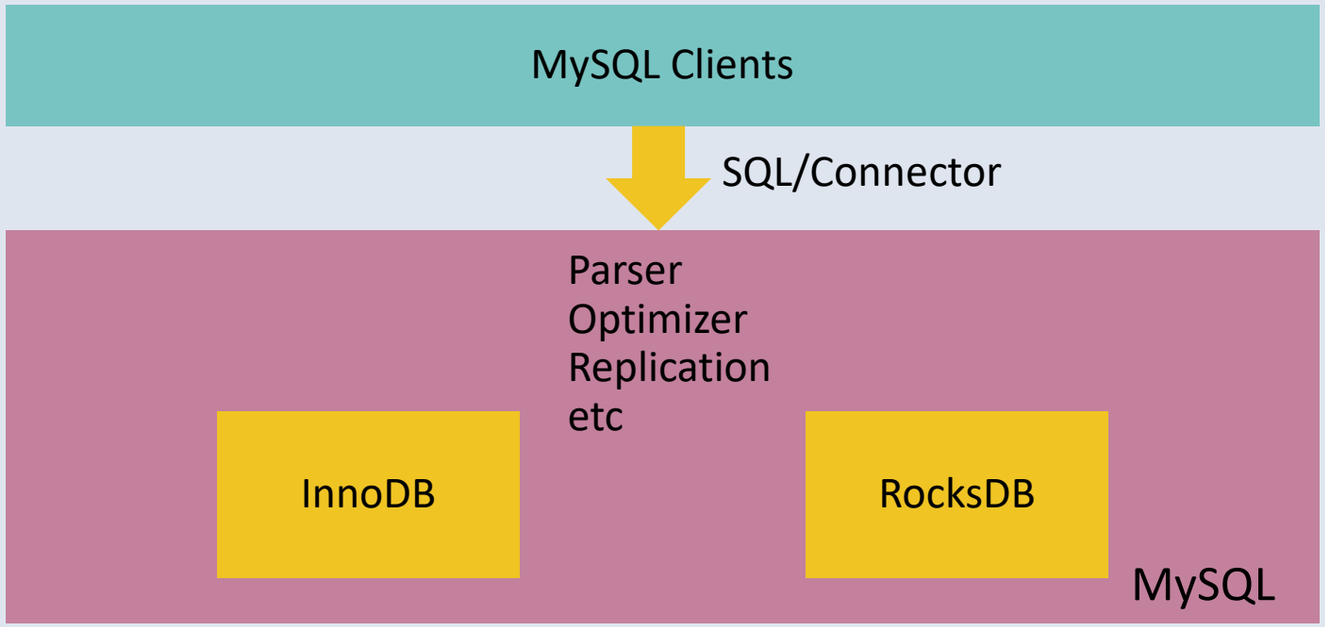
facebook

MyRocks in the Real World

Yoshinori Matsunobu
Production Engineer / MySQL Tech Lead, Facebook
Nov 2018

What is MyRocks

- MySQL on top of RocksDB (Log-Structured Merge Tree Database)
- Open Source, distributed from MariaDB and Percona as well

