Note 18: Multilevel Caches

Kevin Moy

Abstract

This note attempts to finalize our focus on caches by analyzing the tradeoff of different parameter choices for different cache designs. First, we will cover some metrics for how "good" our cache is performing- i.e. measurements of cache quality. Then we can generalize our cache design to a full memory hierarchy of multilevel caches. Finally, we can look at how, in practice, different tradeoffs prioritize different things, and see some examples of current cache designs.

1 Average Memory Access Time

Let's now move on to measuring cache performance, which depends entirely on finding data from lower levels of memory.

For every memory access, we first need to search the first cache. This is usually very quick. If we miss, however, we must continue to memory. We want to represent the *average* memory access time for ALL accesses (hits and misses) with this formula:

$$\text{AMAT} = \text{Hit Time} + \text{Miss Rate} * \text{Miss Penalty}$$

The **hit time** is the time to check if our requested block is in the cache; it is also called the *tag check* time. The **miss rate** is the proportion of accesses that result in misses. Finally, the **miss penalty** is the extra time cost by going to memory and loading our block from there.

Of course, we ideally want to make our AMAT as low as possible for any given cache. Since AMAT is a general statistic, knowing our AMAT is useful for two main things:

1. Examining the quality of various cache designs (parameters).
2. Improving code performance

2 Hit Rate

Let's first focus on hit time. We know hit time consists of time taken to go to the cache + time to check tags in that cache. What could affect hit time?

- **Smaller cache size: Hit time decreases**, since we have less tags to check.
- **Associativity increases: Hit time increases**, since we have less sets with the same number of blocks, so more blocks per set, which means more tag comparisons needed and more time taken to do so. The worst, hit time, then, comes from a fully associative cache- where we must search through everything.

3 Miss Rate

What could affect miss rate?

- Specific access pattern. Ideally we want to access memory sequentially instead of at random to exploit spatial locality!
- Larger block sizes or caches: bring in more data- again, more spatial locality. But we don't want to make it too big- miss penalty gets larger.

There are three main types of cache misses:

- **Compulsory Miss**: First time seeing block, unavoidable miss.
- **Capacity Miss**: We access more blocks in our program than our cache allows- not even full associativity will hold all blocks.
- **Conflict Miss**: Multiple addresses map to same cache slot or set, which causes miss + need for replacement. If we increase cache associativity (less sets), block size increases so this becomes less of a problem.

Let's analyze some ways to fix these misses.

A way to fix compulsory misses is to increase block size, which brings in more potential (first-time) memory addresses in a bigger block. However, this also increases miss penalty and also potentially miss rate.

A way to improve capacity misses are to simply increase cache size.

Finally, conflict misses can be solved by increasing associativity, as we stated above. More blocks per set. However, also note higher associativity means higher hit time (more tag checks).

4 Miss Penalty

Finally, we have miss penalty, which is the cost from going from the cache to some lower memory level. Usually this is main memory, but with **multilevel caches** it can be going from a cache to another LOWER level cache!

Contributing factors to our miss penalty include the **size** of the memory hierarchy, which means more tag comparisons at each (extra) level of the hierarchy (basically, add hit time of each level). However, do note this total time (on average) is actually still less than going directly to memory. Additionally, smaller block size means less shit to load back and thus lower miss penalty.

Now we'll cover multilevel cache designs, and we'll see that the AMAT calculation is still gonna be the same for all of them.

5 Multilevel Caches

The big goal, of course, is the **minimize AMAT** for a given cache design. To do this, we ideally want to minimize hit time, miss rate, and miss penalty. However, there are unfortunately tradeoffs between all three.

A DM cache yields smallest hit time- only one tag check is needed. A fully associative cache yields smallest miss rate, since we can stick whatever we want and wherever into our cache, only needing to replace whenever the entire cache is full.

Caches with smaller block sizes lower miss penalty, but also don't exploit spatial locality as much, so miss rate is actually heightened a bit.

We now have room for more caches. Let's add some more! L2, L3 caches, like the secondary on an NFL team, can serve as "backup" caches. **Multilevel caching reduces miss penalty**: now instead of going to main memory upon a miss, we go to the next cache level- a LOT lower cost.

Let's see a multilevel cache diagram which displays the access mechanism for multilevel caching. Now the write-allocate policy becomes even more emphasized here: if I miss at L1 but hit at L2, we want to write back to L1 as well.

How do we want to organize these cache levels? The L1 cache focuses on minimizing hit time, i.e. super fast access. The miss penalty for L1 is significantly reduced by the presence of L2 (instead of main memory). So a high(er) miss rate is actually not as bad here. So we want a **small and fast L1 cache**. So a DM cache might be a good L1 cache.

L2 and L3 caches, on the other hand, are super close to memory- so we want those to focus on minimizing miss rate (maximizing hit rate). So these should be larger N-way associative caches, for large N.

So now we can update the AMAT equation to specify *cache level*:

$$\text{AMAT} = \text{L1 HT} + \text{L1 MR} * \text{L1 MP}$$

But now L1 miss penalty is the time taken to traverse the rest of the memory hierarchy, which is actually represented as the AMAT for the *L2 cache*:

$$\text{L1 MP} = \text{L2 HT} + \text{L2 MR} * \text{L2 MP}$$

We can generalize this for miss penalty at an arbitrary level:

$$\text{MP}_i = \text{AMAT}_{i+1}$$

Of course, the furthest we can go is to calculate the AMAT for main memory.

6 Local and Global Miss Rates

A **local miss rate** is the proportion of misses for a **specific** cache level. For example, the L2 local miss rate would be the number of L2 misses / L1 misses, since the number of L1 misses will be the same number of accesses to L2!

On the other hand, the **global miss rate** is the fraction of misses for ALL cache levels- basically out of all accesses, the number of misses that miss ALL the caches UP TO THAT LEVEL. So by definition, the global MR is the product of all local MR, and thus must be less than or equal to any local MR. For example, the L2 global miss rate is the fraction of accesses that miss at L2 and L3.

7 Improving Cache Performance

Let's analyze cache design, as well as the tradeoffs present when designing memory hierarchy.

Remember there are many factors and parameters in cache design:

- Cache size, block size, associativity (lower number of sets)
- Policy choices: replacement, write

We want to choose an optimal set of parameters and policies such that we optimize access speed as well as limiting cost and overhead. Simplicity, like in many other areas of life, often wins.

If block size is too LOW, miss rate high (lots of compulsory misses). If block size too HIGH, though, the miss rate will also start to increase- if block size becomes close to cache size, we won't be able to access different parts of memory without kicking a bunch of blocks out (unnecessarily).

Miss rate goes down as you increase set-associativity, or DECREASE the number of sets. More associativity reduces the number of conflict misses because we can store blocks into the cache (sets) flexibly. With DM caches, we cannot do this.

L1 caches are quite small, and associativity is small as well. L2 caches are larger and have larger associativity (higher hit time), but also smaller miss rate.