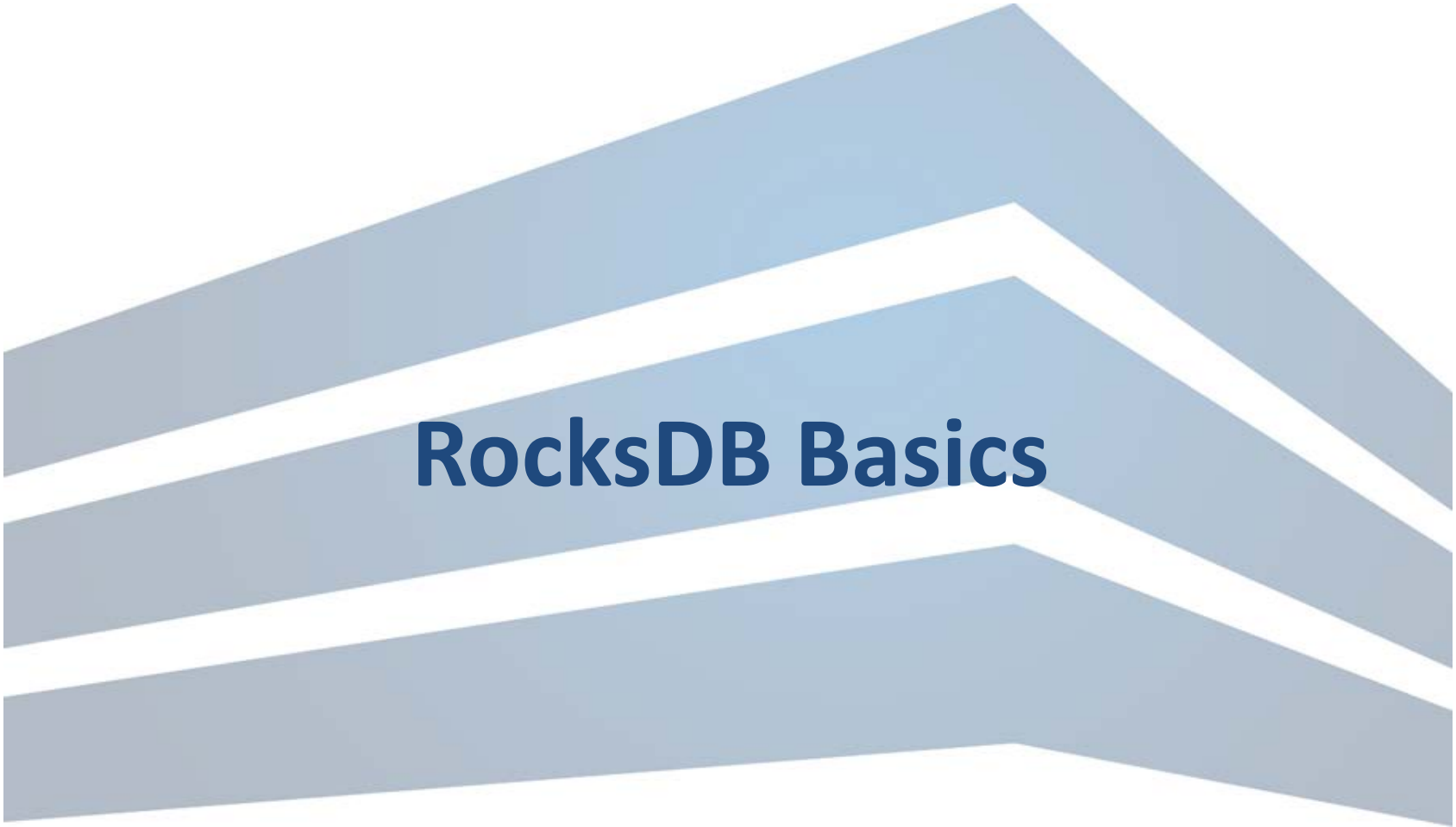
KVRocks: Alternative RocksDB Implementation

