CSE240A Project 2 Cache Tuning
Hao Zhuang
hazhuang@eng.ucsd.edu
A53029734 UCSDCSE

1 Methodology and design decisions.

1.1 Replacement strategy is Least Recently Used (LRU):

It is based on the value of Cache.accessed[index][set_id], where Cache is L1, victim or L2. index is based on the address, set_id is the value we need to choose for replacement.

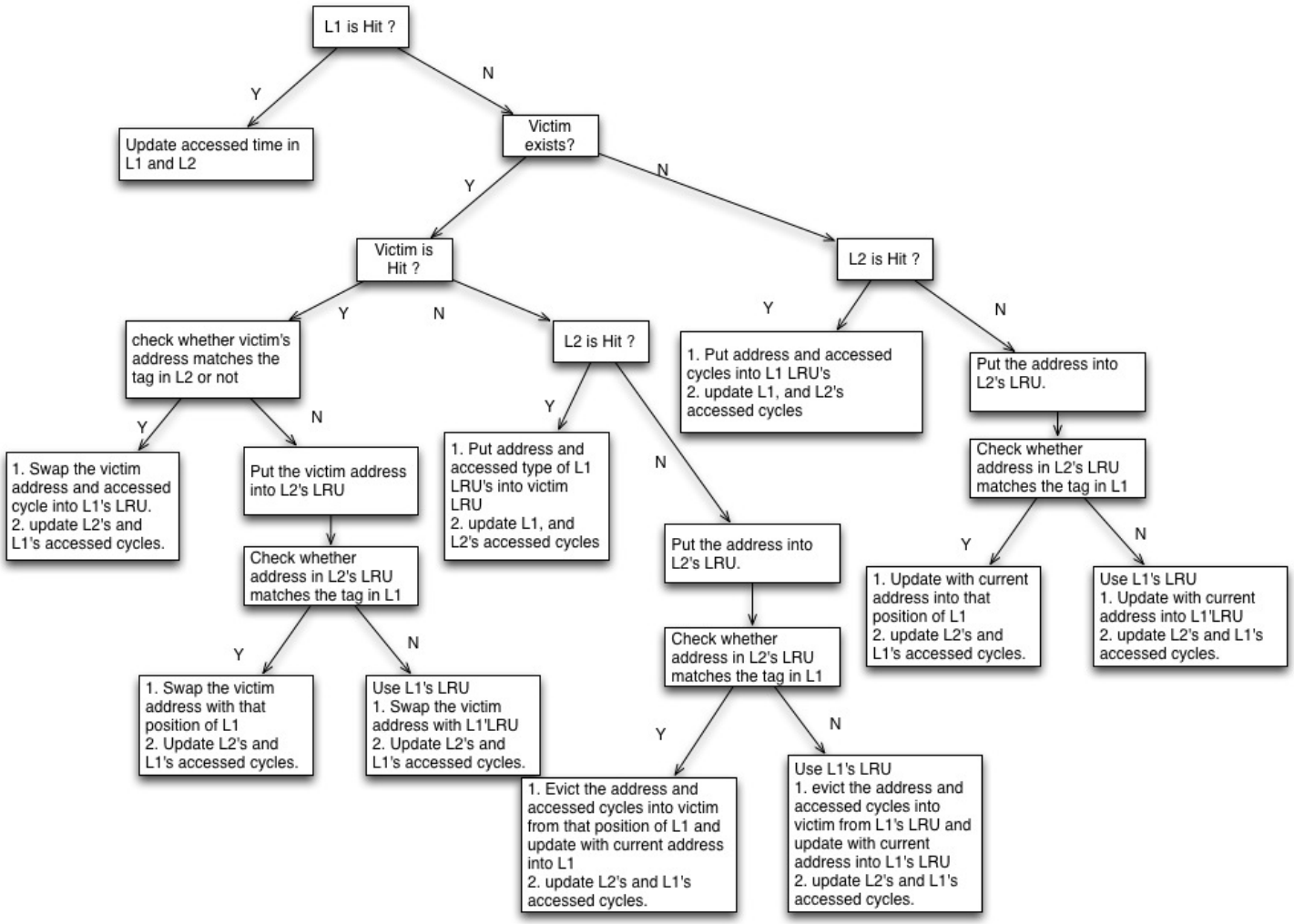
For one index, the smaller value of Cache.accessed[index][set_id], the less recently bit it is. Therefore, the smallest number is the place of LRU.

