A Framework for Benchmarking Consistency in Distributed key-Value Storage Systems

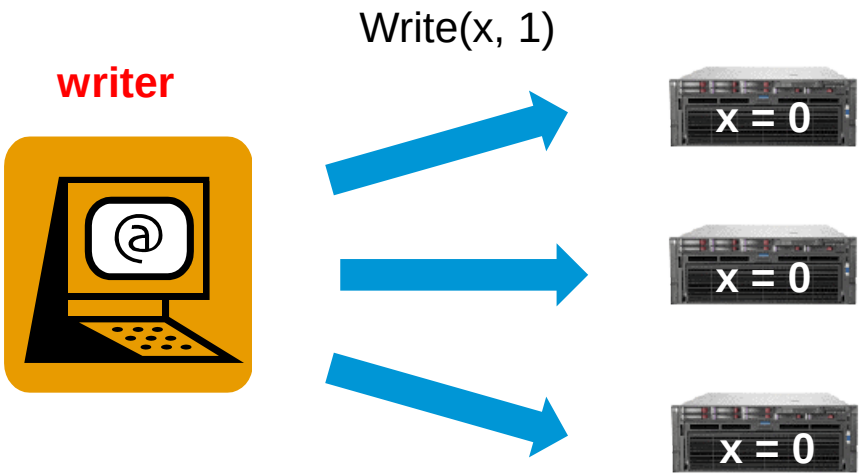
Summer intern: Muntasir R. Rahman (HP Labs / UIUC)

Presenter: Wojciech Golab (HP Labs / University of Waterloo)

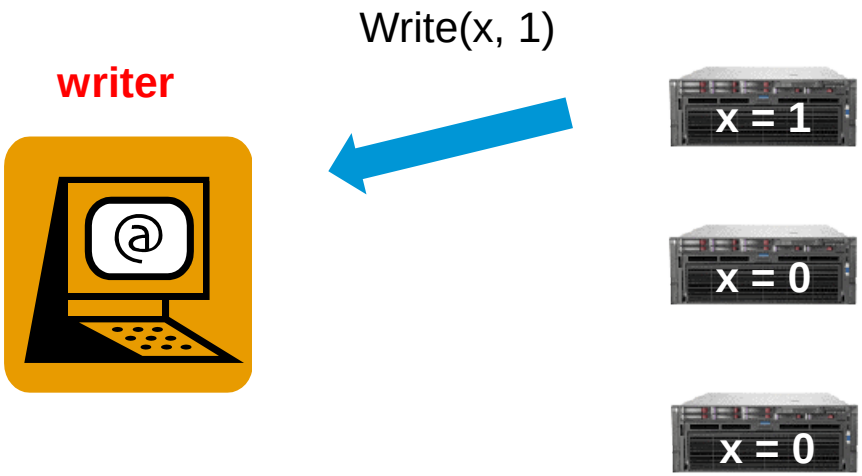
Collaborators: Alvin AuYoung, Kimberly Keeton, Jay J. Wylie (HP Labs, HP Labs, HP Labs / LinkedIn)

