Automated YCSB Benchmarking

Miroslav Cupák

04/04/2013

Project Presentation

Outline

- Introduction
- YCSB
- Approach
- Design & Implementation
- Future Work
- Related Work
- Conclusion

Introduction

- problem
  - monitoring the results of cloud systems benchmarks together with the configuration
- motivation
  - recent explosion of cloud data-serving systems
  - real performance vs "sweet spot" performance
  - benchmarking tools don't offer a sufficient level of automation
  - many experiments on different configurations
  - results interpreted in an ad-hoc way

YCSB

- Yahoo! Cloud Serving Benchmark
- tool for benchmarking of cloud data management systems
- open source
- easily extensible (systems, workload)
- Cassandra, DynamoDB, HBase, Infinispan, JDBC, MapKeeper, MongoDB... (12)
- good level of automation

YCSB Benchmarks

- Tier 1: Performance
  - latency of requests when the database is under load
  - focused on measuring latency as we increase throughput until saturation
- Tier 2: Scalability
  - impact on performance as machines are added
  - scaleup - load servers with data, run the workload, delete the data, add more servers, load more data, run the workload again

YCSB Workloads

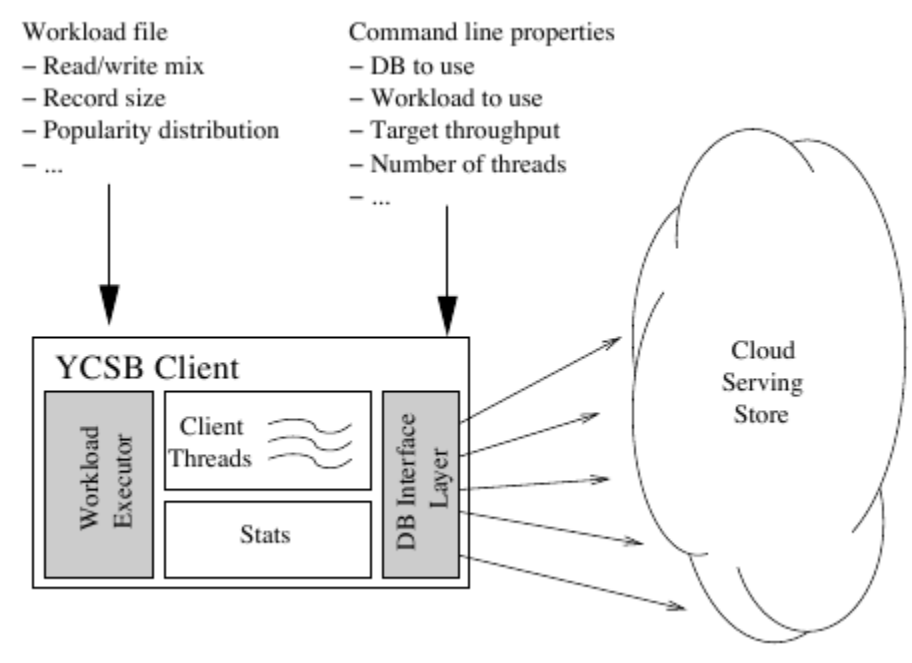
- workload generator (YCSB client)
  - loads data sets and executes workloads
- workloads
  - update heavy, read mostly, read only, read latest, short range, read-modify-write
  - read/update ratio, record count, operation count, data size, record structure, request distribution

