Lab 3 Report

CSCI 4707

Rain Xuanyu Zhang (zhan2223), Hao Wang (wang5167)

November 24, 2015

Following are the results by running each test scripts with both 10K data file and 100K data file, all numbers are in second:

Least Recent Used (LRU) strategy:

Buffertest1		Buffertest2		
10K values	100K values	10K values	100K values	
0.048035	0.247159	0.255279	0.485715	
0.050190	0.193025	0.290991	0.416763	
0.055026	0.173583	0.266439	0.337105	
0.050177	0.166627	0.284896	0.489926	
0.047828	0.164230	0.256378	0.269561	
0.050375	0.169665	0.262623	0.264017	
0.052187	0.163964	0.257342	0.260963	
0.062766	0.194455	0.247287	0.367094	
0.052073	0.184089	0.256154	0.361339	AVERAGE Time used

Clock:

Buffertest1		Buffertest2		
10K values	100K values	10K values	100K values	
0.070572	0.197671	0.254632	0.257316	
0.063651	0.192616	0.257550	0.253413	
0.072142	0.193395	0.260403	0.259468	
0.064894	0.184834	0.286215	0.257591	
0.067303	0.176073	0.256203	0.262193	
0.050717	0.174550	0.269183	0.311044	
0.049288	0.171822	0.284213	0.286561	
0.050563	0.164843	0.268098	0.256053	
0.061141	0.181976	0.267026	0.267955	AVERAGE time used

From above numbers, we can see that the time consumed to run different scripts are identically the same with two buffer strategies, except clock strategy will consume less time to run 100k values when having range queries (buffertest2.sql).

Based on our conjecture, the reason that made this minimal or even negligible difference is the nature of two strategies: both of which are selecting the older buffer frames to replace, and keep track of the time at which each buffer frame in the buffer pool most recently became available/replaceable. There shouldn't be a dramatic difference in terms of efficiency in that those two strategies has similar complexities.