Motivation

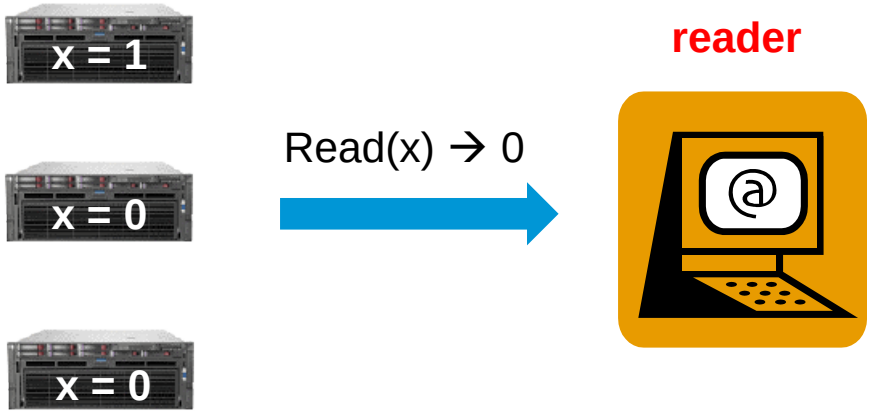
Eventual consistency



Eventual consistency

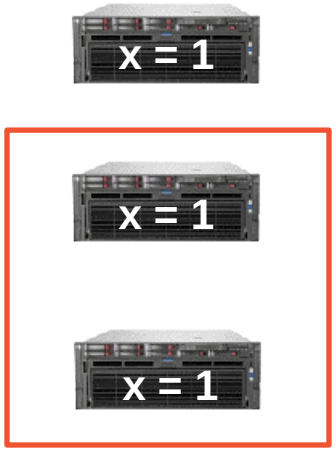


Eventual consistency

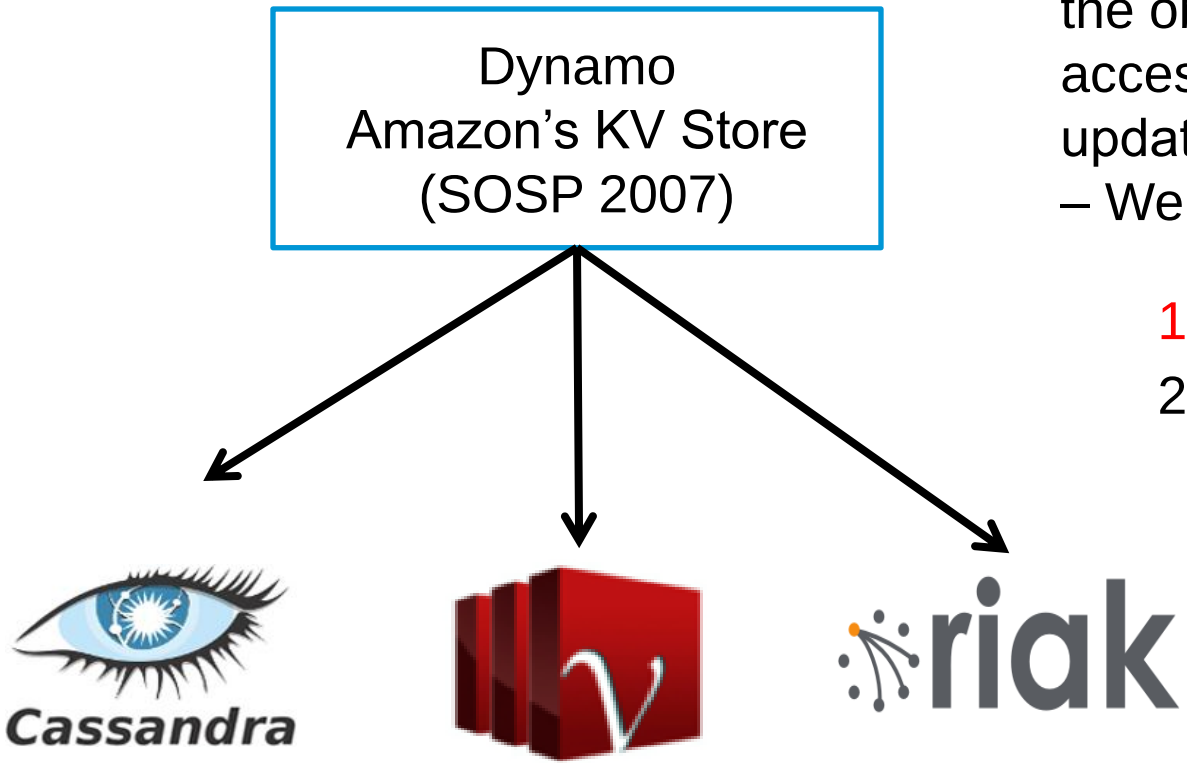


Eventual consistency

eventually…



Eventual consistency



"If no new updates are made to the object, eventually all accesses will return the last updated value"
– Werner Vogels, CACM 2008

1. How soon is eventual?
2. What happens if updates are made continuously?

Goal

Quantify the actual consistency observed by clients accessing an EC key-value store in arbitrary workloads.