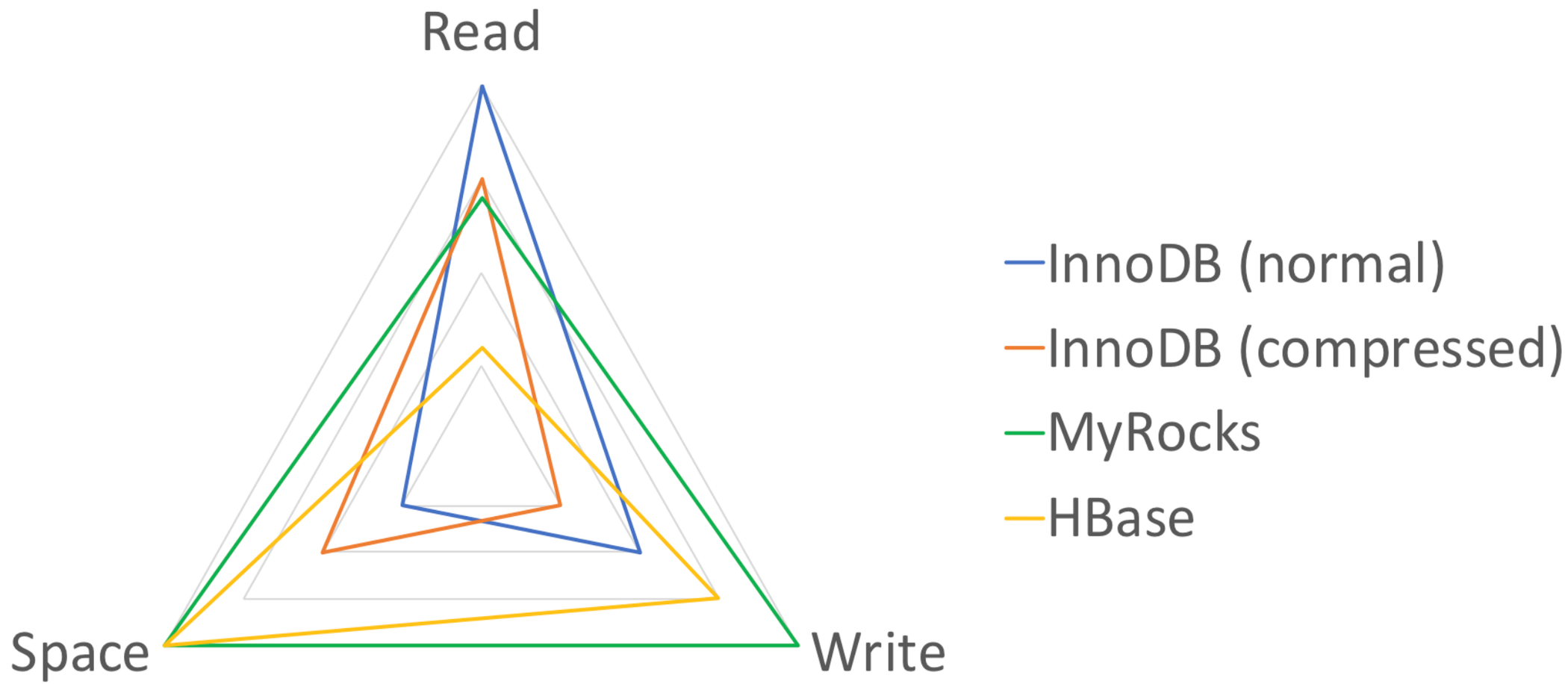


http://myrocks.io/

Read, Write and Space Performance/Efficiency

- Pick two of them
- InnoDB/B-Tree favors Read at cost of Write and Space
- For large scale database on Flash, Space is important
- Read inefficiency can be mitigated by Flash and Cache tiers
- Write inefficiency can not be easily resolved
- Implementations matter (e.g. ZSTD > Zlib)

