Lab4:CacheLab

Junhao Yu

May 23, 2022

1 Problems I Met and How I Solved Them

1.1 simulator

At first, I didn't quite understand how to use the function "getopt". After I carefully read the handout and searched the usage of "getopt", I figured out how it works and wrote a function called "get_parameter" to pack it.

Then, I was quite obfuscated by the offset part. I didn't realize it doesn't matter since we only care about the index and tag, which truly determine where the address store. Then I use a loop to get every char of offset part but somehow it didn't work. Finally, I used a variable "bin" store it, since it's really useless.

To materialize the cache, there is really nothing difficult. I only need to do as Proffessor Lu taught in class and used a variable called "Recent_Use" to store last time it had been used.

1.2 transpose

At first, I recalled that Proffessor Lu mentioned blocking method to minimize the miss counts, but using it still causes 300+ misses. Then, I realiezd that diagonal elements actually caused double misses(perhaps because A and B's tags are the same?). So I used 8 local variables to store a cache line and solved the problem.

64×64 matrix really caused me problem. I tried several different block sizes, but with little progress. Then, after disscussing this problem with classmates, I realized that the main problem was that I hadn't took full advantage of a cache line. Then, I used a "detour" method to use a cache line completely. Then, I copied the "detour" part to the right part. By using this detouring method, I finally solved the problem.

The last matrix was actully easier, since there is no so-called diagonal. Hence, I only need to try different block sizes to minimize the misses.

2 What I Have Learned

1. Get a better knowledge of how the cache works and the LRU strategy.
2. Get a better knowledge of how to take full advantage of cache, making my programme run faster.