Stronger Semantics for Low-Latency Geo-Replicated Storage

Wyatt Lloyd[*], Michael J. Freedman[*], Michael Kaminsky[†], and David G. Andersen[‡]

[*] Princeton University, [†] Intel Lab, [‡] Carnegie Mellon University

Presented by Mina Tahmasbi Arashloo

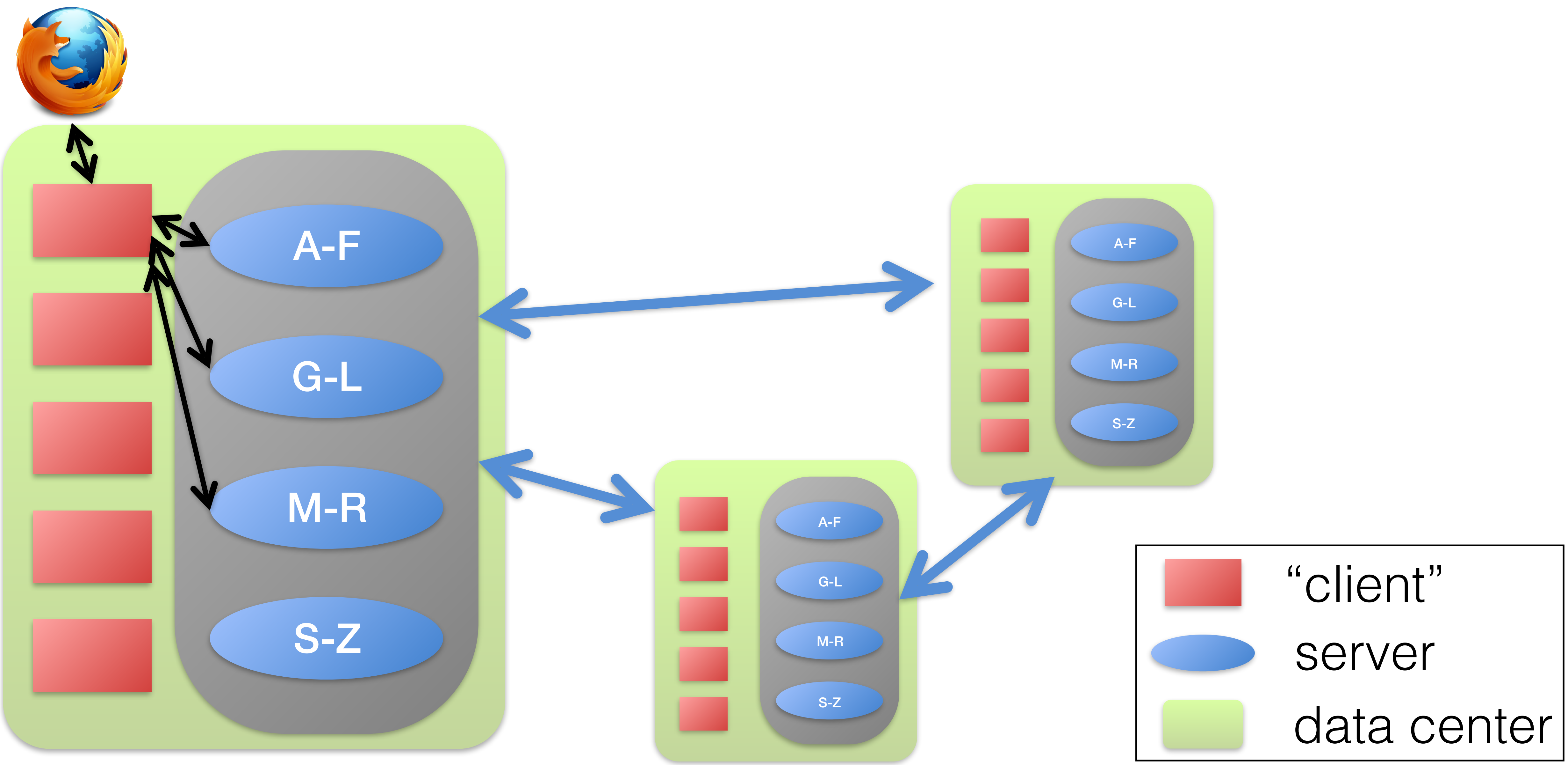
Motivation
and Problem Statement

- It is **provably impossible** to have the strongest forms of consistency and low latency in geo-replicated setting
- Key-value data model is **too simple**
- Can we build a system that provides **low latency**
  - with **stronger consistency** than eventual
  - for a **richer data model** ?

