



Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Μάθημα: Προχωρημένα Θέματα Αρχιτεκτονικής
Εργασία Ακ. Έτους 2018-2019

| Ονοματεπώνυμο | AM |
|------------------------|------|
| Παναγιώτης Σταυρινάκης | 6217 |
| Βασίλειος Αϊβαλιώτης | 5987 |

Project 2. Replacement Policies for Last Level Caches

Σε αυτό το project καλούμαστε να αναπτύξουμε έναν δυναμικό αλγόριθμο αντικατάστασης για επίπεδα της κρυφής μνήμης. Ο αλγόριθμος έχει την ικανότητα να προσαρμόζεται δυναμικά στα working sets των εφαρμογών που εκτελούνται. Πιο συγκεκριμένα, ο αλγόριθμος επιλέγει δυναμικά να αντικαταστήσει είτε το LRU (Least Recently Used) είτε το MRU (Most Recently Used) line ενός cache set. Η υλοποίηση βασίζεται στο paper των M. K. Qureshi et al[1]. Ο εξομοιωτής που χρησιμοποιούμε είναι ο HotLeakage ο οποίος βασίζεται στον εξομοιωτή SimpleScalar.

Οι μόνες αλλαγές που πραγματοποιήθηκαν βρίσκονται μέσα στα αρχεία **cache.h** & **cache.c**. Στο **header** αρχείο ορίζουμε μεταβλητές που θα χρειαστεί η cache για να υπολογίζει μετρήσεις όπως είναι τα cache misses. Στο αρχείο **cache.c** γίνεται αρχικοποίηση των μεταβλητών αυτών στις συναρτήσεις `clear_cache_stats()` & `cache_create()` και συνεχίζουμε με την υλοποίηση του αλγορίθμου αντικατάστασης στην LRU policy case. Η μεταβλητή `repl` περιλαμβάνει το line που βρίσκεται στην ουρά του cache set και με την χρήση του μηχανισμού Dynamic Insertion Policy (DIP) επιλέγουμε δυναμικά το καταλληλότερο Insertion Policy (BIP ή traditional LRU) για το πρόγραμμα που εκτελείται την στιγμή αυτή. Ο αλγόριθμος αυτός λειτουργεί για **data caches επιπέδου 1** και αριθμό σετ μεγαλύτερο του 64, σε διαφορετική περίπτωση η αντικατάσταση εκτελείται όπως ήταν προγραμματισμένη να

εκτελεστεί. Για την επιλογή του κατάλληλου Insertion Policy ορίζουμε τον ίδιο αριθμό αφοσιωμένων σετ ανάμεσα στις δύο πολιτικές, συγκεκριμένα 32 σετ για κάθε πολιτική, και απόσταση αφοσιωμένων σετ ίδιας πολιτικής τέτοια ώστε να επεκτείνονται σε όλο το μέγεθος της cache. Εάν το σετ που τροποποιούμε είναι LRU_dedicated εκτελείται η LRU πολιτική, εάν το σετ είναι BIP_dedicated εκτελείται BIP πολιτική, αλλιώς αν είναι follower set τότε ο Policy Selector(PSEL) επιλέγει ποια από τις δύο θα εκτελεστεί αναλόγως τον αριθμό των misses κάθε πολιτικής. Πιο συγκεκριμένα, αν η LRU έχει περισσότερα misses εκτελείται BIP πολιτική και αντιστρόφως. Η LRU πολιτική τοποθετεί το line του cache set στην MRU θέση ενώ η BIP πολιτική στην LRU θέση. Επιπλέον, για να διαθέτει η BIP μηχανισμό γήρανσης τοποθετεί το line στην MRU θέση ανά 32 misses της συγκεκριμένης πολιτικής.

Με αυτόν τον αλγόριθμο καλύπτουμε τις περιπτώσεις των memory-intensive workloads που έχουν working set μεγαλύτερο από το διαθέσιμο μέγεθος της cache και οδηγούν σε συνεχή paging και paging faults. Συνεχίζουμε όμως να καλύπτουμε και τα spatial/temporal locality friendly προγράμματα χάρη στον DIP μηχανισμό. Παρακάτω βλέπουμε πως επηρεάζεται το miss rate από τον αλγόριθμό μας για κάθε benchmark χρησιμοποιώντας data cache 512 set και 2 set associativity. Πρώτα για την εκτέλεση 10 εκατομμυρίων max instances και περιπτώσεις διάφορων BIPCTR και έπειτα για την εκτέλεση 100 εκατομμυρίων max instances και BIPCTR = 32.

Max:inst = 10000000

Spec2000.176.gcc.ref.166.eio

| | Trad.LRU | DIP |
|-------------|----------|--------|
| BIPCTR = 16 | 0.0167 | 0.0164 |
| BIPCTR = 32 | 0.0167 | 0.0165 |
| BIPCTR = 64 | 0.0167 | 0.0165 |

Spec2000.164.gzip.ref.graphic.eio

| | Trad.LRU | DIP |
|-------------|----------|--------|
| BIPCTR = 16 | 0.0606 | 0.0606 |
| BIPCTR = 32 | 0.0606 | 0.0606 |
| BIPCTR = 64 | 0.0606 | 0.0606 |

Spec2000.197.parser.ref.ref.eio

| | Trad.LRU | DIP |
|-------------|----------|--------|
| BIPCTR = 16 | 0.0016 | 0.0017 |
| BIPCTR = 32 | 0.0016 | 0.0017 |
| BIPCTR = 64 | 0.0016 | 0.0017 |

Spec2000.253.perlbmk.ref.makerand.eio

| | Trad.LRU | DIP |
|-------------|----------|--------|
| BIPCTR = 16 | 0.0018 | 0.0016 |
| BIPCTR = 32 | 0.0018 | 0.0016 |
| BIPCTR = 64 | 0.0018 | 0.0016 |

Παρατηρούμε πως για αυτόν τον όγκο δεδομένων ανεξαρτήτως το μέγεθος του BIPCTR παρουσιάζεται μηδαμινή βελτίωση στο miss rate.

Max:inst = 10000000

| | Trad.LRU | DIP |
|---------------------------------------|----------|--------|
| Spec2000.176.gcc.ref.166.eio | 0.0148 | 0.0143 |
| Spec2000.164.gzip.ref.graphic.eio | 0.0417 | 0.0417 |
| Spec2000.197.parser.ref.ref.eio | 0.0036 | 0.0037 |
| Spec2000.253.perlbmk.ref.makerand.eio | 0.0018 | 0.0016 |

Παρατηρούμε πως κάθε benchmark έχει διαφορετικές απαιτήσεις. Δύο από τα benchmarks παρουσιάζουν σημαντική μείωση του miss rate (Spec2000.176.gcc.ref.166.eio, Spec2000.253.perlbmk.ref.makerand.eio), ένα ελαφρά αύξηση (Spec2000.197.parser.ref.ref.eio) και ένα καμία αλλαγή (Spec2000.164.gzip.ref.graphic.eio).