Methods of benchmarking NoSQL database systems

Ilya Bakulin
webmaster@kibab.com, kibab@FreeBSD.org

SMS Traffic

LVEE 2011

Why benchmarking NoSQL is nessesary

- No guides / FAQs about performance are generally available, or are outdated
- NoSQL systems are actively developed
- Nobody wants to end up with crashed DB in production right before 2-week vacation

Why benchmarking NoSQL is complex

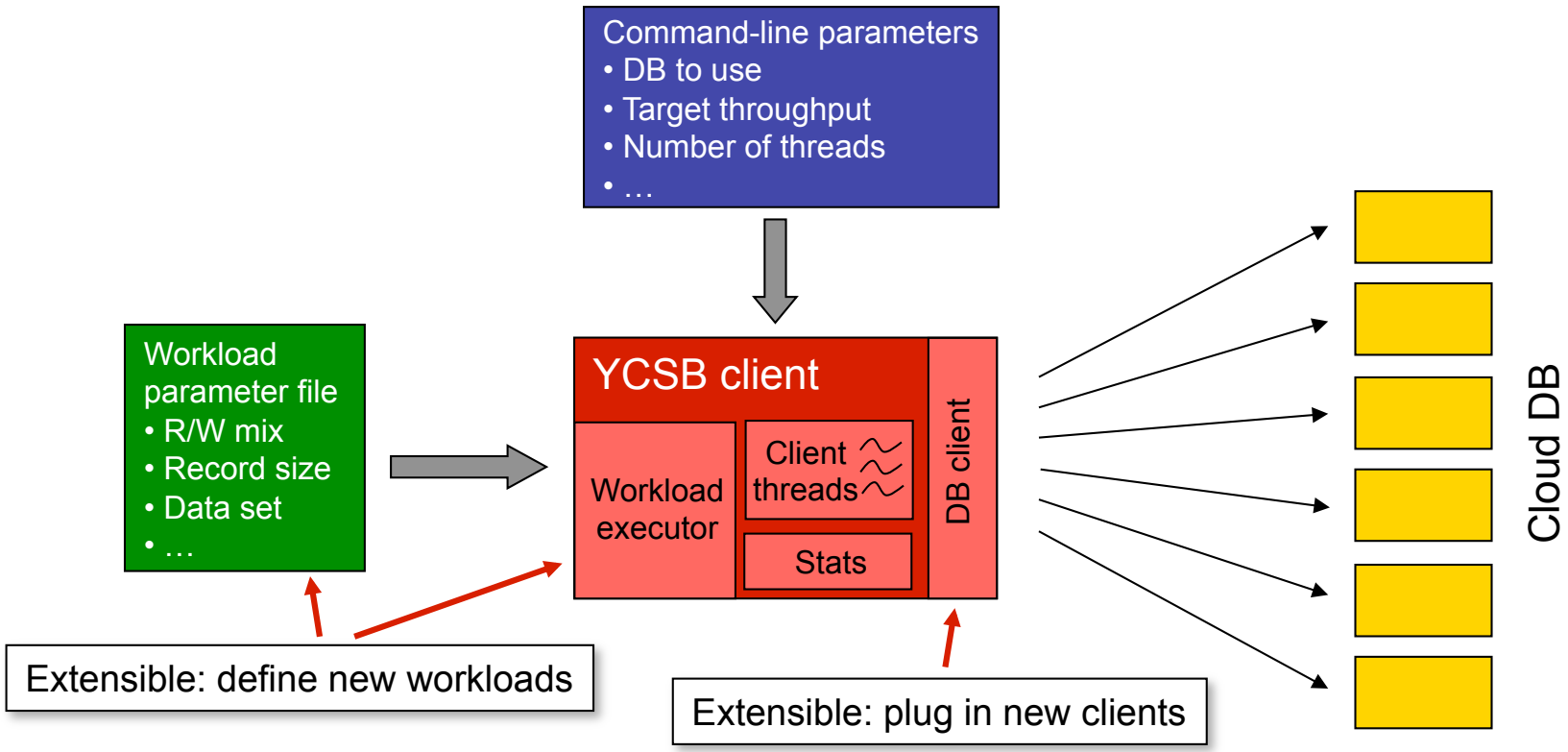
- RDBMS use SQL to provide access to data stored in them, while NOSQL systems don't
- Each NoSQL uses different protocol (Thrift, Memcached-style, own protocols)
- Existing benchmarks require SQL to work with database under inspection.

What is YCSB?