1.2 L1.isHit(ldst, address, cycles)

Compare the tag, if there is the same tag of address as tag of Cache.tag[index][set_id], return true, which is hit. But before that, we should update the Cache.accessed[index][set_id] with current "cycles".

Otherwise, return false.

1.3 My design contains the flow of accessCache(ldst, address, cycles):



Pseudo Code:

```
---------------------------------------------------------------------------------------------------------------
 L1Accesses++;
TotalMemRef++;
 if L1.isHit() == true, then
       update the accessed cycle L2; // inclusion property of L1 and L2, and L1.isHit has
                                          // update the L1.access already.
       return L1.getHitLatency;
 else // L1 miss
        L1Misses++;
       if victim cache exists
          VictimeAccesses++;
          if victim.isHit() == true // victim hit
               update the corresponding values (address/tag, accessed cycles) into L2's LRU,
               which has address/tag l2a;
               check whether l2a has tag in L1; // this is for the inclusion property of L1 and L2
               if l2a's tag in L1, then
                    swap corresponding values between L1 and victim;
               else
                    find L1's LRU, and swap corresponding values between L1 and victim;
               update the accessed time in L1 and L2;
               return Victim.getHitLatency();
          else // victim miss
               VictimeMisses++ ;
               L2accesses++;
               if L2.isHit() == true
                    update corresponding values into L1's LRU, and put evictions into victim's LRU;
                    Return L2.getHitLatency() + L1L2Transfer;
               else L2Misses++;
     else // victim cache does not exist
          // try L2
          L2accesses++;
          if L2.isHit() == true
               update corresponding values into L1's LRU    ;
               Return L2.getHitLatency() + L1L2Transfer;
          else L2Misses++;
          // No return means L1, victim, L2 miss;
update corresponding values from memory into L2's LRU, which has address/tag l2a;
check whether l2a has tag in L1, // this is for the inclusion property of L1 and L2;
if l2a's tag in L1, then
     update corresponding values into L1;
     if there is victim
          put evictions into victim's LRU;
else
```

find the L1 LRU, update corresponding values into L1;
if there is victim
put evictions into victim's LRU;
update the accessed time in L1 and L2;
Return memoryAccessTime; // which is 350 cycles here

---

2 Two baselines for the test mode

2.1 Baseline one

	art.trace.gz	swim.trace.gz
CPI	35.18	15.67
AMAT	89.67	48.54
L1 Miss Rate	0.25	0.18
Victim Local Miss Rate	1.00	1.00
L2 Local Miss Rate	0.98	0.70
L2 Global Miss Rate	0.25	0.13

2.1.1 gunzip –c art.trace.gz | cache - sim - t 16384 2 32 512 4 32 262144 8 32

Cache Config:
L1 size=16 KB, assoc=2, LS=32 bytes, lat=3 cyc
Victim size=512 bytes, assoc=4, LS=32 bytes, lat=1 cyc
L2 size=256 KB, assoc=8, LS=32 bytes, lat=24 cyc
Total size of all caches 279040 bytes
Total CPI=35.18
Average memory access time=89.67
L1 Miss Rate=0.25
Victim Local Miss Rate=1.00
L2 Local Miss Rate=0.98
L2 Global Miss Rate=0.25

2.1.2 gunzip –c swim.trace.gz | cache - sim - t 16384 2 32 512 4 32 262144 8 32

Cache Config:
L1 size=16 KB, assoc=2, LS=32 bytes, lat=3 cyc
Victim size=512 bytes, assoc=4, LS=32 bytes, lat=1 cyc
L2 size=256 KB, assoc=8, LS=32 bytes, lat=24 cyc
Total size of all caches 279040 bytes
Total CPI=15.76
Average memory access time=48.54
L1 Miss Rate=0.18
Victim Local Miss Rate=1.00