(in)consistency = staleness

Related Work

Measuring consistency

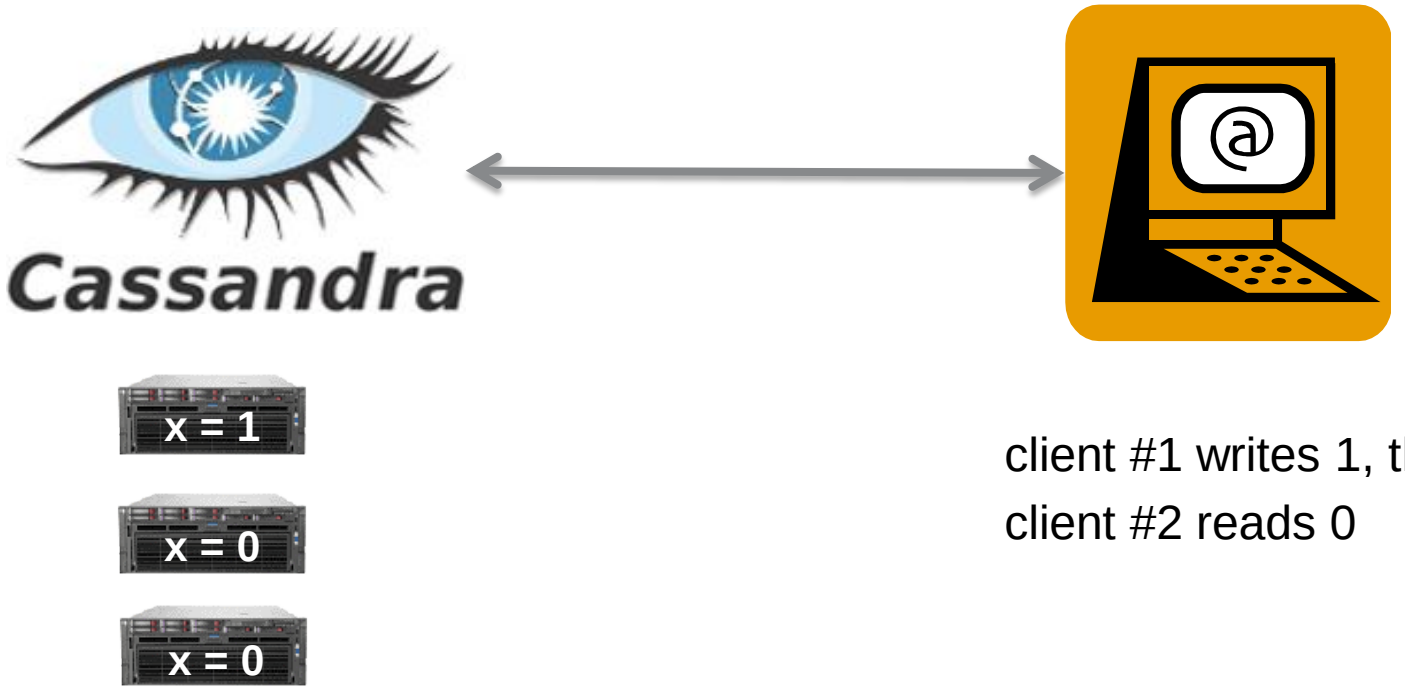
How much consistency do EC key-value stores provide in practice?

[BT11] D. Bermbach and S. Tai. Eventual consistency: How soon is eventual? An evaluation of Amazon S3's consistency behavior. MW4SOC 2011.

[Pa11] S. Patil et al. YCSB++: benchmarking and performance debugging advanced features in scalable table stores. SOCC 2011.