- YCSB stands for **Y**ahoo **C**loud **S**erving **B**enchmark
- Developed by Yahoo! Research group
- Open Source project, hosted on GitHub (178 watchers, 42 forks)

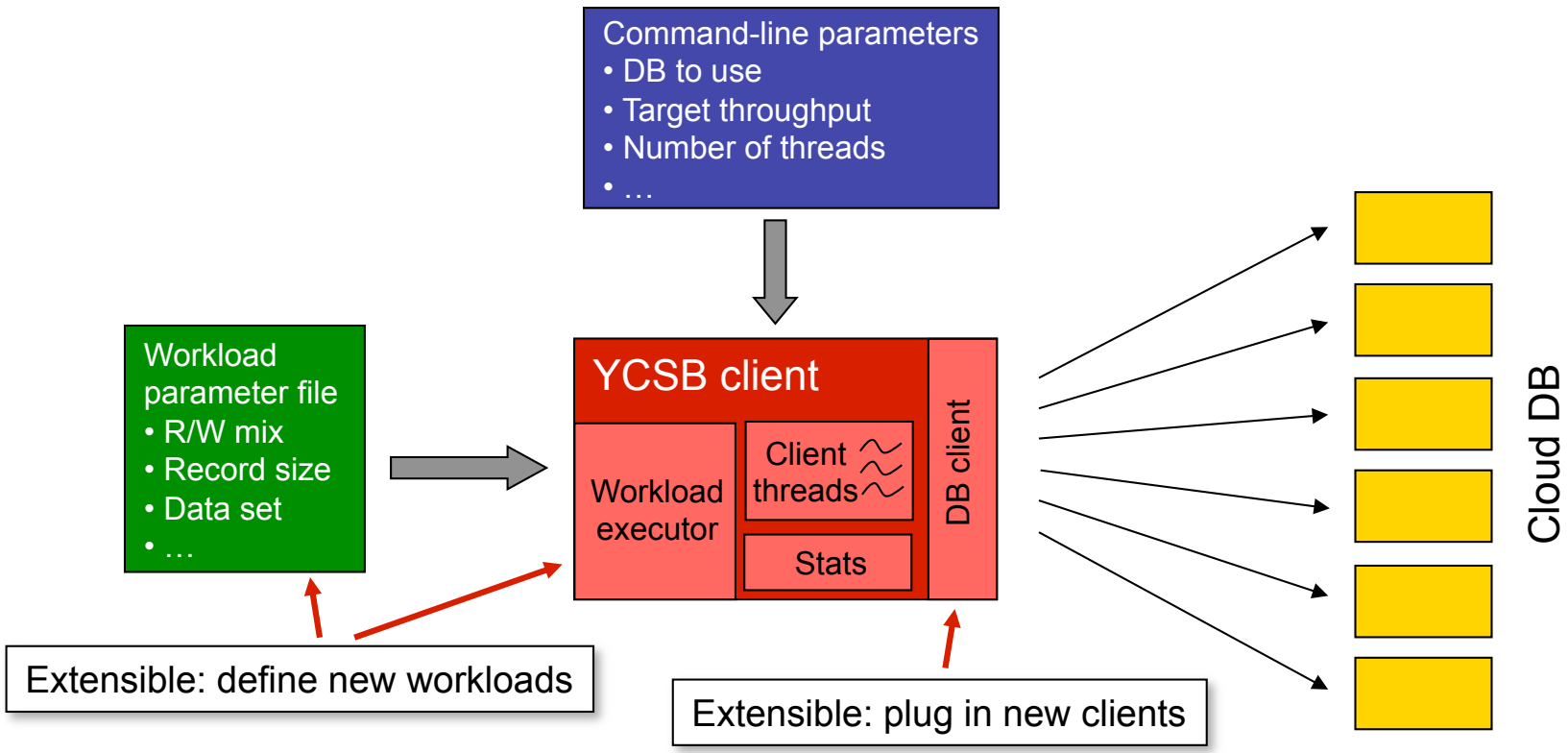
Architecture

- Java application
- Shipped with ready-to-use adapters for several popular Opensource databases