10/24/2018

Samsung Semiconductor Inc.
Sr. Director/Principal Engineer
YANG SEOK KI
yangseok.ki@samsung.com

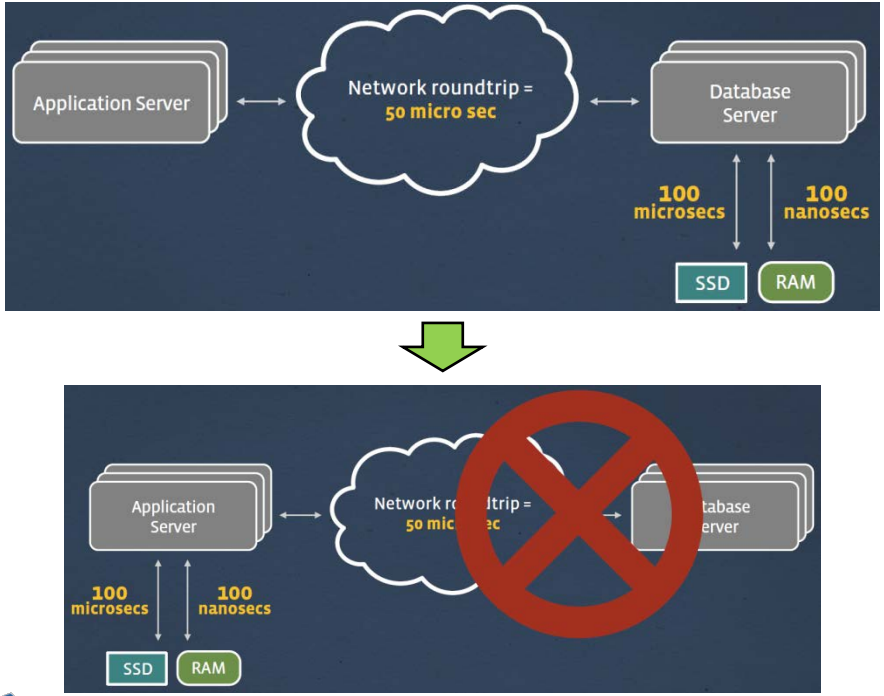
Outline

- RocksDB Basics
- Project Goal & Scope
- KVRocks Core Engine Overview
- MyRocks

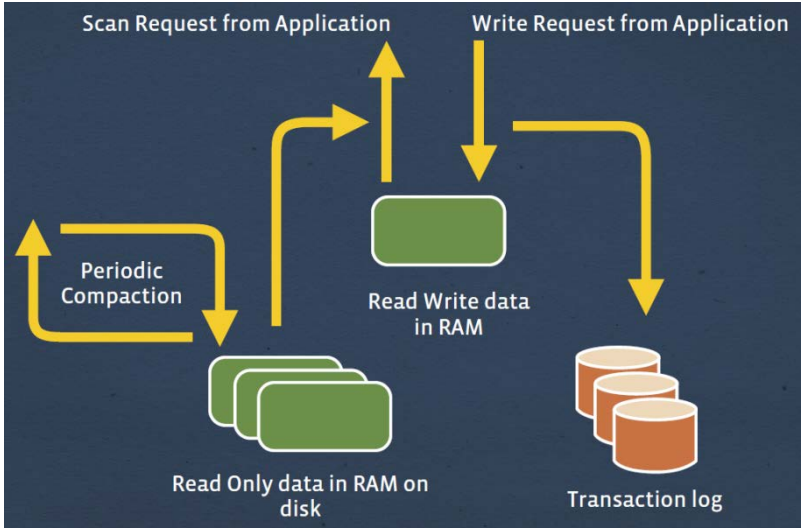

RocksDB Basics

RocksDB: Key Value Database