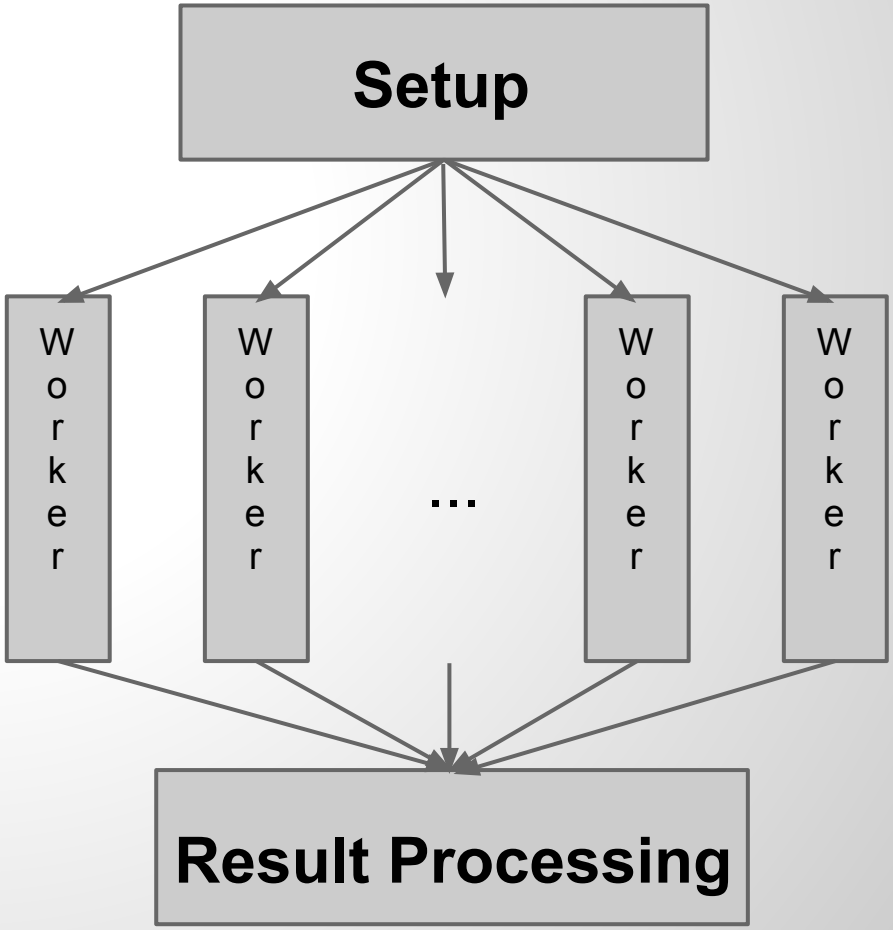
YCSB Workloads

- example - update heavy workload

```
#   Application example: Session store
#   Records: 1 KB (10 fields, 100 bytes each, plus key)
recordcount=1000
operationcount=1000
workload=com.yahoo.ycsb.workloads.CoreWorkload
readallfields=true
readproportion=0.5
updateproportion=0.5
scanproportion=0
insertproportion=0
requestdistribution=zipfian
```

YCSB Client Execution





YCSB Output

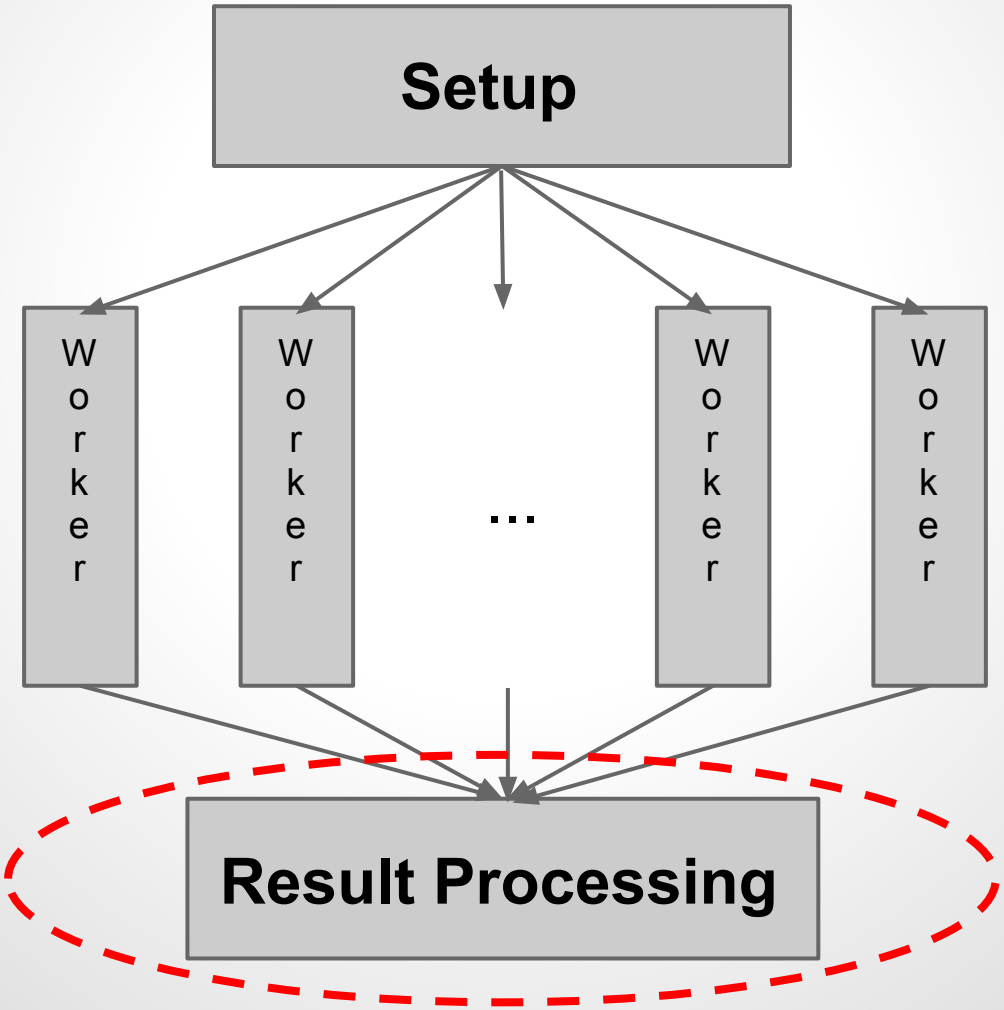
```
$ ycsb run cassandra-10 -p hosts=127.0.0.1 -P workloada
  -p recordcount=100000000 -s -threads 10

 10 sec: 1000 operations; 60.61 current ops/sec; [UPDATE AverageLatency
(us)=1055.57] [READ AverageLatency(us)=4200.67] [CLEANUP AverageLatency
(us)=102]
[OVERALL], RunTime(ms), 10183.0
[OVERALL], Throughput(ops/sec), 98.20288716488265
[UPDATE], Operations, 493
[UPDATE], AverageLatency(us), 1246.6247464503042
[UPDATE], MinLatency(us), 525
[UPDATE], MaxLatency(us), 48089
[UPDATE], 95thPercentileLatency(ms), 1
[UPDATE], 99thPercentileLatency(ms), 6
[UPDATE], Return=0, 493
[UPDATE], 0, 268
[UPDATE], 1, 207
```