Read, Write and Space Performance/Efficiency

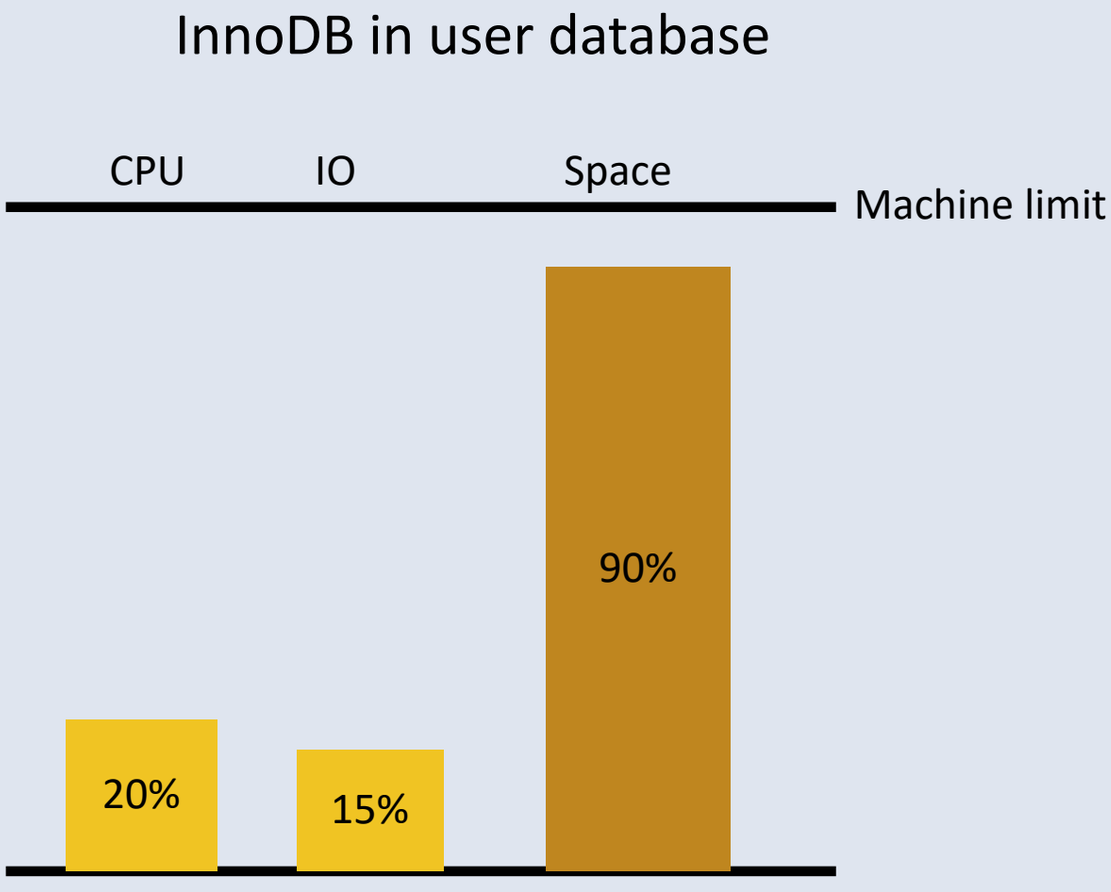


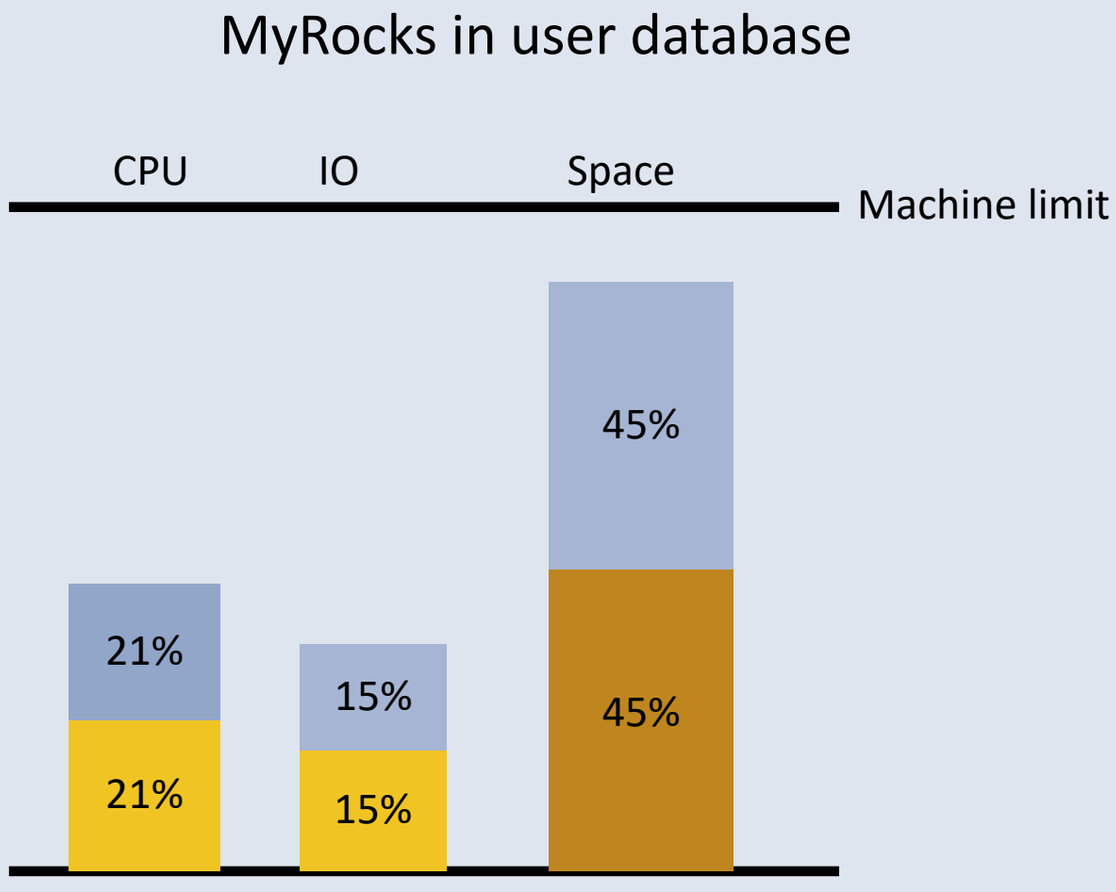
- Compressed InnoDB is roughly 2x smaller than uncompressed InnoDB, MyRocks/HBase are 4x smaller
- Decompression cost on read is non zero. It matters less on i/o bound workloads
- HBase vs MyRocks perf differences came from implementation efficiencies rather than database architecture

UDB – Migration from InnoDB to MyRocks

- UDB: Our largest user database that stores social activities
- Biggest motivation was saving space
  - 2X savings vs compressed InnoDB, 4X vs uncompressed InnoDB
- Write efficiency was 10X better
- Read efficiency was no worse than X times
- Could migrate without rewriting applications

