LAB 4 REPORT

1003839484 Yue Huang
1003936592 Jiayi Zhao

Q1

To verify the implementation of the nextline prefetcher, we build a loop. Every iteration of the loop accesses an address of the array. If the distance between two accesses is the cache block size, we get a dl1 miss rate of 1.56%. If the distance between two accesses is two times the block size, we get a dl1 miss rate of 99.21%.
The configuration file for this is the same as the provided file.

Q2

To verify the implementation of the stride prefetcher, we build a loop. Every iteration of the loop accesses an address of the array. If the distance between two accesses is fixed 2 times the cache block size, we get dl1 miss rate of 0.01%. If the distance between two accesses changes every iteration, we get dl1 miss rate of 99.40%.
The configuration file for this is the same as the provided file.

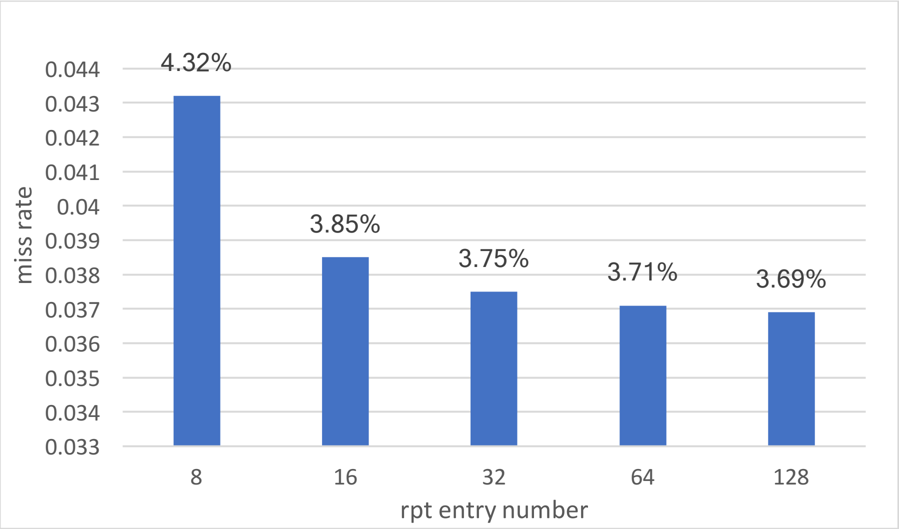
Q3

Average access time =
T access(L1Data) + L1miss rate * (T access(L2Data) + L2miss rate * T hit (Memory))

Config	L1 Miss Rate	L2 Miss Rate	Average access time
baseline	4.16%	11.40%	1.89024
next-line	4.19%	8.38%	1.770122
stride	3.85%	5.78%	1.60753

Q4

Entries in RPT	8	16	32	64	128
Dl1 Miss rate	4.32%	3.85%	3.75%	3.71%	3.69%



The graph shows that as we increase the entries in RPT, the miss rate of the dl1 cache is dropped. As the entries in RPT become bigger, the dl1 miss rate tends to converge.

Q5

We could add prefetcher accuracy and prefetcher coverage in sim-cache to study the performance of the prefetcher.

Prefetcher accuracy is to measure of all the prefetches, how many prefetches are actually useful.

Prefetcher coverage is to measure of all the misses, how many of the misses are covered by the prefetcher.

We could count the number of useful prefetches and use it to calculate prefetcher accuracy and prefetcher coverage.

Q6

First we run a loop with same stride every iteration, we get a dl1 miss rate of 0.01%.

Then we try run the following loop:

```
      for (j=0;j<8192&&i<size;j++){
            a=a+test[i];
            if(j%2==0){
                  i=i+BSIZE+1;

            else
                  i=i+BSIZE*2+1;
            }
```

Our correlation prefetcher has an entry of 8192, so we set the loop iteration to be 8192 for simplicity of our test. If we run the loop once, it sets up the tag and address of correlation table in the prefetcher. There are 8233 dl1 misses and the dl1 miss rate is 67.49%. If we run the loop 10 times, we get a dl1 miss rate of 32.33%.

The configuration file used for mbq6.c is the same as cache-lru-open.cfg

Open-ended prefetcher description

The open-ended prefetcher combined stride prefetcher and miss prefetch table (an idea from correlation prefetcher). The prefetch address can be generated by the stride prefetcher part and the miss prefetch table concurrently. We changed the stride prefetcher as follows, only when the state is Steady we use the prefetch address given by stride prefetcher. If the state is otherwise, we use the prefetch address given by miss prefetch table. We update our miss prefetch table only when a miss happens. We index by its previous access address and save the miss address in the table.
The RPT entries for our stride prefetcher is 32. The entries for our miss prefetch table is 8192. The tag is 17 bits and address is 32 bits. The table can be implemented as a directed map cache

Using CACTI, we get results for RPT with 32 entries:
Access time(ns): 0.285431
Cycle time (ns) : 0.103623
Cache height * width(mm): 0.0316458 * 0.0307544

Using CACTI, we get results for miss prefetch table if table entry is 8192 as follows:
Access time(ns): 0.704074
Cycle time (ns) : 0.391299
Cache height * width(mm): 0.257616 * 0.303171

If miss prefetch table entry is 1024:
Access time(ns): 0.425516
Cycle time (ns) : 0.183885
Cache height * width(mm): 0.0754907 * 0.159559

Therefore, if we choose RPT with 32 entries and miss prefetch table with 1024 entries, the access time of the prefetch address can be within 0.28 ns if a PC is in steady state and 0.42ns otherwise. The area overhead is reasonable considering the whole chip size. If we use miss prefetch table of 1024 entries, we get average dl1 miss rate of 1.88% across the three microbenchmarks.
If we choose RPT with 32 entries and miss prefetch table with 8192 entries. The access time of prefetch address is 0.28ns if RPT entry of the PC is in steady state and 0.70ns if a PC is not in steady state. Both the access time and area overhead can be too big. However, we get an average dl1 miss rate of 1.75% across the three microbenchmarks.