Who Watches the Watchmen?

On the Lack of Validation in NoSQL Benchmarking

Wolfram Wingerath, Steffen Friedrich, Felix Gessert, Norbert Ritter

University of Hamburg
Department of Informatics
Databases and Information Systems

March 5th, 2015

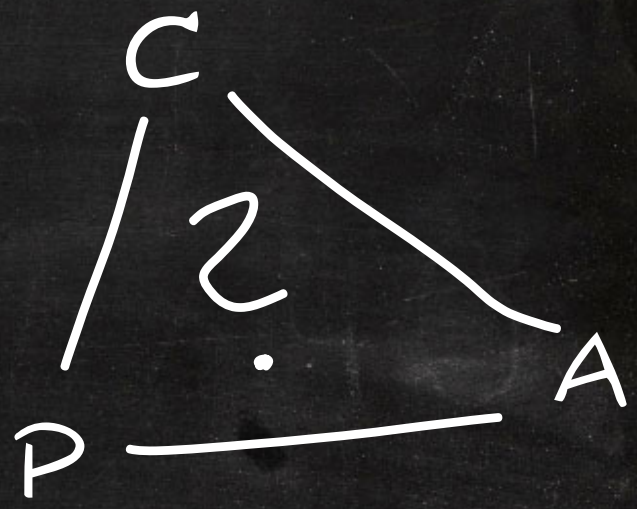
Motivation

Why Validate Benchmarking Tools?

Benchmarking NoSQL is Hard

Dimensions of Interest

- latency and throughput
- availability
  - direct
  - steady-state
- consistency
  - staleness (version-based, time-based)
  - ordering guarantees
  - durability
  - transactions



Existing Benchmarks

- Wada et al. [WFZ+11]
- Bermbach et al. [BT11, BT14, Ber14]
- YCSB++ [PPR+11]
- BG Benchmark [BG13]
- HP Labs [RGA+12, GLS11]
- YCSB+T [DFNR14]

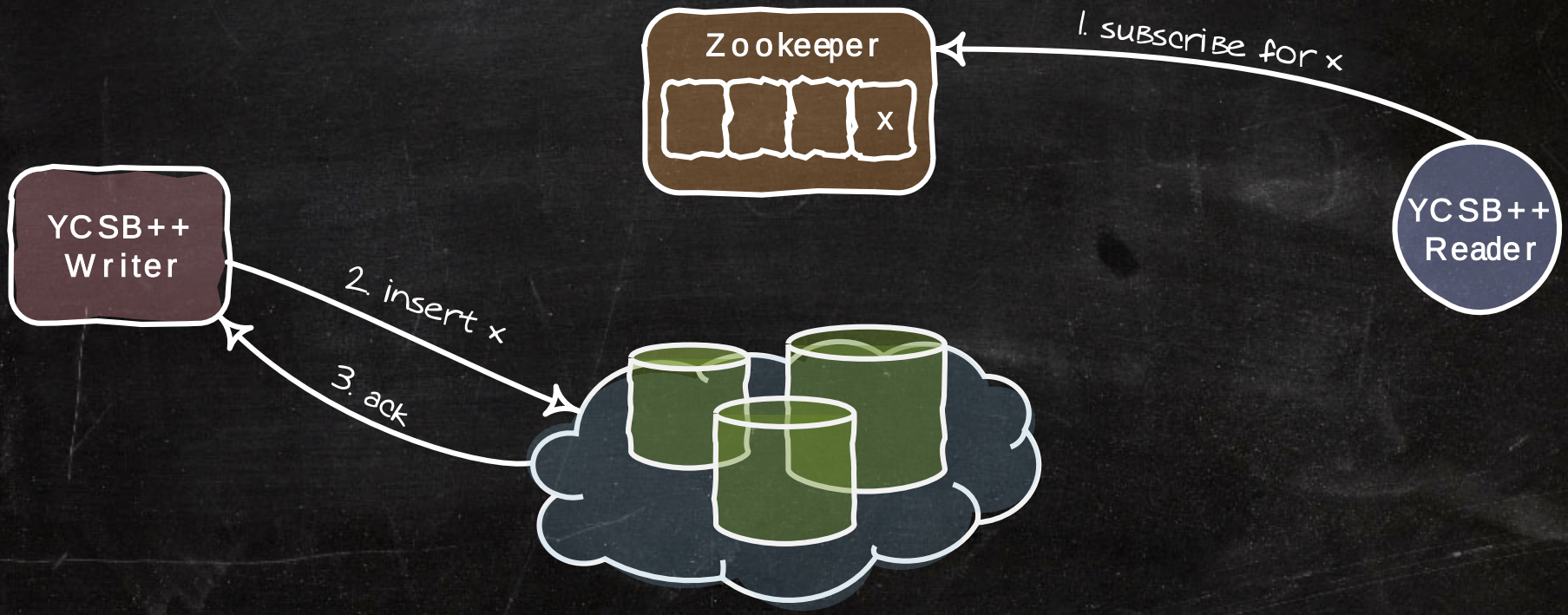
General problems:

→ no error bounds

→ often no validation, i.e. you have to trust them...

