Amar Budhiraja 201303009
Rohita Darga 201201104

DATABASE SYSTEMS DELIVERABLE 2

Parameters like the the number of pages and the page size affect LRU caching. A page holds data. The bigger the page is, the greater amount of data it can hold. The number of pages shows how many pages the cache can hold. Each parameter alters an aspect of how information is handled and how its is called. Let's concentrate on one parameter at a time.

Let's begin with the size of the page. We know that the number of frames is equal to the size of the memory divided by page size. By increasing the page size, the number of frames decrease. This can be both a good and bad thing. Having fewer frames will decrease the replacement choices. Large pages also waste space due to internal fragmentation. For example, if only two-thirds a page is filled for each page, then thirty pages would have wasted space equivalent to ten pages. This is no small amount.

On the positive side, when using large pages, the number of faults may decrease since there is extra space where information can easily be filled in a page. For example, when using clustered indexes, more and more records can be added to one page reducing the risk of using an overflow bucket. The larger buckets also reduce the number of blocks needed to store information which makes it faster to access data when using primary index or another form of information storage because the number of block accessed decrease.

For example, a system which has a smaller page size needs a larger page table since the memory needs to be held in more pages. Another system with the same memory and with a large page size will have a smaller page table making the algorithms used to detect the page needed faster. This is especially helpful when using B+ trees as the number of levels decrease.

Also, when larger pages are used, it increases the efficiency of information retrieval. Memory is stored on disks and since most of the delay in the retrieval of data is caused by seek time, when a larger amount of data is transferred, it reduces this seek time. If the pages were smaller, more time would be spent finding those individual pages in order to get all the information.

So, as can be seen, the perfect size for the page size must be found. A size that is neither too large or too small in order to increase efficency and decrease wastage of storage space. We shall see how this affects our own DBMS below, but first let's explore how the parameter of number of pages affects LRU caching.

The number of pages should have an obvious effect on LRU caching. The fewer the number the pages, the more MISSES we have since there are fewer possible page options in the LRU. The more MISSES there are, the more times we will have to go down the memory heirarchy in order to access our needed memory. This takes a massive amount of time since the lower the memory is in the heirarchy, the longer it will take to access it. The larger the LRU is, the more likely we will find our block within the cache. This is beneficial especially

for random block accesses versus sequential block accesses. In sequential, the cache will always be missing the next block while in random block accesses, it is likely to be visited again.

However, on the flip side, the greater number of blocks there are in the cache, the longer it will take to search within the cache for a certain block which will reduce effectiveness. Also, as cache's get larger, they are also more expensive. Also, a larger cache will not be effective if the data is sequential as mentioned before. Like before, there should be a reasonable size must be found that is neither too large or too small.

Our Data:

This shows the hit-miss ratio for the number of pages

(Number of Records44)-Pages 2^4
1-0.23529412
2-0.35352942
3-0.47235295
4-0.9991177

Number of Records ~500 Pages Required 2^5
0.14117648
0.21294117
0.45668491
0.7717647
0.99941176

Number of Records ~2000 Pages Required 2^6
0.108490564
0.16061321
0.1721698
0.43620282
0.6627358
0.99622643
PageSize is 100bytes