User Database at Facebook

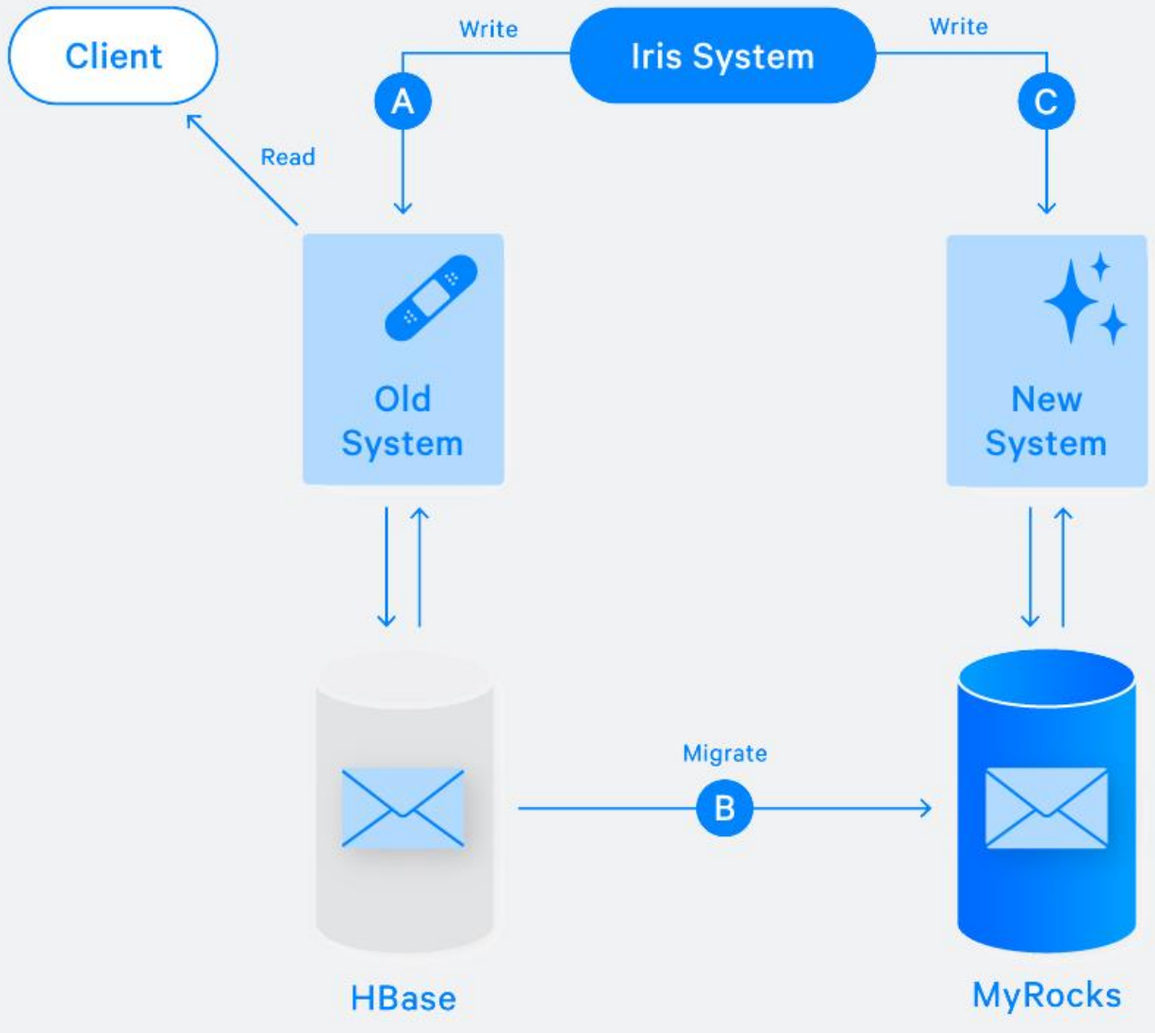




MyRocks on Facebook Messaging

- In 2010, we created Facebook Messenger and we chose HBase as backend database
  - LSM database
  - Write optimized
  - Smaller space
  - Good enough on HDD
- Successful MyRocks on UDB led us to migrate Messenger as well
  - MyRocks used much less CPU time, worked well on Flash
  - p95~99 latency and error rates improved by 10X
- Migrated from HBase to MyRocks in 2017~2018

FB Messaging Migration from HBase to MyRocks



Our current status

- Our two biggest database services (UDB and Facebook Messenger) have been reliably running on top of MyRocks
- Efficiency wins : InnoDB to MyRocks
- Performance and Reliability wins : HBase to MyRocks
- Gradually working on migrating long tail, smaller database services to MyRocks

Future Plans

- MySQL 8.0

- Pushing more efficiency efforts
  - Simple read query paths to be much more CPU efficient
  - Working without WAL, engine crash recovery relying on Binlog

- Towards more general purpose database
  - Gap Lock and Foreign Key
  - Long running transactions
  - Online and fast schema changes
  - Mixing MyRocks and InnoDB in the same instance

