Cache Simulator

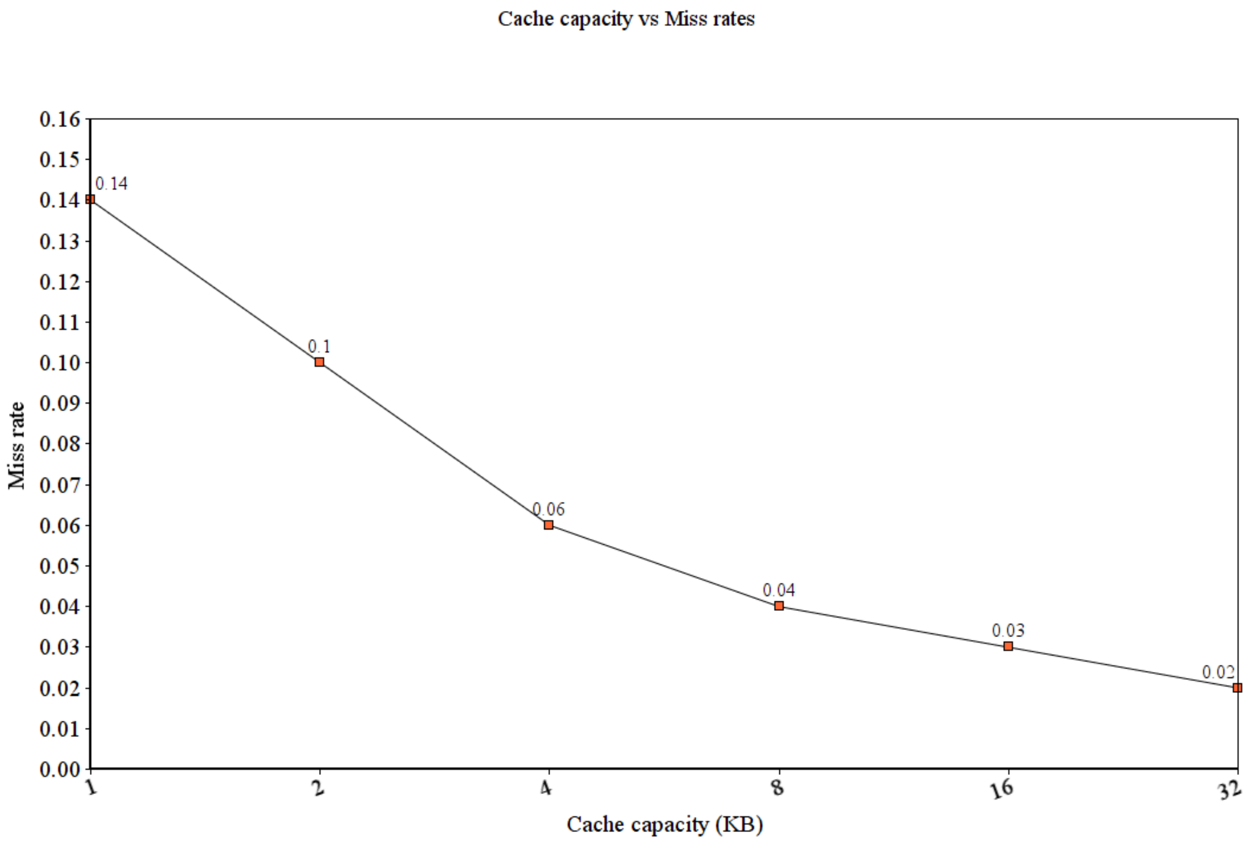
Implementation

In this project, I implemented a cache in C language. Inside cache_access function, I, firstly, find out how many bits are used for the sets in order to determine the number of bits for the tag. In the meantime, I find the starting point of the set we need to access so as to check if the data exists in the set. The first for loop goes through the lines of the set and it finds out that there exists the same tag with valid data, it means it is a hit so I update the time it is accessed to the current time and return. If it does not exist, it means a miss so I increment total_cache_misses variable. Then, I look for a line in a given set where there is no data (means its status is invalid). If there is, then I update its tag with current tag as well as the time with current time and make its status to be valid, then return. However, if all lines' status is valid, I look for least recently used line in the set by comparing the time of each line with one another. After finding the line, as usual, I update its time and tag with current ones.

