1. Analyze/Interpret any “interesting” trends that you notice in the behaviour of the LRU, LFU, and MRU replacement policies. You have to explain when and why one policy is better than the other.

Answer: Interesting trends we notice is that, for sequential scan and index scan respectfully, the average buffer hit rates of LRU, LFU and MRU are all the same when the buffer cache size is large enough (about 100 or 120)

1. Try to estimate the size of your table Data in number of blocks, based on the buffer cache hit rates, with increasing size of the table.

Answer: The size of the table Data is roughly 30 blocks.

I call the following command to do the analysis.

cat ScanQueries.sql | postgres -B 20 -D ~/data/ -d 1 -s mydatabase 2>&1 | grep “Shared blocks”

! Shared blocks: 109 read, 0 written, buffer hit rate = 62.41%

! Shared blocks: 16 read, 0 written, buffer hit rate = 55.56%

! Shared blocks: 16 read, 0 written, buffer hit rate = 46.67%

! Shared blocks: 16 read, 0 written, buffer hit rate = 46.67%

! Shared blocks: 16 read, 0 written, buffer hit rate = 46.67%

! Shared blocks: 16 read, 0 written, buffer hit rate = 46.67%

! Shared blocks: 16 read, 0 written, buffer hit rate = 46.67%

! Shared blocks: 16 read, 0 written, buffer hit rate = 46.67%

! Shared blocks: 16 read, 0 written, buffer hit rate = 46.67%

! Shared blocks: 16 read, 0 written, buffer hit rate = 46.67%

As we know, system blocks (catalog etc.) are loaded when the database starts, so the statistic of the 1st time read is not accurate. I will simply ignore it. For the rest data, we use the number of blocks it reads from the disk divided by (1 – buffer hit rate), then we can get the size of the table in terms of blocks. For example: 16 / (1 – 46.67%) = 30. We definitely need to change the buffer cache size to verify the result.