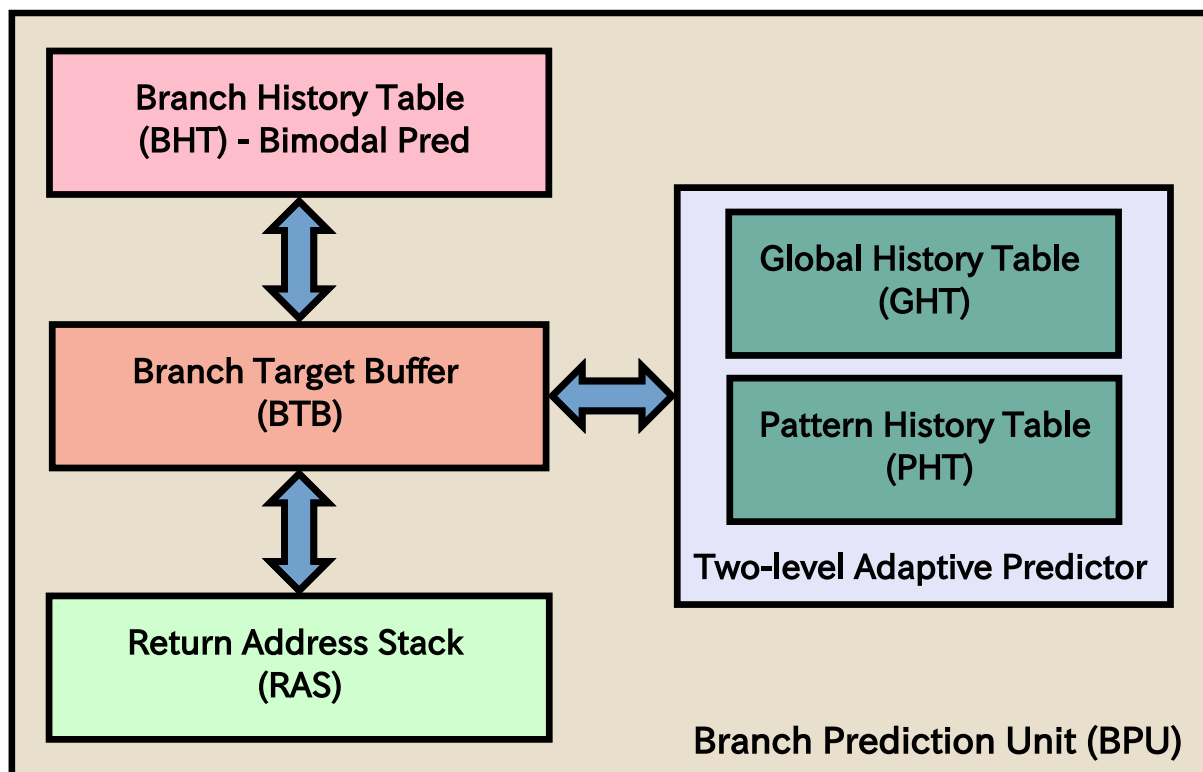


Άσκηση 2: Προσομοίωση Πρόβλεψης Διακλάδωσης

Μάθημα: Αρχιτεκτονική Υπολυστών II

Καραμπέρης Παναγιώτης ΑΜ: 5241



1. Εισαγωγή

Η παρούσα εργασία έχει ως στόχο την υλοποίηση και αξιολόγηση διαφορετικών προβλεπτών διακλάδωσης (branch predictors) στο πλαίσιο ενός προσομοιωτή βασισμένου στο εργαλείο Intel PIN Tool. Πιο συγκεκριμένα, ζητήθηκε να υλοποιηθούν μερικοί στατικοί και δυναμικοί προβλέπτες διακλάδωσης καθώς και να γίνει σύγκριση της επίδοσης του καθενός. Οι προσομοιώσεις πραγματοποιήθηκαν πάνω σε εφαρμογές του PARSEC benchmark suite, με στόχο τη μέτρηση της ακρίβειας κάθε προβλέπτη. Η μέτρηση βασίστηκε στον υπολογισμό του MPKI (Mispredictions Per 1000 Instructions) για κάθε προβλέπτη και benchmark.

2. Υλοποίηση προβλεπτών διακλάδωσης

- **Στατικός not taken**

Ο συγκεκριμένος αποτελεί τον πιο απλό προβλέπτη καθώς υποθέτει ότι όλες οι διακλαδώσεις δεν ακολουθούνται.

