Micah Jeffries

Professor Lupo

CPE 315-09

6/6/20

1) If you are building a processor and have to do static branch prediction (meaning you have to assume at compile time whether a branch is taken or not), how should you do it? You can make a different decision for branches that go forward or backward.

Based on my statistics that I gathered from Shang, I would build a processor that always make the static branch prediction taken for both branches that go forward and backward. In both branch statistics, the number of taken branches far outnumbered the not taken branches in all optimization levels which is why taken is generally a better static branch prediction.

2) If you are building a 256-byte direct-mapped cache, what should you choose as your block (line) size?

Based on my statistics that I gathered from Shang, I would build a 256-byte direct-mapped cache with a block size of 16 bytes since this cache had the highest overall hit rate in all optimization levels. Other small block sizes had a high hit rate as well such as 8 bytes and 32 bytes. These would be fine choices, but as we continue to increase the block size, the hit rate continually decreases. Therefore, it is generally a bad idea to build a cache with a large block size.

3) What conclusions can you draw about the differences between compiling with no optimization and -O2 optimization?

Based on my statistics that I gathered from Shang, compiling with -O2 optimization is clearly better for performance than compiling with no optimization. The number of instructions executed with -O2 optimization is drastically lower than the number of instructions with no optimization. This is because many instructions are combined together to produce more complex instructions. This actually increases the CPI, but the lower number of instructions far outweighs the increased CPI. The hit rate for all the caches were lower but that is only because the total number of cache references was lowered while the number of cache misses stayed the same. This results in a lower hit rate but does not result in a worse performance.