Cache capacity vs Miss rates

In theory, if the size of the cache is increased, it should decrease the miss rate, since the cache will store more information than before. The results of the experiment also prove it. Here is the summary:

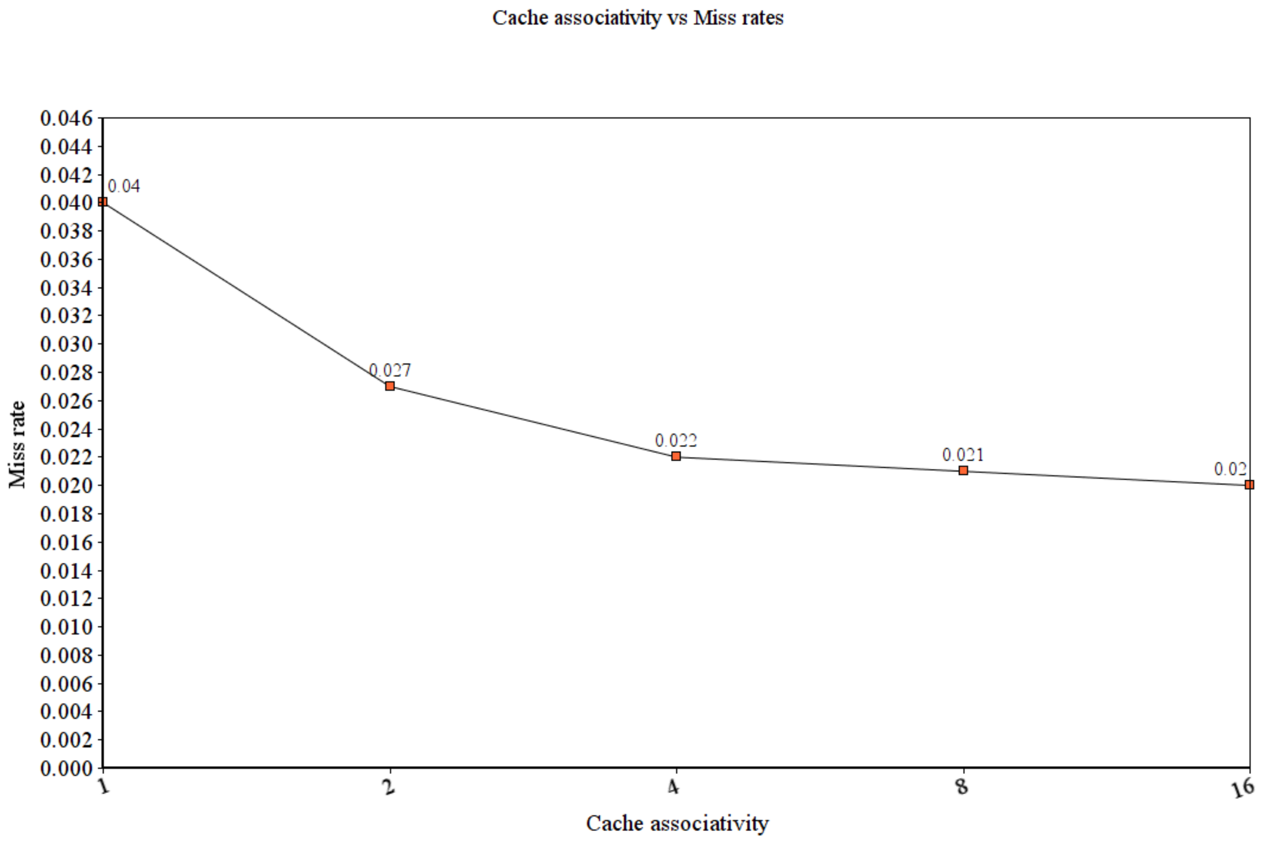
Capacity	1 KB	2 KB	4 KB	8 KB	16 KB	32 KB
Miss rate	0.14	0.10	0.06	0.04	0.03	0.02
Misses	52635	36520	24118	15351	10836	9032



Cache associativity vs Miss rates

In theory, increasing the associativity will decrease the number of misses since for each set, we have more options to store information which will decrease the conflict misses. The results of the experiment also confirm it. Here is the summary:

