

Conrado Uraga

PA3 Cache Simulator Analysis

After doing a couple of simulation with only an L1 cache in my cache simulator, I noticed that as you increase the associativity with the block size, the hit rate goes up. Mostly the block determines the hit rate. So if you want a 95% hit rate, a cache size of 32768 with associativity of 32 and block size of 16 for a long trace file will give you 95% hit rate with an LRU replacement policy. However, a better parameter would be a cache size of 65536 with associativity of 64 and block size of 32 to get a 96% hit rate. However, the associativity level is way too high, as well as the block size. Instead, let's lower the both of them and only increase the cache size.

After doing a few more tests, I found out that an associativity of 4 and a block size of 16 seems to be give a good hit rate. As I increased the size, it seems to verify that the hit rate is steadily increasing. Thus a cache size with 16384 and the assoc:4 and block size 16 for the long trace file will give a 95% hit rate. If you want a better hit rate for smaller trace files, it seems like increasing the cache size while keeping associativity of 4 and block size 16 does the trick

Now for a 99% hit rate, we have to increase everything: the cache size, the associativity and the block size. To get this, I made the cache size 1048576, assoc: 256 and block size 128 with an LRU policy using the long trace file. Now this is way too big and would be rather expensive to implement.

Now if you want 95% hit rate for both L1 and L2 caches, what cache is known for producing the best hit rates? A fully associative cache. Thus make both L1 and L2 caches fully associative caches and make L1 smaller than L2. L2 should be 16 times bigger than L1. I chose L1 to be 16384 and L2 to be 262144. The block size however has to be huge so make the block size 512. Run it with a long trace file and the hit percentages for both caches will be a bit above 95%(I got L1 to be 96% while L2 to be 95%). Thus we can see that associative cache are better for hit rates but require more space(check).

So in the end, associative caches provide a better hit rate than direct, way better. This is due to not always mapping to the same index, rather we can either all the spots open for the cache or with a set associative cache, we have a little more breathing room since it's like a mini fully associative in that index. However, we're trading off speed since you have to check each and every block, not really good in performance. Thus if you really care about the hit rate, you need to consider the blocks size, the associative that correlates with the block size(like assoc:4 with block size 16 and a good enough cache size) or set the caches as fully associative. Although direct is fast and simple, the hit rates are more likely to go down as we read more inputs.