L2 Local Miss Rate=0.70
L2 Global Miss Rate=0.13

2.2 Baseline 2 (For reference, The miss rates and CPI for the following baseline cache design: L1 (16K 2-way 32-byte blocks), victim (16 32-byte lines, 4-way), L2 (256K, 4 - way, 32 - byte).)

	art.trace.gz	swim.trace.gz
CPI	35.18	15.67
AMAT	89.67	48.54
L1 Miss Rate	0.25	0.18
Victim Local Miss Rate	0.00 (no victim)	0.00 (no victim)
L2 Local Miss Rate	0.98	0.70
L2 Global Miss Rate	0.25	0.13

2.2.1 gunzip -c art.trace.gz | cache-sim -t 16384 2 32 16 4 32 262144 8 32

Cache Config:
L1 size=16 KB, assoc=2, LS=32 bytes, lat=3 cyc
Victim size=16 bytes, assoc=4, LS=32 bytes, lat=0 cyc
L2 size=256 KB, assoc=8, LS=32 bytes, lat=24 cyc
Total size of all caches 278544 bytes
Total CPI=35.18
Average memory access time=89.67
L1 Miss Rate=0.25
Victim Local Miss Rate=0.00
L2 Local Miss Rate=0.98
L2 Global Miss Rate=0.25

2.2.2 gunzip -c swim.trace.gz | cache-sim -t 16384 2 32 16 4 32 262144 8 32

Cache Config:
L1 size=16 KB, assoc=2, LS=32 bytes, lat=3 cyc
Victim size=16 bytes, assoc=4, LS=32 bytes, lat=0 cyc
L2 size=256 KB, assoc=8, LS=32 bytes, lat=24 cyc
Total size of all caches 278544 bytes
Total CPI=15.76
Average memory access time=48.54
L1 Miss Rate=0.18
Victim Local Miss Rate=0.00
L2 Local Miss Rate=0.70
L2 Global Miss Rate=0.13

3 For the real mode (-r)

We use the parameters hard coded in function

```
-----------------------------------------------
c->setL1Attributes( 16*1024, 1, 512);
c->setVictimAttributes(8*1024, 1, 512);
c->setL2Attributes(276, 16, 512);
c->setEvictionPolicy(0);
-----------------------------------------------
```

The parameters derive from the idea that, in L1 and victim, I focus more on the hit time, for L2 I focus more on the miss rate.

	art.trace.gz	swim.trace.gz
CPI	7.30	3.37
AMAT	16.54	7.79
L1 Miss Rate	0.21	0.01
Victim Local Miss Rate	0.44	1.00
L2 Local Miss Rate	0.53	0.38
L2 Global Miss Rate	0.03	0.00

3.1.1 gunzip -c art.trace.gz | cache-sim -r

```
Cache Config:
   L1 size=16 KB, assoc=1, LS=512 bytes, lat=6 cyc
   Victim size=8192 bytes, assoc=1, LS=512 bytes, lat=5 cyc
   L2 size=276 KB, assoc=16, LS=512 bytes, lat=37 cyc
  Total size of all caches 307200 bytes
Total CPI=7.30
Average memory access time=16.54
L1 Miss Rate=0.12
Victim Local Miss Rate=0.44
L2 Local Miss Rate=0.53
L2 Global Miss Rate=0.03
```

3.1.2 gunzip -c swim.trace.gz | cache-sim -r

```
Cache Config:
   L1 size=16 KB, assoc=1, LS=512 bytes, lat=6 cyc
   Victim size=8192 bytes, assoc=1, LS=512 bytes, lat=5 cyc
   L2 size=276 KB, assoc=16, LS=512 bytes, lat=37 cyc
  Total size of all caches 307200 bytes
Total CPI=3.37
Average memory access time=7.79
L1 Miss Rate=0.01
Victim Local Miss Rate=1.00
```

L2 Local Miss Rate=0.38
L2 Global Miss Rate=0.00