Approach

- tasks
  - modify YCSB to store results of experiments in a database
  - create an application showing results in graphs
  - collect configuration of the client and the system and associate it with data points in graphs
  - evaluate the approach

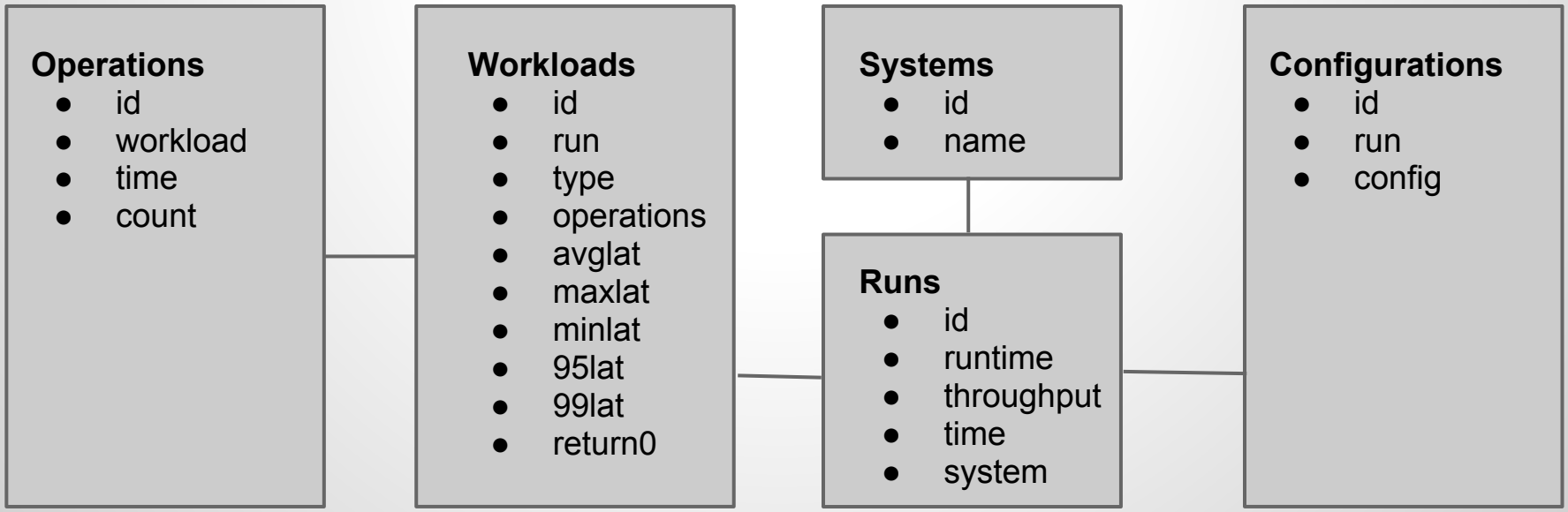
Design & Implementation - Saving

- storing results in a DB



Design & Implementation - Saving

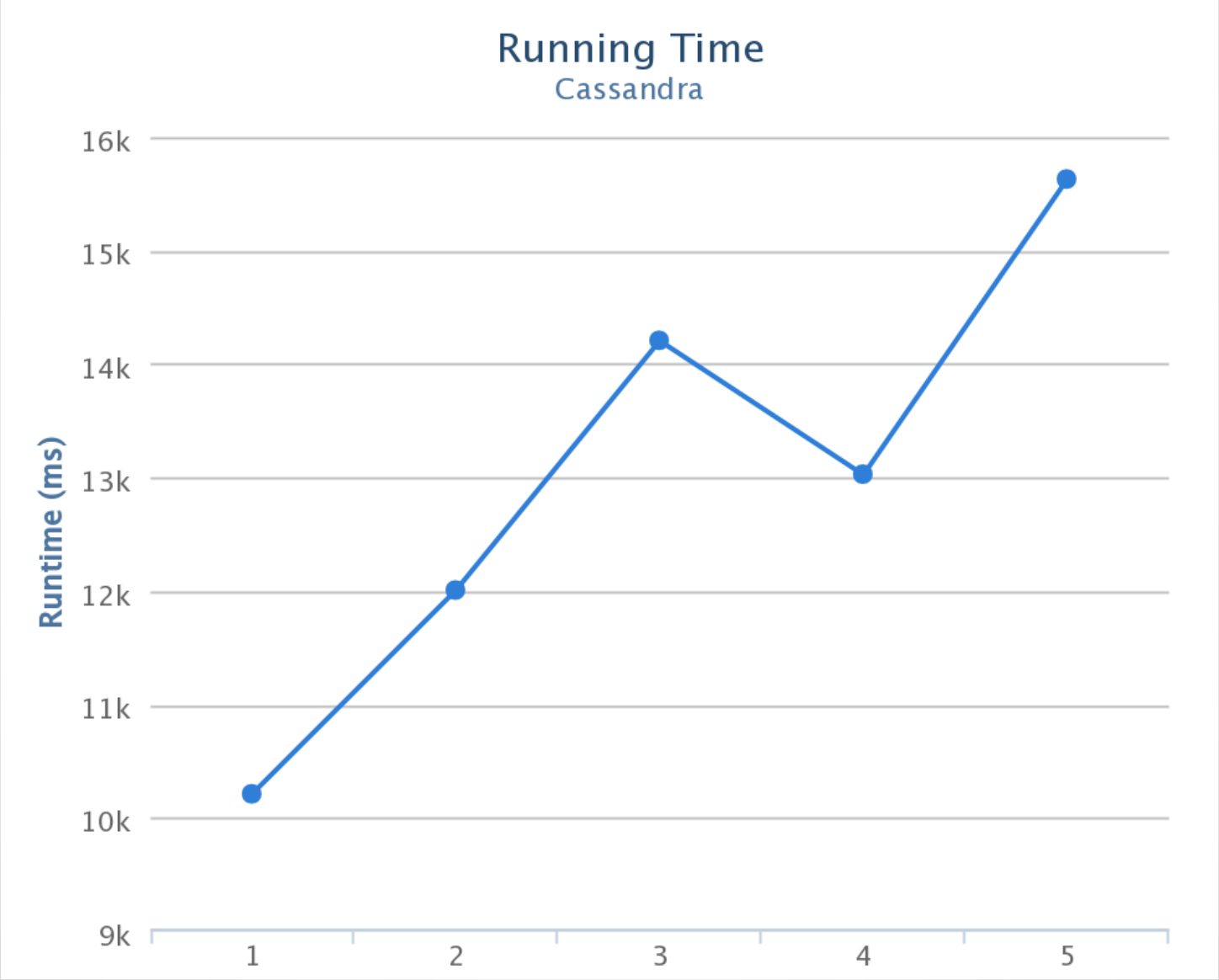
- storing results in a DB
  - implementation of MeasurementsExporter
  - requires non-trivial query construction
  - `-p exporter=...`



