RocksDB Festival

RF3_Team_WAL

Supported by IITP, StarLab.

July 5, 2021
김민준, 이빈
alswnssl0528@naver.com,
32183118@dankook.ac.kr
Team Name

RocksDB Festival

- Contents
  - ✓ 팀 구성 : Team Info
  - ✓ RocksDB db_bench 설치 및 실험 결과 : db_bench
  - ✓ 선택한 주제와 선택한 이유 : WAL
  - ✓ 이번 스터디 동안 기대하는 결과

RocksDB Festival : Team Info

- Team Member (Github ID)
  - ✓ 김민준(Sadsprin)
  - ✓ alswnssl0528@naver.com

  - ✓ 이빈(Bynnn)
  - ✓ 32183118@dankook.ac.kr

RocksDB Festival : db_bench

- RocksDB db_bench 설치 및 실험 결과
  - ✓ 실험 환경 : A, B, C

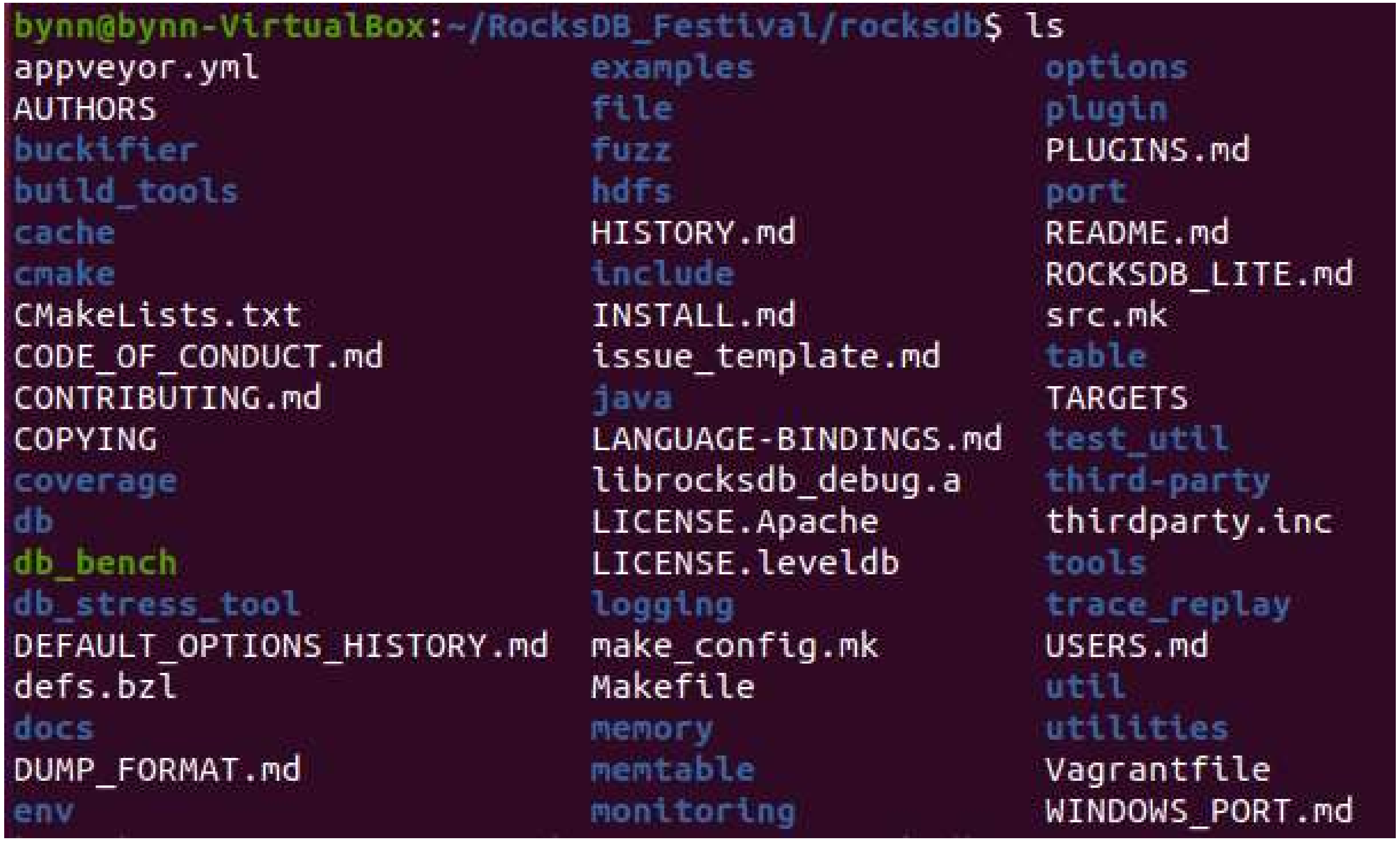
A	
CPU	4 * Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHZ
OS	Ubuntu 20.04.2 LTS
	Real Machine