- **Application database**
  - Eliminate network roundtrip for data access

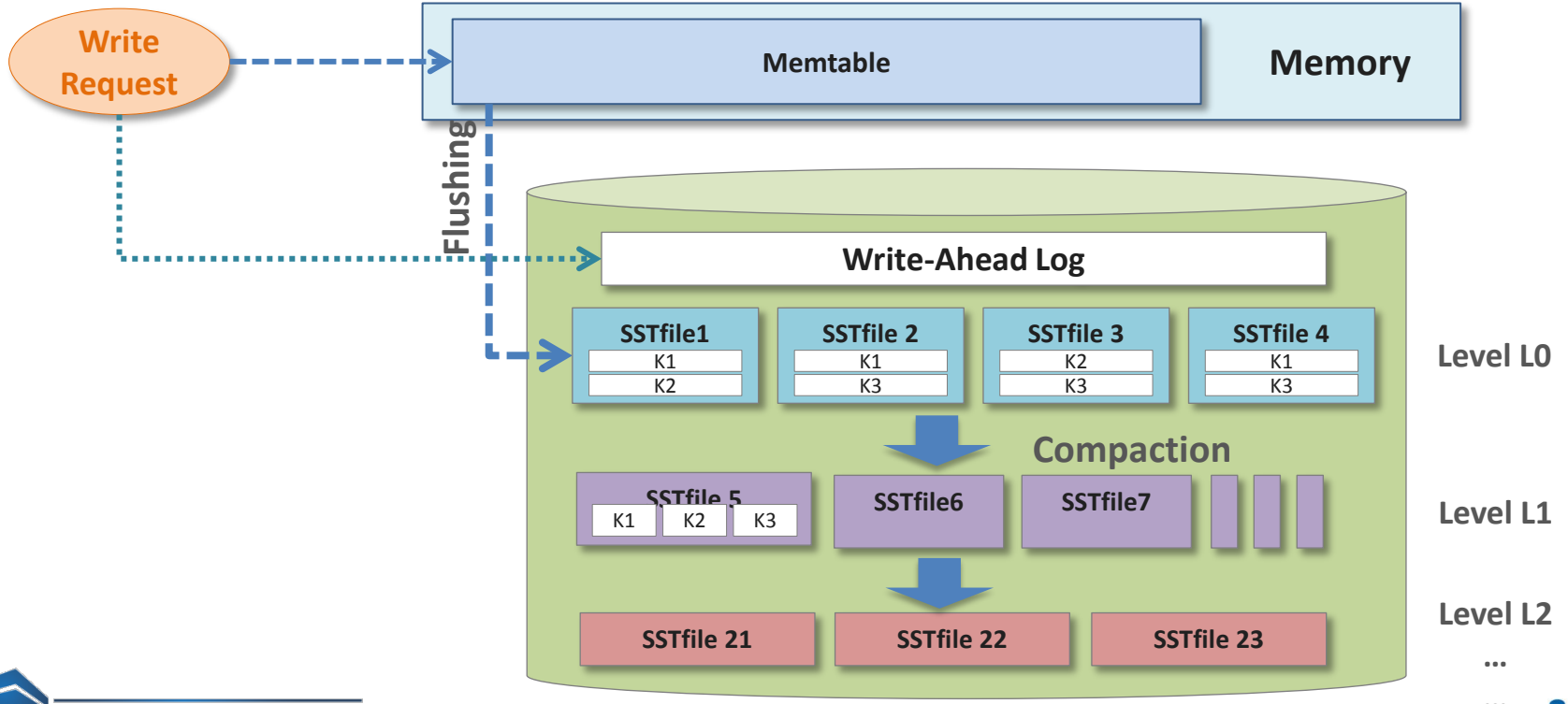


- **Write-optimized database**
  - Fast append in foreground, slow tree restructuring task in background



RocksDB Performance Challenges

- **IO amplification due to background compaction**
  - Performance
  - SSD endurance