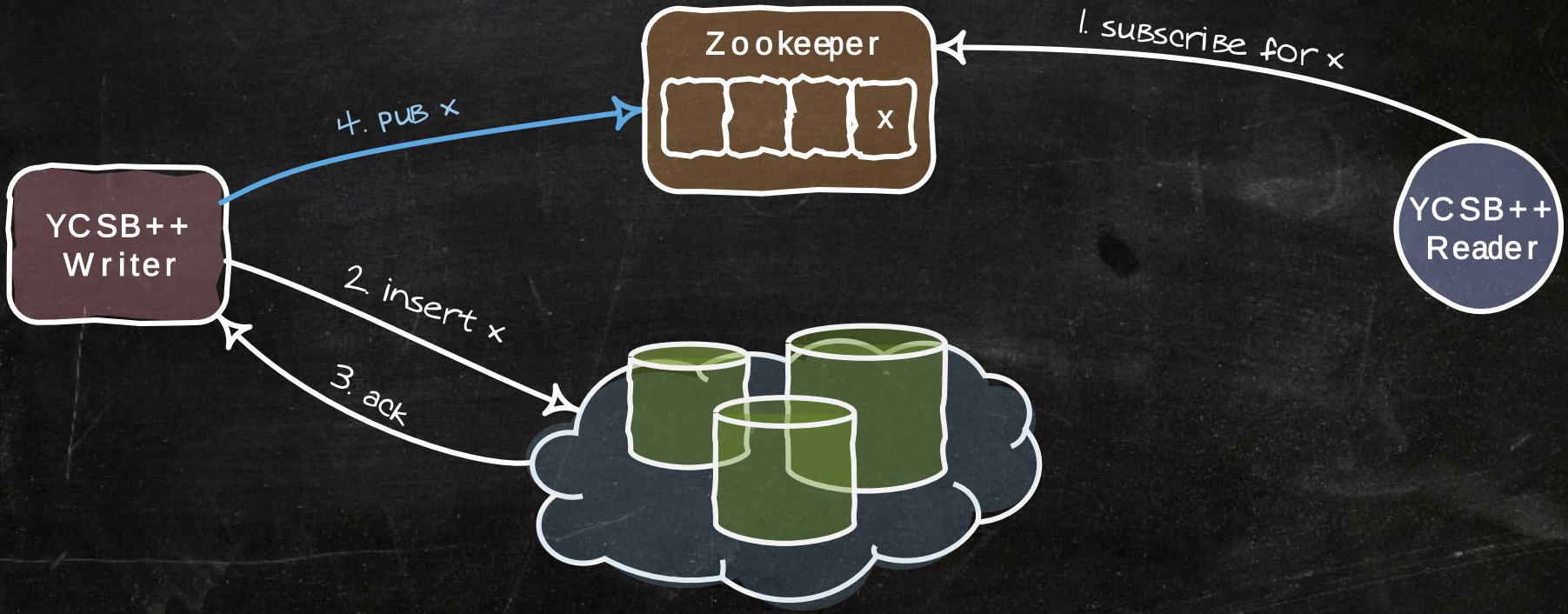
Existing Benchmarks

An example: YCSB++



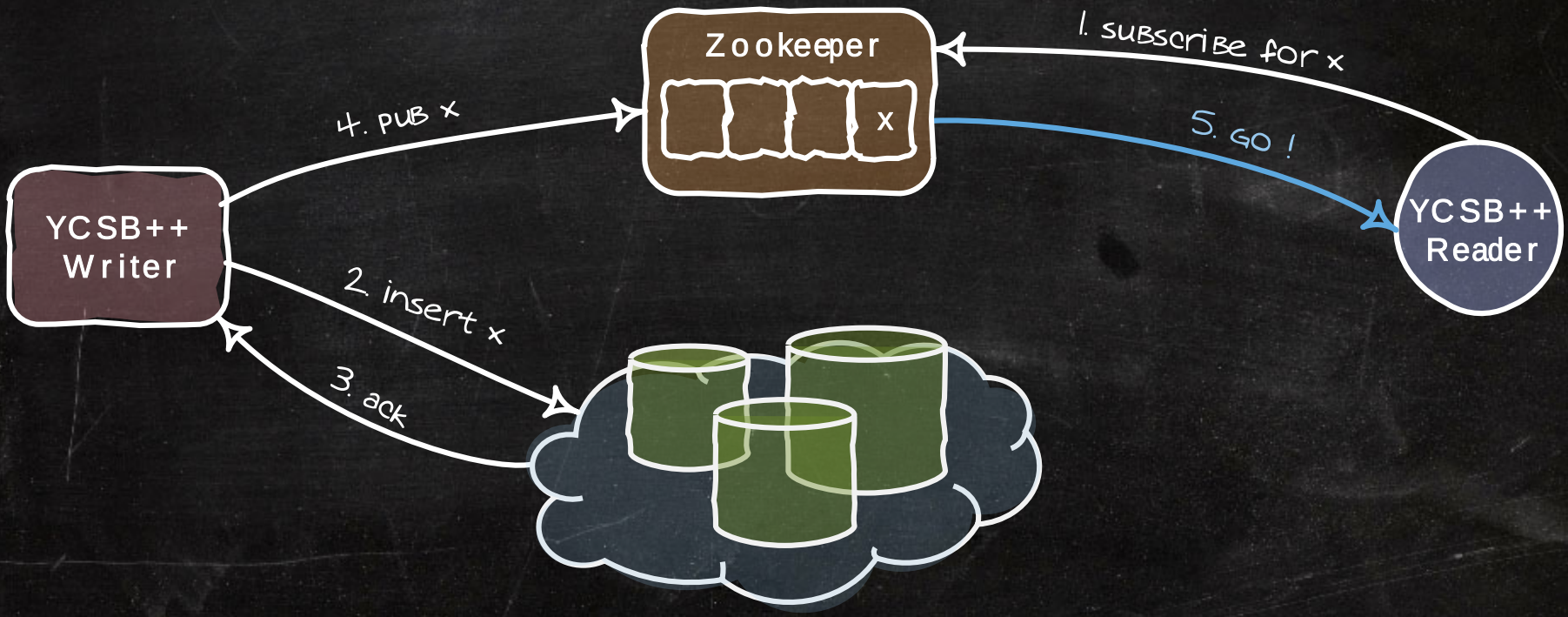
Existing Benchmarks

An example: YCSB++



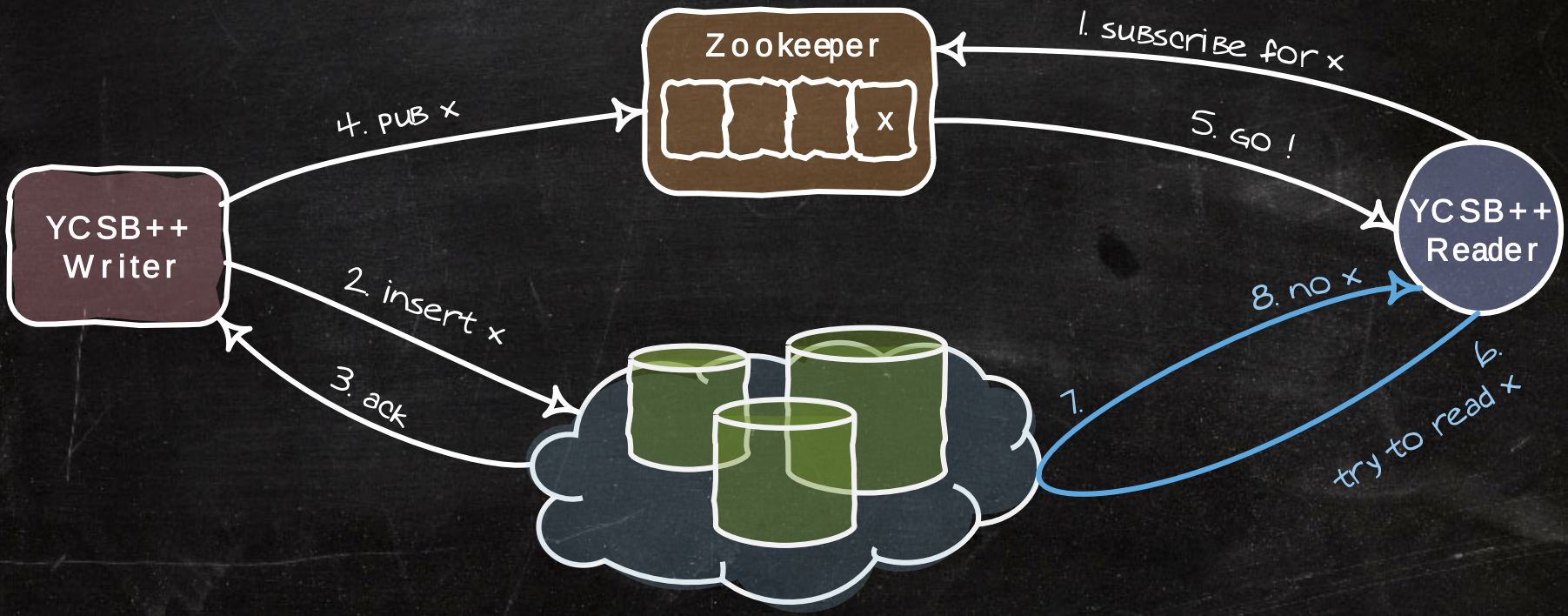
Existing Benchmarks

An example: YCSB++



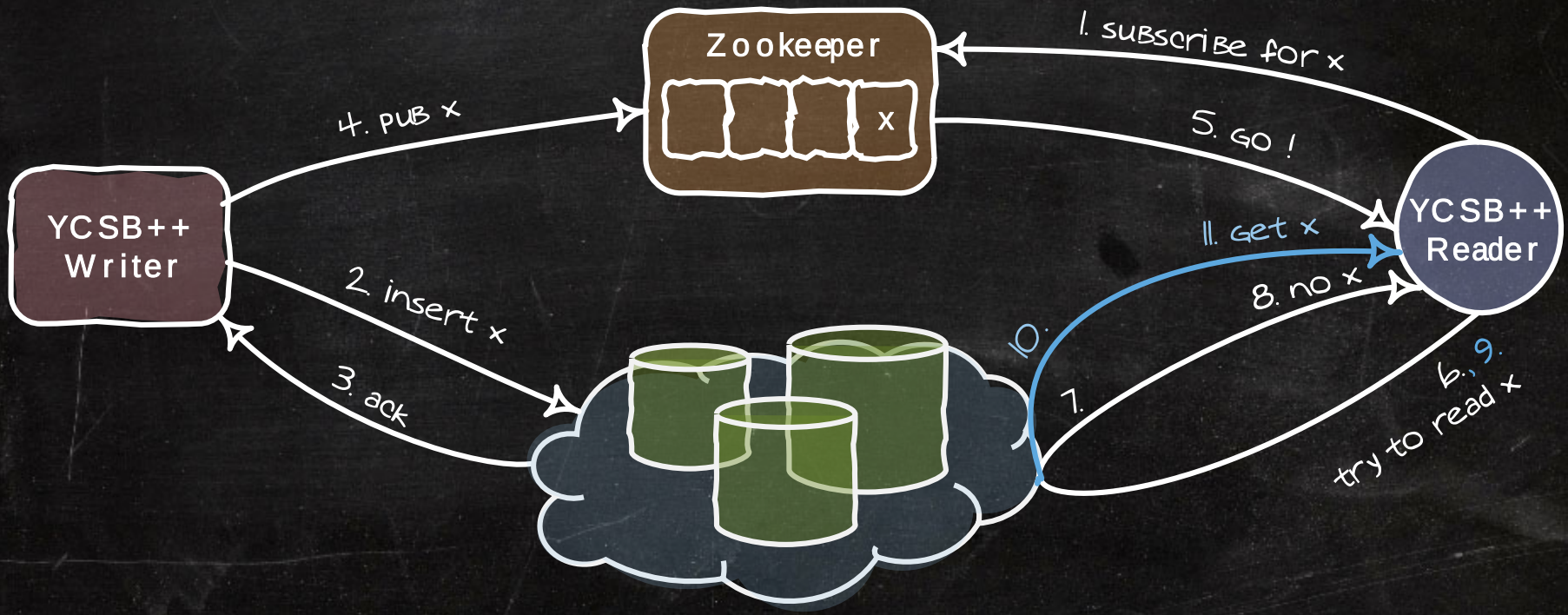
Existing Benchmarks

An example: YCSB++



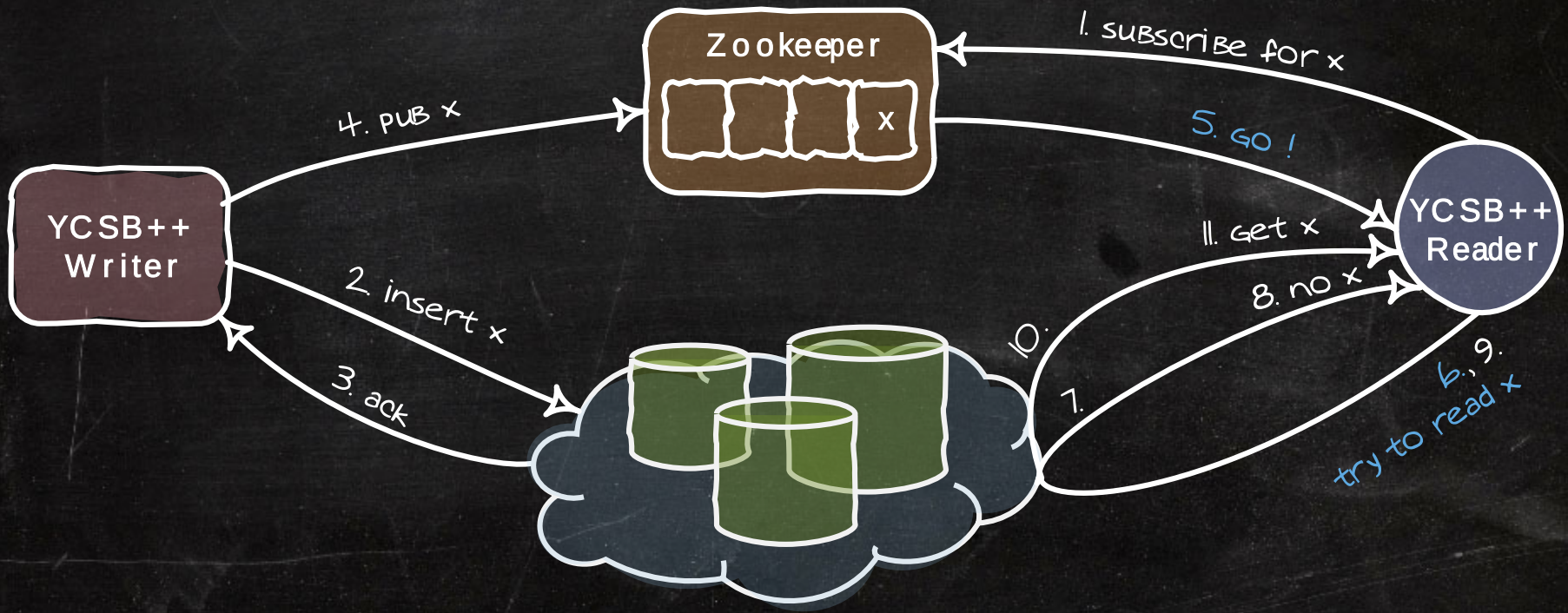
Existing Benchmarks

An example: YCSB++



Existing Benchmarks

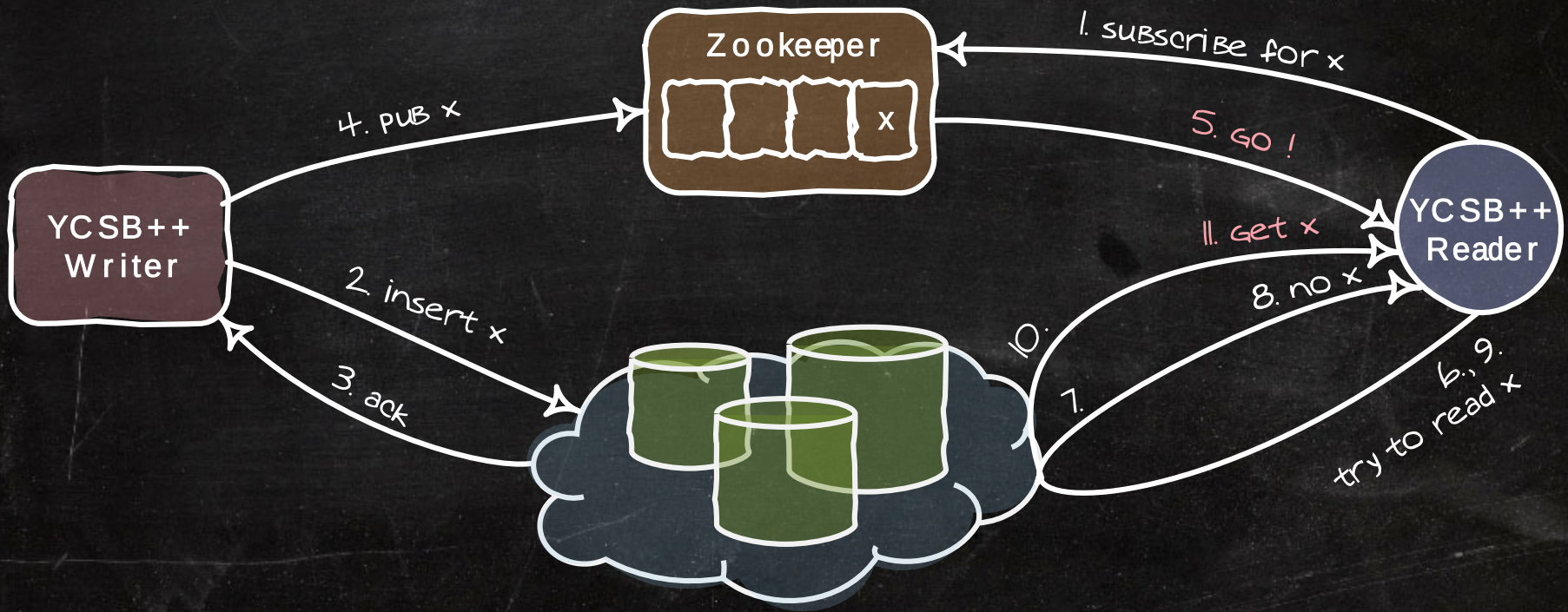
An example: YCSB++



possibly consistent between 7 and 10
but lower bound: 6 − 5 = 1