- **Στατικός BTBNT**

Ο BTBNT εξετάζει την θέση του στόχου: αν ο στόχος είναι προς τα πίσω ($\text{target} < \text{ip}$) προβλέπεται ως taken, ενώ αν ο στόχος είναι πιο μπροστά ($\text{target} > \text{ip}$) ως not taken.

- **Δυναμικός 1-bit, 16K**

Η διακλάδωση προβλέπεται ότι θα κατευθυνθεί όπως και την προηγούμενη φορά.

- **Δυναμικός 2-bit, 8K**

Ο συγκεκριμένος προβλέπτης αποτελεί βελτίωση του 1-bit. Διατηρεί 2-bit counters, σε πίνακα 8K εγγραφών. Κάθε counter έχει 4 καταστάσεις:

- 00 – Strongly Not Taken
- 01 – Weakly Not Taken
- 10 – Weakly Taken
- 11 – Strongly Taken

Μετά από δύο συνεχόμενες λανθασμένες (ή σωστές) προβλέψεις αλλάζει η πρόβλεψη (Taken-Not Taken)

- **Gshare**

Ο Gshare είναι ένας δυναμικός προβλέπτης διακλάδωσης ο οποίος αξιοποιεί τον Global History Register (GHR). Η βασική του λειτουργία βασίζεται στον συνδυασμό της διεύθυνσης της διακλάδωσης (IP) με το περιεχόμενο του GHR, με χρήση της πράξης XOR. Το αποτέλεσμα αυτής της πράξης χρησιμοποιείται ως δείκτης σε έναν πίνακα προβλέψεων Pattern History Table (PHT), ο οποίος περιέχει 2-bit counters. Κάθε counter προβλέπει εάν η αντίστοιχη διακλάδωση θα εκτελεστεί ή όχι, με βάση την τιμή του. Μετά από κάθε διακλάδωση, ο GHR ενημερώνεται προσθέτοντας το πραγματικό αποτέλεσμα στο ιστορικό και ταυτόχρονα ενημερώνεται ο αντίστοιχος μετρητής στο PHT. Η χρήση XOR επιτρέπει την

αποφυγή συγκρούσεων στο PHT

- **BTB + RAS**

Ο προβλέπτης BTB (Branch Target Buffer) σε συνδυασμό με RAS (Return Address Stack) εστιάζει στην πρόβλεψη τόσο της κατεύθυνσης (taken/not taken) όσο και του στόχου (target) της διακλάδωσης, με έμφαση στις call και return εντολές. Ο BTB λειτουργεί ως ένας πίνακας, στον οποίο αποθηκεύονται ζεύγη της μορφής IP → Target για διακλαδώσεις που έχουν προβλεφθεί ως taken. Κατά την πρόβλεψη, αν η διακλάδωση αναγνωριστεί ως taken και υπάρχει έγκυρη καταχώρηση στο BTB, τότε χρησιμοποιείται το αποθηκευμένο target. Σε περίπτωση λάθους στόχου, ενημερώνεται η καταχώρηση ή αντικαθίσταται μέσω LRU. Παράλληλα, η RAS λειτουργεί ως στοίβα για την πρόβλεψη return εντολών: στην call εντολή αποθηκεύεται στη RAS η διεύθυνση επιστροφής (IP + 4), ενώ στην ret, γίνεται pop από τη RAS και η τιμή που εξάγεται συγκρίνεται με το πραγματικό target. Αν η πρόβλεψη είναι λανθασμένη, μετρίεται ως Incorrect_RAS.

3. Τεχνική ανάλυση BTB και Gshare

- **Gshare**

Ο Gshare χρησιμοποιεί έναν πίνακα προβλέψεων (Pattern History Table – PHT) μεγέθους 8K εγγραφών, δηλαδή 2^{13} θέσεις, με 2-bit counters σε κάθε θέση. Το index στον πίνακα υπολογίζεται μέσω XOR μεταξύ των 13 κατώτερων bits της διεύθυνσης της διακλάδωσης (IP) και του Global History Register (GHR), ο οποίος επίσης έχει μέγεθος 13 bits. Ο GHR ενημερώνεται δυναμικά με κάθε νέα διακλάδωση, καταγράφοντας εάν ήταν taken ή not taken, επιτρέποντας έτσι στον προβλέπτη να προσαρμόζεται σε πρόσφατα μοτίβα εκτέλεσης.