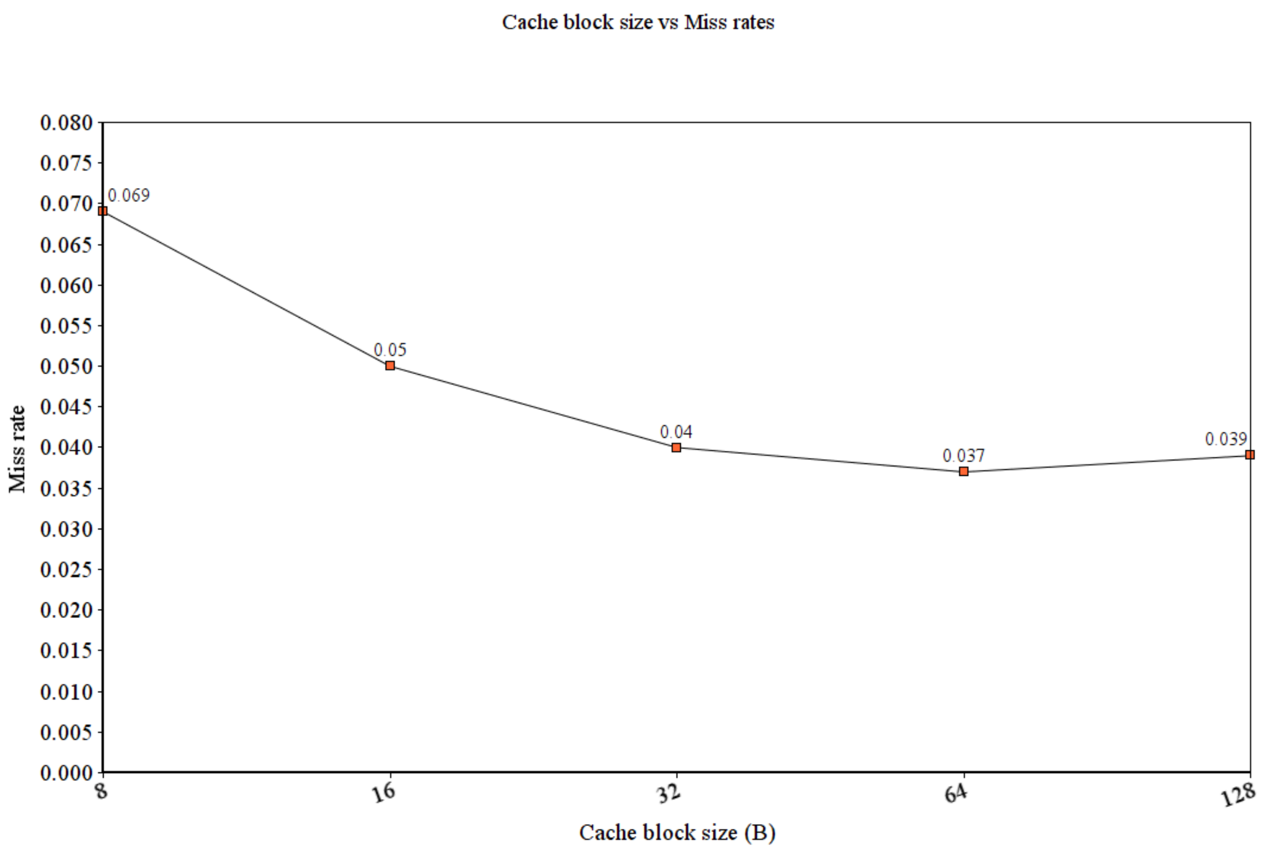
Associativity	1	2	4	8	16
Miss rate	0.040	0.027	0.022	0.021	0.020
Misses	15351	10434	8319	7881	7699



Cache block size vs Miss rates

In theory, increasing the cache block size will improve spatial locality, hence might decrease the miss rate depending on the order of memory access. The results of the experiment can also approve of this. Here is the summary:

Line size (B)	8	16	32	64	128
Miss rate	0.069	0.050	0.040	0.037	0.039
Misses	26338	19190	15351	14255	14760


Cache block size vs Miss rates
Miss rate
Cache block size (B)