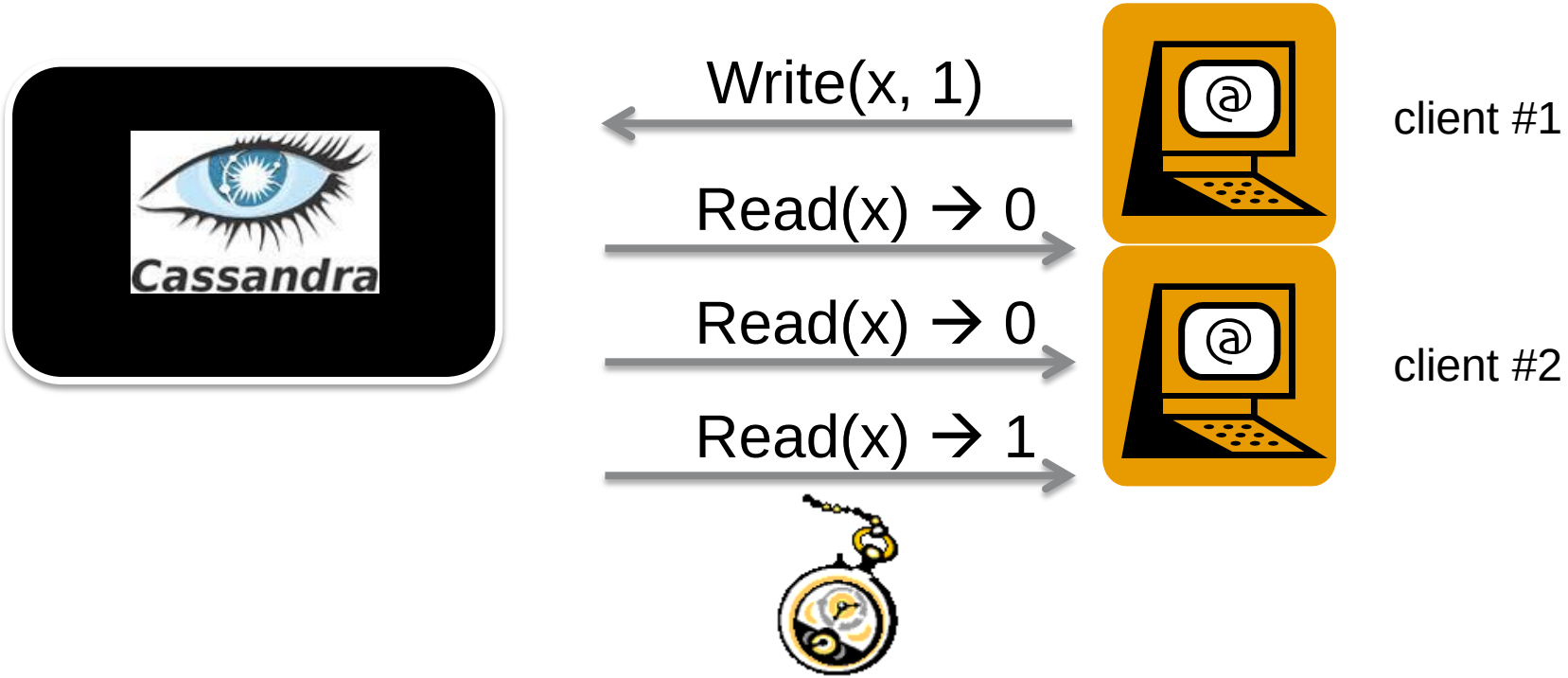
[Wa11] H. Wada et al. Data consistency properties and the trade-offs in commercial cloud storage: the consumers' perspective. CIDR 2011.

Data-centric vs. client-centric consistency

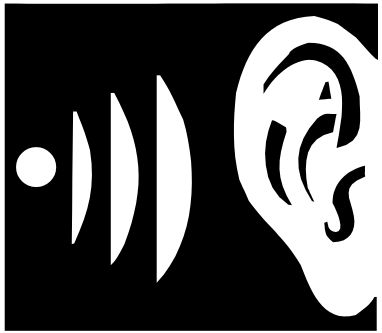


client #1 writes 1, then
client #2 reads 0

Measurement technique [BT11, Pa11, Wa11]

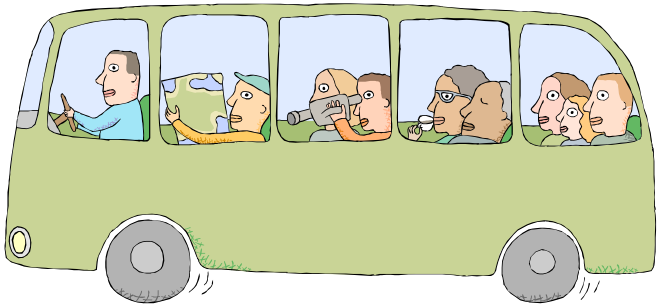


If a tree falls…



client reads x=1

Observer effect



Our Approach

Passive measurement

config. parameters

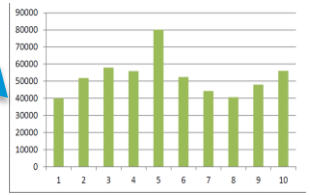
workload

failure pattern

Trace: start + end time of every operation (collected at clients)

