

Ernie Lee  
CS320  
Project Report

astar_memory_trace:	gcc_memory_trace:	mcf_memory_trace:
90 92 92 92	76 85 95 97	89 91 91 91
92 92 92 92	97 97 98 98	91 91 91 91
92	98	91
92	97	91
62 62 62 62	93 94 94 94	65 65 65 65

For the astar\_memory\_trace, the first three caches had about the same cache hit rates. The hit rates for the set associative without allocation on a write miss had the worst performance. The gcc\_memory\_trace had the worst performance using the direct mapped cache. The set associative and fully associative caches had the best performance having the hit rates above 97%. The mcf\_memory\_trace had very consistent hit rates aside from a 1KB size direct mapped cache. The set associative cache with no allocation on a write miss was the worst performer out of all the caches.

The fully associative cache had the best performance out of the other caches. The reason being that there are no indexes in the cache. Using the LRU replacement policy, only the data that was used the least was replaced. However, using this cache is expensive in terms of its use of resources.

The optimal configuration for the cache is to have the cache large enough to hold all the addresses such that there is no need for a replacement policy. Data will not be replaced so the only misses occur when the data is first put into the cache. It is also best to search each way of the cache concurrently to speed up the time it takes to search.