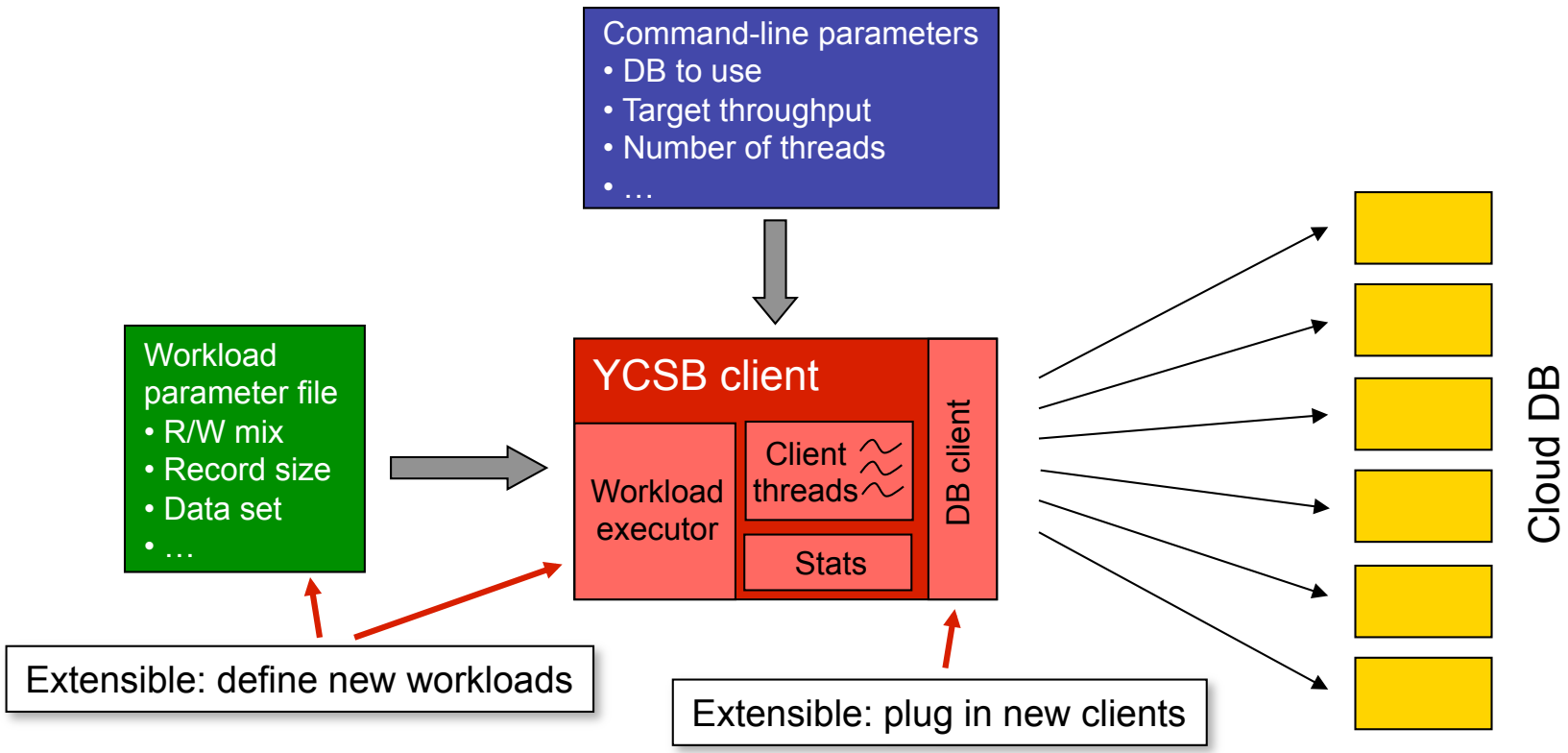
More on DB interface

- Simple operations: INSERT, UPDATE, REPLACE, DELETE, SCAN
- Does not use SQL
- ... but SQL support is avaible through contributed JDBC driver
- ... Even sharding configurations are possible