staleness calculation

Chi	Delta
54	143
23	60
678	1432



Definitions

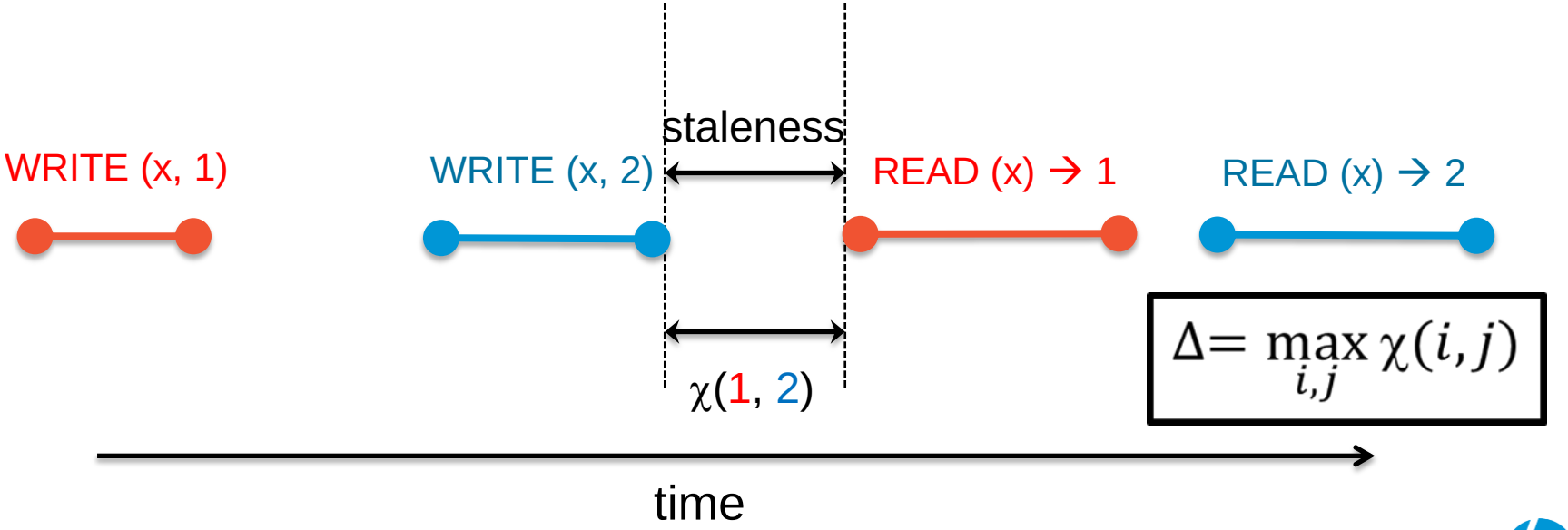
***Fresh*:** a value is considered fresh from the "moment" it's written until it's overwritten.

Staleness: how long ago was the value read last considered fresh?
(e.g.,100ms ago or 0ms ago)

Note: not looking at causality

Calculating staleness

[GLS11] W. Golab, X. Li, M. A. Shah. Analyzing consistency properties for fun and profit. PODC 2011.



