CSCI4707 Lab03 Report

Clock Sweep

Value10K Value100K

Test1 Test2 Test1 Test2

00:00:00.15308 00:00:00.29209 00:00:00.311469 00:00:00.37201

00:00:00.148313 00:00:00.297079 00:00:00.306113 00:00:00.419014

00:00:00.159175 00:00:00.306932 00:00:00.314106 00:00:00.424416

00:00:00.162407 00:00:00.281892 00:00:00.323929 00:00:00.40384

average

00:00:00.155744 00:00:00.294498 00:00:00.313904 00:00:00.404820

FIFO

Value10K Value100K

Test1 Test2 Test1 Test2

00:00:00.180262 00:00:00.34845 00:00:00.410539 00:00:00.503271

00:00:00.182693 00:00:00.34859 00:00:00.393551 00:00:00.522399

00:00:00.176537 00:00:00.365331 00:00:00.403578 00:00:00.492329

00:00:00.175072 00:00:00.378696 00:00:00.421712 00:00:00.545548

average

00:00:00.178641 00:00:00.369267 00:00:00.407345 00:00:00.515887

*Note: Because the default buffer pool is too large, so executing our test data*

*doesn’t use replace policy. So I reset the buffer pool to 80, using the following*

*command ‘$HOME/lab03/bin/postgres –B 80 –D $HOME/lab03/data’. And then*

*we executed the given test case, we got the following results.*

**Analysis**

From the above result, we can clearly find that, FIFO algorithm executed always slower

than clock sweep. Firstly, we find that both test1 and test2 always try to find different

data with different given value, which means replace policy will be called a lot of times.

Because of this, we think the reason that FIFO runs faster than clock sweep is FIFO

algorithm always return the first unpinned buffer. But Clock Sweep attempt to find the oldest unpinned buffer and replace it. It means that this algorithm leave the more frequently used buffer in the buffer pool. So, according to to the test case that select several different data lots of times, FIFO Clock Sweep should run faster than FIFO.