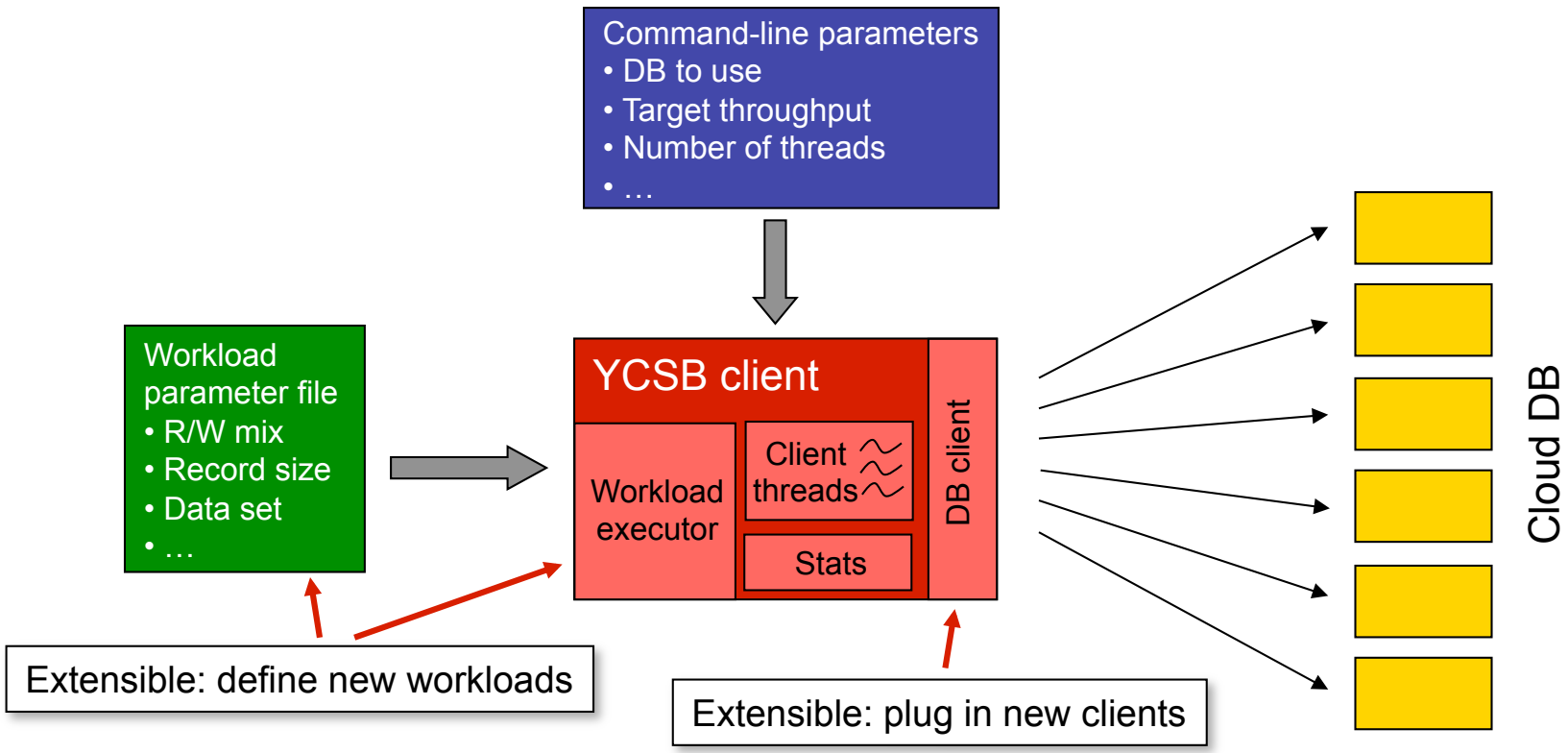
More on DB interface

- Simple operations: INSERT, UPDATE, REPLACE, DELETE, SCAN
- Does not use SQL
- ... but SQL support is avaible through contributed JDBC driver
- ... Even sharding configurations are possible



