

Tölvutækni og Forritun Verkefni 2

Ragnar Björn Ingvarsson, rbi3

Yi Hu, yih2

10. nóvember 2024

I Skrifu hermi fyrir skyndiminni

```
void initCache()
{
    cache = (cache_set_t*) malloc(S*sizeof(cache_set_t));
    for (int i = 0; i < S; i++) {
        cache[i] = (cache_line_t*) malloc(E*sizeof(cache_line_t));
    }
}

void freeCache()
{
    free(cache);
}

void accessData(mem_addr_t addr)
{
    long set = (addr >> b) & ((1ULL << s) - 1);
    long tag = addr >> (s+b);

    int has_hit = 0;
    int insert_place = -1;
    int lowest_value = policy_code == 1 ? cache[set][0].lru : cache[set][0].fifo;
    int lowest_index = 0;

    for (int i = 0; i < E; i++) {

        cache_line_t line = cache[set][i];

        if (policy_code == 1) {
            if (lowest_value > line.lru) {
                lowest_value = line.lru;
                lowest_index = i;
            }
        } else if (policy_code == 2) {
            if (lowest_value > line.fifo) {
                lowest_value = line.fifo;
                lowest_index = i;
            }
        }

        if (line.valid && line.tag == tag) {
            line.lru = ++lru_counter;
            hit_count++;
            has_hit = 1;
            cache[set][i] = line;
        }
    }
}
```

```

        break;
    }
    if (!line.valid) {
        insert_place = i;
    }
}
if (!has_hit) {
    miss_count++;
    if (insert_place == -1) {
        insert_place = policy_code == 3 ? rand() % E : lowest_index;
        eviction_count++;
    }
    cache_line_t line_to_be_added;
    line_to_be_added.tag = tag;
    line_to_be_added.fifo = fifo_counter++;
    line_to_be_added.lru = lru_counter++;
    line_to_be_added.valid = 1;
    cache[set][insert_place] = line_to_be_added;
}
}

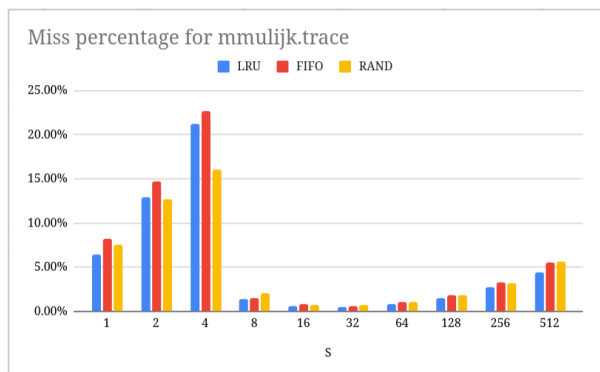
```

```

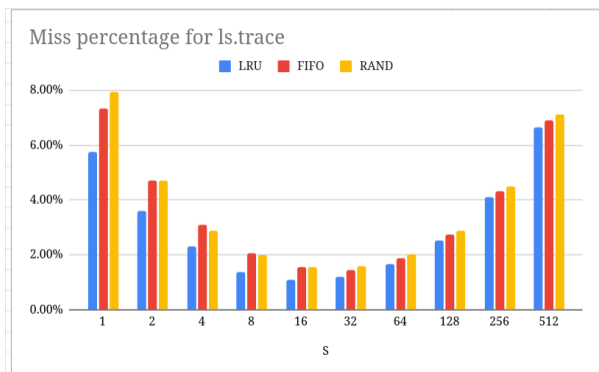
ragnar@gamer ~/school/tolfor/v2/verk2 ◆ ./csim -S 16 -B 64 -E 4 -p LRU -t traces/transpose.trace
hits: 233 misses: 5 evictions: 0
miss ratio: 2.10%
ragnar@gamer ~/school/tolfor/v2/verk2 ◆ ./csim -S 16 -B 64 -E 4 -p LRU -t traces/yi.trace
hits: 6 misses: 3 evictions: 0
miss ratio: 33.33%
ragnar@gamer ~/school/tolfor/v2/verk2 ◆ ./csim -S 16 -B 64 -E 4 -p LRU -t traces/yi2.trace
hits: 16 misses: 1 evictions: 0
miss ratio: 5.88%
ragnar@gamer ~/school/tolfor/v2/verk2 ◆ █

```

II Besta uppsetning á skyndiminni



Mynd 1: mmulijk.trace



Mynd 2: ls.trace

II.i Hver er munurinn á bestu uppsetningu á milli rakningaskráanna?

Við sjáum af súluritunum að það er töluverður munur á keyrslu milli rakningaskráanna. Fyrir mmulijk.trace kemur langverst út að hafa S mjög lítið, en svo er langbest að hafa það einhversstaðar í miðjunni um 16-32, þar sem það virðist ekki vera betra að vera með of mörg mengi.

Með ls.trace þá sést að það myndast dalur í ritinu, svo það er best sð vera aftur, í kring um 16-32.

Munurinn hér er væntanlega vegna þess að þar sem ls.trace er mjög óreglulegt og út um allt, þá er best að setja upp minnið okkar þannig að við séum með jafnan fjölda mengja og stærð línanna, þar sem ef við förum of langt í aðra hvora áttina þurfum við að leita svo djúpt í staðinn fyrir að leita grunnt á báða ása. Þetta hjálpar að mynda eins mikið "bil" á milli gagnanna svo við lendum ekki í skellum.

mmulijk.trace er hins vegar mjög reglulegt og þess vegna viljum við notfæra okkur að gögnin eru alltaf hlið við hlið. Þess vegna er svo mikilvægt að hafa ekki of stórar línur því þá flækjast gögn kannski saman í línur og valda fleiri skellum, svo við viljum hafa nógu litlar línur til að bara halda utan um gögnin sem við erum nú þegar að vinna með en nógu stórar svo við séum heldur ekki að henda út gögnum sem við erum að nota.

Svo besta uppsetningin fyrir mmulijk er $S = 32$ á LRU og fyrir ls er $S = 16$ á LRU.

II.ii Hvernig kemur uppsetning Intel Core i7 út í hermunum?

Við sjáum að þetta er sama uppsetningin og við vorum að prófa og það sést á súluritunum að þetta er ekki besta uppsetningin hvorki fyrir mmulijk.trace né ls.trace en hún er samt alls ekki sú versta og kemur bara mjög vel út í báðum tilfellum.

Ég myndi þá halda að ástæðan fyrir að velja svona uppsetningu sé vegna þess að hún sé kannski best ef við prófum yfir fleiri tilfelli af rakningaskráum og líka að það sé kannski betra fyrir orkusparnað eða pláss eða kostnað og svoleiðis.