- **BTB + RAS**

Το πλήθος των sets για την τον BTB προβλέπτη υπολογίζεται ως:

$$num_sets = \frac{btb_lines}{btb_assoc}$$

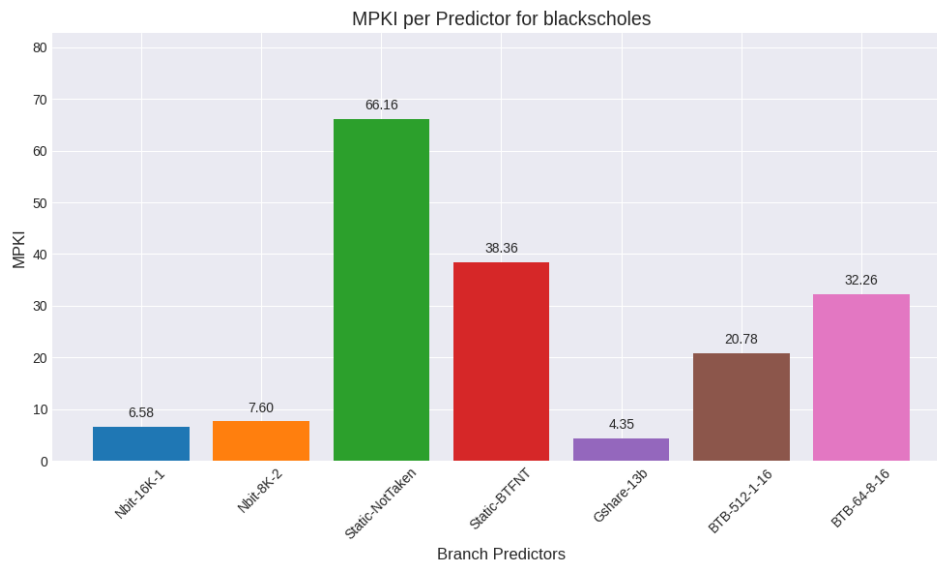
Κάθε σύνολο υλοποιείται ως λίστα που αποθηκεύει ζεύγη της μορφής IP->Target και διαχειρίζεται τις εγγραφές με πολιτική LRU.

Η Return Address Stack (RAS) χρησιμοποιείται για την πρόβλεψη των ret εντολών. Κατά την call, αποθηκεύουμε τη διεύθυνση επιστροφής (IP + 4), ενώ στην ret, κάνουμε pop και συγκρίνουμε με το actual target.

4. Αποτελέσματα

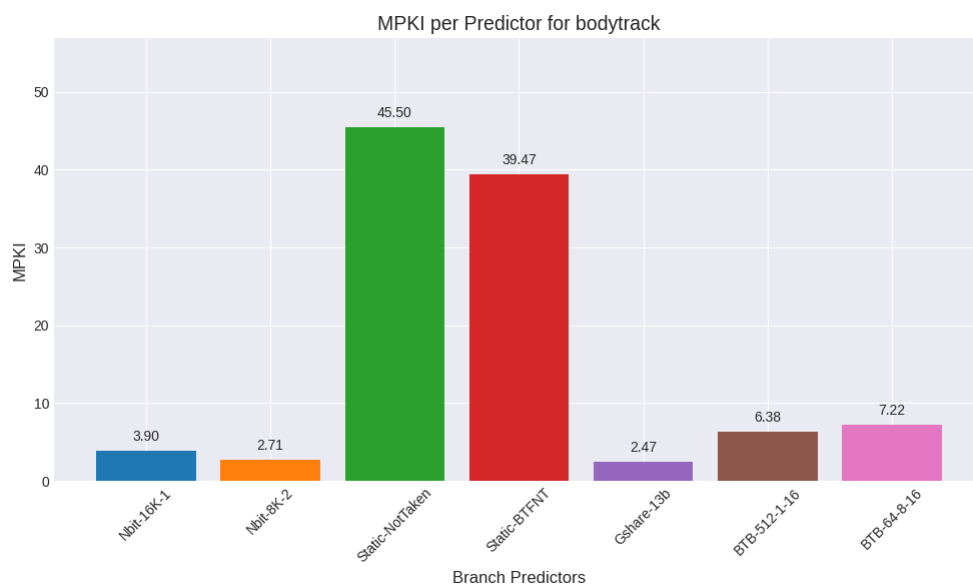
Αφού υλοποιήθηκαν όλοι οι προβλέπτες, τρέξαμε τα benchmarks και πήραμε τα ακόλουθα αποτελέσματα:

- **blackscholes**



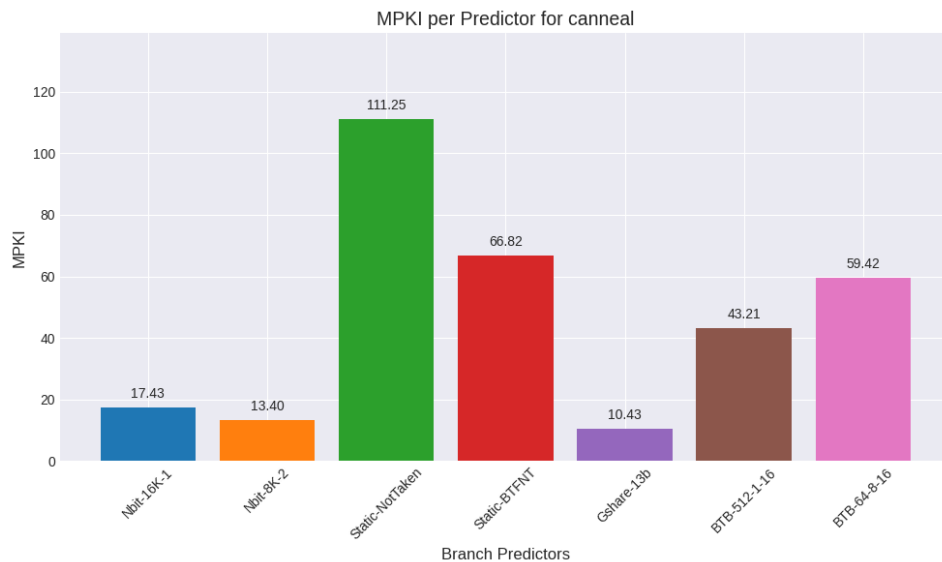
Συμπεράσματα: Ο Gshare εμφανίζει τη χαμηλότερη τιμή MPKI, γεγονός που υποδηλώνει προβλέψιμο μοτίβο διακλαδώσεων που επωφελείται από τη χρήση ιστορικού. Ο 2-bit δυναμικός προβλέπτης ακολουθεί με παρόμοια απόδοση, ενώ ο BTB κινείται σε μέτρια επίπεδα. Οι στατικοί προβλέπτες εμφανίζουν σημαντικά υψηλό MPKI, ιδιαίτερα ο Static-NotTaken, δείχνοντας ότι η στατική προσέγγιση είναι ανεπαρκής για το συγκεκριμένο πρόγραμμα.

- **bodytrack**



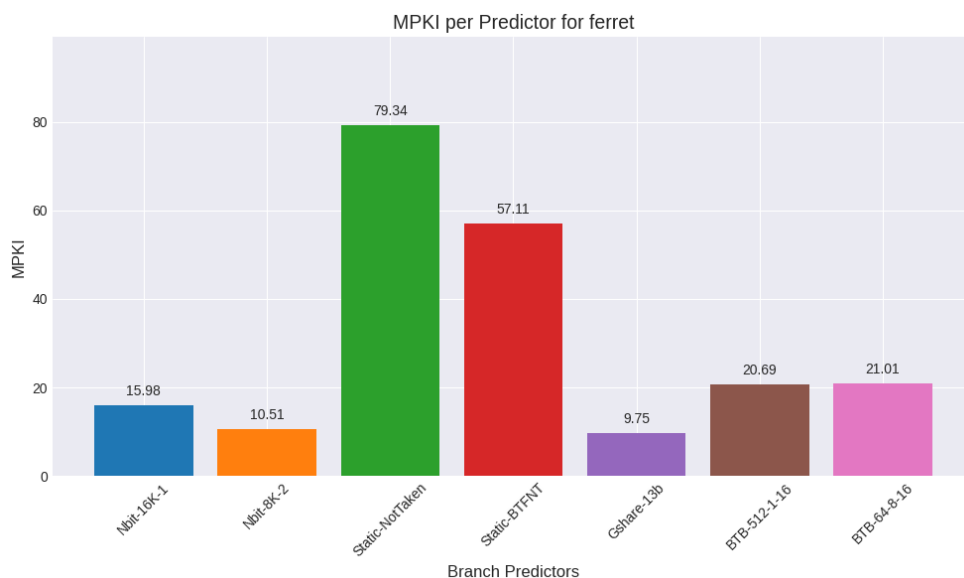
Συμπεράσματα: Αρκετά καλά αποτελέσματα παρατηρούνται και εδώ με τους δυναμικούς προβλέπτες, κυρίως για τον Gshare καθώς και για τον BTB. Οι στατικοί προβλέπτες παρουσιάζουν σαφώς χειρότερη απόδοση, κάτι που δείχνει πιο σύνθετη διακλάδωση που επωφελείται από την ιστορικότητα των δυναμικών μεθόδων.