facebook

MyRocks in the Real World

Vadim Tkachenko, CTO Percona

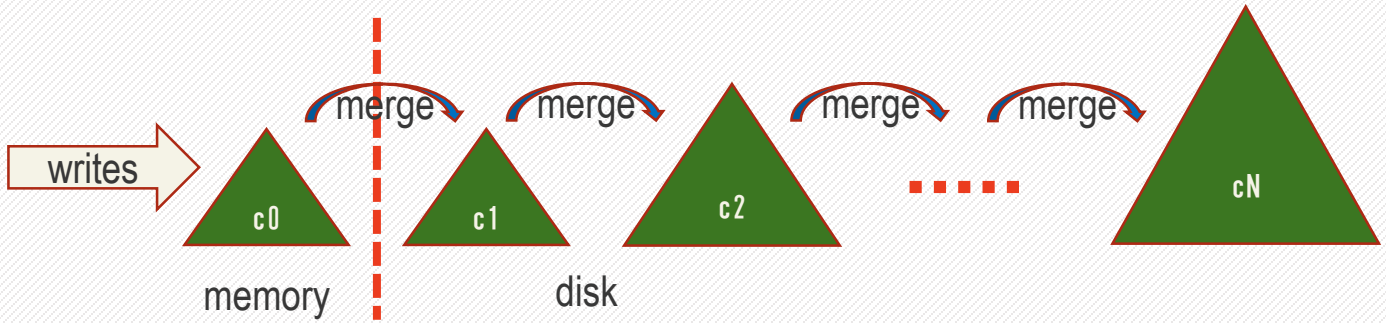
Facebook-scale engine comes with

Percona Server 5.7 and 8.0 RC

PERCONA
Server for MySQL

MyRocks is based on a Log Structured Merge Tree data structure

MyRocks



LSM-tree is Industry accepted

APACHE HBASE

LEVELDB

cassandra

influxdata

ClickHouse

TiDB
