



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
www.cslab.ece.ntua.gr

ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ακ. έτος 2010-2011, 8ο εξάμηνο, Σχολή ΗΜ&ΜΥ

2η ΕΡΓΑΣΙΑ

Τελική Ημερομηνία Παράδοσης: 27 Μαΐου 2012 (δεν θα δοθεί παράταση)

1. Εισαγωγή

Αντικείμενο της εργασίας αποτελεί η μελέτη της επίδρασης διαφόρων παραμέτρων του συστήματος πρόβλεψης διακλαδώσεων στην απόδοση του καθώς και η αξιολόγηση διαφορετικών επιλογών για την υλοποίηση του μηχανισμού πρόβλεψης δεδομένου συγκεκριμένου διαθέσιμου χώρου πάνω στο τσιπ γι' αυτόν το σκοπό.

Για τους σκοπούς της άσκησης αυτής θα χρησιμοποιήσετε το περιβάλλον προσομοίωσης του *Championship Branch Prediction 2006* (<http://taco.cs.utsa.edu/camino/cbp2/>). Το framework αυτό είναι γραμμένο σε C++ και παρέχει τις διάφορες classes που χρειάζονται για την προσομοίωση των συστημάτων πρόβλεψης, από τον μηχανισμό ανάγνωσης των traces των benchmarks μέχρι τις περιγραφές διαφορετικών predictors.

2. Simulation Framework

Το framework αυτό στηρίζεται στη χρήση traces, τα οποία έχουν αποκτηθεί κατά την εκτέλεση benchmarks από τη σουίτα SPEC2000. Για την εξοικονόμηση χώρου, τα traces αυτά δεν περιέχουν όλες τις εντολές του κάθε προγράμματος, παρά μόνο τις εντολές άλματος (conditional branches, direct και indirect branches, function calls, function returns). Πιο συγκεκριμένα, περιλαμβάνουν τις εντολές άλματος που πραγματοποιούνται σε ένα κομμάτι της εκτέλεσης διάρκειας 100M εντολών. Κάθε εγγραφή περιέχει το PC της εντολής άλματος, το είδος της εντολής (opcode και flags), τη διεύθυνση προορισμού καθώς και το αποτέλεσμα, δηλαδή αν το άλμα εκτελέστηκε ή όχι.

Η υλοποίηση του κάθε predictor αποτελείται βασικά από 2 μεθόδους, τις **predict** και **update**. Η πρώτη καλείται να προβλέψει με βάση το PC της εντολής, αν το άλμα θα εκτελεστεί (Taken / Not Taken) καθώς και τον προορισμό του. Προφανώς, το τι προβλέπεται εξαρτάται από το είδος του predictor, αφού υπάρχουν predictors που προβλέπουν μόνο τον προορισμό (π.χ. RAS) ή μόνο το αποτέλεσμα (π.χ. n-bit). Αντίστοιχα, η δεύτερη μέθοδος καλείται να αποθηκεύσει τις πληροφορίες εκείνες που απαιτούνται για τις μελλοντικές προβλέψεις.

Η αρχικοποίηση και η χρήση των predictors γίνεται στο **predict.cc**, το οποίο μπορείτε και να διαμορφώσετε ανάλογα με το ποιους predictors θέλετε να χρησιμοποιήσετε και τι αποτελέσματα

θέλετε να τυπώσετε. Το framework αυτό παρέχει επίσης την υλοποίηση ενός gshare predictor, η οποία μαζί με τις υλοποιήσεις που σας παρέχουμε εμείς μπορούν να λειτουργήσουν ως παράδειγμα για την ανάπτυξη των δικών σας υλοποιήσεων. Μπορείτε να κατεβάσετε τον κώδικα από :

<http://www.cslab.ece.ntua.gr/courses/advcomparch/files/askiseis/branch-pred-src.tar.gz>

Ο κώδικας που σας παρέχουμε περιέχει διορθώσεις / προσθήκες σε σχέση με τον κώδικα που μπορείτε να κατεβάσετε από τη σελίδα του Championship. Για το λόγο αυτό, για τους σκοπούς της άσκησης καλό είναι να χρησιμοποιήσετε αυτόν που σας δίνουμε εμείς.

3. Μελέτη των n-bit predictors

Θα μελετήσετε την απόδοση των n-bit predictors, χρησιμοποιώντας την υλοποίηση που σας παρέχουμε (**nbit_predictor.h**). Ο predictor κατά την αρχικοποίηση του απαιτεί ως όρισμα το μήκος του n-bit counter.

A.1) Διατηρώντας σταθερό τον αριθμό των BHT entries και ίσο με 16K, προσομοιώστε τους n-bit predictors, για $N=1, 2, \dots, 7$ χρησιμοποιώντας τα 6 traces που σας παρέχουμε. Συγκρίνετε τους predictors χρησιμοποιώντας τα direction Mispredictions Per Thousand Instructions (direction MPKI).

