Week6: RocksDB

Siyeol Choi
2019315530

1. INTRODUCTION

RocksDB is embedded database for key-value data written in C++. RocksDB has features of transactions, backups and snapshots, column families, TTL, merge operators, static collection, and geospatial indexing. This week, set the environment variable of rocksdb and run benchmarks of the PC

2. METHODS

At WSL2 linux ubuntu system, cloned rocksdb from the facebook repository and run the make file. Then run the db_bench with the option of

--benchmarks="readrandomwriterandom" \N threads doing random-read, random-write

-db="/path/to/datadir" \Use the db with the following name.

-use_direct_io_for_flush_and_compaction=true \

-use_direct_reads=true \ Use O_DIRECT for reading data

-duration=600 \ Set duration

-statistics \Database Statistics

-stats_interval_seconds=10 \Report stats every N seconds. This overrides stats_interval when both are > 0

This will produce results of readrandomwriterandom and block cache.

3. Performance Evaluation

3.1 Experimental Setup

[Benchmark setup]

RocksDB: version 7.8.0

CPU: 8 * Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz

CPUCache: 6144 KB

Keys: 16 bytes each (+ 0 bytes user-defined timestamp)

Values: 100 bytes each (50 bytes after compression)

Entries: 1000000

Prefix: 0 bytes

Keys per prefix: 0

RawSize: 110.6 MB (estimated)

FileSize: 62.9 MB (estimated)

Write rate: 0 bytes/second

Read rate: 0 ops/second

Compression: Snappy

Compression sampling rate: 0

Memtablerep: SkipListFactory

Perf Level: 1

[System Setup]

OS Ubuntu 20.04.1 LTS (GNU/Linux 4.19.104-microsoft-standard

cpu cores 4

Vendor: Msft

Product: Virtual Disk

Revision: 1.0

Compliance: SPC-3

User Capacity: 274,877,906,944 bytes [274 GB]

Logical block size: 512 bytes

Physical block size: 4096 bytes

Type Configuration

3.2 Experimental Results

[Result of readrandomwriterandom]

- 63.155 micros/op
- 15833 ops/sec (9503999 operations / 600.228 seconds)
- reads:8553600

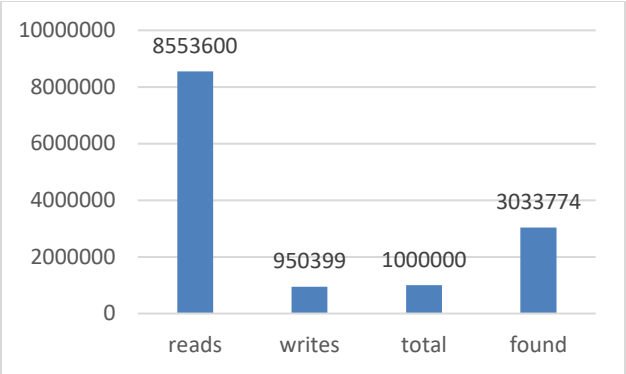
  writes:950399

  total:1000000

  found:3033774

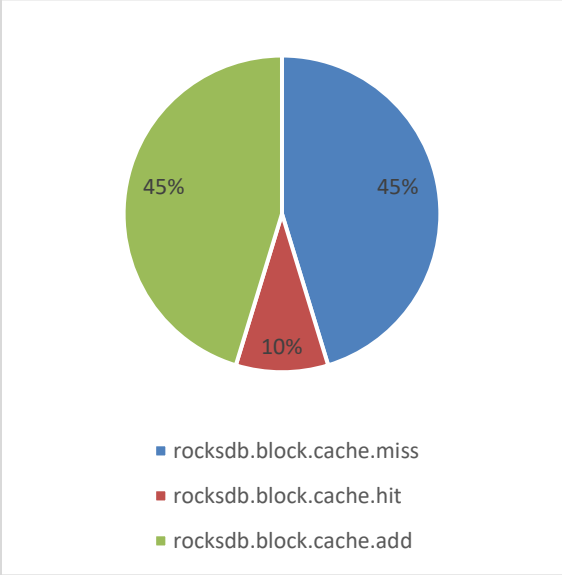
* micros/op: Microseconds spent processing one operation

* ops/sec: Processed operations per second



[Result of Block Cache]

- rocksdb.block.cache.miss COUNT : 3054159 (45%)
  rocksdb.block.cache.hit COUNT : 635652 (10%)
  rocksdb.block.cache.add COUNT : 3054159 (45%)



4. Conclusion

Microseconds spent processing one operation was 63.155 and Processed operations per second was 15833. The ratio of cache miss/hit/add was 45%/10%/45%.

In the following iterations of experiments, we will be amending parameters and functions to monitor rooms of improvement on micros/op and ops/second.

5. REFERENCES

[1] RocksDB https://github.com/facebook/rocksdb

[2] db_bench option https://github.com/facebook/rocksdb/wiki/Benchmarking-tools

[3] Report format https://github.com/LeeBohyun/SWE3033-F2022/blob/main/report-guide.md