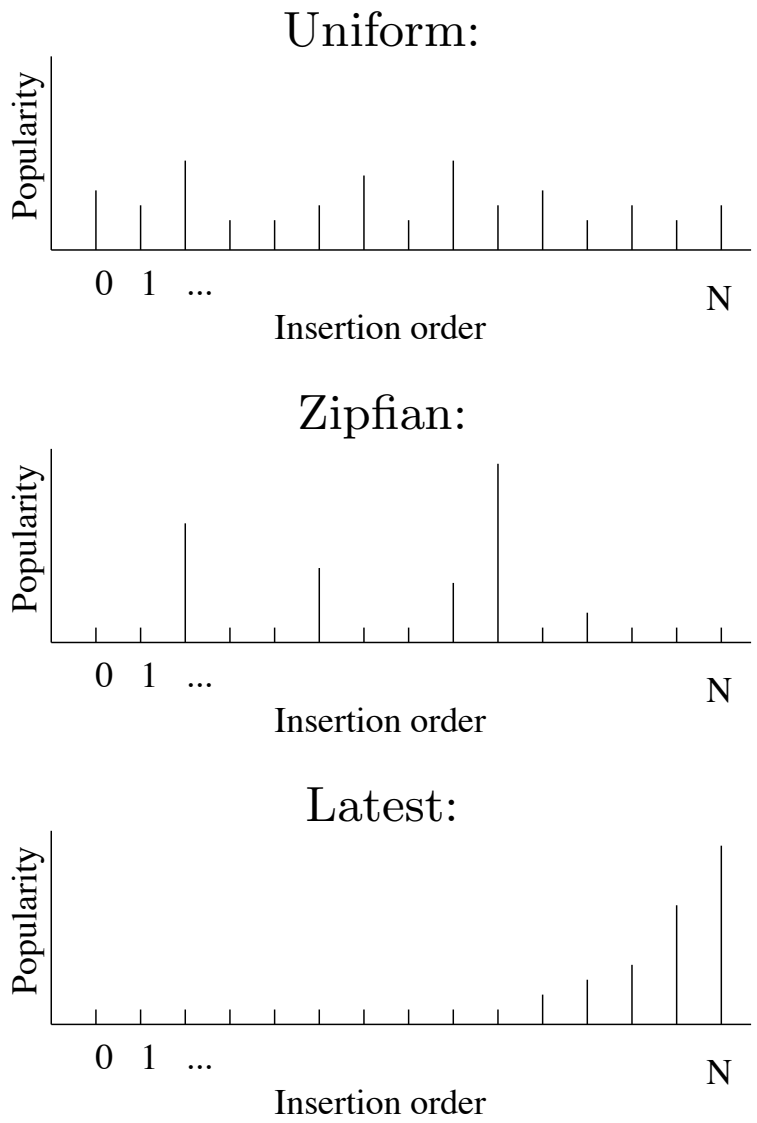
More on DB interface

- Simple operations: INSERT, UPDATE, REPLACE, DELETE, SCAN
- Does not use SQL
- ... but SQL support is avaible through contributed JDBC driver
- ... Even sharding configurations are possible



More on workload

- Specifies what DB operations are used by application
- Also defines request distribution
- It is possible to specify record size
- It's possible to specify number of records and operations



/* TODO: Remove this crap */

SMS Traffic: workload construction

SMS service provider, several gateways, big clients (such as banks)

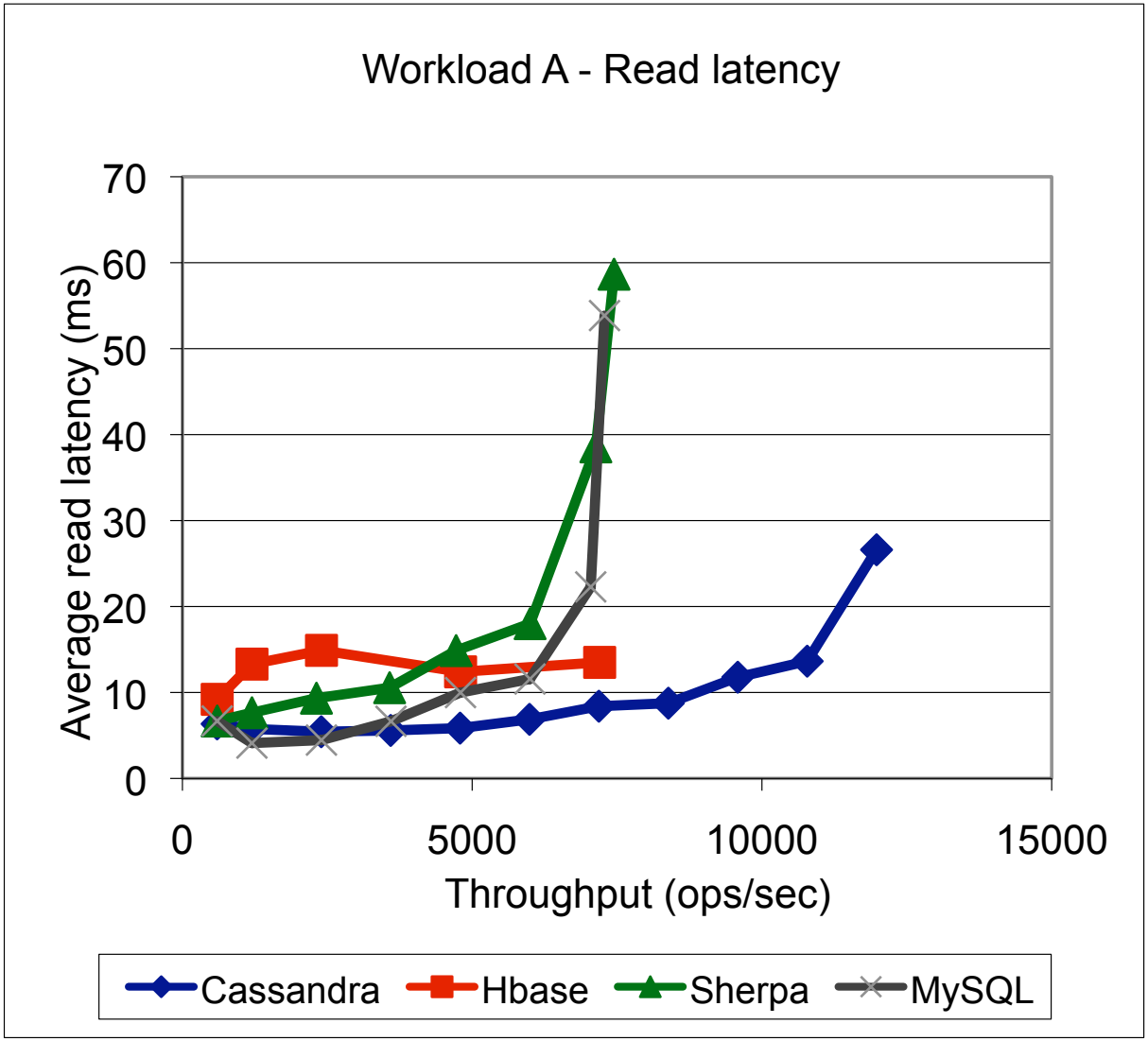
- 15% inserts, 65% updates, 15% reads
- Request distribution: **latest** SMS messages are the "hottest" ones
- Evaluated Cassandra and sharded MySQL as DB storage for the next generation of SMS sending platform