B	
CPU	2 * Intel(R) Core(TM) i5-4690 CPU @ 3.50GHZ
OS	Ubuntu 20.04.2 LTS
	Virtual Machine

C	
CPU	4 * AMD Ryzen 7 3700X 8-Core Processor
OS	Ubuntu 20.04.2 LTS
SSD	PM981a
	Virtual Machine

RocksDB Festival : db_bench

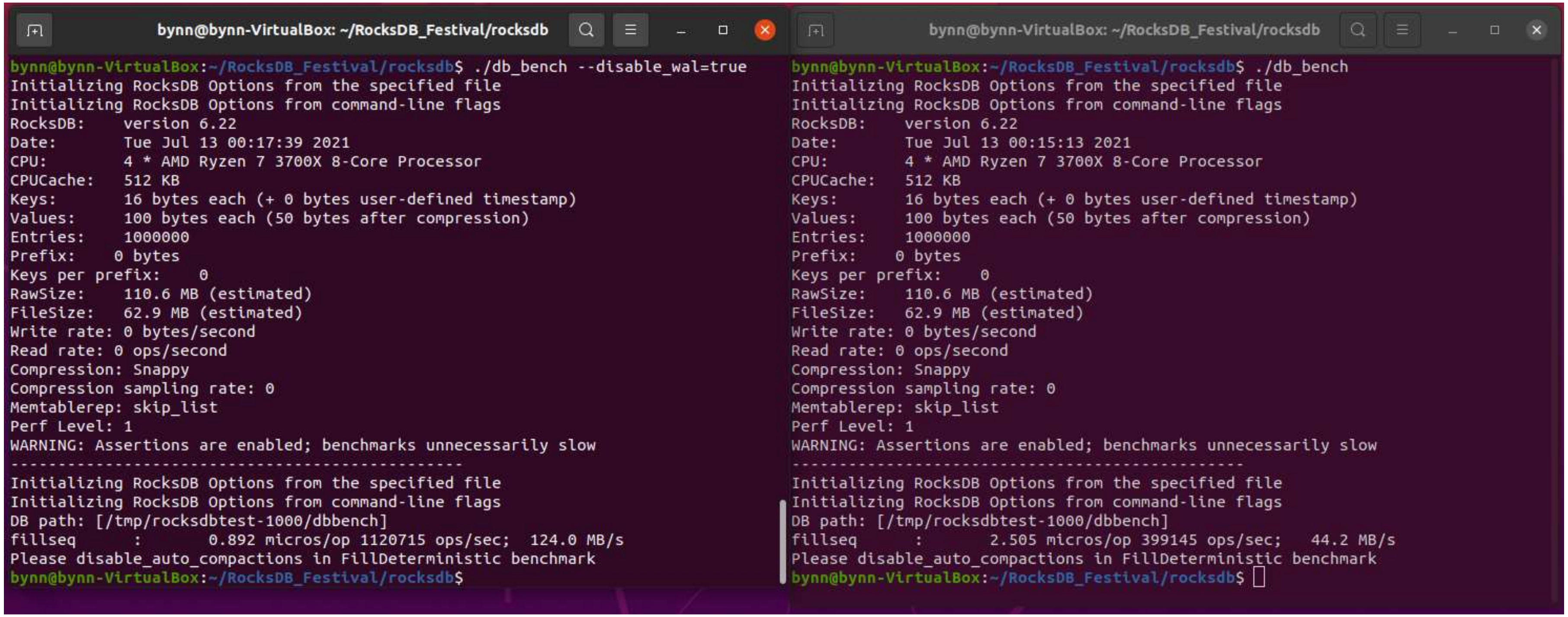
- RocksDB db_bench 설치 및 실험 결과



```
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$ ls
appveyor.yml                  examples              options
AUTHORS                       file                  plugin
buckifier                     fuzz                  PLUGINS.md
build_tools                   hdfs                  port
cache                         HISTORY.md            README.md
cmake                         include               ROCKSDB_LITE.md
CMakeLists.txt                INSTALL.md            src.mk
CODE_OF_CONDUCT.md            issue_template.md     table
CONTRIBUTING.md               java                  TARGETS
COPYING                       LANGUAGE-BINDINGS.md  test_util
coverage                      librocksdb_debug.a    third-party
db                            LICENSE.Apache        thirdparty.inc
db_bench                      LICENSE.leveldb       tools
db_stress_tool                logging               trace_replay
DEFAULT_OPTIONS_HISTORY.md    make_config.mk        USERS.md
defs.bzl                      Makefile              util
docs                          memory                utilities
DUMP_FORMAT.md                memtable              Vagrantfile
env                           monitoring            WINDOWS_PORT.md
```