Existing Benchmarks

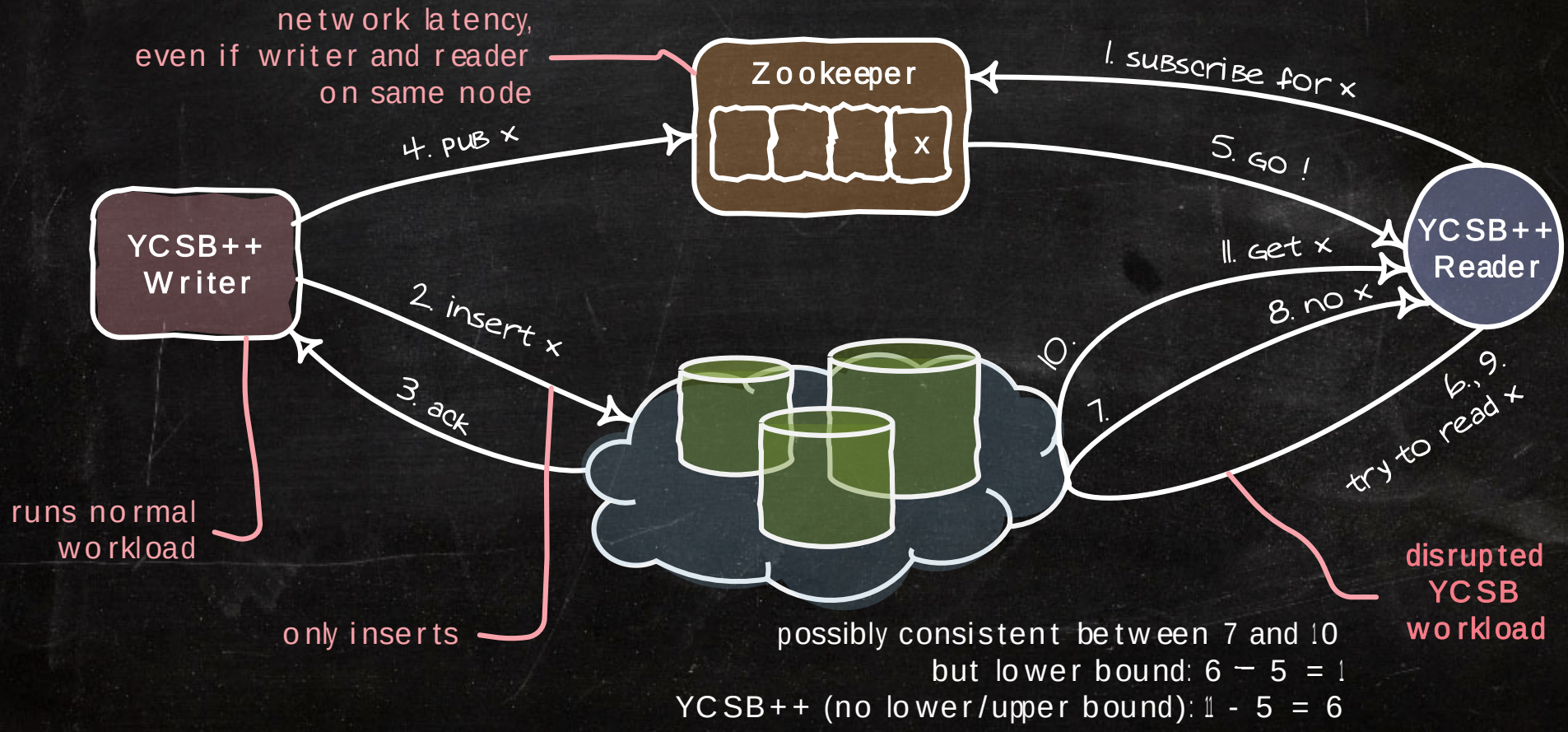
An example: YCSB++



possibly consistent between 7 and 10
but lower bound: 6 − 5 = 1
YCSB++ (no lower/upper bound): 11 - 5 = 6

Existing Benchmarks

An example: YCSB++



Existing Benchmarks

But how Bad is it?

Existing Benchmarks

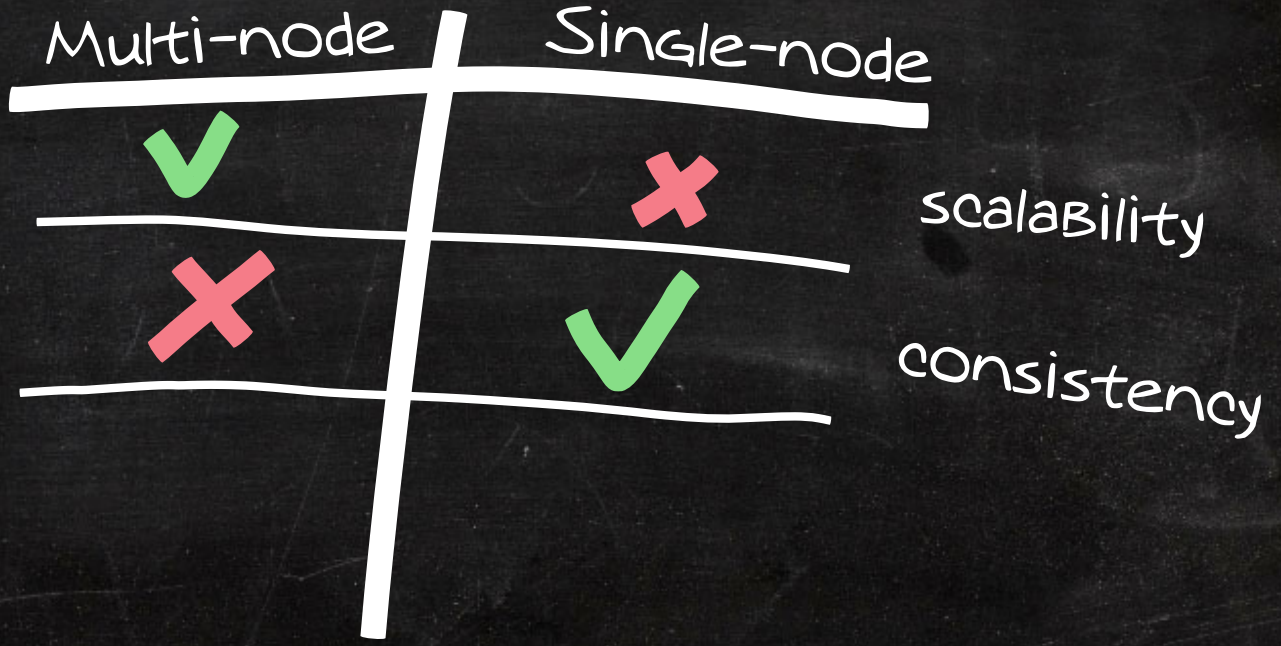
But how Bad is it?

hard to say, Because measurements are taken on the client side only → actual system state is unknown

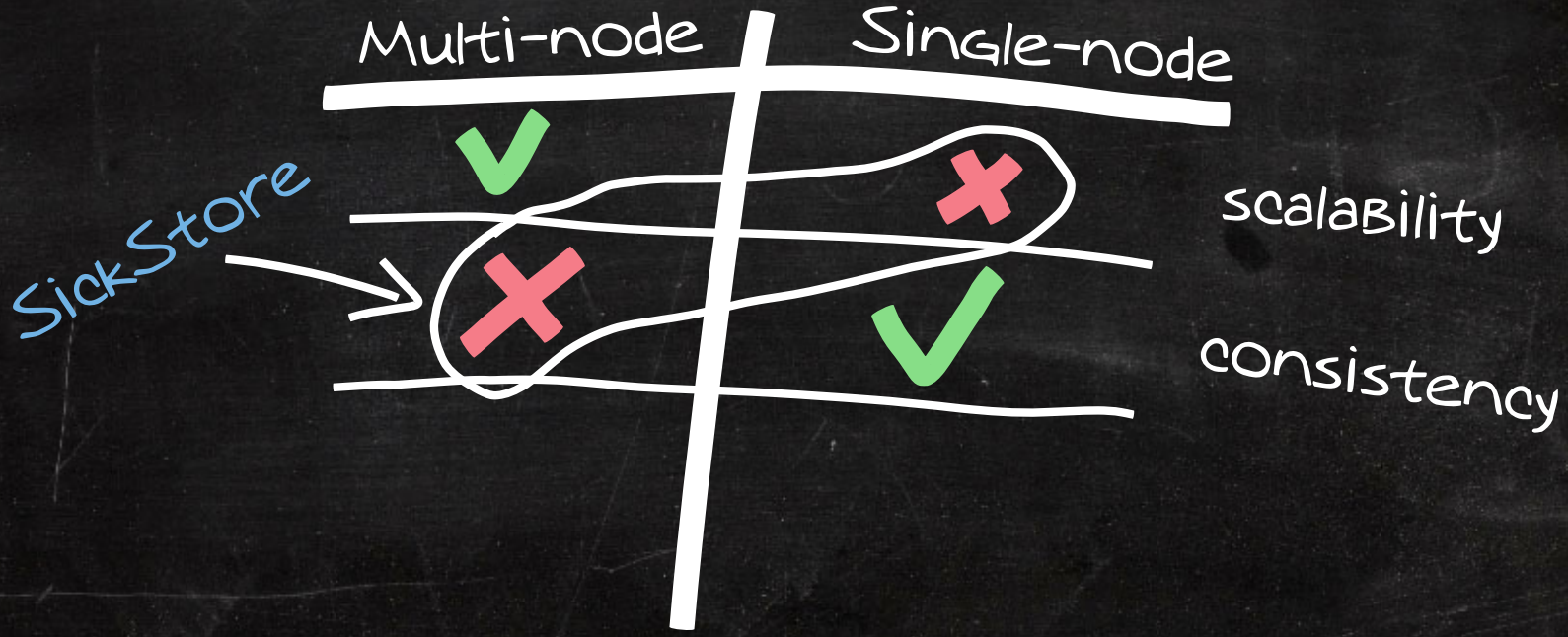
Single-Node Inconsistency to the rescue!