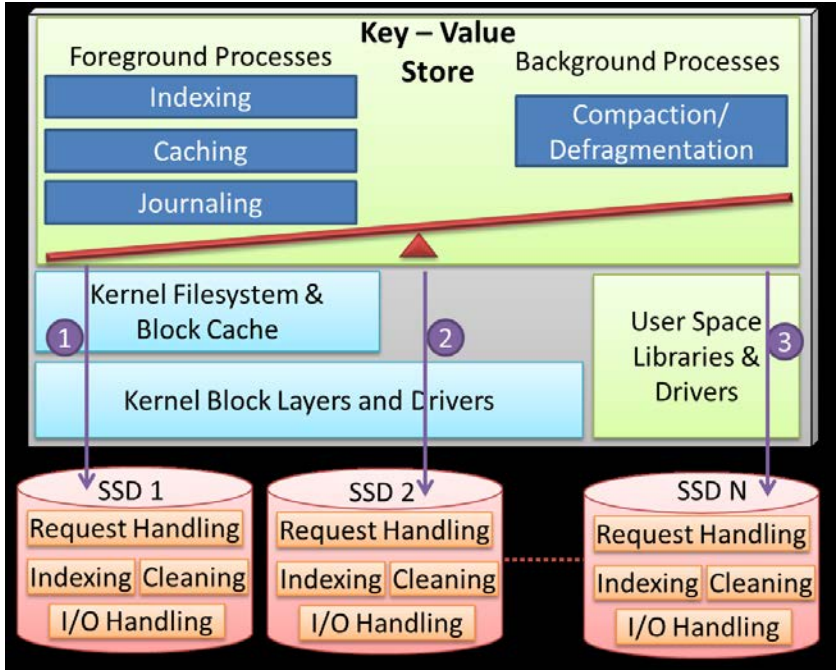


Project Goal & Scope

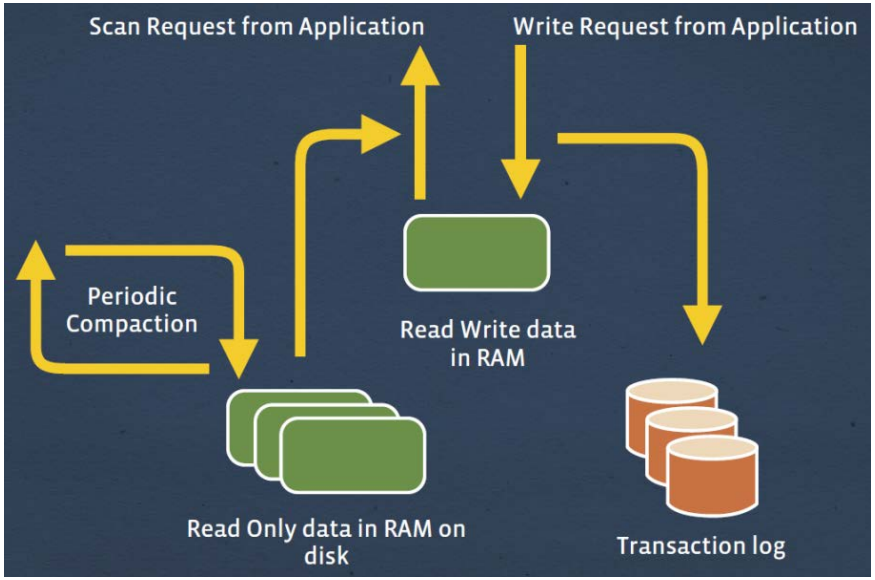
KVRocks Goals

- **Drop-in replacement**
  - RocksDB-compatible for easy adoption
- **High Performance**
  - Balance between foreground and background tasks
  - No application-level compaction
  - No WAL (Write Ahead Log)