RocksDB Festival : db_bench

- RocksDB db_bench 설치 및 실험 결과
  - ✓ --disable_wal=[boolean]

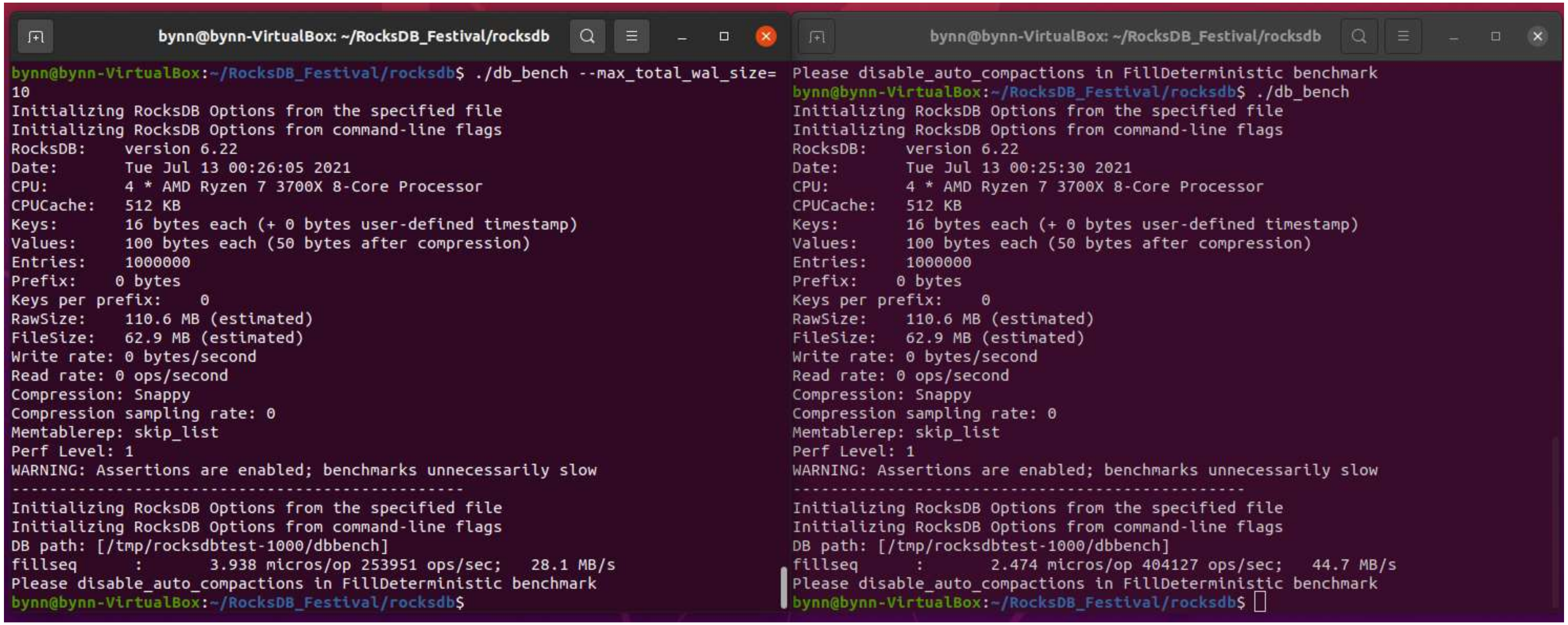

bynn@bynn-VirtualBox: ~/RocksDB_Festival/rocksdb
bynn@bynn-VirtualBox: ~/RocksDB_Festival/rocksdb

```
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$ ./db_bench --disable_wal=true
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 6.22
Date:       Tue Jul 13 00:17:39 2021
CPU:        4 * AMD Ryzen 7 3700X 8-Core Processor
CPUCache:   512 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: skip_list
Perf Level: 1
WARNING: Assertions are enabled; benchmarks unnecessarily slow
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/tmp/rocksdbtest-1000/dbbench]
fillseq      :       0.892 micros/op 1120715 ops/sec;  124.0 MB/s
Please disable_auto_compactions in FillDeterministic benchmark
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$
```

```
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$ ./db_bench
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 6.22
Date:       Tue Jul 13 00:15:13 2021
CPU:        4 * AMD Ryzen 7 3700X 8-Core Processor
CPUCache:   512 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: skip_list
Perf Level: 1
WARNING: Assertions are enabled; benchmarks unnecessarily slow
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/tmp/rocksdbtest-1000/dbbench]
fillseq      :       2.505 micros/op 399145 ops/sec;   44.2 MB/s
Please disable_auto_compactions in FillDeterministic benchmark
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$
```

RocksDB Festival : db_bench