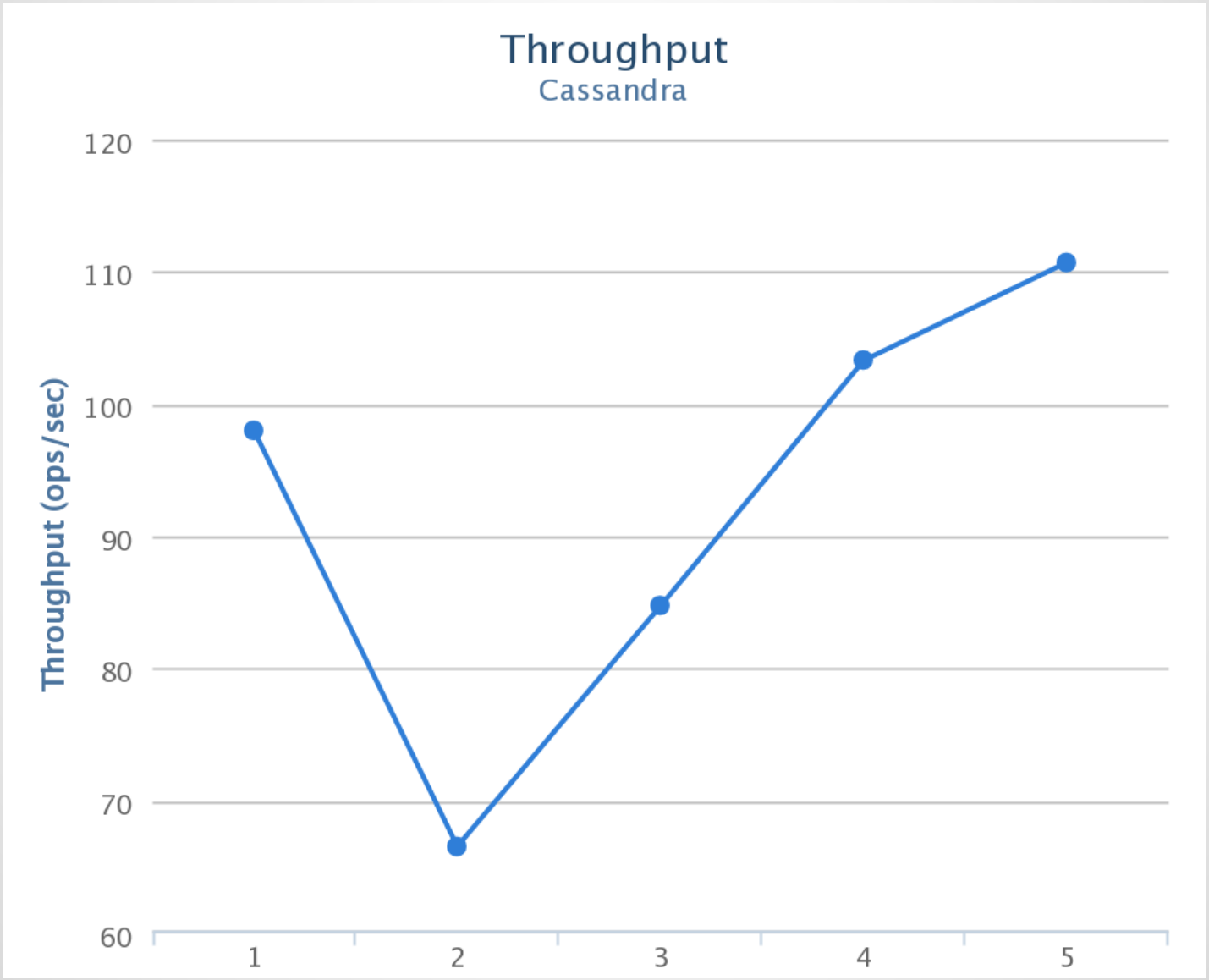
Design & Implementation - Graphs

- generating graphs
  - dashboard-like comparison of multiple runs
  - results over the last few runs to see if a configuration change lead to improvements in performance
  - Java EE 6 web application + Highcharts
  - runtime, throughput, number of operations, latency (avg/min/max/99%/95%)
- user/group management
  - Cassandra developers see only Cassandra results

