

F24 CSE220 Lab2

Andrew Bruce
acbruce@ucsc.edu

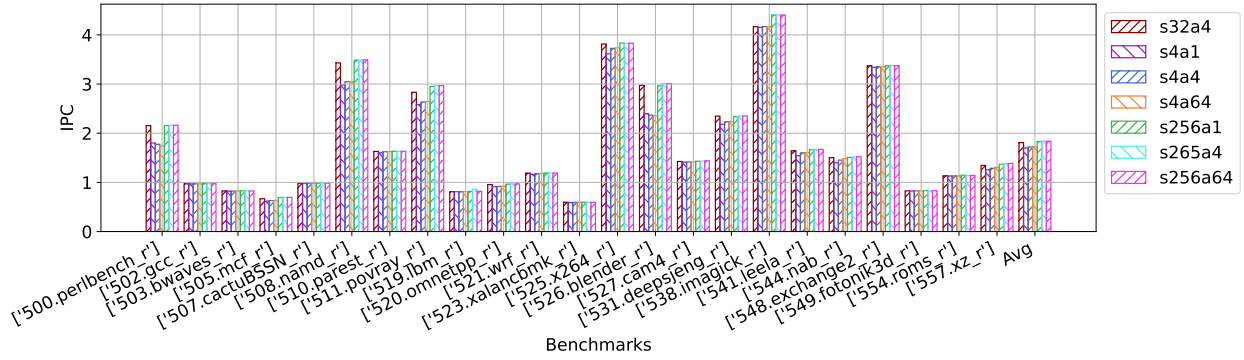
Mohit Agrawal
mmagrawa@ucsc.edu

November 1, 2024

Github Link: <https://github.com/MohitAgrawal404/220Assgn2>

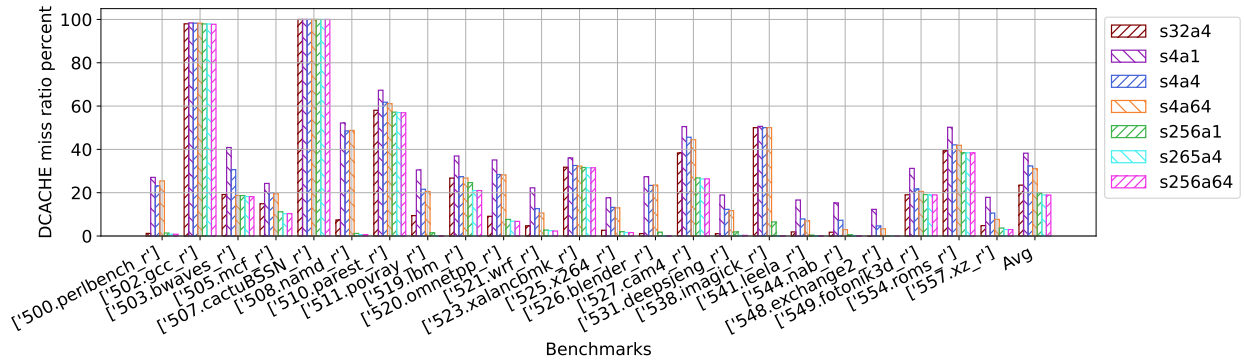
Part1

IPC



In this plot increasing both the size and associativity of the cache increases the IPC for almost every single test. Increasing the associativity from 1 to 4 was not as effective as increasing the size from 4kb to 256kb at increasing IPC.