- RocksDB db_bench 설치 및 실험 결과
  - ✓ --max_total_wal_size=[int value]



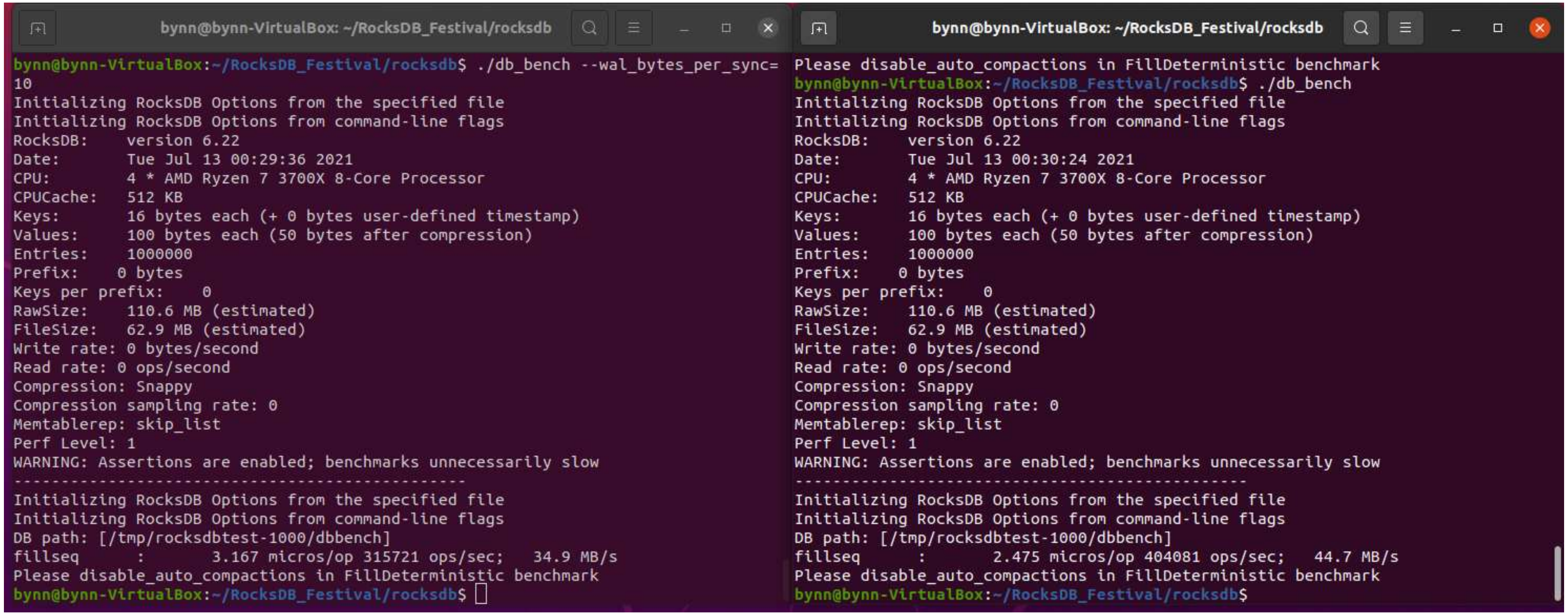
bynn@bynn-VirtualBox: ~/RocksDB_Festival/rocksdb
bynn@bynn-VirtualBox: ~/RocksDB_Festival/rocksdb

```
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$ ./db_bench --max_total_wal_size=
10
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 6.22
Date:       Tue Jul 13 00:26:05 2021
CPU:        4 * AMD Ryzen 7 3700X 8-Core Processor
CPUCache:   512 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: skip_list
Perf Level: 1
WARNING: Assertions are enabled; benchmarks unnecessarily slow
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/tmp/rocksdbtest-1000/dbbench]
fillseq      :       3.938 micros/op 253951 ops/sec;   28.1 MB/s
Please disable_auto_compactions in FillDeterministic benchmark
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$
```

```
Please disable_auto_compactions in FillDeterministic benchmark
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$ ./db_bench
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 6.22
Date:       Tue Jul 13 00:25:30 2021
CPU:        4 * AMD Ryzen 7 3700X 8-Core Processor
CPUCache:   512 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: skip_list
Perf Level: 1
WARNING: Assertions are enabled; benchmarks unnecessarily slow
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/tmp/rocksdbtest-1000/dbbench]
fillseq      :       2.474 micros/op 404127 ops/sec;   44.7 MB/s
Please disable_auto_compactions in FillDeterministic benchmark
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$
```