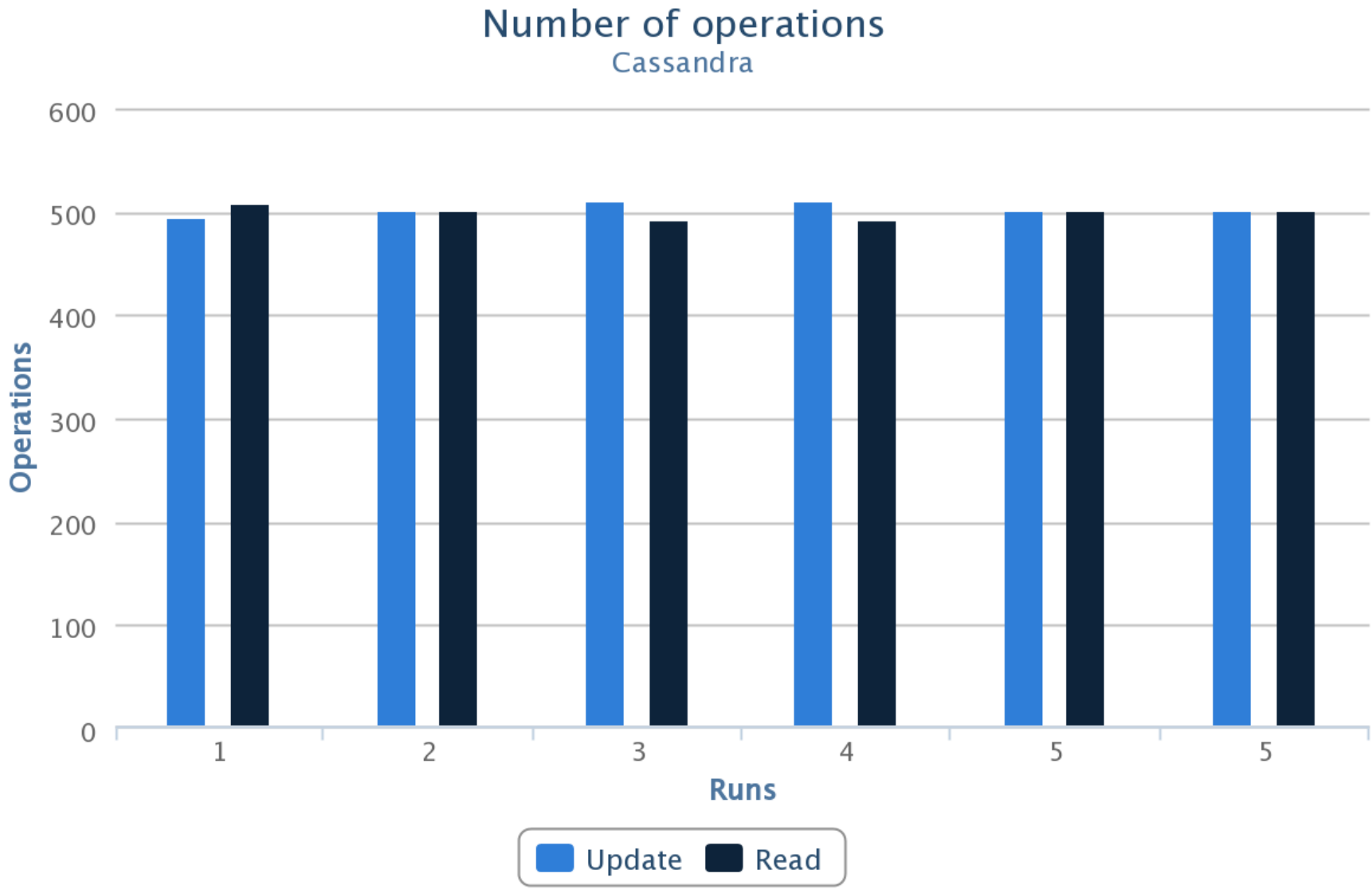
Design & Implementation - Graphs



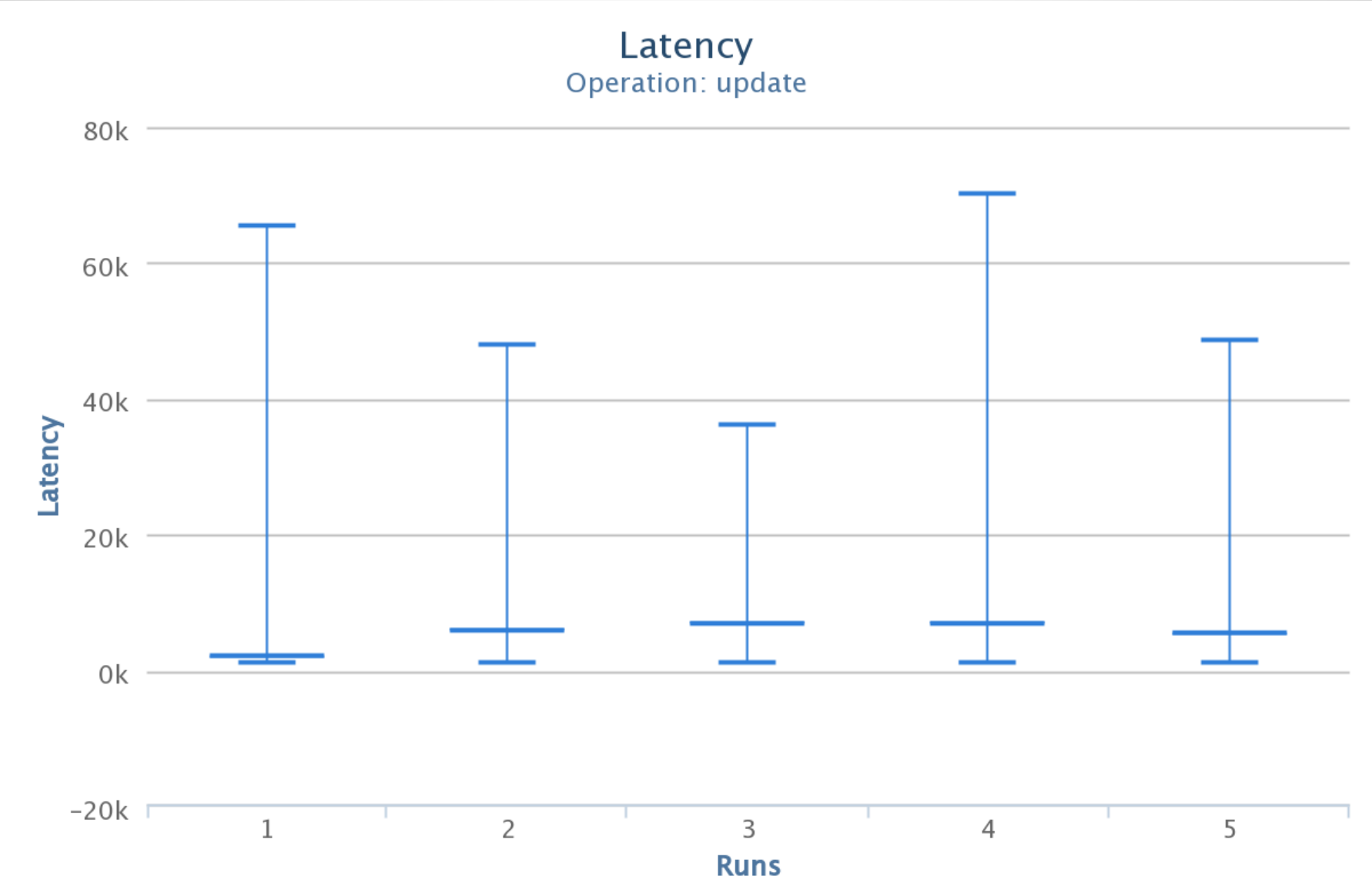
Design & Implementation - Graphs



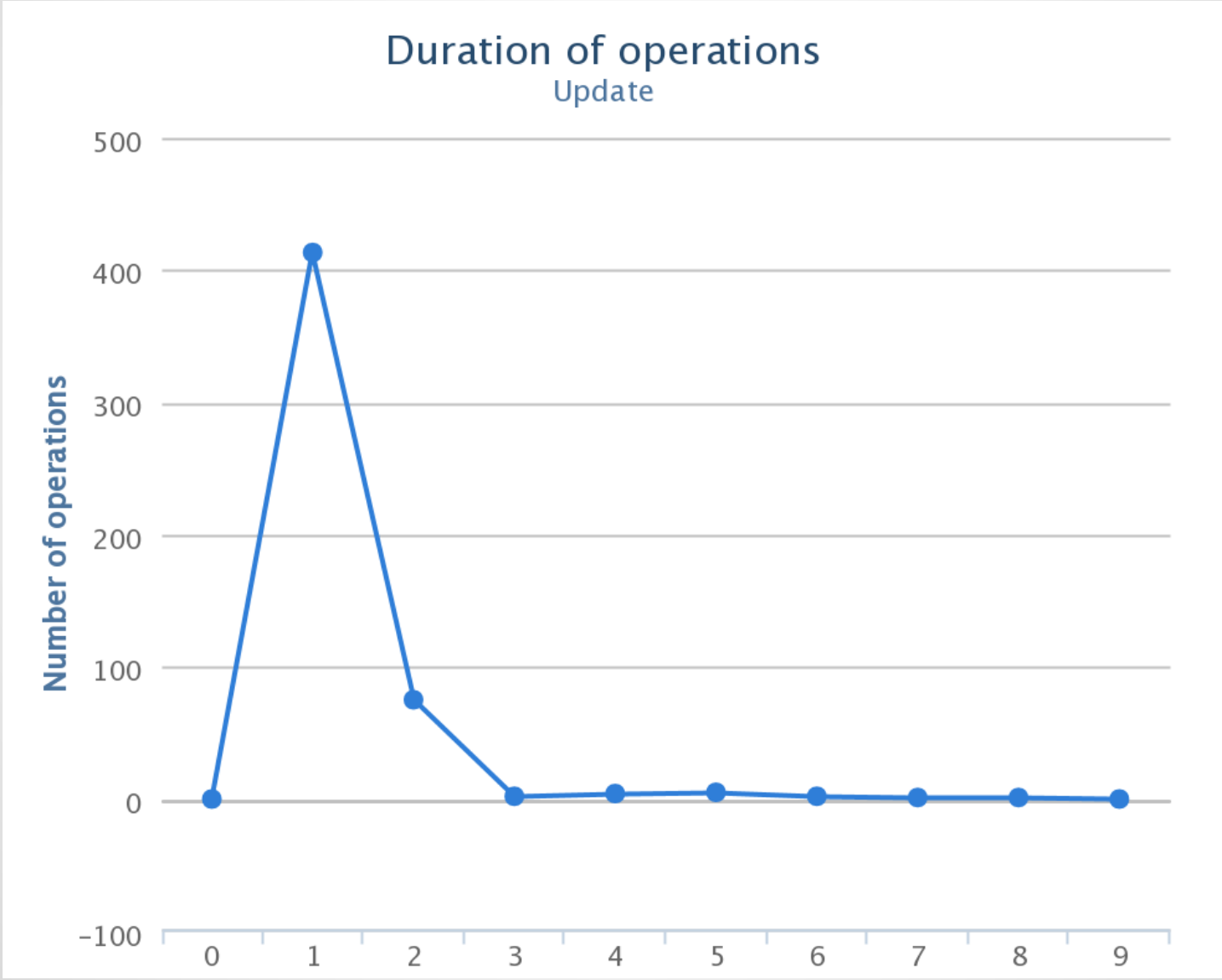
Design & Implementation - Graphs



Design & Implementation - Graphs

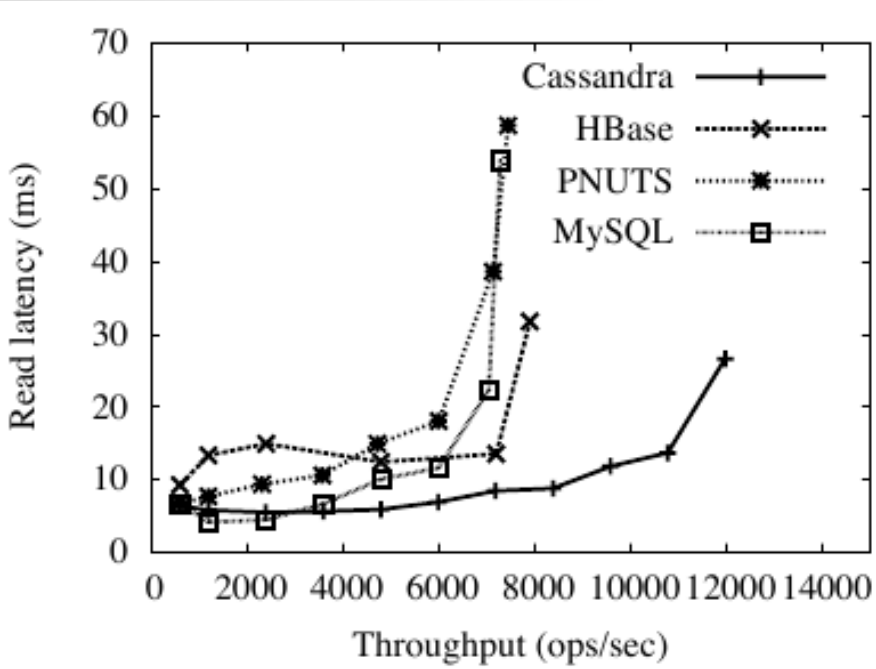


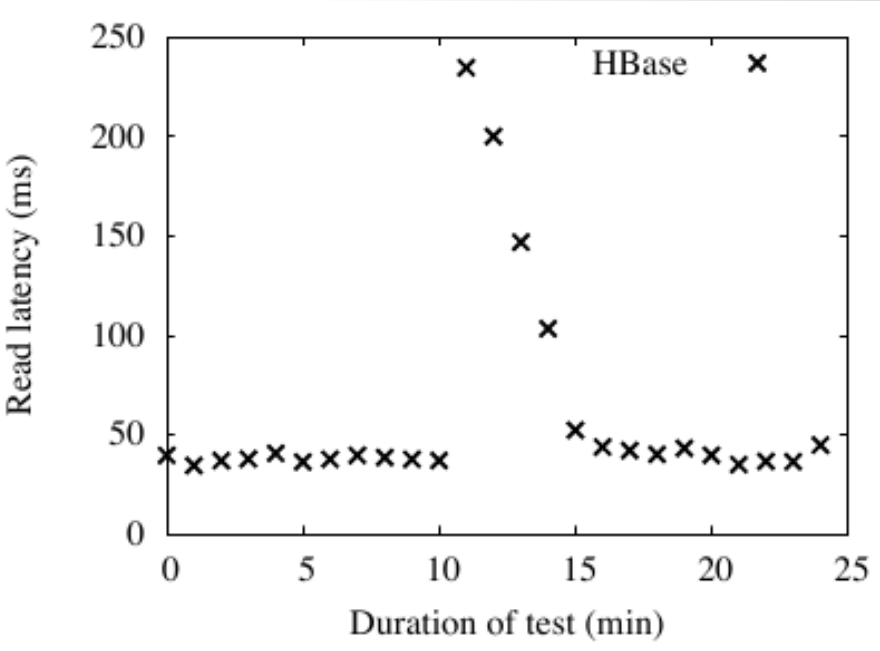
Design & Implementation - Graphs



