WEEK6

김우진
2017314712

1. INTRODUCTION

This week you will install RocksDB and run DB_Bench on our own system.

2. METHODS

1) Install pre-requisites

Linux - Ubuntu

- Upgrade gcc version at least 4.8
- gflags: `sudo apt-get install libgflags-dev` If this doesn't work, here's a nice tutorial: tutorial link
- snappy: `sudo apt-get install libsnappy-dev`
- zlib: `sudo apt-get install zlib1g-dev`
- bzip2: `sudo apt-get install libbz2-dev`
- lz4: `sudo apt-get install liblz4-dev`
- zstandard: `sudo apt-get install libzstd-dev`
- java: `sudo apt install default-jdk sudo apt install default-jre`

2) Build and install RocksDB

```
$ git clone https://github.com/facebook/rocksdb
$ cd rocksdb
$ make static_lib -j8
$ make db_bench DEBUG_LEVEL=0 -j8
```

3) Run DB_Bench

```
$ ./db_bench
```

You can see options of `db_bench` using below command (Details):

```
$ ./db_bench --help
```

For the report, run the below command and measure the performance of RocksDB on your system:

> Update `-db="/path/to/datadir"` path according to your experimetal environment

```
$ ./db_bench --benchmarks="readrandomwriterandom" \
  -db="/path/to/datadir" \
  -use_direct_io_for_flush_and_compaction=true \
  -use_direct_reads=true \
  -duration=600 \
  -statistics \
  -stats_interval_seconds=10 2>&1 | tee result.txt
```

4) Record the experimental result

At the end of the benchmark, you can see the below result:

```
...
readrandomwriterandom :      53.084 micros/op 18838 ops/sec; ( reads:10172700 writes:1130299 total:1000000 found:4076910)
...
```

- `micros/op` : Microseconds spent processing one operation
- `ops/sec` : Processed operations per second

3. Performance Evaluation

3.1 Experimental Setup

Type	Specification
OS	Ubuntu 18.04.65 LTS
CPU	Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz
Memory	3994720 kB
Kernel	Linux ubuntu 5.4.0-144-genericcat /proc

3.2 Experimental Results

DB path: [/home/nicholasbear/Desktop/rocksdb/db]
readrandomwriterandom : 38.050 micros/op
26281 ops/sec
600.083 seconds
15770999 operations;
(reads:14193900 writes:1577099
total:1000000 found:7053120)

4. Conclusion

RocksDB 는 다음과 같은 성능을 제공합니다.

빠른 읽기 및 쓰기 성능: LSM 트리는 읽기와 쓰기 성능 모두에서 매우 우수한 성능을 제공합니다. 또한 RocksDB 는 Memtable 과 SSTable 등의 다양한 데이터 구조를 사용하여 데이터를 저장하므로, 메모리와 디스크의 효율적인 사용을 가능케 합니다.

고성능 컴팩션: LSM 트리는 특정 시간 간격으로 Segments 를 병합하는 컴팩션 작업을 수행합니다. RocksDB 는 이러한 컴팩션 작업을 효율적으로 수행하여 디스크 공간의 최적화와 성능 향상을 동시에 달성할 수 있습니다.

데이터 안정성: LSM 트리는 데이터의 안정성과 복구 기능을 제공합니다. RocksDB 는 WAL(Write-Ahead-Log)을 사용하여 데이터를 안정적으로 저장하고, 복구 기능을 제공합니다.

확장성: RocksDB 는 분산 환경에서 높은 확장성을 제공합니다. 또한 RocksDB 는 다양한 언어와 플랫폼에서 사용할 수 있으며, 다양한 오픈소스 프로젝트에서 사용되고 있습니다.

RocksDB 는 대용량 데이터 처리에 특화되어 있으며, 높은 읽기와 쓰기 성능, 안정성, 확장성을 제공합니다. 따라서 RocksDB 는 대규모 데이터 처리 및 분산 시스템에서 매우 효과적으로 사용될 수 있습니다.