Testing process

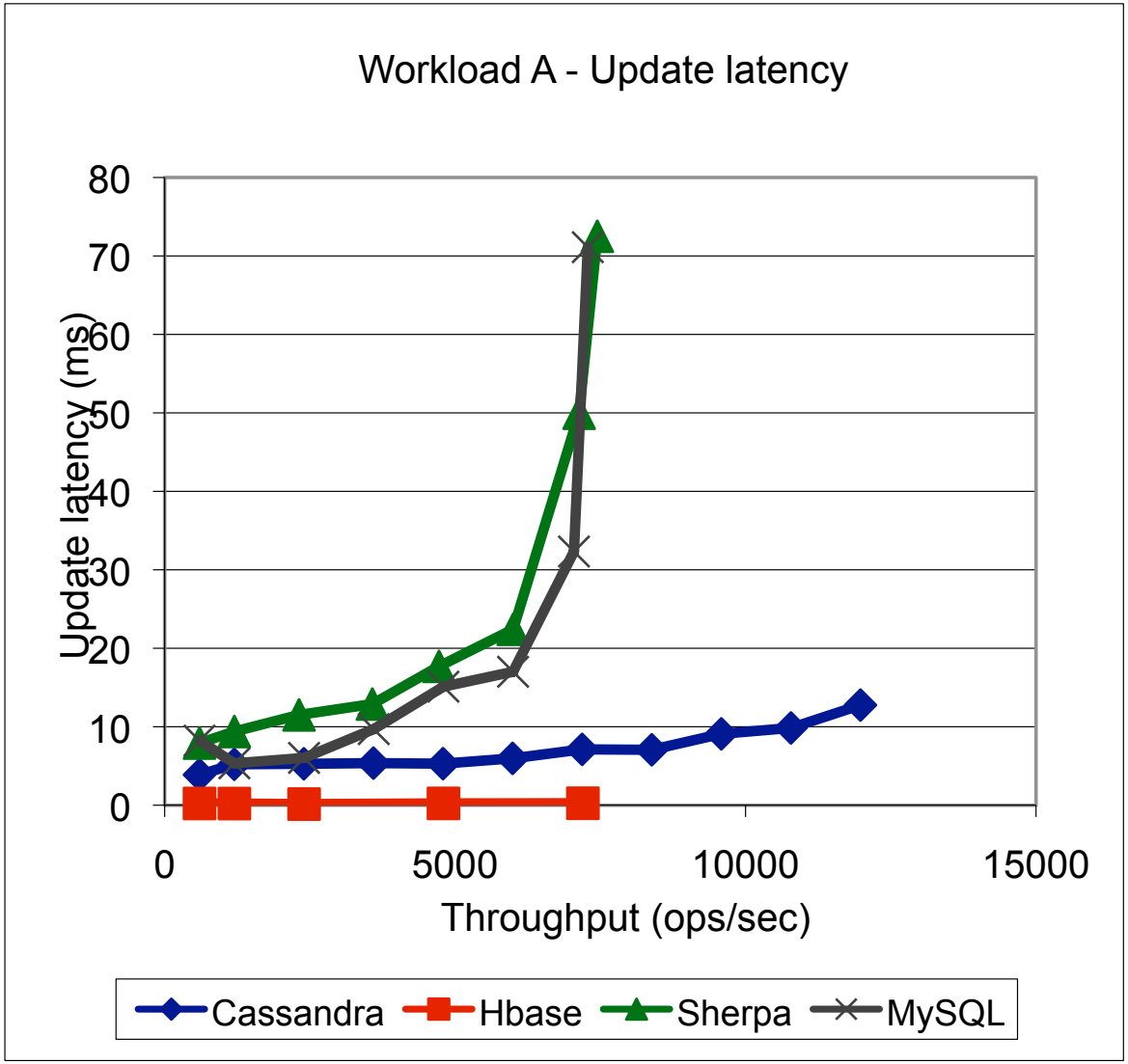
- 3 instances of DBMS system on one server (Core Quad Q9400, 4GB RAM, SATA-II HDD, FreeBSD 8.2-amd64)
- Cassandra 0.7.4 (1GB Java heap / instance)
- MySQL 5.1 + InnoDB engine (1GB InnoDB buffer pool size / instance)
- Client: separate machine, 1Gb/s connection