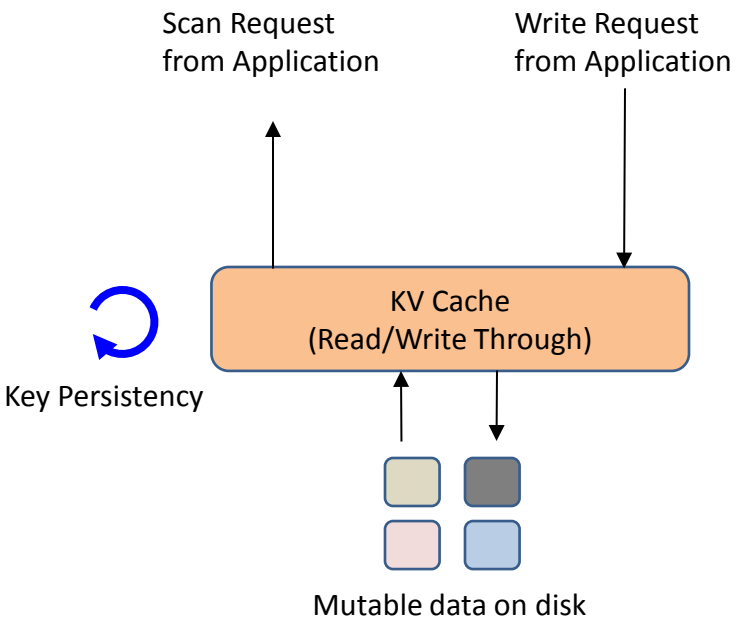
Foreground	Background
SKTable Read	SKTable Read/Write/Flush
Synchronous Put	Asynchronous Put
Get	Prefetching
Indexing	No compaction
Caching	
No WAL	



Key Architecture Differences



LevelDB/RocksDB



KVRocks

Feature Comparison (KvRocks Ver. 1.0, 2018)

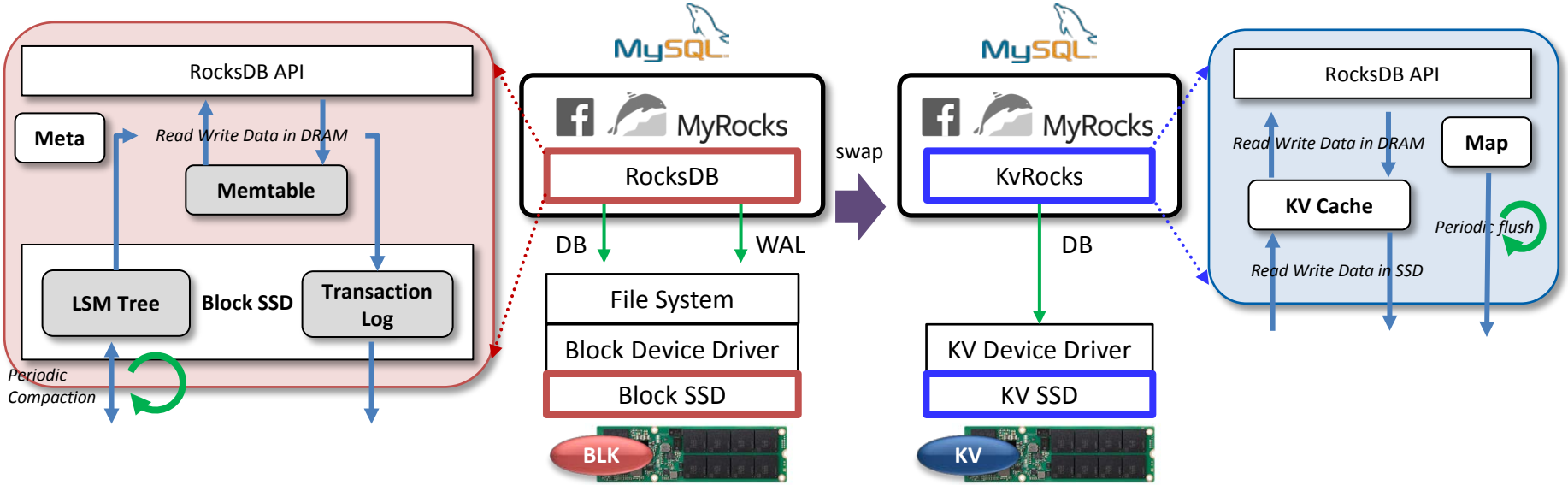
Feature	LevelDB	RocksDB	KVSSD	KvRocks Ver. 1.0 (2018)
Put	Yes	Yes	Yes	Supported
Get	Ephemeral snap-based	Ephemeral snap-based	Current-based	Supported (snap-based)
Delete	Yes	Yes	Yes	Supported
Iterator	Ephemeral snap-based	Ephemeral snap-based	Prefix-based	Supported (snap-based)
Exist	Ephemeral snap-based	Ephemeral snap-based	Current-based	Supported (snap-based)
Range query	Yes	Yes	No	Supported
Snapshot	Ephemeral	Ephemeral	No	Supported (ephemeral)
Comparator	Per DB	Per column family	No	Supported (Per DB)
Transaction	Yes	Yes	No	Supported
Data Compression	Yes	Yes	No	Supported
CRC	Yes	Yes	No	Supported
Column Family	No	4K (MyRocks)	No	Supported (2K)
TTL	No	Yes	No	Supported

