# Week#3 I/O stat and hit ratio by varying the buffer size

Jaehun Lee
2017314626

## 1. INTRODUCTION

Buffer Size is critical for DBMS performance, since if the requested page is in the pool, which means buffer hit, there is no need for read/write between buffer frame and storage. However if the requested page is not in the pool, the data of the frame should be eliminated and even if victim is dirty, system should write victim to storage and read the requested page from storage into the victim buffer frame. In this exercise, buffer size will be changed and I/O stat, hit ratio and TpmC depending on the buffer size will be observed.

## 2. METHODS

The purpose of this exercise is to check performance of DBMS with I/O stat, hit ratio and TpmC depnding on buffer size. TPC-C is used for benchmark tool and MySQL is used for DBMS. Benchmark will take 600 seconds. The buffer size will be 10, 20 30, 40, 50 % of DB size (2GB). As the buffer size increases, the buffer miss is expected to decrease, and accordingly, the number of disks i/o is expected to decrease. Therefore, the number of read/write per transaction is expected to decrease. In addition, if buffer misses are reduced and performance is improved, the total TpmC is also expected to increase.

## 3. Performance Evaluation

### 3.1 Experimental Setup

**[Table 1] System Setup**

| Type | Specification |
|---|---|
| OS | Ubuntu 20.04.3 LTS |
| CPU | Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz (6 cores) |
| Memory | 8GB |
| Kernel | 5.11.0-43-generic |
| Device | samsung ssd 860 evo 500GB |

**[Table 2] Benchmark Setup**

| Type | Configuration |
|---|---|
| DB size | 2GB (20 warehouse) |
| Buffer Pool Size | 200MB, 400MB, 600MB, 800MB, 1GB (10,20,30,40,50% of DB size) |
| Benchmark Tool | tpcc-mysql |
| Runtime | 600s |
| Connections | 8 |

### 3.2 Experimental Results

**[Table 3] Performance depending on Buffer Size**

| Buffer Size | TpmC | Hit Ratio | | | I/O | |
|---|---|---|---|---|---|---|
| | | Start | 50% | End | r/s | w/s |
| **200M** | 9691.5 | 96.6 | 96.1 | 96 | 5048 | 2429 |
| **400M** | 17071.0 | 98.6 | 98.7 | 98.6 | 2772 | 2327 |
| **600M** | 19477.4 | 99.1 | 99.2 | 99.1 | 2149 | 1856 |
| **800M** | 23563.4 | 99.4 | 99.6 | 99.6 | 1178 | 2068 |
| **1G** | 22785.9 | 99.4 | 99.7 | 99.7 | 1206 | 2010 |

As the buffer size increased, the hit ratio also increased. When buffer size was 200MB, hit ratio was about 96.23% (avg), but when buffer size was 1GB, hit ratio was about 99.6% (avg). As the hit ratio increased, TpmC also increased, so it can be seen that the performance of mysql improved. However, there was little difference in hit ratio between 800MB and 1GB, and also, TpmC decreased at 1GB compared to 800MB. This shows that the buffer size is already sufficient at 800 MB. The increase in buffer size over 800MB did not show any significant performance improvement. This seems to be because the hit ratio has already recorded a high figure of 99.6% in the buffer size of 800MB. The number of I/O per transaction is calculated as follows. (divide r/s and w/s by TpmC)

**[Table 4] I/O per transaction on Buffer Size**

| | 200M | 400M | 600M | 800M | 1G |
|---|---|---|---|---|---|
| **r/s** | 0.52 | 0.16 | 0.11 | 0.05 | 0.05 |
| **w/s** | 0.25 | 0.13 | 0.10 | 0.09 | 0.09 |

I/O per transaction also decreased, since buffer size increased and as buffer hit ratio increased, there was less disk I/O. In addition, even in this case, there was no significant difference in the results of 800MB and 1GB.

## 4. Conclusion

Through this experiment, it was confirmed that the buffer hit ratio increased as the buffer size increased. In addition, as the hit ratio increased, the number of disk I/O decreased, and the performance of DBMS also improved, resulting in an increase in TpmC. However, if sufficient hit ratio improvement has already occurred due to the increase in buffer size, it can also be confirmed by the case of 800M->1G that further improvement in performance is difficult to expect due to the increase in buffer size.

## 5. REFERENCES

[1] meeeejin, “SWE3033-F2021”, Github repository, https://github.com/meeeejin/SWE3033-F2021/