Enter SickStore

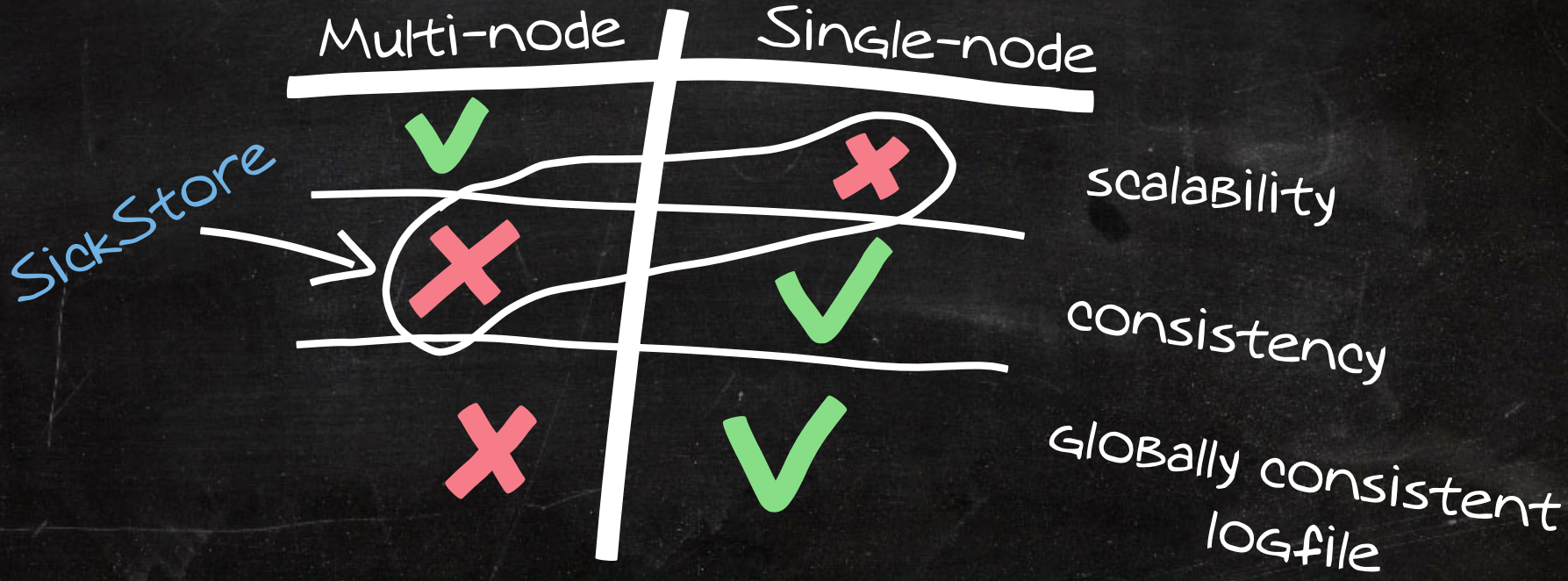
Single-Node Inconsistency to the rescue!



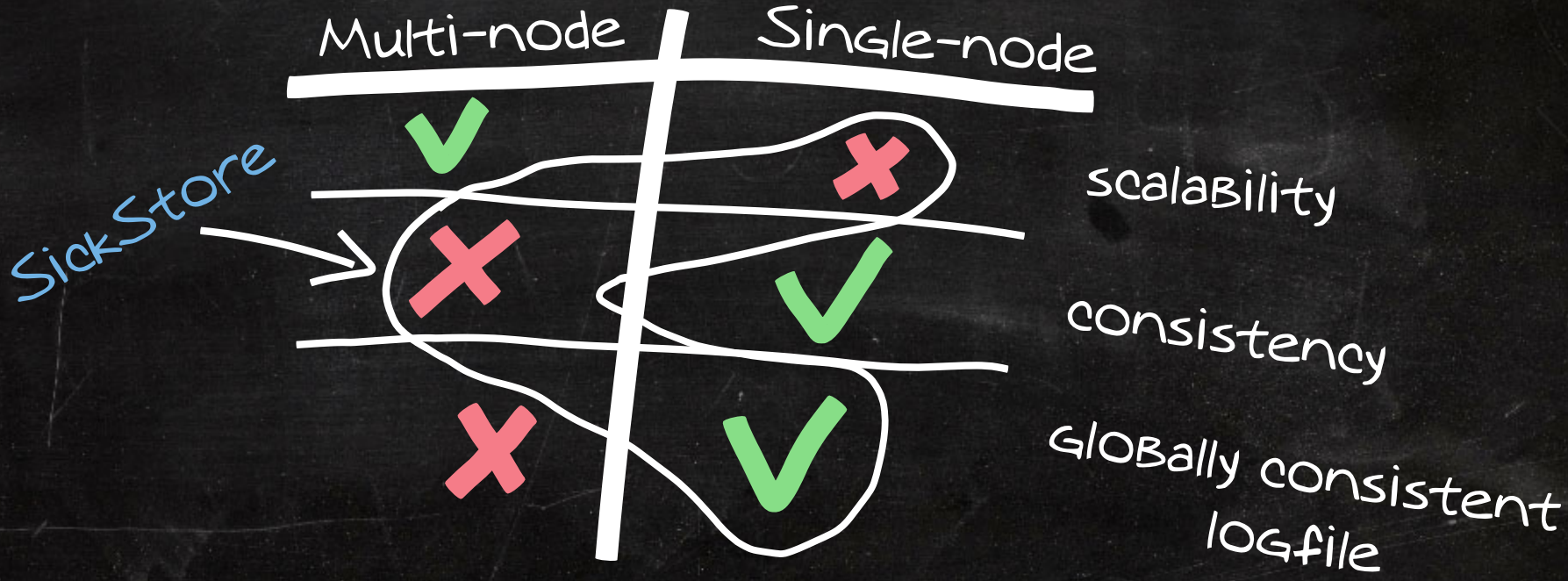
Single-Node Inconsistency to the rescue!



Single-Node Inconsistency to the rescue!



Single-Node Inconsistency to the rescue!

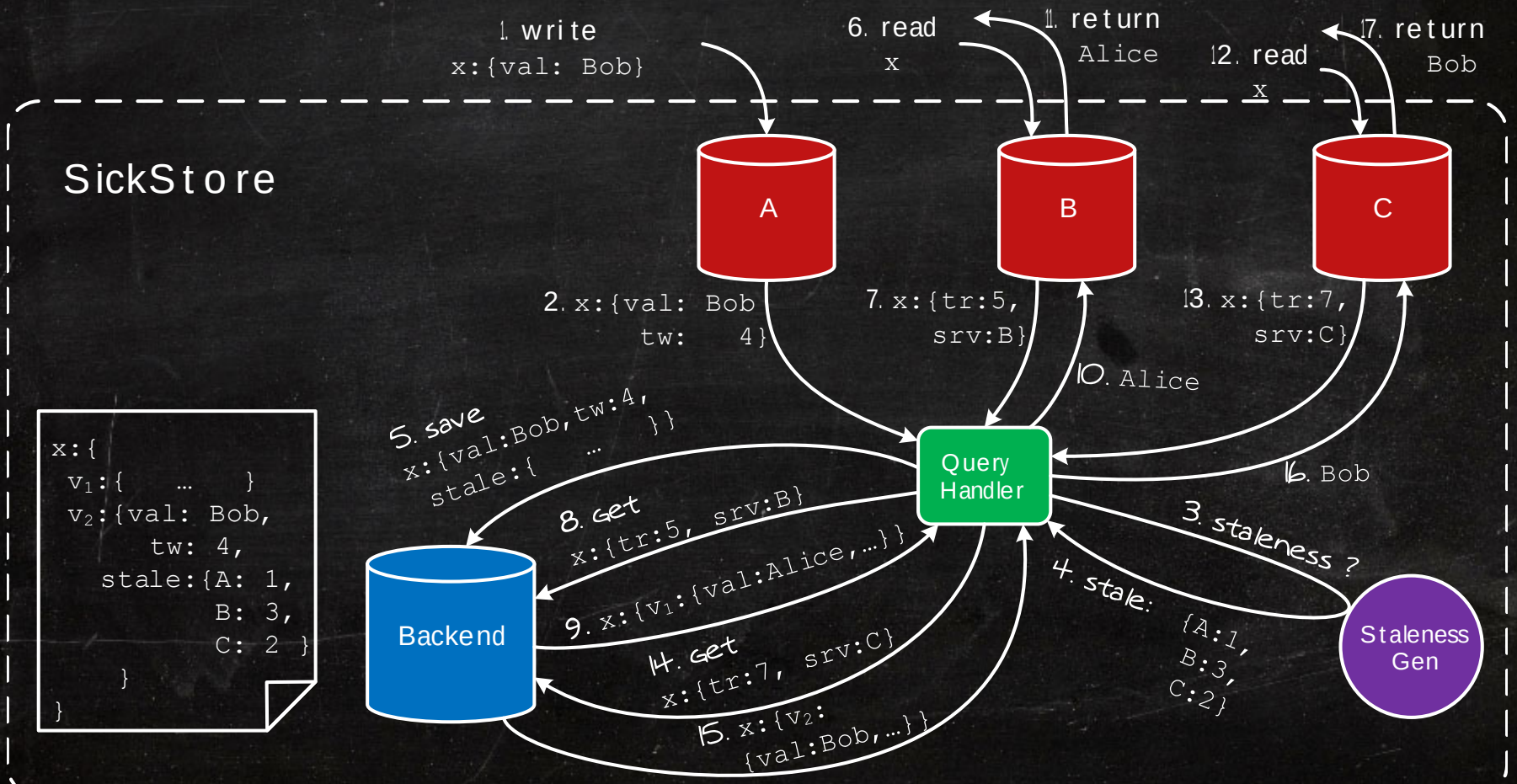


SickStore

Single-node inconsistent key-value Store

- consistent Backend: write requests are executed in their actual order (last-write-wins)
- multi-node Behaviour: queries are served By virtual storage nodes suBject to controlled anomalies
- tunaBle staleness: the delay By which data Become visiBle to storage nodes is configuraBle
- GloBally consistent logfile: no clock drift Between storage nodes → unamBiguous system state