A.2) Στο προηγούμενο ερώτημα η αύξηση του αριθμού των bits ισοδυναμούσε με αύξηση του απαιτούμενου hardware, αφού κρατούσαμε σταθερό τον αριθμό των entries του BHT. Διατηρώντας τώρα σταθερό το hardware και ίσο με 32K bits, επαναλάβετε τις προσομοιώσεις για τα 6 benchmarks, θέτοντας $N=1,2,4$ και τον κατάλληλο αριθμό entries. Δώστε το κατάλληλο διάγραμμα και εξηγήστε τις μεταβολές που παρατηρείτε. Ποιον predictor θα διαλέγατε ως την βέλτιστη επιλογή;

4. Μελέτη του BTB

Χρησιμοποιώντας την υλοποίηση του BTB (**btb.h**) που παρέχεται μελετήστε την ακρίβεια πρόβλεψης για τις ακόλουθες περιπτώσεις:

btb_lines	btb_assoc
512	1
256	2
128	4
64	8

Προσομοιώστε για τα benchmarks που παρέχονται και δώστε όπως και πριν τα κατάλληλα διαγράμματα. Υπενθυμίζεται ότι για τον BTB υπάρχουν 2 περιπτώσεις misses. Η πρώτη είναι direction misprediction και η δεύτερη target misprediction στην περίπτωση direction hit. Πώς θα εξηγούσατε τη διαφορά επίδοσης ανάμεσα στις διαφορετικές οργανώσεις; Διαλέξτε την καλύτερη οργάνωση για το BTB.

5. Σύγκριση διαφορετικών predictors

Στο κομμάτι αυτό θα συγκρίνετε τους παρακάτω predictors (οι predictors σε **bold** δε δίνονται):

- **Static Not-Taken.**
- **Static BTFNT (Backward Taken Forward Not Taken).**
- N-bit predictor που επιλέξατε στο A2.
- gshare predictor, θέτοντας κατάλληλο αριθμό entries ώστε το hardware overhead να είναι ίσο με 32K (στον κώδικα που σας δίνουμε κάθε entry του PHT έχει μήκος 2 bits).
- **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 = ZΥπολογίστε 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 = 4, 8Υπολογίστε 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 (4, 8 bits) θεωρείται αμελητέο.
- **Tournament Hybrid predictors** (βλ. διαφάνειες μαθήματος) με τα εξής χαρακτηριστικά:
 - Ο meta-predictor M είναι ένας 2-bit predictor με 512 entries (το overhead του μπορείτε να το αγνοήσετε στην ανάλυση σας).
 - Οι P₀, P₁ μπορούν να είναι 2-bit, gshare ή global-history predictors (με 2-bit PHT counters και BHR length = 2 ή 4).
 - Οι P₀, P₁ έχουν overhead 16K ο καθένας.

Προσομοιώστε και συγκρίνετε την ακρίβεια πρόβλεψης (direction MPKI) των συστημάτων για τις παραπάνω περιπτώσεις. Δώστε τα κατάλληλα διαγράμματα. Ποιον predictor θα διαλέγατε τελικά να υλοποιήσετε;

Επιλογή benchmarks

Η ανάθεση των benchmarks έχει γίνει με βάση το τελευταίο ψηφίου του AM σας.

(i) <http://www.cslab.ece.ntua.gr/courses/advcomparch/files/askiseis/team03.tar.gz>

για AM που τελειώνουν σε 0,1,2,3

(ii) <http://www.cslab.ece.ntua.gr/courses/advcomparch/files/askiseis/team46.tar.gz>

για AM που τελειώνουν σε 4,5,6

(iii) <http://www.cslab.ece.ntua.gr/courses/advcomparch/files/askiseis/team79.tar.gz>

για AM που τελειώνουν σε 7,8,9

Παραδοτέο θα είναι ένα ηλεκτρονικό κείμενο (pdf, doc ή odt) που θα περιέχει την αναφορά με τα διαγράμματα και τα συμπεράσματά σας, καθώς και τον κώδικα που υλοποιήσατε. Μη ξεχάσετε να αναφέρετε τα στοιχεία σας (Όνομα, Επώνυμο, AM). Η άσκηση θα παραδοθεί μόνο ηλεκτρονικά στην ιστοσελίδα:

<http://www.cslab.ece.ntua.gr/courses/advcomparch/submit>

Δουλέψτε ατομικά. Έχει ιδιαίτερη αξία για την κατανόηση του μαθήματος να κάνετε μόνοι σας την εργασία. Μην την αντιγράψετε απλά από άλλους συμφοιτητές σας.