MyRocks
RocksDB storage engine for MySQL

Mark Callaghan

Member of Technical Staff, Facebook

MySQL is thriving

- Features
- Performance
- Usage
- Community

Why MyRocks?

- Best space efficiency
- Better write efficiency
- Good read efficiency
- Effective with SSD & disk

For an important web-scale workload

- Uses 50% the space vs compressed InnoDB
- Uses 25% the space vs InnoDB
- Write rate to storage is 10% the InnoDB rate

When to consider MyRocks?

- You are using InnoDB
- The database is larger than RAM

Goal is performance similar to InnoDB with much better storage efficiency. Verified with production and benchmarks.

Progress

Achieved in 2016

- Efficient performance
- Deployed
- Started ports to Percona & MariaDB Server

Planned for 2017

- Better documentation
- More production deployments
- Usable in Percona & MariaDB Server
- Performance improvements & results
- Features

Efficiency: RocksDB vs a B-Tree

Space efficiency

- Fragmentation
- Fixed page size
- Per-row metadata
- Key prefix encoding

Write efficiency

- Uses more space = more data to write
- Working set larger than cache
- sizeof(page) / sizeof(row)
- Double write buffer (InnoDB)

Read efficiency

- More data in cache & less data to cache
- Bloom filter
- Spend less on writes, use more for reads
- Read-free index maintenance

Problems: my.cnf options

- Fixed - get good enough performance with default my.cnf
- Set these to get great performance
  rocksdb_block_cache_size
  rocksdb_max_background_compactions

Insert benchmark	inserts/ second	queries/ second
default	13979	11986
block cache	13610	44604
block cache & background threads	86501	44581

Problems: long range scans

- Visible with concurrent, long range scans
- Fixed: problem was memory system contention

Sysbench	range scans / second
InnoDB 5.6.26	6403
old MyRocks	3090
new MyRocks	6093

Problems: group commit

- Binlog crash safety costs 5% to 20% of throughput with MyRocks
- Not fixed yet: design discussion in progress

Problems: large transactions & OOM

The problem

- Uncommitted changes buffered in memory
- Temporarily double-buffered on commit

The solution

- Commit early: rocksdb_commit_in_middle
- Prevent large trx: rocksdb_max_locks (old), rocksdb_max_write_batch_size (new)
- Tolerate large trx: design discussion in progress

Evaluate performance with Linkbench

Throughput, hardware efficiency and QoS

	TPS	iostat r/t	iostat wKB/t	CPU usecs/t	Size (GB)	p99 update (ms)
MyRocks+zlib	**28965**	**1.03**	**1.25**	999	**374**	**1**
InnoDB	21474	1.16	19.70	**914**	14xx	6
InnoDB+zlib	20734	1.07	14.59	1199	880	6

The value of write efficiency

- InnoDB depends more on fast SSD
- MyRocks spends less on writes to enable more reads & writes

Insert benchmark	Fast SSD	Slow SSD
InnoDB 5.7.10	268873	124782
InnoDB 5.6.26	111111	66251
MyRocks	102712	83766

Linkbench	Fast SSD	Slow SSD	Disk
InnoDB 5.6.26	21414	10143	414
MyRocks	28965	23484	2195

Thank you

myrocks.io
rocksdb.org
mongorocks.org

smalldatum.blogspot.com
twitter.com/markcallaghan