- **canneal**



Συμπεράσματα: Το canneal έχει από τις υψηλότερες τιμές MPKI στους στατικούς προβλέπτες, ειδικά στον Static-NotTaken, ενώ και οι BTB προβλέπτες παρουσιάζουν σχετικά υψηλά MPKI. Αυτό δείχνει ότι πρόκειται για ένα απαιτητικό πρόγραμμα σε επίπεδο πρόβλεψης διακλάδωσης, με καλύτερη απόδοση από τους Nbit και Gshare.

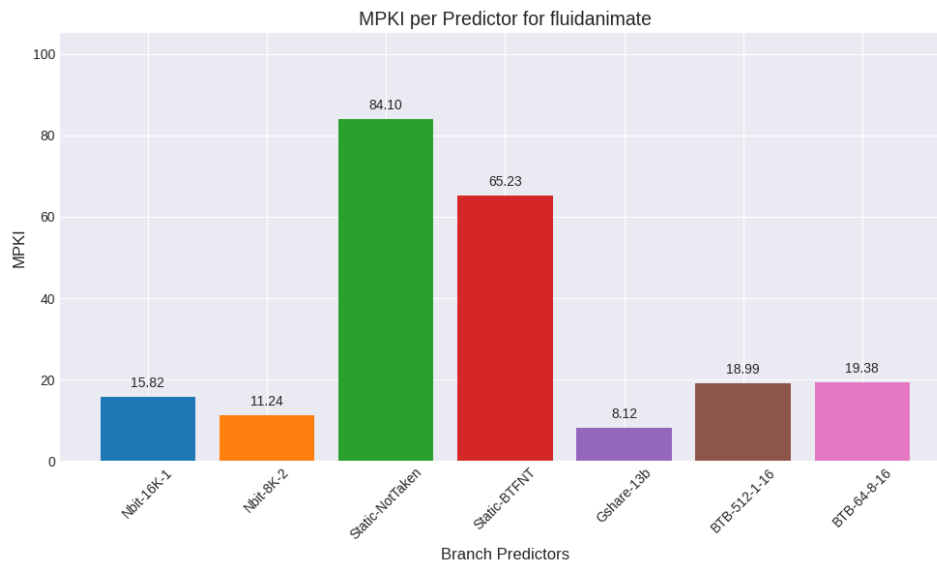
- **ferret**



Συμπεράσματα: Το ferret παρουσιάζει υψηλά MPKI στους στατικούς προβλέπτες αλλά και στους BTB-64 και BTB-512, ενώ οι Nbit και Gshare επιτυγχάνουν τις καλύτερες επιδόσεις. Αυτό δείχνει

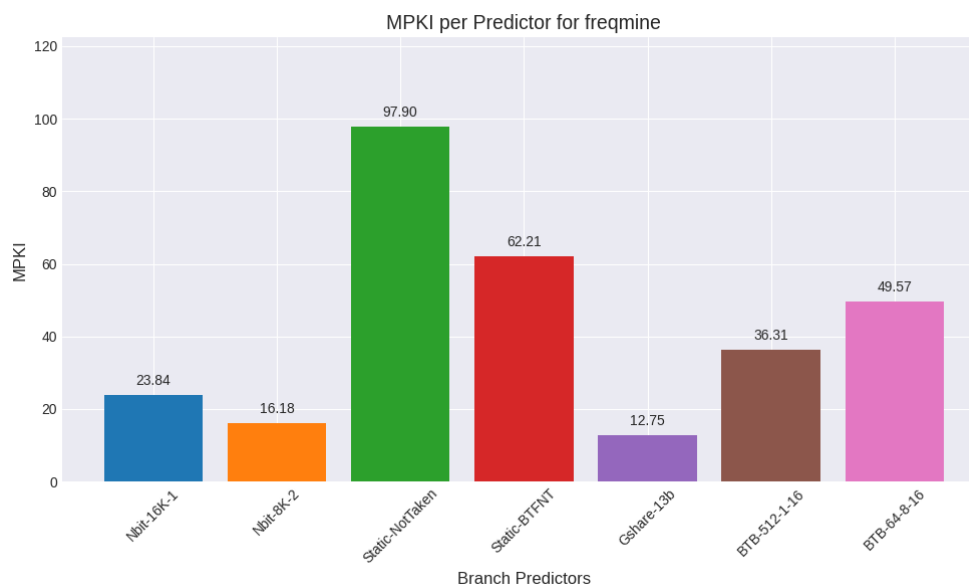
πως το πρόγραμμα έχει πολύπλοκα μοτίβα διακλάδωσης που ευνοούν προβλέπτες με ιστορικότητα.