RocksDB Festival : db_bench

- RocksDB db_bench 설치 및 실험 결과
  - ✓ --wal_bytes_per_sync=[int value]

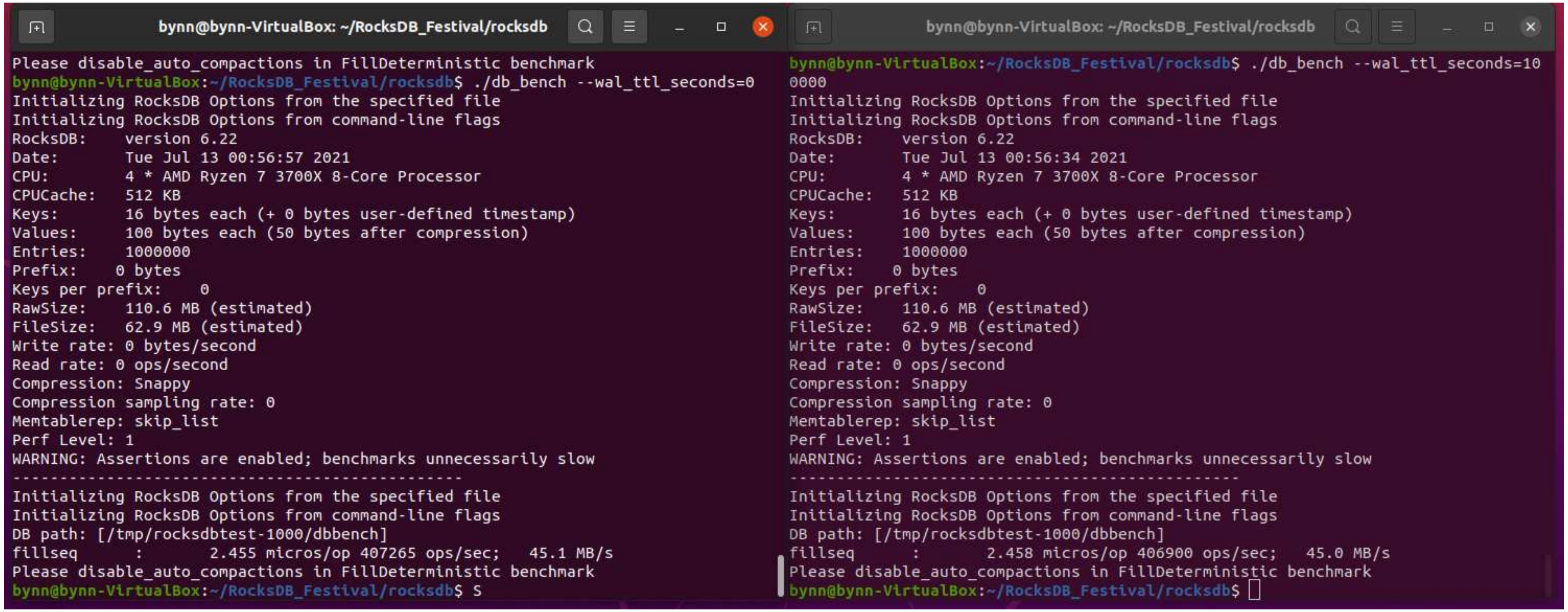


```
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$ ./db_bench --wal_bytes_per_sync=
10
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 6.22
Date:       Tue Jul 13 00:29:36 2021
CPU:        4 * AMD Ryzen 7 3700X 8-Core Processor
CPUCache:   512 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: skip_list
Perf Level: 1
WARNING: Assertions are enabled; benchmarks unnecessarily slow
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/tmp/rocksdbtest-1000/dbbench]
fillseq      :       3.167 micros/op 315721 ops/sec;   34.9 MB/s
Please disable_auto_compactions in FillDeterministic benchmark
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$
```

```
Please disable_auto_compactions in FillDeterministic benchmark
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$ ./db_bench
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 6.22
Date:       Tue Jul 13 00:30:24 2021
CPU:        4 * AMD Ryzen 7 3700X 8-Core Processor
CPUCache:   512 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: skip_list
Perf Level: 1
WARNING: Assertions are enabled; benchmarks unnecessarily slow
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/tmp/rocksdbtest-1000/dbbench]
fillseq      :       2.475 micros/op 404081 ops/sec;   44.7 MB/s
Please disable_auto_compactions in FillDeterministic benchmark
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$
```