Should avoid swapping and disk IO saturation

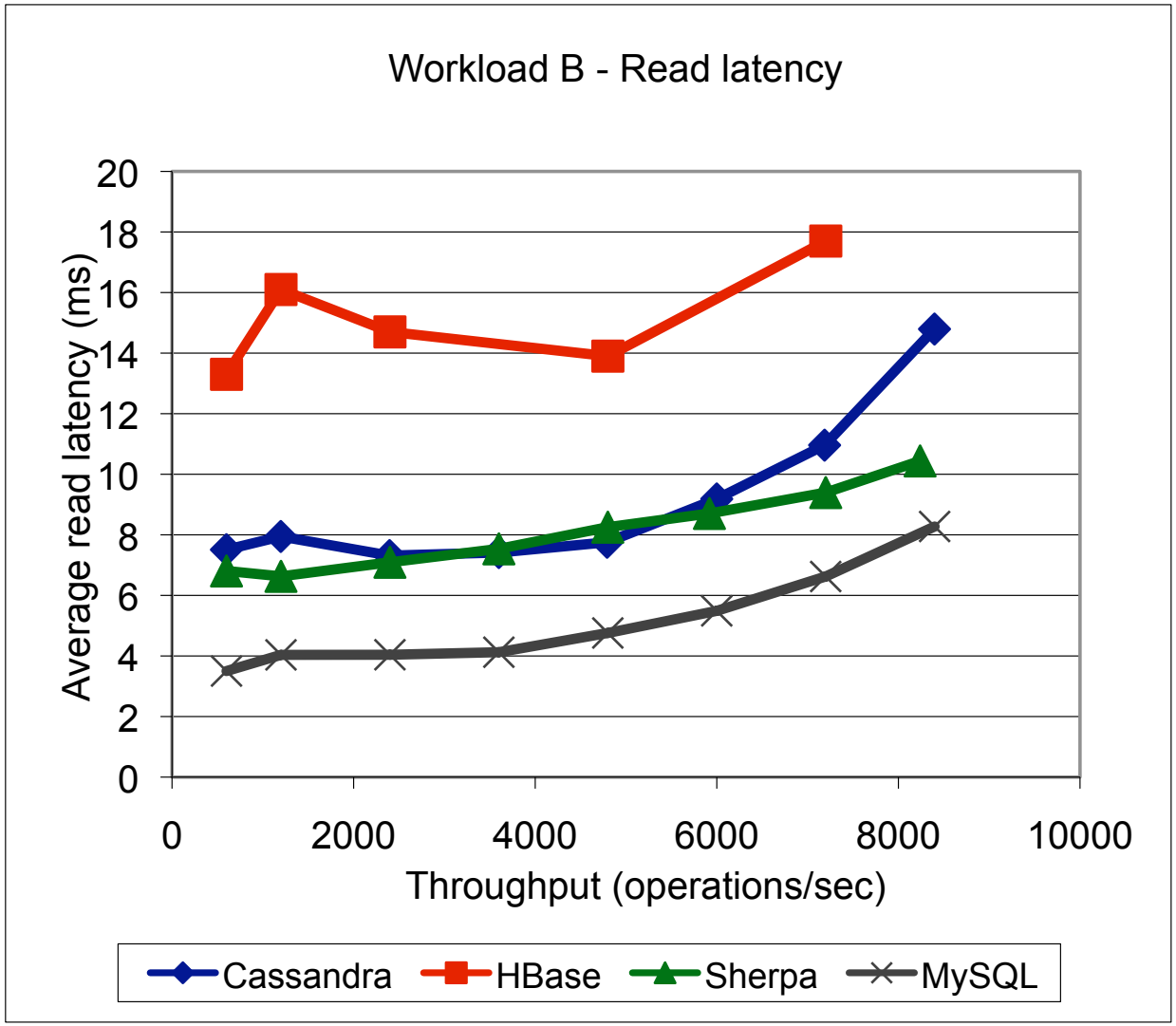
Some resuts: Workload "A": 50% read / 50% write