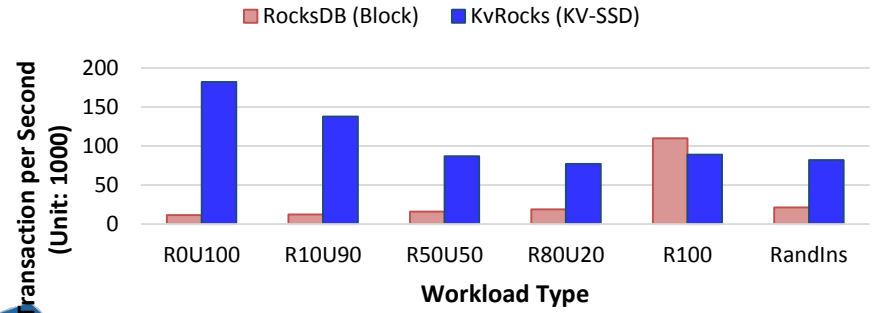
Feature Comparison (KvRocks Ver. 1.0, 2018)

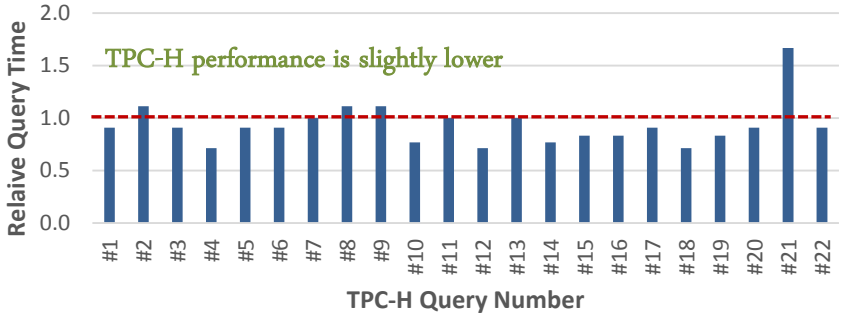
Feature	LevelDB	RocksDB	KVSSD	KvRocks Ver. 1.0 (2018)
Checkpoint	No	Yes	No	Yes (11/2018)
Single Delete	No	Yes	No	Supported
GetApproximateSize	No	Yes	No	Supported
Full Backup	No	Yes	No	Yes (11/2018)
Incremental Backup	No	Yes	No	No (2.0)
Replication	No	Yes	No	No (2.0)
Multiple DBs per process	No	Yes	No	No (2.0)
Non-Blocking database access	No	Yes	No	No (2.0)
Rate Limiter	No	Yes	No	Yes (TechDay)
Backup & Restoring	No	Yes	No	Yes (11/2018)
Simulation Cache	No	Yes	No	Yes (TechDay)
Tailing Iterator	No	Yes	No	Yes (12/2018)
Prefix_extractor	No	Yes	No	Yes (12/2018)
ReadOnly Mode Open	No	Yes	No	Yes (12/2018)


MyRocks

KvRocks Architecture







* Testing was done on a server with 2 x Intel Xeon E5-2695 v4 servers with 128 GB of DRAM, and a PM983 (in block or KV mode) SSD

