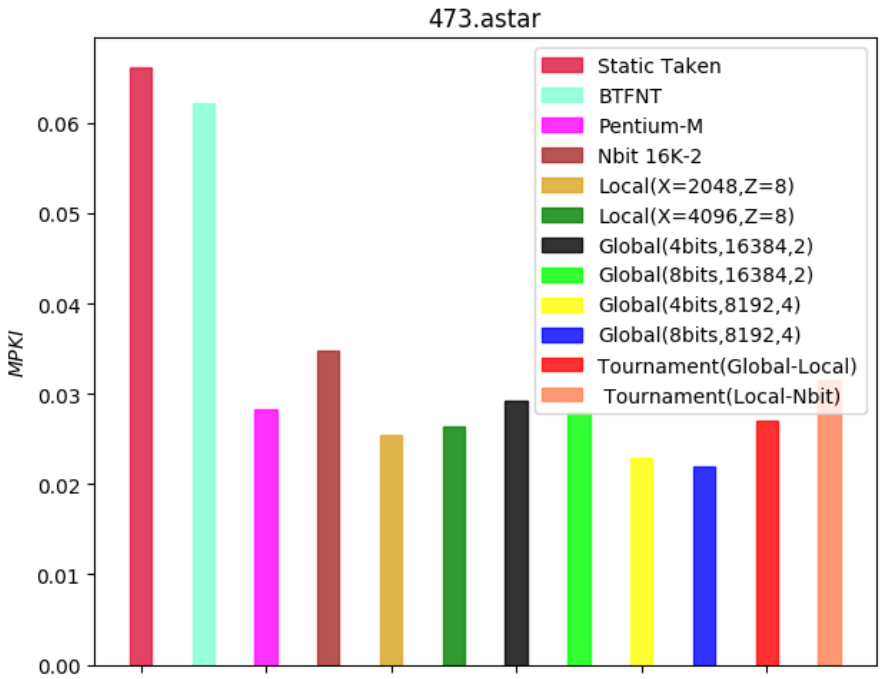

445.gobmk
MPKI
0.00
0.02
0.04
0.06
0.08
Static Taken
BTFNT
Pentium-M
Nbit 16K-2
Local(X=2048,Z=8)
Local(X=4096,Z=8)
Global(4bits,16384,2)
Global(8bits,16384,2)
Global(4bits,8192,4)
Global(8bits,8192,4)
Tournament(Global-Local)
Tournament(Local-Global)
Tournament(Global-Nbit)
Tournament(Local-Nbit)