SickStore
Architecture

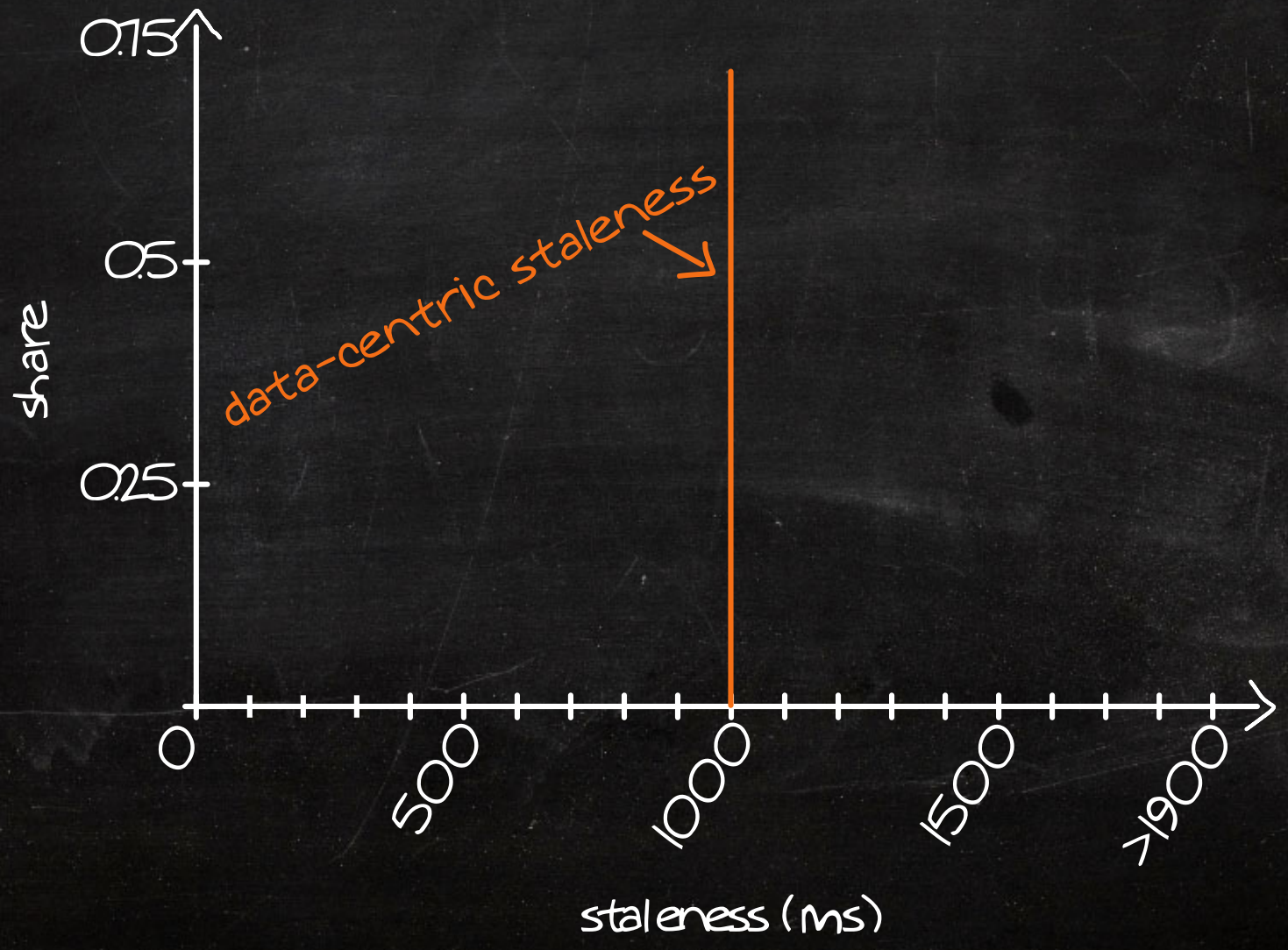


Experimental Validation

YCSB++

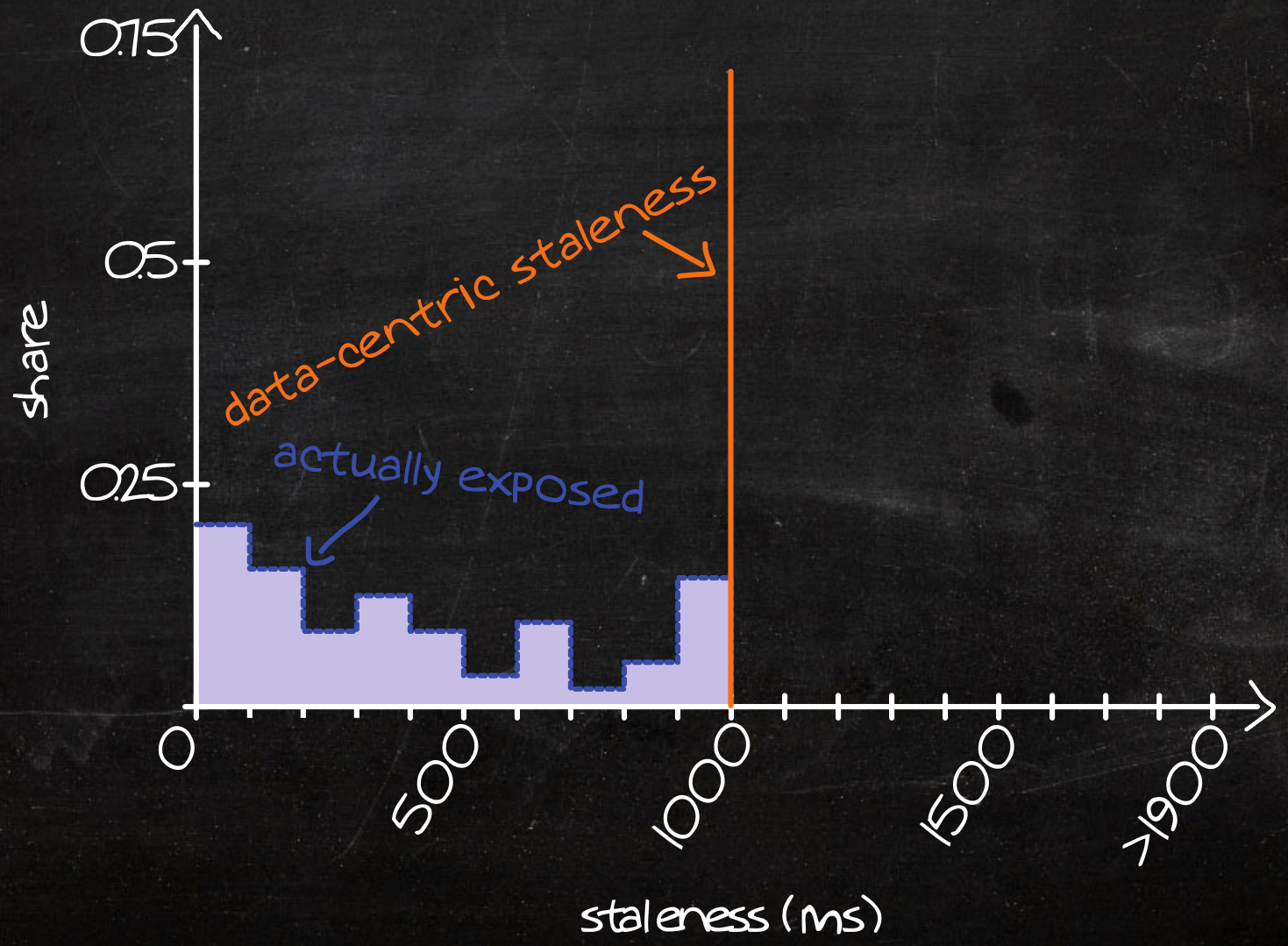
Experimental Validation: YCSB++

staleness 1 sec fixed @ 5 ops/sec over 40 seconds



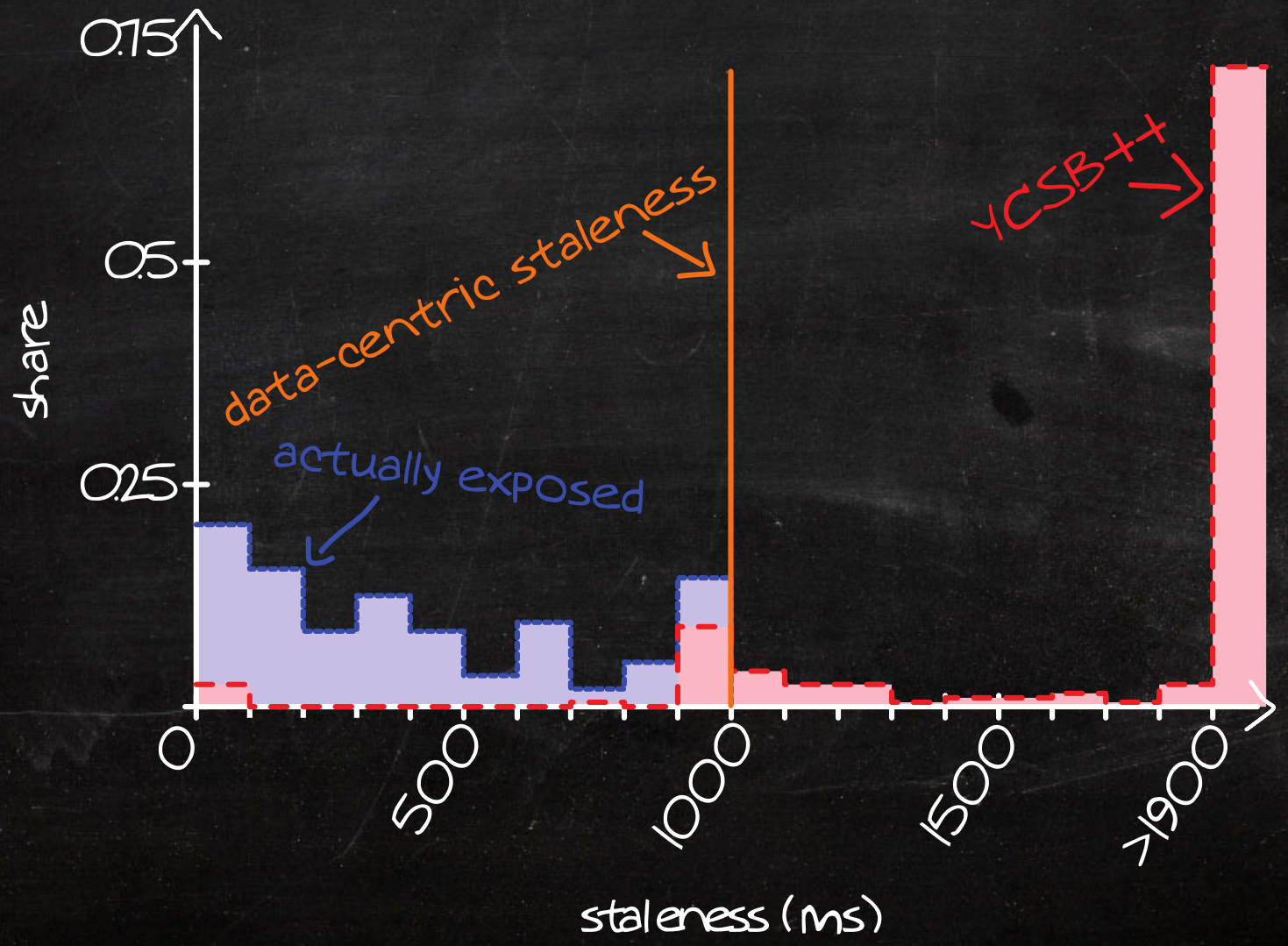
Experimental Validation: YCSB++

staleness 1 sec fixed @ 5 ops/sec over 40 seconds



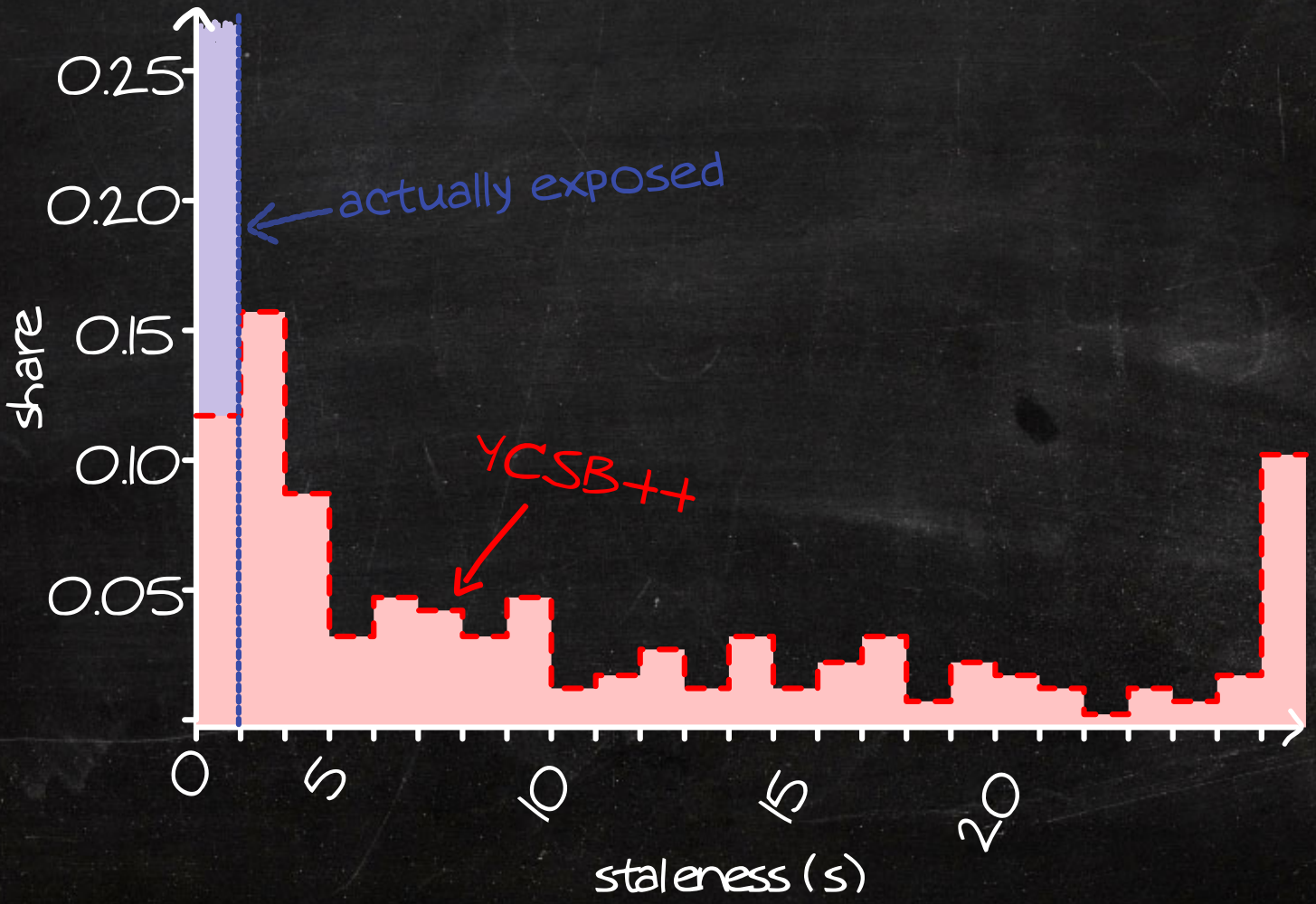
Experimental Validation: YCSB++

staleness 1 sec fixed @ 5 ops/sec over 40 seconds



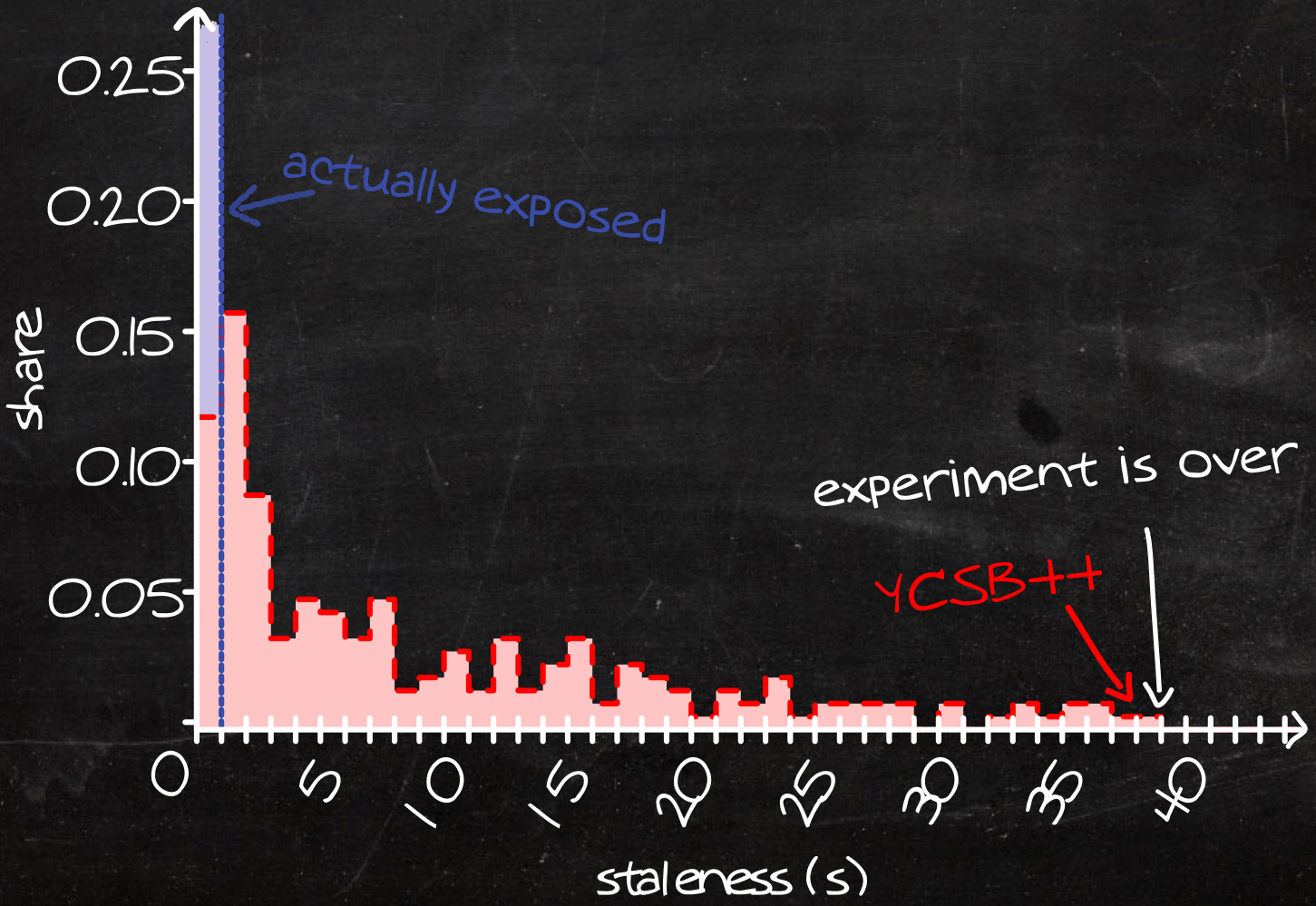
Experimental Validation: YCSB++

Not even by a long shot...



Experimental Validation: YCSB++

Not even by a long shot...



Experimental Validation: YCSB++

What went wrong?

Implementation error:

ZooKeeper Queue Basically was a stack!