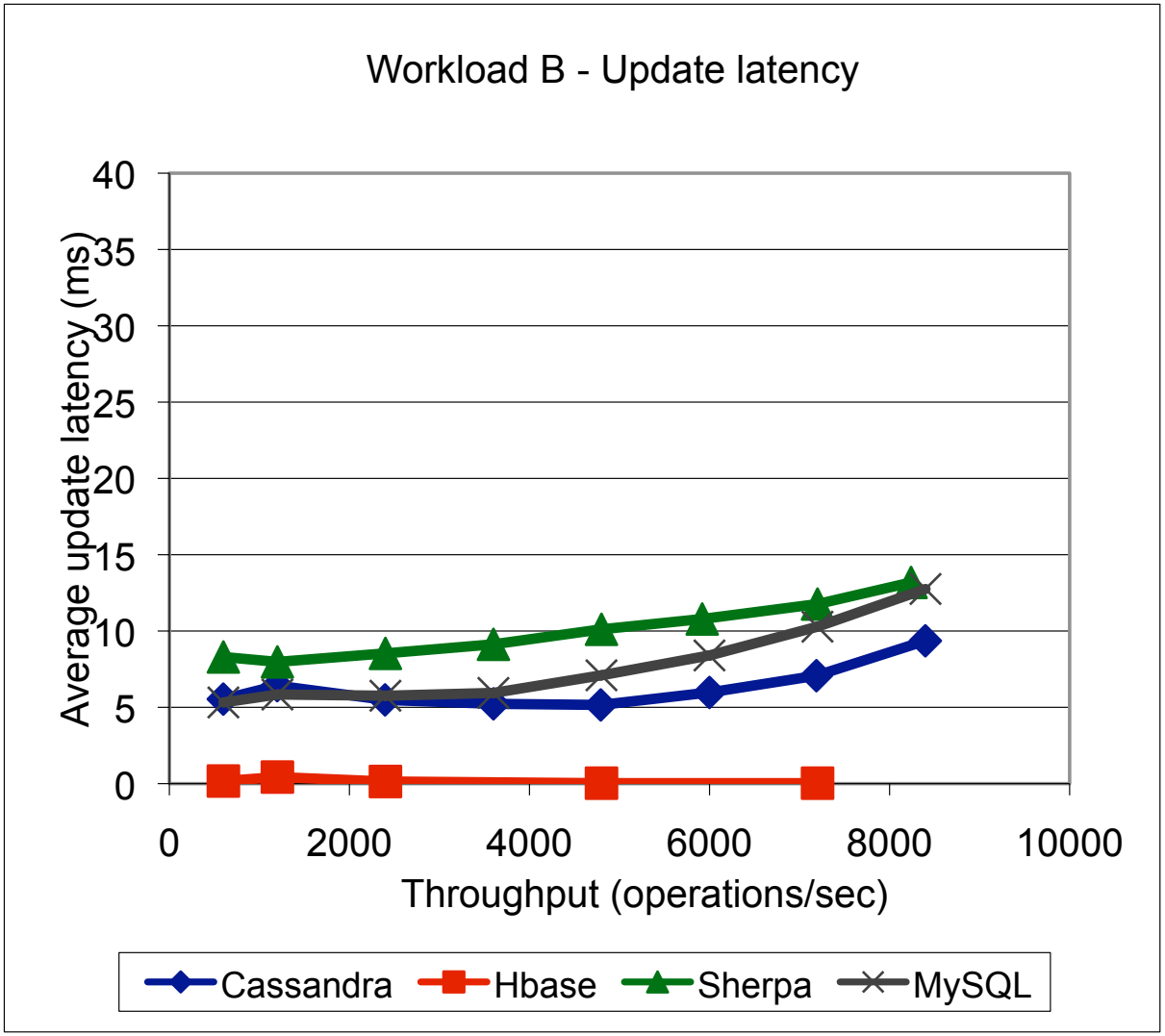
Some resuts: Workload "A": 50% read / 50% write



Some resuts: Workload "B": 95% read / 5% write



Some resuts: Workload "B": 95% read / 5% write



Links

- Yahoo Cloud Serving Benchmark: https://github.com/brianfrankcooper/YCSB
- google://
- webmaster@kibab.com, kibab@FreeBSD.org