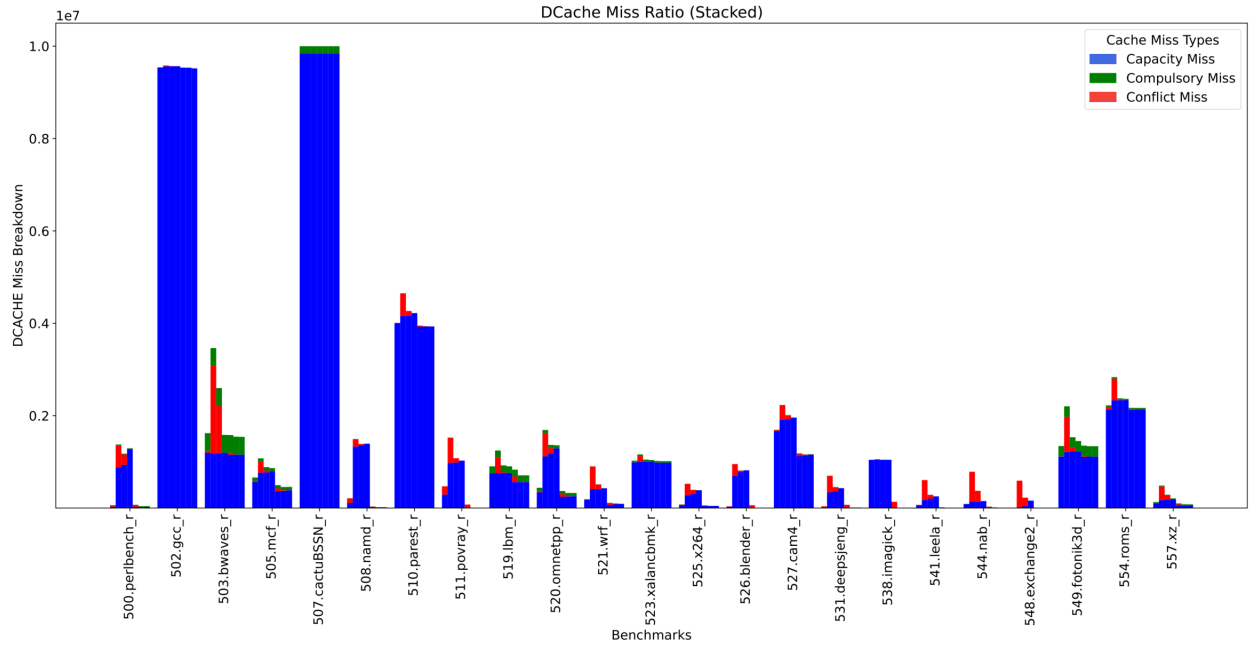
DCache miss rate



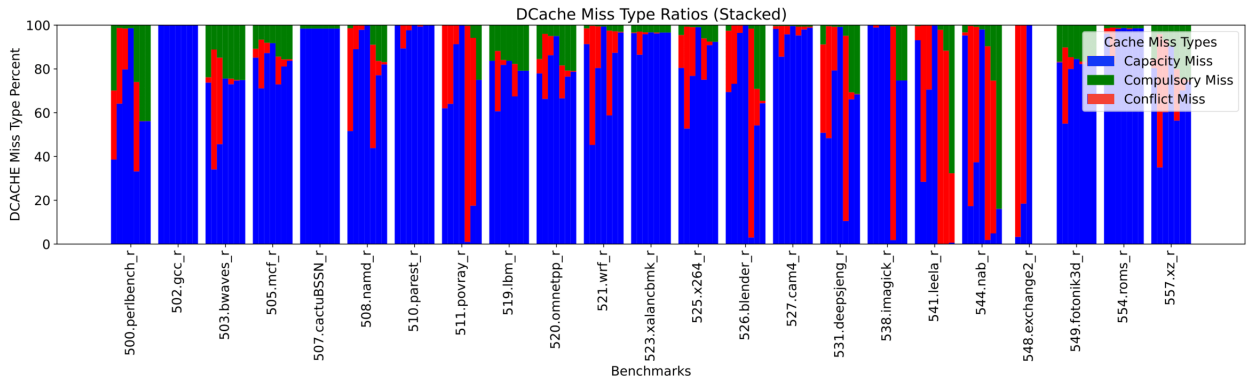
Outside of GCC and cactuBSSN, increasing the cache size to 256 kb universally decreased the miss ratio of the dcache. Increasing associativity also lowered the miss ratio. We also notice that the size 32 configuration performed surprisingly well compared to the 264 configurations showing the ability to get similar performance with less complexity by having a good associativity to capacity ratio.

Part3

DCache miss types



This graph is the same as the dcache miss rate but with stacked bars to show the cache miss type.



This graph shows the ratio of the types of misses for each configuration.

Overall we notice that the vast majority of misses are capacity misses with compulsory misses being the least common, possibly because they only happen once per block. We notice that the Capacity misses for the configurations with the same size is roughly the same due to capacity misses primarily being due to how limited the size of the cache is. The Conflict misses are more affected by the associativity as we can see between the s4a1 and s4a4 we notice a bigger red bar for s4a1, showing that a lower associativity can lead to more data fighting to be in the same spot. We believe that to improve our cache more it would be beneficial to increase the cache size more and also scale the associativity a little bit with it seeing as most of our misses are caused by Capacity misses and we notice a significant decrease in those misses once the size is increased and almost no conflict misses when our associativity is scaled well with the size of the cache.