A Distributed SQL Database

Cockroach DB

YugaByte DB

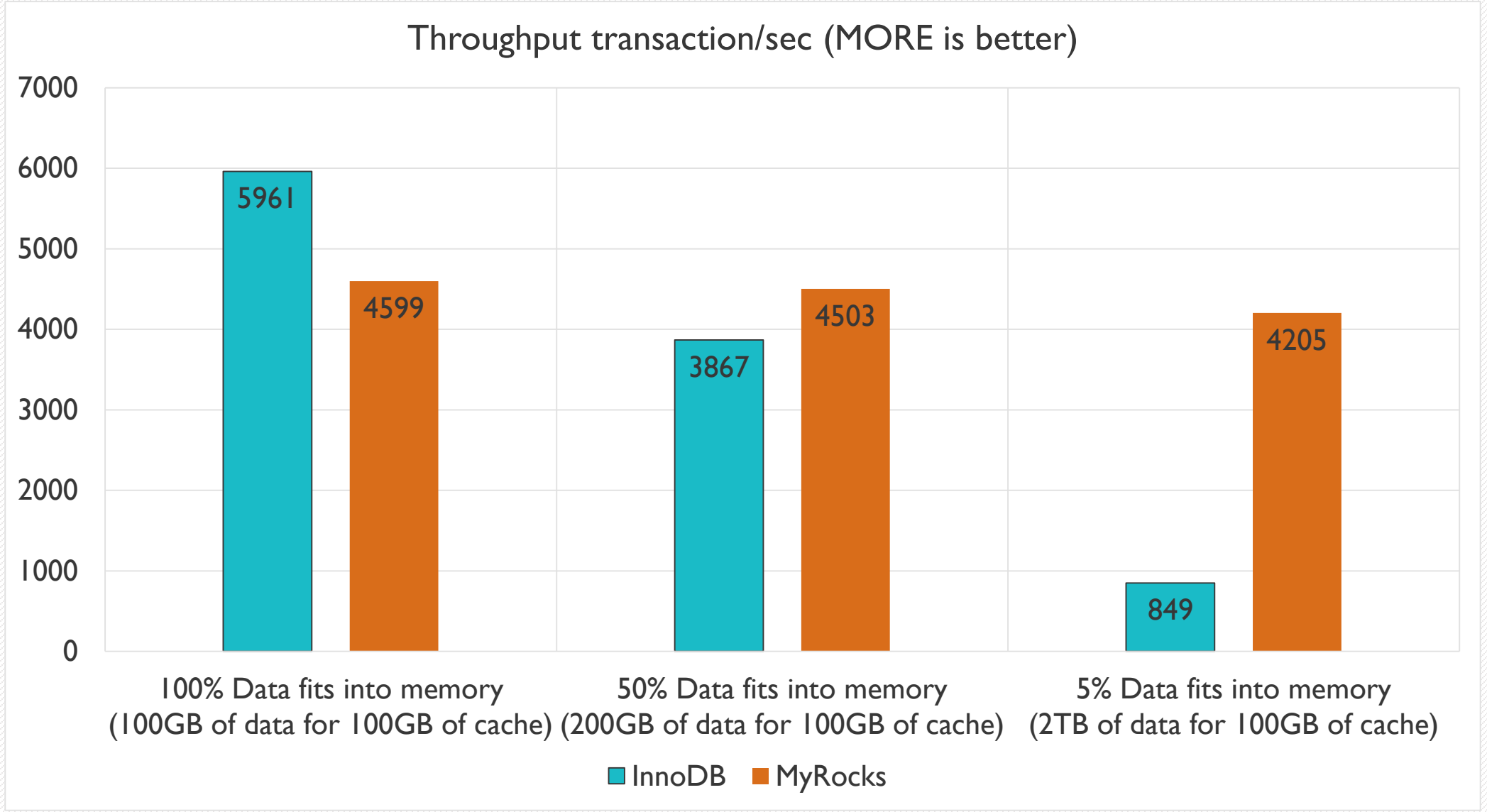
MyRocks is designed for big data sets

MyRocks is Cloud efficient

MyRocks is good for SSD lifetime

MyRocks is compression friendly

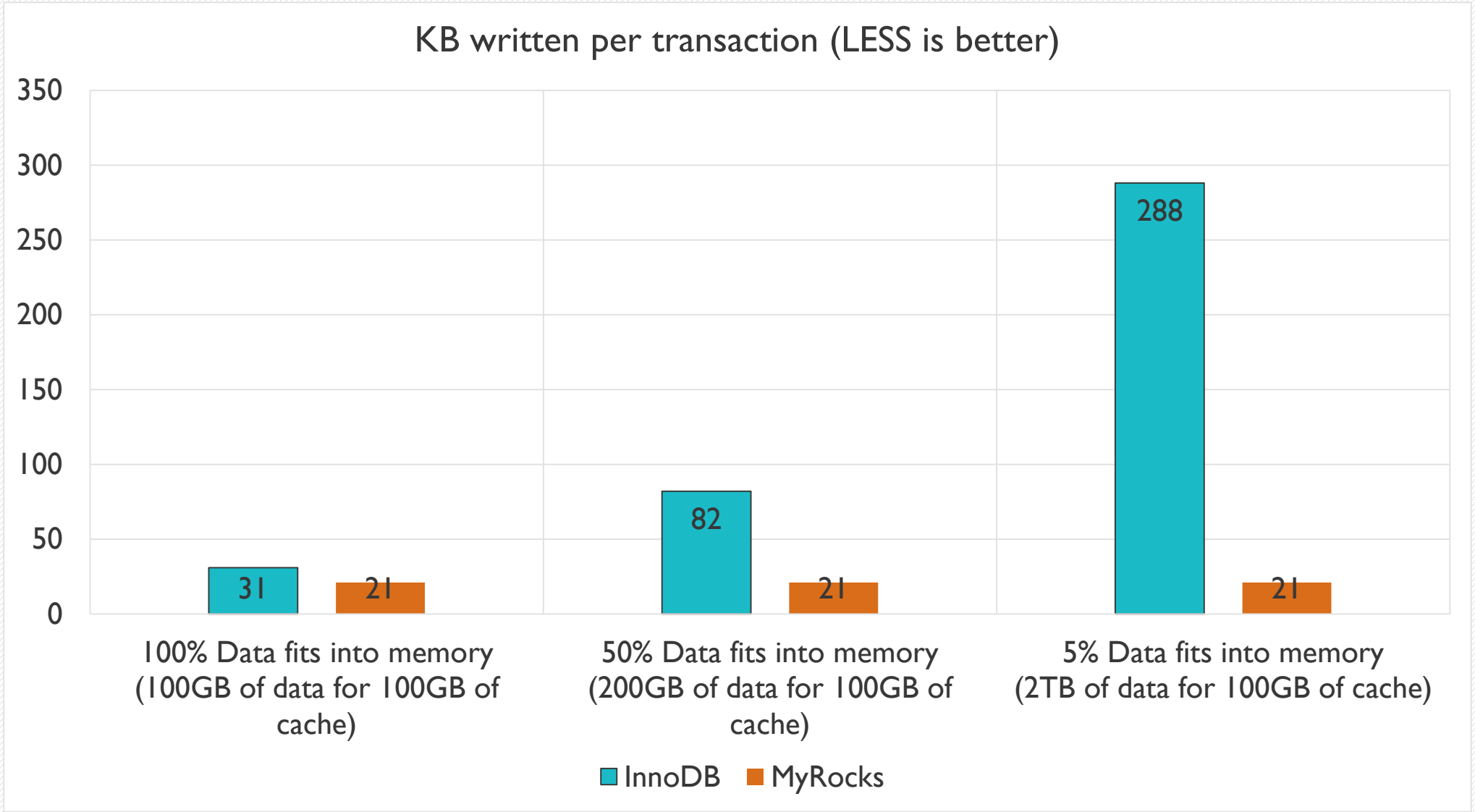
MyRocks is designed for big data sets



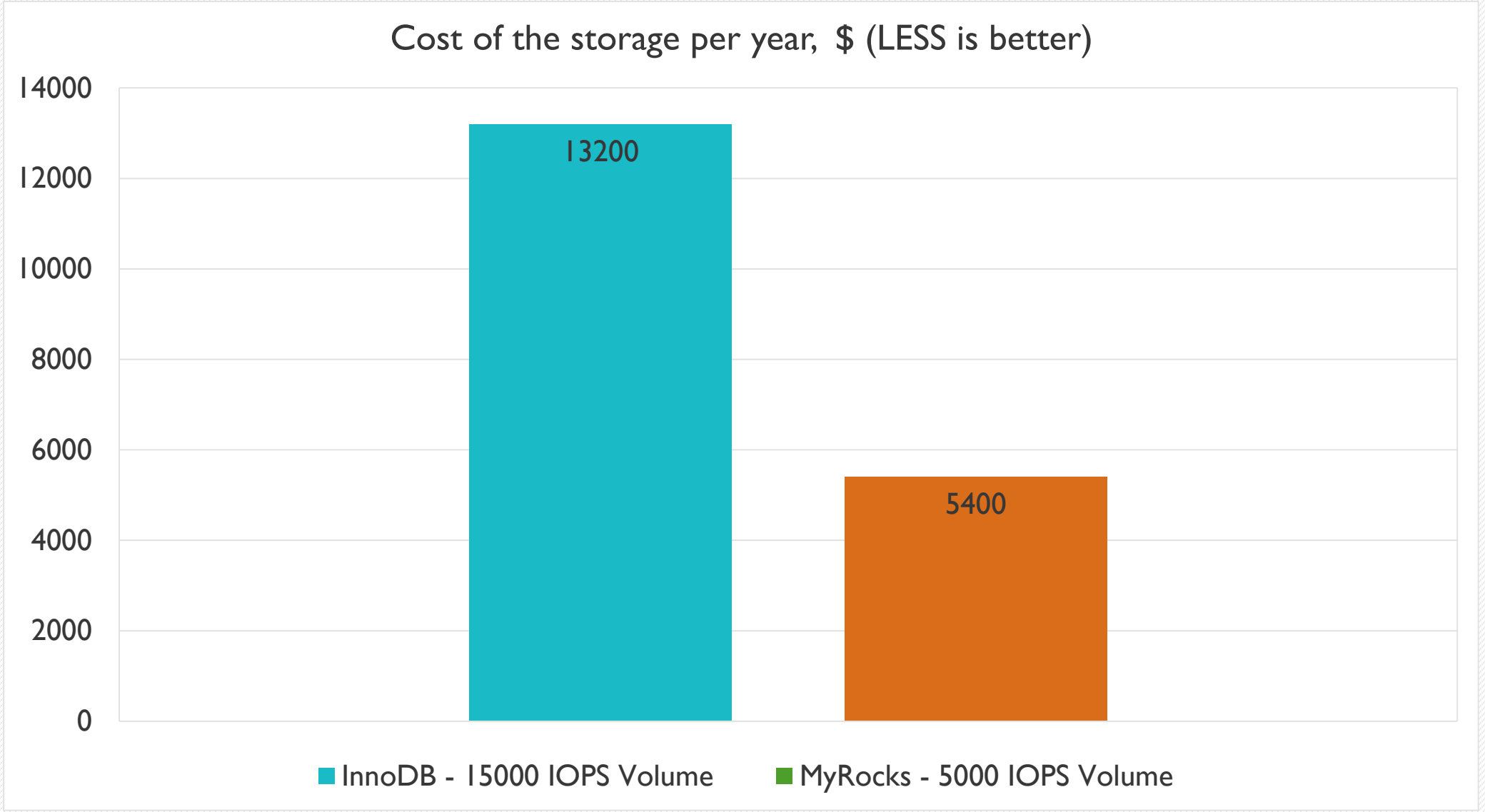
MyRocks is designed for big data sets