Experimental Results

Experimental setup: Cassandra + YCSB + Δ

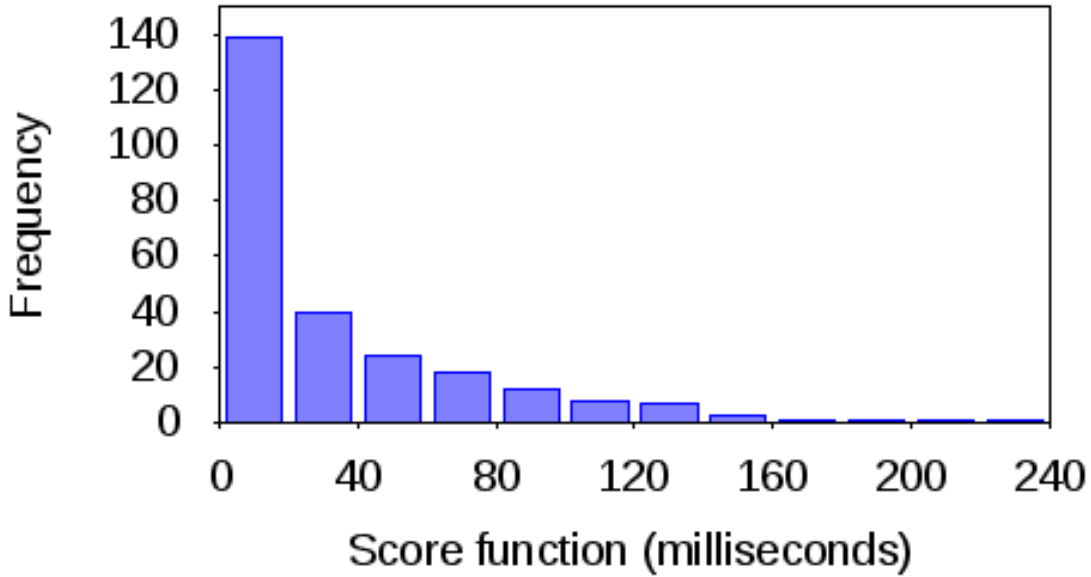
10 servers

- dual-socket Xeons
- ample DRAM
- 1 GbE network
- SSD

each server runs:

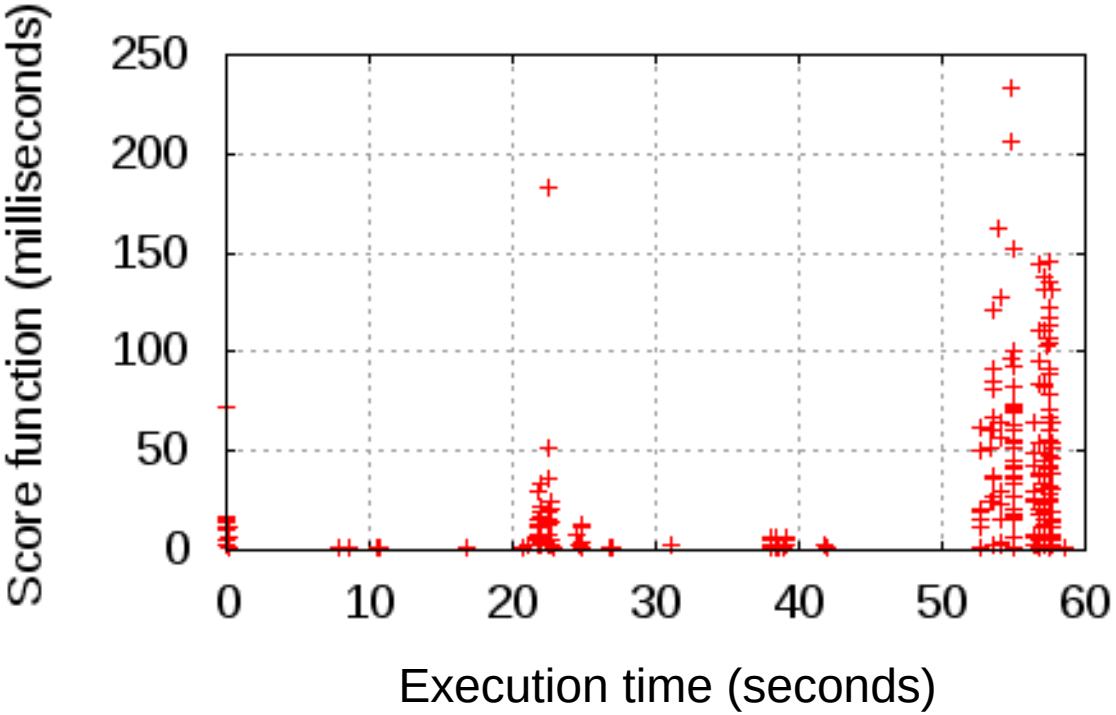
- 1 Cassandra node
- 1 YCSB client
  (“hot spot” distribution,
  80% read, 20% write)

Histogram of score function (χ) values



- χ computed for all keys
- only positive χ shown
- 255 positive scores among approx. 324k writes and 1.3M reads

Time series plot of score function (χ) values



- χ computed for all keys
- only positive χ shown
- 255 positive scores among approx. 324k writes and 1.3M reads

Conclusion

Summary

Methodology for passive benchmarking of consistency in EC key-value stores that captures faithfully the consistency actually observed by clients.

Ongoing and future work

Science:

- understanding the actual behavior of EC key-value storage systems in practice (workloads, configurations, failures)

Engineering:

- defining and enforcing consistency SLAs
- consistency “amplification”

Thank You!

Muntasir R. Rahman (mrahman2@illinois.edu)
Wojciech Golab (wgolab@uwaterloo.ca)
Alvin AuYoung (alvin.auyoung@hp.com)
Kimberly Keeton (kimberly.keeton@hp.com)
Jay J. Wylie (jay.j.wylie@gmail.com)