



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 2008-2009, 8ο εξάμηνο, Σχολή ΗΜ&ΜΥ

2η ΕΡΓΑΣΙΑ

Τελική Ημερομηνία Παράδοσης: 3 Ιουνίου 2009 (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' αυτόν το σκοπό.

Για τους σκοπούς της άσκησης αυτής θα χρησιμοποιήσετε το περιβάλλον προσομοίωσης του *Championship Branch Prediction 2006* (<http://cava.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, 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. Προφανώς, το τι προβλέπεται εξαρτάται από το είδος του predictor, αφού υπάρχουν predictors που προβλέπουν μόνο προορισμό (π.χ. RAS) και άλλοι που προβλέπουν 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 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 το κατεβάσετε από :

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

Ο κώδικας που σας δίνουμε περιέχει διορθώσεις / προσθήκες σε σχέση με τον κώδικα που μπορείτε να κατεβάσετε από τη σελίδα του Championship. Για 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 32K, 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

Προσομοιώστε και για τα 6 benchmarks 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:

- **Static Not-Taken**
- **Static Taken**

- **Static BTFNT (Backward Taken Forward Not Taken)**
- αυτός που βρήκατε στο A2
- Ένας BTB με hardware overhead 32K bits (το PC και το target έχουν μέγεθος 32bits).
- **Local-History two-level predictor** (βλ. διαφάνειες μαθήματος) με τα εξής χαρακτηριστικά :
 - PHT entries = 4096
 - PHT n-bit counter length = 2
 - BHT entries = X
 - BHT entry length = Z
 Υπολογίστε το Z ώστε το απαιτούμενο hardware να είναι σταθερό και ίσο με 32K, για X=2048 και 4096 .
- gshare predictor, θέτοντας κατάλληλο αριθμό entries ώστε το hardware overhead να είναι ίσο με 32K.

Οι predictors σε **bold** δε δίνονται και πρέπει να υλοποιηθούν.

Προσομοιώστε για τα 6 benchmarks και συγκρίνετε την ακρίβεια πρόβλεψης (direction MPKI) των συστημάτων για τις παραπάνω 8 περιπτώσεις. Δώστε τα κατάλληλα διαγράμματα. Ποιον predictor θα διαλέγατε να υλοποιήσετε;

Παραδοτέο της άσκησης 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. Στο ηλεκτρονικό κείμενο να αναφέρετε στην αρχή τα στοιχεία σας (Όνομα, Επώνυμο, ΑΜ).

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 προσπαθήσετε να την αντιγράψετε απλά από άλλους συμφοιτητές σας.