MyRocks is Cloud efficient

MyRocks is good for SSD lifetime



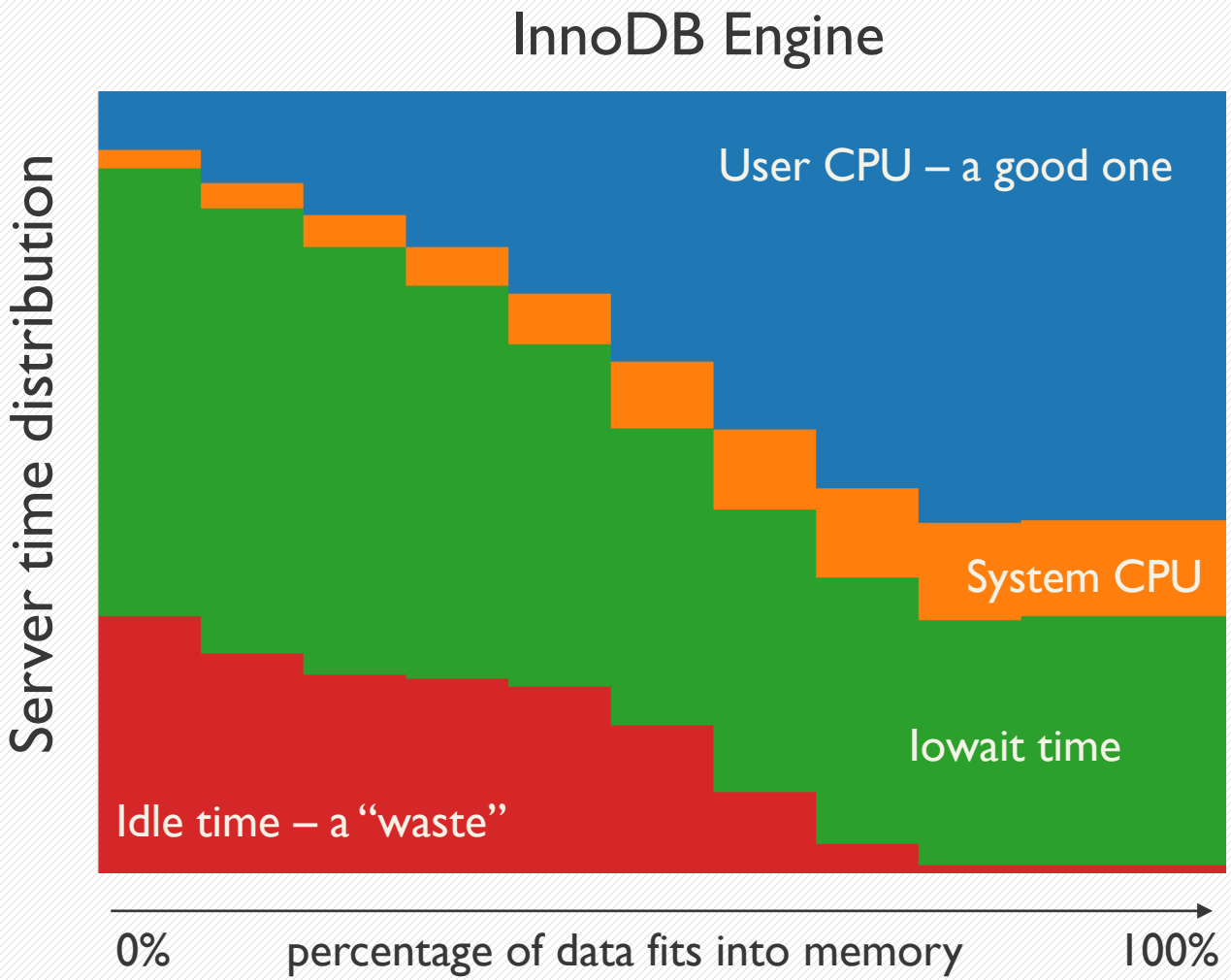
MyRocks is Cloud efficient



Volumes are provisioned to
achieve the same performance

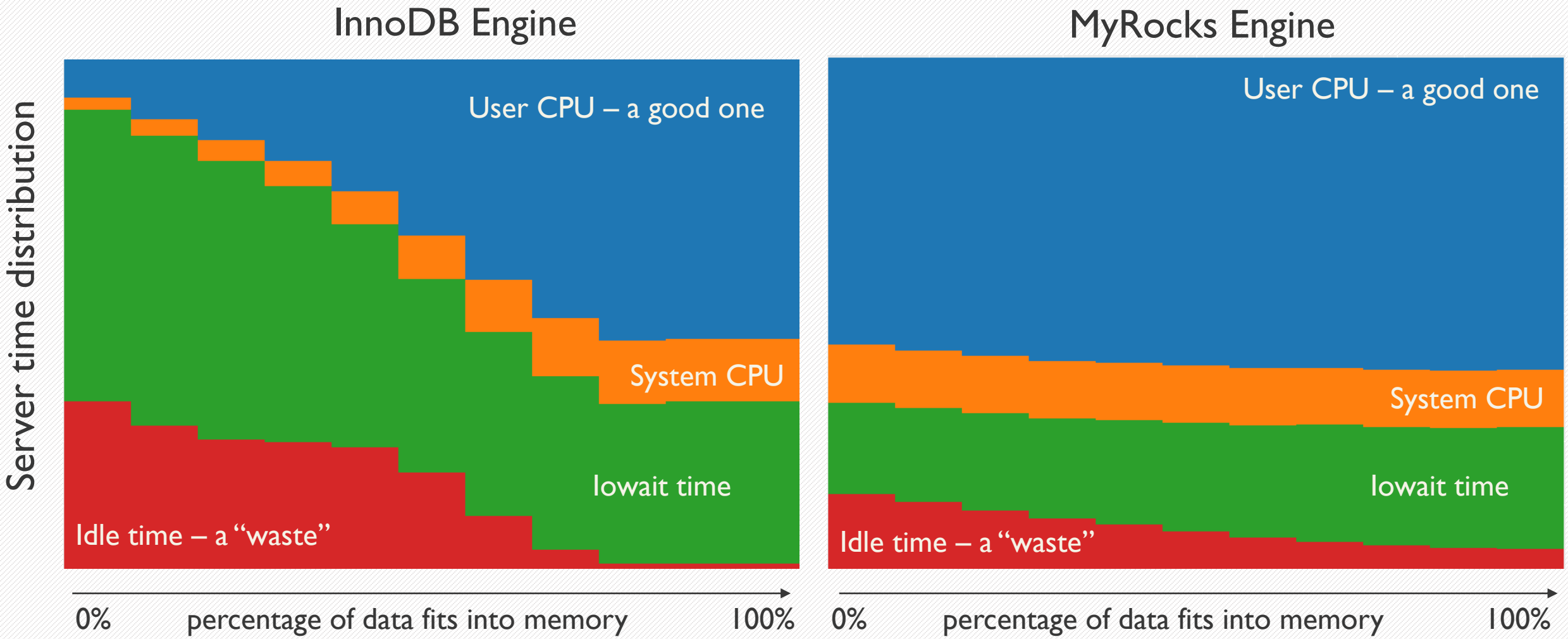
MyRocks is designed for big data sets

MyRocks is Cloud efficient

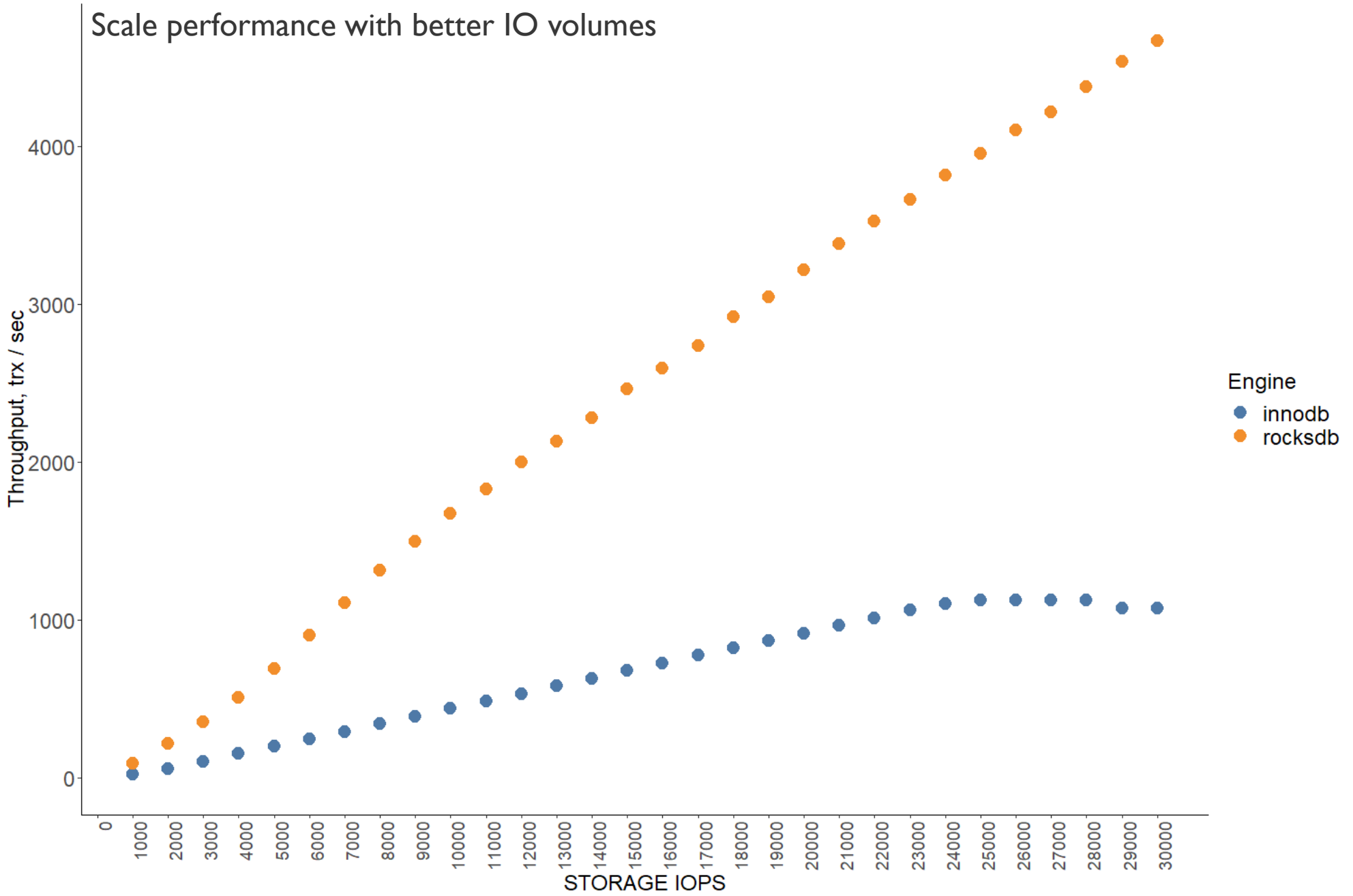


MyRocks is designed for big data sets

MyRocks is Cloud efficient



MyRocks is Cloud efficient



Modern compressions methods

Method	Compression Ratio	Compression speed	Decompression speed
LZ4 (default)	2.1	750 MB/s	3700 MB/s
Zstandard	2.8	470 MB/s	1380 MB/s

So is MyRocks perfect?
Are there downsides?
Want to know more?

My talk today: 12:20pm, room @Jones
Yoshinori's talk: 3:30pm, room @Dax