- fluidanimate



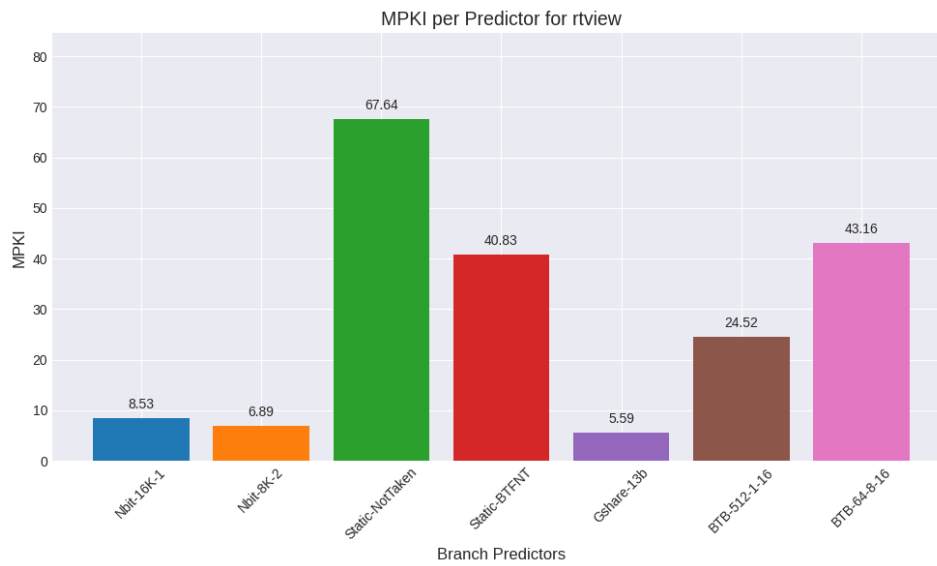
Συμπεράσματα: Παρόμοια εικόνα με το ferret, το fluidanimate δείχνει υψηλά MPKI στους στατικούς προβλέπτες αλλά πιο χαμηλά για Gshare και Nbit. Ο Gshare έχει πολύ καλή απόδοση, υποδεικνύοντας υψηλό βαθμό προβλεψιμότητας μέσω ιστορικών δεδομένων.

- freqmine



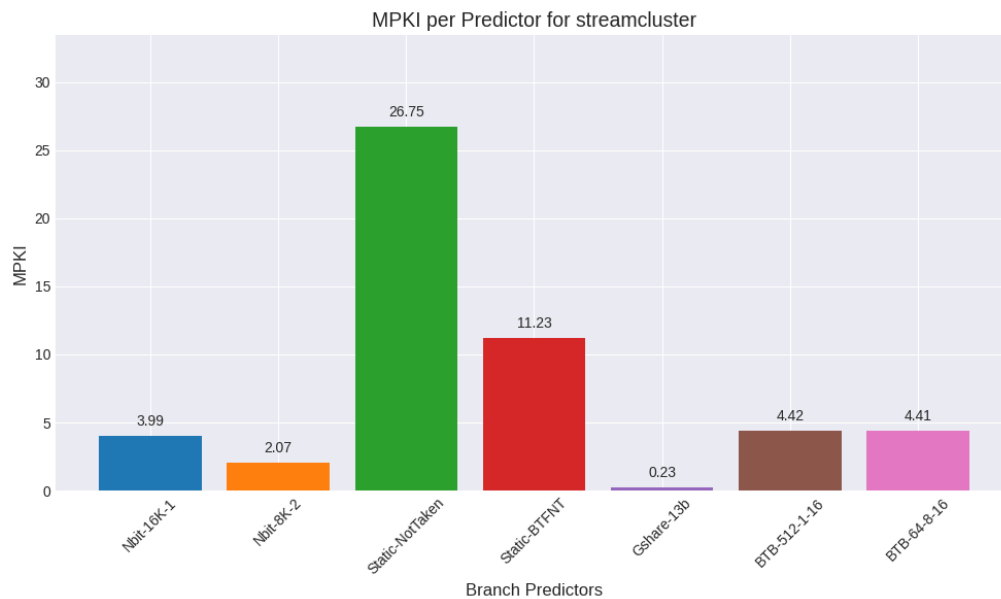
Συμπεράσματα: Το freqmine έχει επίσης υψηλά MPKI για στατικούς και BTB προβλέπτες, ενώ οι Gshare και Nbit εμφανίζουν πολύ καλύτερη συμπεριφορά. Αυτό υποδηλώνει ένα πρόγραμμα με αρκετές διακλαδώσεις που αξιοποιείται αποτελεσματικά από δυναμικούς προβλέπτες.