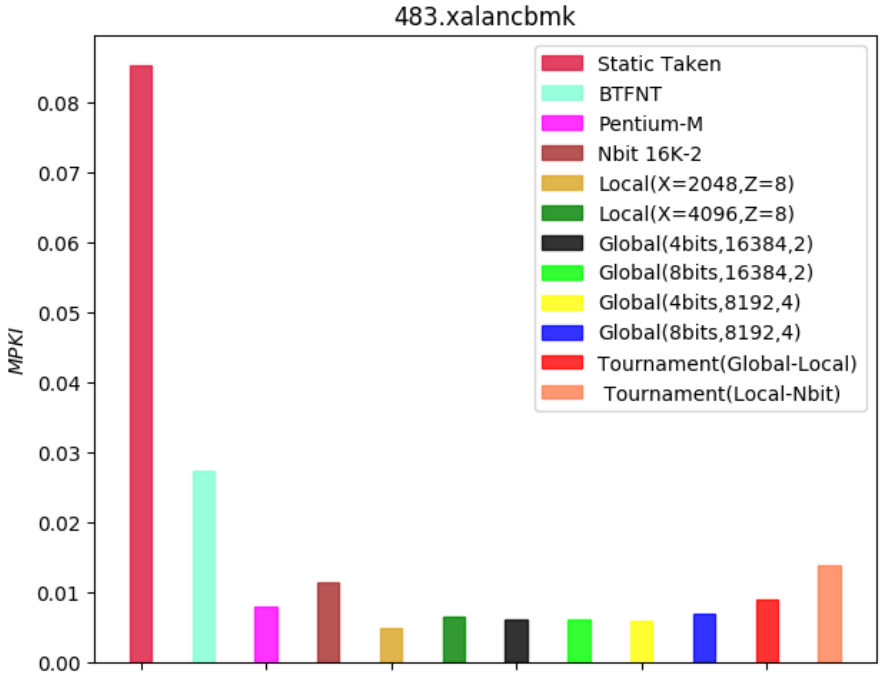

450.soplex
MPKI
0.00
0.02
0.04
0.06
0.08
Static Taken
BTFNT
Pentium-M
Nbit 16K-2
Local(X=2048,Z=8)
Local(X=4096,Z=8)
Global(4bits,16384,2)
Global(8bits,16384,2)
Global(4bits,8192,4)
Global(8bits,8192,4)
Tournament(Global-Local)
Tournament(Local-Global)
Tournament(Global-Nbit)
Tournament(Local-Nbit)


458.sjeng
MPKI
0.10
0.08
0.06
0.04
0.02
0.00
Static Taken
BTFNT
Pentium-M
Nbit 16K-2
Local(X=2048,Z=8)
Local(X=4096,Z=8)
Global(4bits,16384,2)
Global(8bits,16384,2)
Global(4bits,8192,4)
Global(8bits,8192,4)
Tournament(Global-Local)
Tournament(Local-Nbit)


459.GemsFDTD
MPKI
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
Static Taken
BTFNT
Pentium-M
Nbit 16K-2
Local(X=2048,Z=8)
Local(X=4096,Z=8)
Global(4bits,16384,2)
Global(8bits,16384,2)
Global(4bits,8192,4)
Global(8bits,8192,4)
Tournament(Global-Local)
Tournament(Local-Nbit)


471.omnetpp
MPKI
0.10
0.08
0.06
0.04
0.02
0.00
Static Taken
BTFNT
Pentium-M
Nbit 16K-2
Local(X=2048,Z=8)
Local(X=4096,Z=8)
Global(4bits,16384,2)
Global(8bits,16384,2)
Global(4bits,8192,4)
Global(8bits,8192,4)
Tournament(Global-Local)
Tournament(Local-Nbit)


473.astar
MPKI
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Static Taken
BTFNT
Pentium-M
Nbit 16K-2
Local(X=2048,Z=8)
Local(X=4096,Z=8)
Global(4bits,16384,2)
Global(8bits,16384,2)
Global(4bits,8192,4)
Global(8bits,8192,4)
Tournament(Global-Local)
Tournament(Local-Nbit)



- **Συμπεράσματα**:

  1. Ο **Static Taken predictor** έχει την χειρότερη δυνατή απόδοση για όλα τα benchmarks , και αυτό γιατί αλγοριθμικά θεωρεί οτι τα branch δεν θα εκτελεστούν ενώ έχουμε μεγάλου βρόγχους.
  2. Ο **BTFNT predictor** παρουσιάζει κατά γενική ομολογία καλύτερη απόδοση για όλα τα benchmarks συγκριτικά με **Static Taken predictor**, αλλά τα επίπεδα του δείκτη MKPI παραμένουν ακόμα αρκετά υψηλά
  3. Εν συνεχεία οι **N-bit**, **Pentium-M** και όλοι οι **Global History** και **Local History** οδηγούν σε πολύ χαμηλότερες τιμές του δείκτη MKPI συγκριτικά με τους 2 προαναφερθέντες, αλλά δεν παρουσιάζεται σαφές pattern για εξαγωγή άμεσου αποτελέσματος για την απόδοση τους, οπότε η ταξινόμηση τους θα γίνεται για την απόδοση του κατά μέσο όρο για όλα τα benchmarks.
  4. Οι **tournament predictors**, παρουσιάζουν επίσης καλή απόδοση , είναι υβριδικοί και χρησιμοποιούν μια συνδυασμένη έκδοση **Global**, **Local** και **N-bit** predictors.

Ταξινόμηση των predictors με αντικειμενικό κριτήριο την κάτα μέσο όρο απόδοση τους πάνω στα 12 benchmarks

1. **Global History (8 bits)** predictor
2. **Local History (8 bits)** predictor
3. **Local History (4 bits)** predictor
4. **Global History (4 bits)** predictor
5. **Tournament (Local - Global)** predictor
6. **Tournament (Global - Local)** predictor
7. **Tournament (Global - Nbit)** predictor
8. **Tournament (Local - Nbit)** predictor
9. **N-bit 16K-2** predictor
10. **Pentium-M** predictor
11. **Static BTFNT** predictor
12. **Static Taken** predictor