RocksDB Festival : db_bench

- RocksDB db_bench 설치 및 실험 결과
  - ✓ --wal_ttl_seconds=[int value]


bynn@bynn-VirtualBox: ~/RocksDB_Festival/rocksdb
bynn@bynn-VirtualBox: ~/RocksDB_Festival/rocksdb

```
Please disable_auto_compactions in FillDeterministic benchmark
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$ ./db_bench --wal_ttl_seconds=0
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 6.22
Date:       Tue Jul 13 00:56:57 2021
CPU:        4 * AMD Ryzen 7 3700X 8-Core Processor
CPUCache:   512 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: skip_list
Perf Level: 1
WARNING: Assertions are enabled; benchmarks unnecessarily slow
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/tmp/rocksdbtest-1000/dbbench]
fillseq      :       2.455 micros/op 407265 ops/sec;   45.1 MB/s
Please disable_auto_compactions in FillDeterministic benchmark
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$ S
```

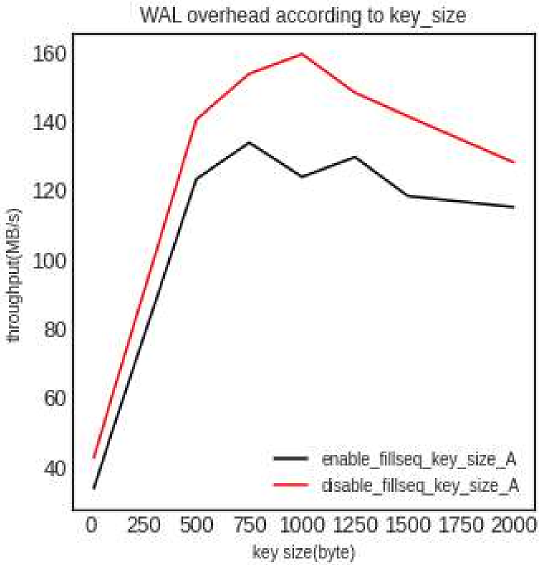
```
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$ ./db_bench --wal_ttl_seconds=10
0000
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 6.22
Date:       Tue Jul 13 00:56:34 2021
CPU:        4 * AMD Ryzen 7 3700X 8-Core Processor
CPUCache:   512 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: skip_list
Perf Level: 1
WARNING: Assertions are enabled; benchmarks unnecessarily slow
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/tmp/rocksdbtest-1000/dbbench]
fillseq      :       2.458 micros/op 406900 ops/sec;   45.0 MB/s
Please disable_auto_compactions in FillDeterministic benchmark
bynn@bynn-VirtualBox:~/RocksDB_Festival/rocksdb$ 
```