Design & Implementation - Graphs

- not yet done

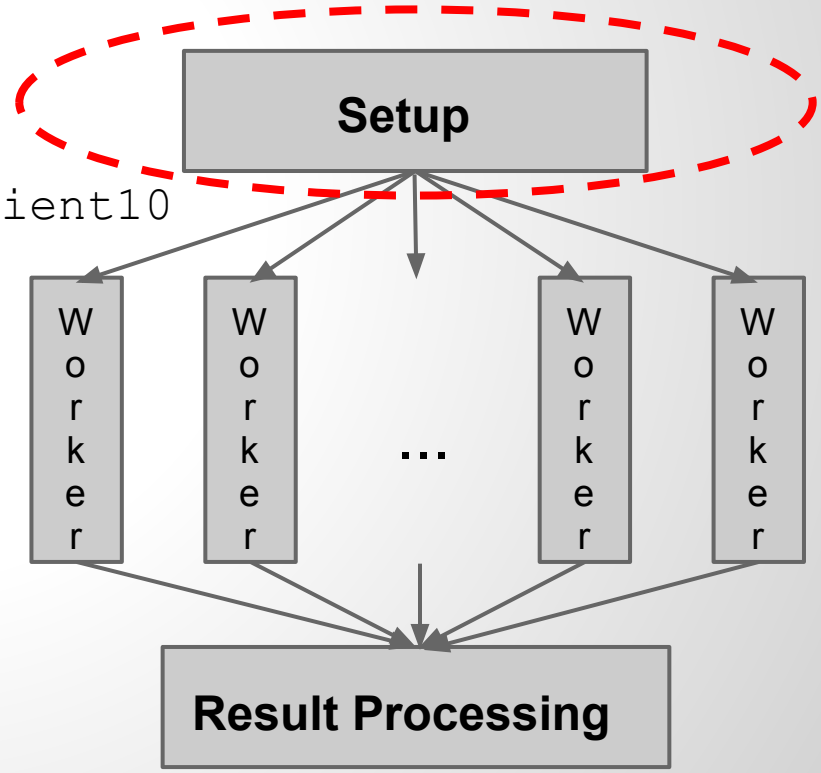




Design & Implementation - Config

- config fetched after selecting a run in a graph
- so far only client configuration

```
com.yahoo.ycsb.Client
-db com.yahoo.ycsb.db.CassandraClient10
-p hosts=127.0.0.1
-P ../workloads/workloada
-p recordcount=100000000
-s
-threads 10
-target 100
-t
...
```



Future Work

- extend graphs
- polling system configuration over SSH
- run comparison (diff-like view)
- evaluation (Cassandra, HBase)

Related Work

- Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, Russell Sears: **Benchmarking cloud serving systems with YCSB**. SoCC 2010.
- Brian F. Cooper: **Yahoo! Cloud Serving Benchmark - Overview and results**. SoCC 2010.
- Tilmann Rabl, Sergio Gómez-Villamor, Mohammad Sadoghi, Victor Muntés-Mulero, Hans-Arno Jacobsen, and Serge Mankovskii. 2012. **Solving big data challenges for enterprise application performance management**. Proc. VLDB Endow 2012.

Conclusions

- YCSB modified to store data in a more suitable form.
- Web application showing the results in graph form implemented.
- Configuration management in progress.

Thank you! Questions?