Experimental Validation: YCSB++

What went wrong?

Implementation error:
Zookeeper Queue Basically was a stack!

Experimental Validation: YCSB++

What went wrong?

Implementation error:
ZooKeeper Queue Basically was a stack!

Experimental Validation: YCSB++

What went wrong?

Implementation error:

ZooKeeper Queue Basically was a stack!

Experimental Validation: YCSB++

What went wrong?

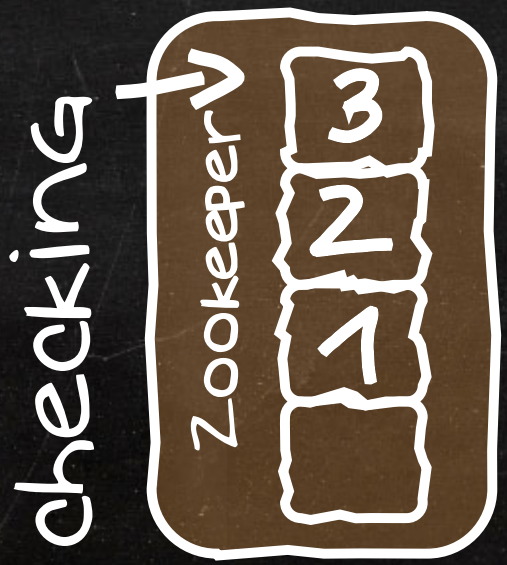
Implementation error:

ZooKeeper Queue Basically was a stack!

Experimental Validation: YCSB++

What went wrong?

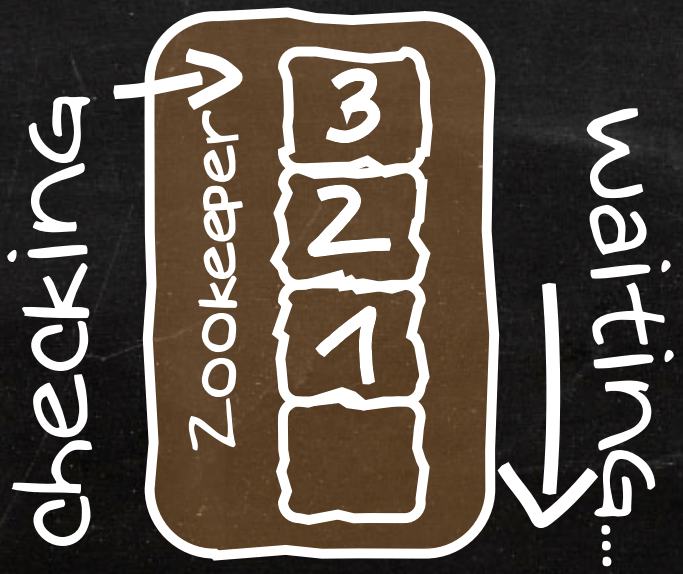
Implementation error:
ZooKeeper Queue Basically was a stack!



Experimental Validation: YCSB++

What went wrong?

Implementation error:
ZooKeeper Queue Basically was a stack!