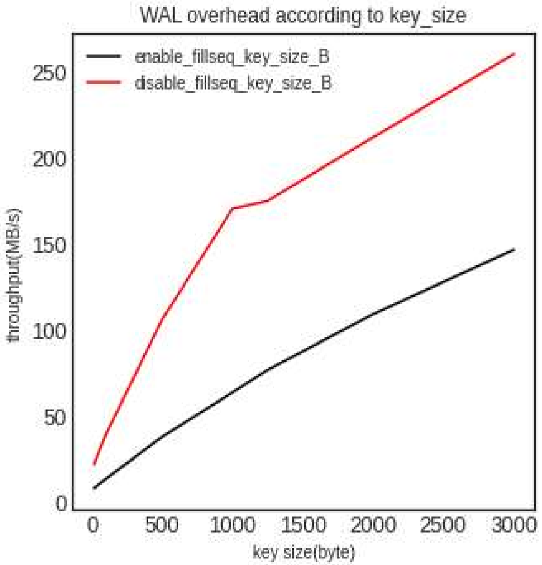
RocksDB Festival : db_bench

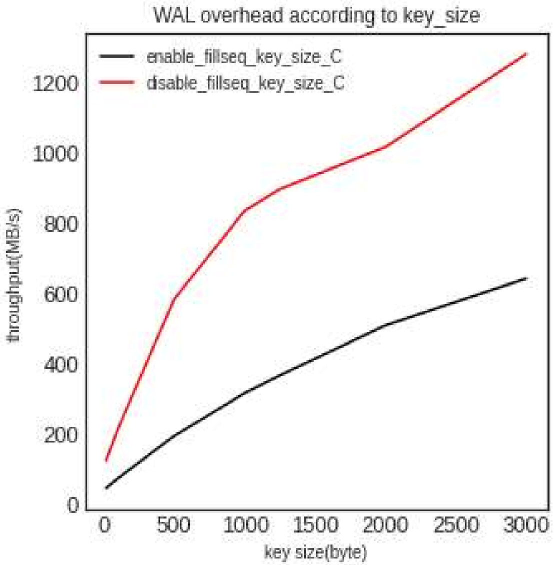
RocksDB db_bench 설치 및 실험 결과

✓ WAL Overhead 측정 실험

- RocksDB에서 WAL 로깅에 의한 overhead를 측정한다.
- Overhead : WAL 사용 유무에 대한 Throughput(ops/sec(MB/s)) 의 차이
- db_bench options
  - --disable_wal=[boolean]
  - --key_size=[int value]

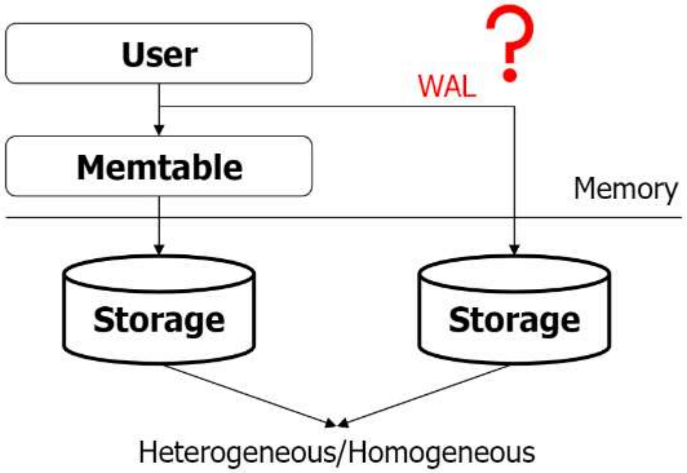






RocksDB Festival : WAL

- 선택한 주제와 선택한 이유
  - ✓ WAL(Write-Ahead-Log)
  - ✓ Every update to RocksDB is written to WAL on disk.
  - ✓ In the event of a failure, write ahead logs can be used to completely recover the data in the memtable, which is necessary to restore the database to the original state.
  - ✓ In the default configuration, RocksDB guarantees process crash consistency by flushing the WAL after every user write.



RocksDB Festival

- 이번 스터디 동안 기대하는 결과

1. WAL의 구조에 따른 오버헤드 분석
2. WAL 오버헤드를 줄이기 위한 방법 고안

   > WAL format, total size of WAL, compression

   > system failure, system crash prediction

Discussion