- **rtview**



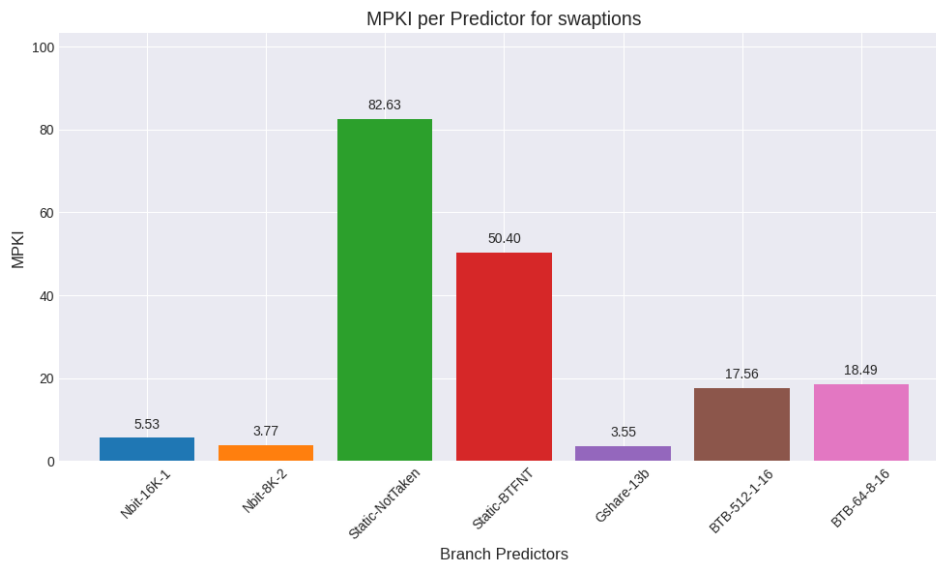
Συμπεράσματα: Το συγκεκριμένο benchmark παρουσιάζει χαμηλό MPKI για όλους τους δυναμικούς προβλέπτες, ειδικά για τον Gshare και τον Nbit. Αυτό δείχνει ότι οι διακλαδώσεις είναι ιδιαίτερα προβλέψιμες και οι δυναμικές μέθοδοι τις χειρίζονται αποτελεσματικά.

- **streamcluster**



Συμπεράσματα: Παρότι οι στατικοί προβλέπτες δεν είναι αποδοτικοί, οι δυναμικοί προβλέπτες καταγράφουν πολύ χαμηλό MPKI, κάτι που φανερώνει υψηλή προβλεψιμότητα των διακλαδώσεων. Ο Gshare έχει την καλύτερη επίδοση.

