CSCI4707 Lab3 Report

We found that with the default number of shared buffers allocated to the database, the code segment of buffer replace policy will not be used at all because there are already enough buffers in the memory pool. So here we limit the number of shared buffers to 100 when starting the database: “./bin/postgres -B 200 -D data/”.

After running the given test, we derive the results shown in the following table:

Test Algorithm	values10k.dat		values100k.dat	
	buffertest1	buffertest2	buffertest1	buffertest2
FIFO	00:00:00.179915	00:00:00.308247	00:00:00.292917	00:00:00.432653
Clock Sweep	00:00:00.18741	00:00:00.399316	00:00:00.300057	00:00:00.536458

From the table, we can see that FIFO algorithm is always a little bit faster than the Clock Sweep algorithm with the given tests. This is because, FIFO is iterating through the buffer pool and return the first buffer that is not pinned. That is, it does not take much time for FIFO to find such a buffer with the given tests. However, Clock Sweep algorithm is approximating the LRU replacement policy in the way that it is trying to replace an older page. In order to do that, it is keeping another count which gives an hint of how long the buffer has not been used. And in order to find a buffer for new page, this algorithm may have to iterate through the buffer pool several times and is creating a lot overhead. And the given tests are simply selecting the data with given values, and the Clock Sweep algorithm's effort to always find an older page does not help much here. Because the tests are always trying to find different data with different given values. So FIFO wins here because it does not have as much overhead in the process of finding the page to replace.