DBBench

```
## kvdb
cd ~/src/kvdb/gflags-v2.2.1
set +e
mkdir build
cd build
cmake ..
make

cd ~/src/kvdb
set +e
mkdir build
set -e
cd build
make clean
cmake .. -DCMAKE_BUILD_TYPE=RelWithDebInfo  -DDOWNLOAD_BOOST=1 \
         -DWITH_BOOST=../..  -DCMAKE_INSTALL_PREFIX=~/install/kvdb
make -j $(nproc)
```

```
cp -f ~/src/kvdb/build/db_bench ~/src/kvdb/build.save/
python ./rocksdb_bench.py --dbpath=/tmp/tmpkvdb --dbbench_path=~/src/kvdb/build.save --dbtype=kvdb --out_dir kvdb4k --automatic #--runall
```

MyRocks based on KVRocks

```
cd ~/src/kv-percona-server
set +e
mkdir build
cd build
#make clean
cmake ..  -DCMAKE_BUILD_TYPE=RelWithDebInfo -
DINSDB_ROOT_DIR=../../src/insdb -DKVDB_ROOT_DIR=../../src/kvdb  -
DDOWNLOAD_BOOST=1 -DWITH_BOOST=../..  -DENABLE_DOWNLOADS=1 -
DCMAKE_INSTALL_PREFIX=~/install/myrocks.kv

make -j $(nproc)

set -e
cd ~/src/kv-percona-server/build/storage/rocksdb
make clean
make -j $(nproc)
sudo cp ~/conf/kvmysqld.cnf /etc/mysql/percona-server.conf.d/mysqld.cnf
sudo cp ~/src/kv-percona-server/build/storage/rocksdb/ha_rocksdb.so
/usr/lib/mysql/plugin/ha_rocksdb.so
```

```
## copied from https://github.com/facebook/mysql-5.6/wiki/my.cnf-tuning
#
plugin-
load=rocksdb=ha_rocksdb.so;rocksdb_cfstats=ha_rocksdb.so;rocksdb_dbstats=ha_rocksdb.so;rocksdb_perf_context=ha_rocksdb.so;rocksdb_perf_conte
xt_global=ha_rocksdb.so;rocksdb_cf_options=ha_rocksdb.so;rocksdb_compaction_stats=ha_rocksdb.so;rocksdb_global_info=ha_rocksdb.so;rocksdb_dd
l=ha_rocksdb.so;rocksdb_index_file_map=ha_rocksdb.so;rocksdb_locks=ha_rocksdb.so;rocksdb_trx=ha_rocksdb.so

#
default-storage-engine=rocksdb
#skip-innodb
default-tmp-storage-engine=MyISAM
binlog_format=ROW
collation-server=latin1_bin
transaction-isolation=READ-COMMITTED

rocksdb_lock_wait_timeout=288000
rocksdb_max_open_files=-1
rocksdb_max_background_jobs=8
rocksdb_max_total_wal_size=4G
rocksdb_block_size=16384
rocksdb_block_cache_size=32G
rocksdb_table_cache_numshardbits=6

# rate limiter
rocksdb_bytes_per_sync=4194304
rocksdb_wal_bytes_per_sync=4194304
rocksdb_rate_limiter_bytes_per_sec=104857600 #100MB/s. Increase if you're running on higher spec machines

# triggering compaction if there are many sequential deletes
rocksdb_compaction_sequential_deletes_count_sd=1
rocksdb_compaction_sequential_deletes=199999
rocksdb_compaction_sequential_deletes_window=200000

rocksdb_default_cf_options=write_buffer_size=128m;target_file_size_base=32m;max_bytes_for_level_base=512m;level0_file_num_compaction_trigger
=4;level0_slowdown_writes_trigger=10;level0_stop_writes_trigger=15;max_write_buffer_number=4;compression_per_level=kLZ4Compression;bottommos
t_compression=kZSTD;compression_opts=-
14:1:0;block_based_table_factory={cache_index_and_filter_blocks=1;filter_policy=bloomfilter:10:false;whole_key_filtering=1};level_compaction
_dynamic_level_bytes=true;optimize_filters_for_hits=true;compaction_pri=kMinOverlappingRatio
```

Thank You!

kvssd@ssi.samsung.com

https://github.com/OpenMPDK/KVSSD