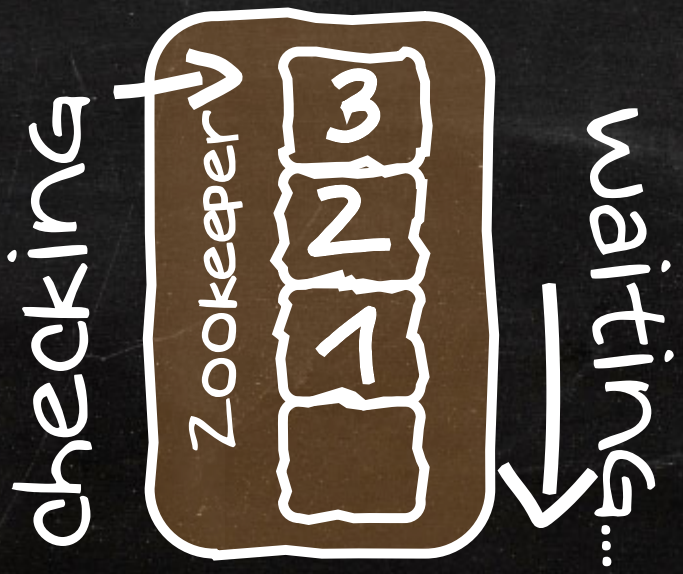


Experimental Validation: YCSB++

What went wrong?

Implementation error:
ZooKeeper Queue Basically was a stack!

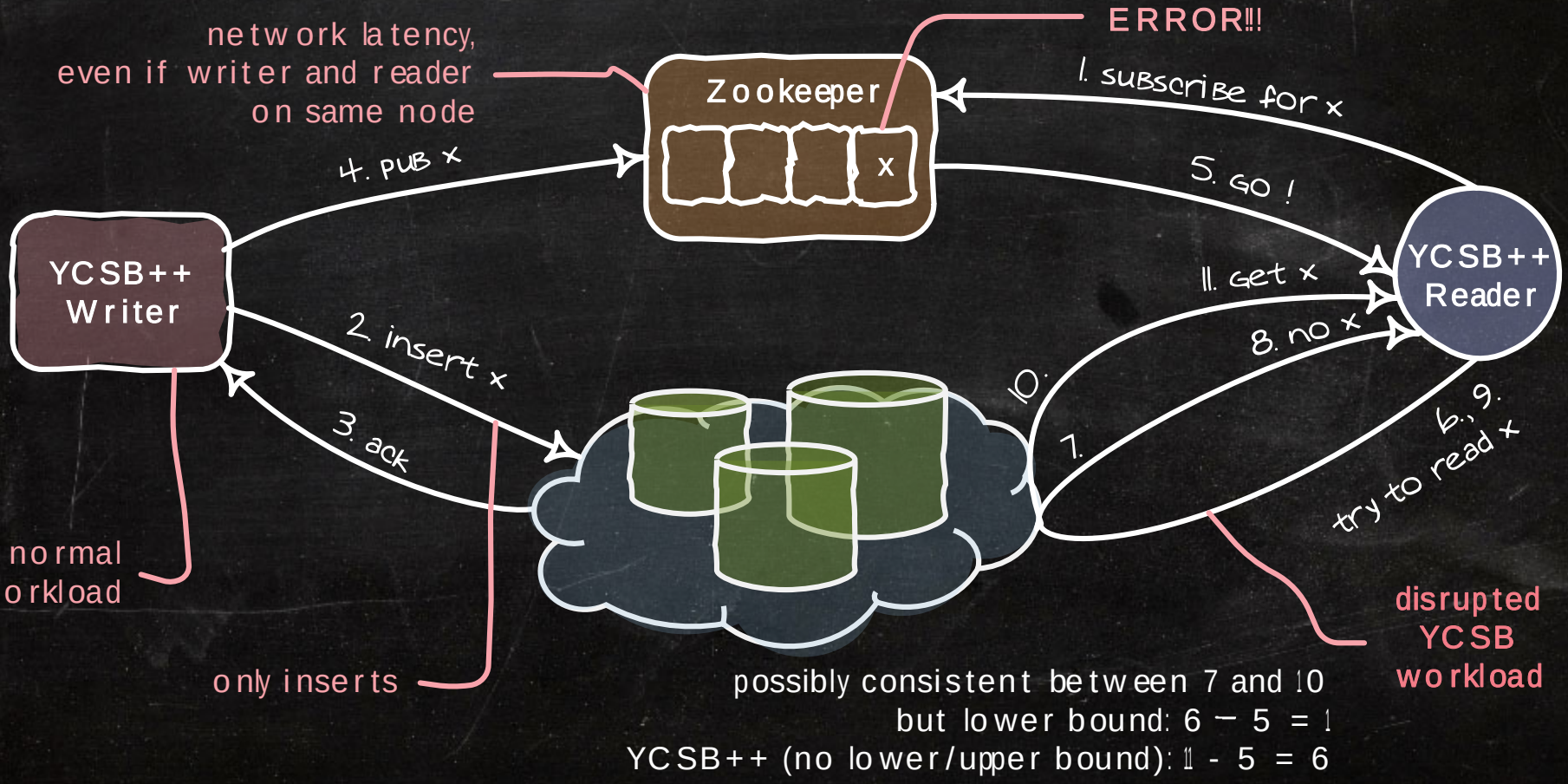
- → Max. staleness = experiment duration!
- → arBitrary Measurement error
- → staleness is detected, But not reasonably Quantified



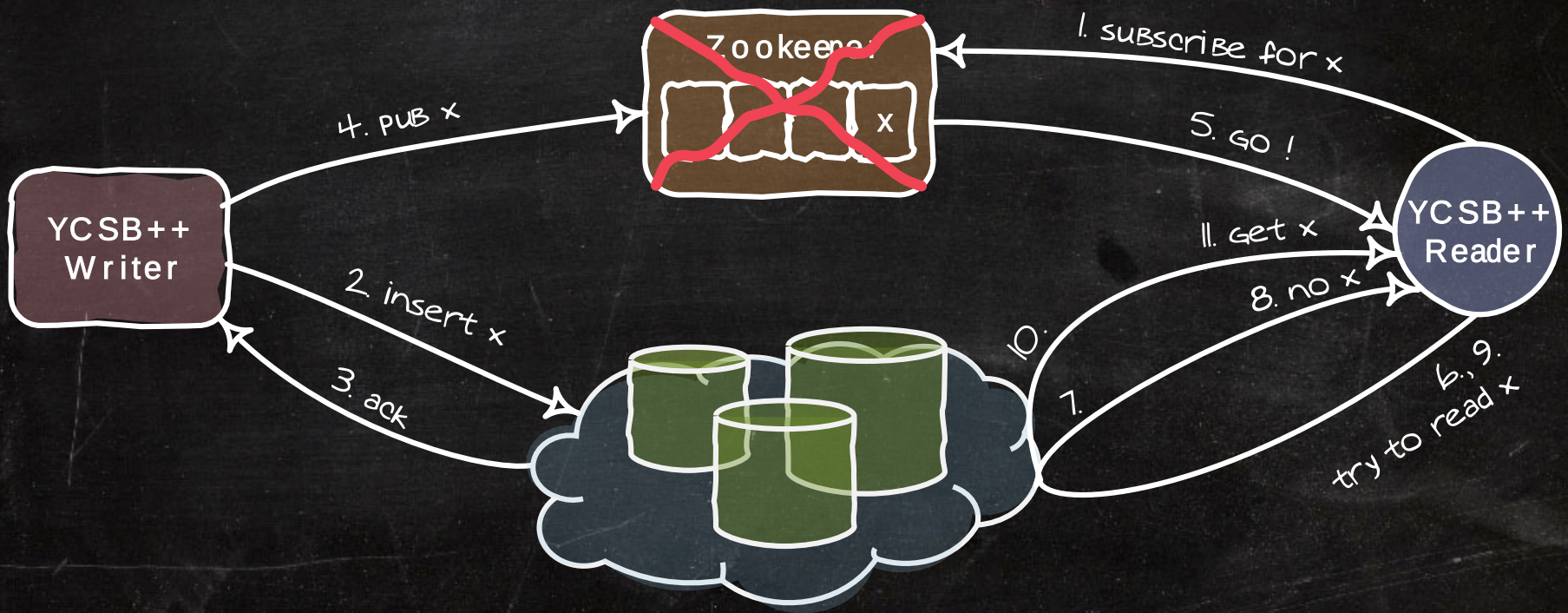
Next on our Agenda:

Valid Benchmarking with NoSQLMark

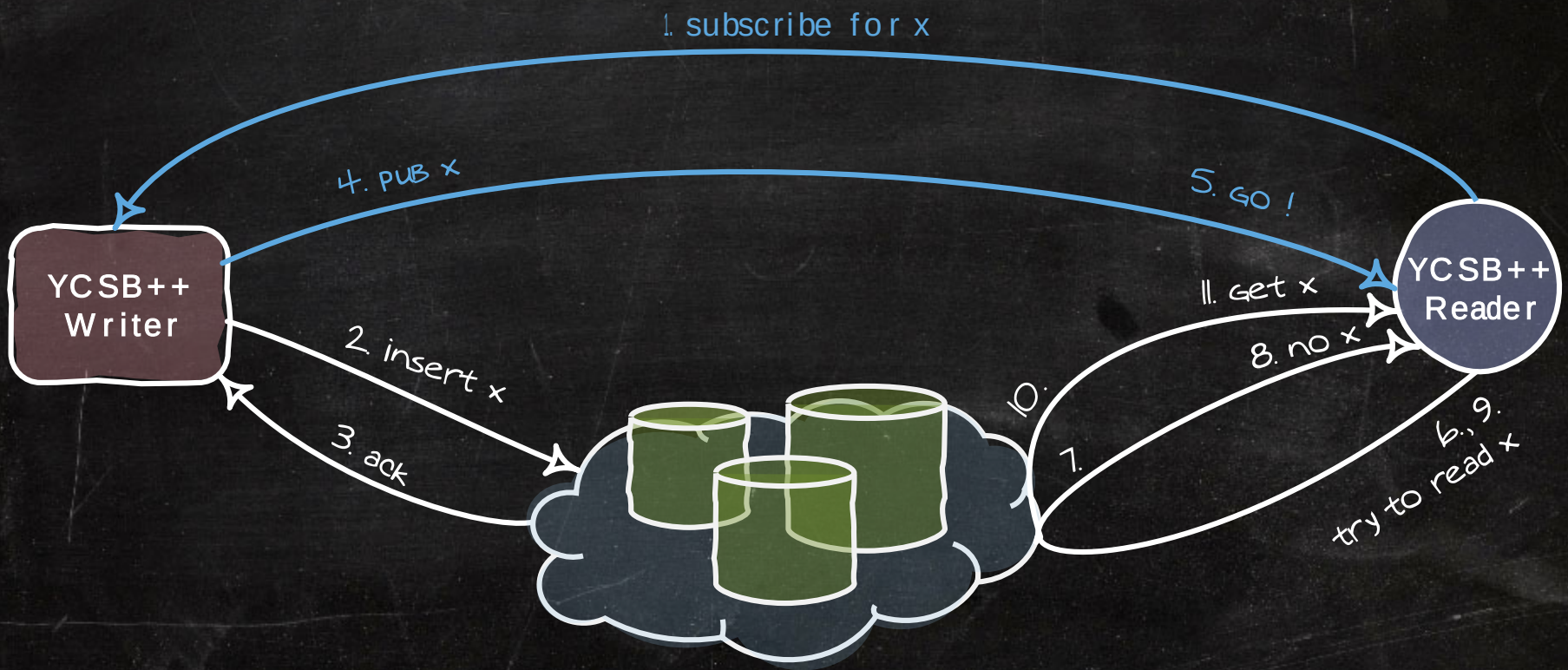
NoSQLMark: Improving on YCSB++



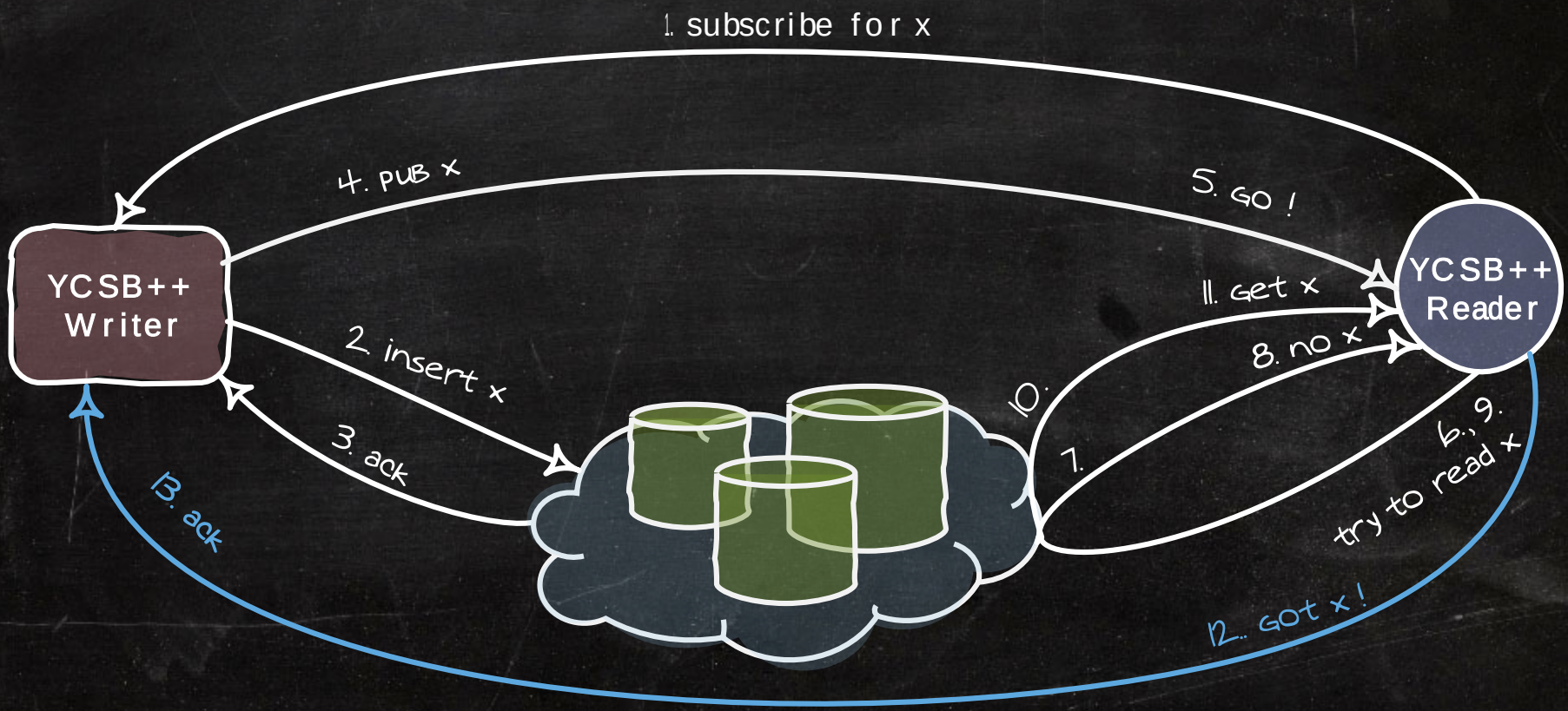
NoSQLMark: Improving on YCSB++



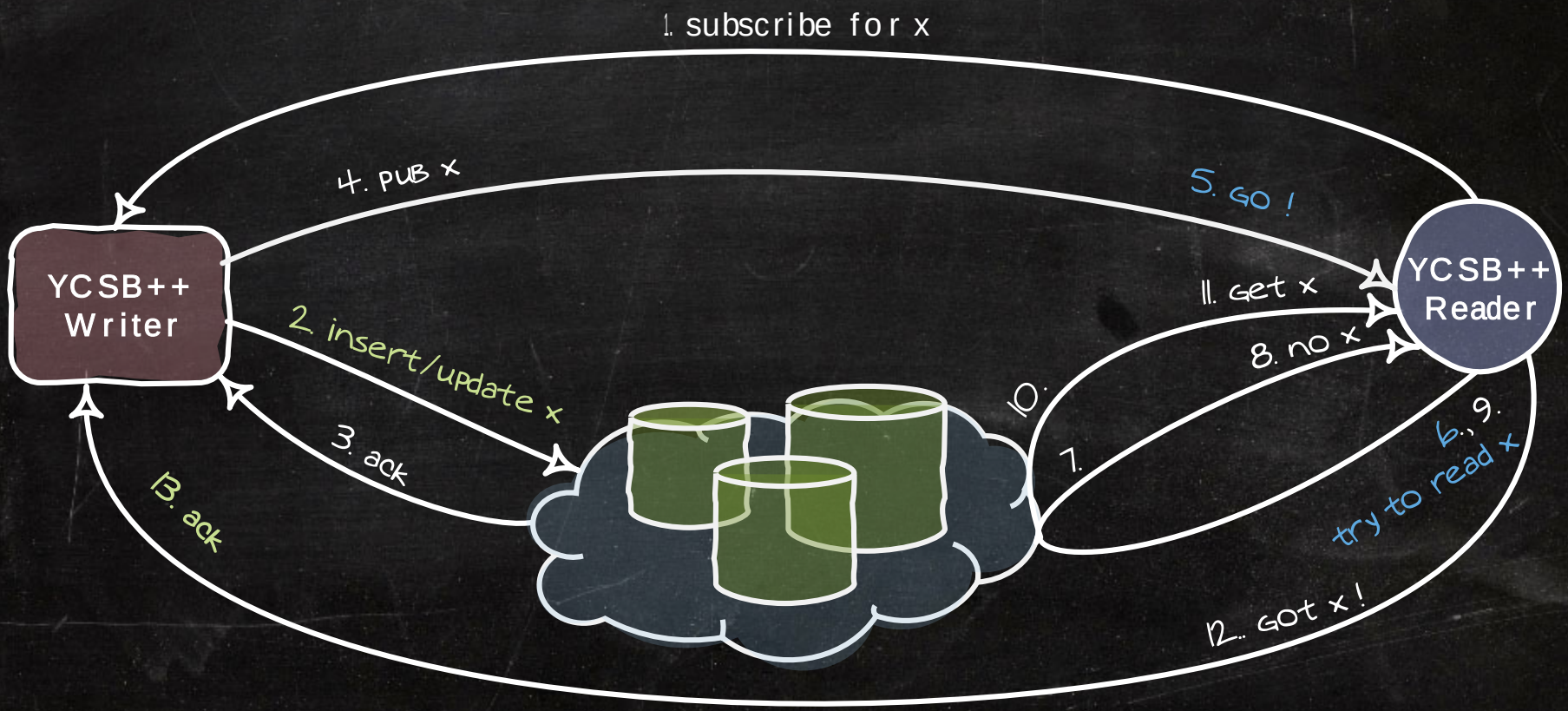
NoSQLMark: Improving on YCSB++



NoSQLMark: Improving on YCSB++



NoSQLMark: Improving on YCSB++



lower bound: 6 − 5 = 1
upper bound: 13 - 2 = 11

NoSQLMark
In a Nutshell

- **embedded pub-sub**: reduced network communication
- **configurable disruption**: proportion, frequency etc. of consistency reads are tunable
- **unified workload**: no separation between workload and consistency measurement operations
- lower and upper **bounds for staleness**
- experimental **verification** through SickStore

References I

Bermbach, David:
Benchmarking Eventually Consistent Distributed Storage Systems.
Karlsruhe, KIT, Fakultät für Wirtschaftswissenschaften, Phdthesis, 2014

Barahmand, Sumita ; Ghandeharizadeh, Shahram:
BG: A Benchmark to Evaluate Interactive Social Networking Actions.
In: CIDR, 2013

Bermbach, David ; Tai, Stefan:
Eventual Consistency: How Soon is Eventual? An Evaluation of Amazon S3's Consistency Behavior.
In: Proceedings of the 6th Workshop on Middleware for Service Oriented Computing.
New York, NY, USA : ACM, 2011 (MW4SOC '11). –
ISBN 978-1-4503-1067-3, 1:1–1:6

Bermbach, David ; Tai, Stefan:
Benchmarking Eventual Consistency: Lessons Learned from Long-Term Experimental Studies.
In: Proceedings of the 2nd IEEE International Conference on Cloud Engineering (IC2E), IEEE, 2014. –
Best Paper Runner Up Award

References II

- Dey, Akon ; Fekete, Alan ; Nambiar, Raghunath ; Röhm, Uwe: YCSB+T: Benchmarking Web-scale Transactional Databases. In: Proceedings of International Workshop on Cloud Data Management (CloudDB'14). Chicago, USA, 2014
- Golab, Wojciech ; Li, Xiaozhou ; Shah, Mehul A.: Analyzing Consistency Properties for Fun and Profit. In: Proceedings of the 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing. New York, NY, USA : ACM, 2011 (PODC '11). – ISBN 978-1-4503-0719-2, 197-206
- Patil, Swapnil ; Polte, Milo ; Ren, Kai ; Tantisiriroj, Wittawat ; Xiao, Lin ; López, Julio ; Gibson, Garth ; Fuchs, Adam ; Rinaldi, Billie: YCSB++: Benchmarking and performance debugging advanced features in scalable table stores. In: Proceedings of the 2nd ACM Symposium on Cloud Computing. New York, NY, USA : ACM, 2011 (SOCC '11). – ISBN 978-1-4503-0976-9, 9:1-9:14

References III

- Rahman, Muntasir R. ; Golab, Wojciech ; AuYoung, Alvin ; Keeton, Kimberly ; Wylie, Jay J.:
  Toward a Principled Framework for Benchmarking Consistency.
  In: Proceedings of the Eighth USENIX Conference on Hot Topics in System Dependability.
  Berkeley, CA, USA : USENIX Association, 2012 (HotDep'12), 8–8
- Wada, Hiroshi ; Fekete, Alan ; Zhao, Liang ; Lee, Kevin ; Liu, Anna:
  Data Consistency Properties and the Trade-offs in Commercial Cloud Storage: the Consumers' Perspective.
  In: CIDR'11, 2011, S. 134–143

Questions?