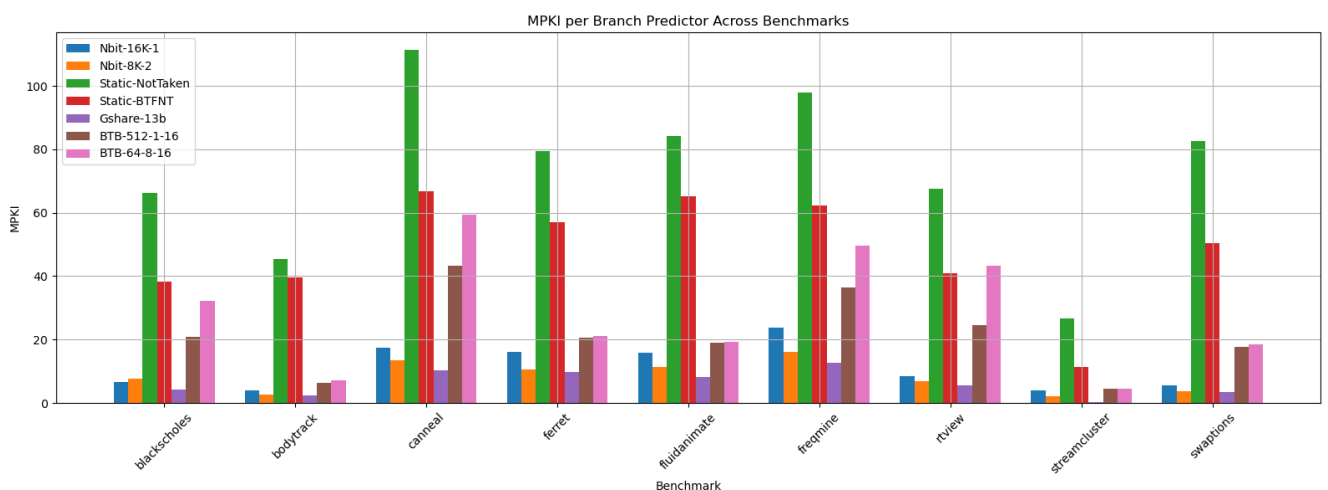
- **swaptions**



Συμπεράσματα: Ο Gshare υπερέχει ξεκάθαρα, με πολύ χαμηλό MPKI, ενώ ο BTB κινείται σε ενδιάμεσα επίπεδα. Οι δυναμικοί Nbit προβλέπτες έχουν ικανοποιητική απόδοση, ενώ οι στατικοί υστερούν σημαντικά, ειδικά ο Static-NotTaken.

5. Ανάλυση Αποτελεσμάτων

Για να αναλύσουμε τα αποτελέσματα με μεγαλύτερη ευκολία φτιάξαμε ένα ενιαίο διάγραμμα στο οποίο αποτυπώνονται συνολικά όλες οι προηγούμενες μετρήσεις για κάθε προβλέπτη του κάθε benchmark.



Συνολικά, από τη σύγκριση των προβλέψεων διακλάδωσης στα διάφορα benchmarks, προκύπτει ότι οι δυναμικοί προβλέπτες, και ιδιαίτερα οι Gshare και Nbit, επιτυγχάνουν σαφώς χαμηλότερες τιμές MPKI σε σχέση με τους στατικούς, αποδεικνύοντας την αποτελεσματικότητά τους σε προγράμματα

με πολύπλοκες και μεταβαλλόμενες διακλάδωσης. Αντίθετα, οι στατικοί προβλέπτες (Static-NotTaken και Static-BTFNT) παρουσιάζουν ιδιαίτερα υψηλά MPKI, αποτυγχάνοντας να ανταποκριθούν στην πολυπλοκότητα των benchmarks. Οι προβλέπτες τύπου BTB έχουν ενδιάμεση απόδοση, με τις επιδόσεις τους να εξαρτώνται σημαντικά από την αρχιτεκτονική του κάθε προγράμματος. Συνεπώς, η χρήση δυναμικών προβλεπτών με υποστήριξη ιστορικού διακλάδωσης αναδεικνύεται ως η πλέον αποδοτική λύση για τη μείωση των αστοχιών διακλάδωσης.

6. Συμπεράσματα

Η εργασία αυτή ανέδειξε τη σημασία της σωστής πρόβλεψης διακλάδωσης για την απόδοση επεξεργαστών. Η σύγκριση στατικών και δυναμικών προβλεπτών ανέδειξε την υπεροχή των προβλεπτών που χρησιμοποιούν ιστορικό (Gshare, Nbit). Οι δυναμικές τεχνικές πέτυχαν δραστική μείωση του MPKI, ιδιαίτερα σε εφαρμογές με σύνθετη ροή ελέγχου. Από την άλλη, οι στατικοί προβλέπτες, λόγω της απλότητάς τους, απέτυχαν να προσαρμοστούν στα διαφορετικά μοτίβα των benchmarks. Η χρήση BTB σε συνδυασμό με RAS είχε ενδιάμεση απόδοση και αποδείχθηκε χρήσιμη κυρίως σε εφαρμογές με call/return.