Contributions

- A scalable geo-replicated data store with:
  - low latency
  - causal consistency
  - support for column-family data model
  - read-only transactions
  - write-only transactions

Background : Web Service Architecture



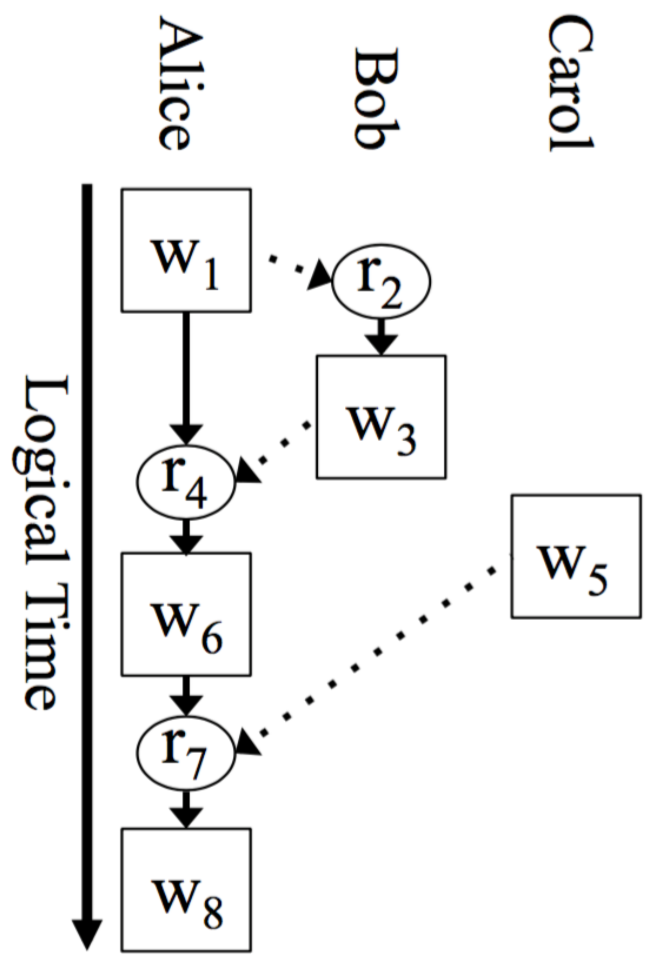
* The figure is adopted from Wyatt Lloyd's slides for his NSDI'13 presentation.

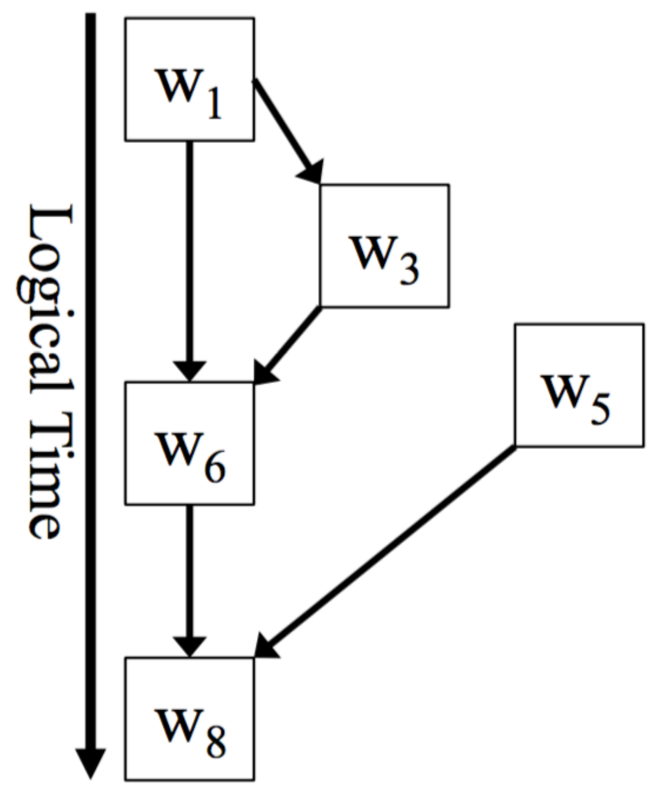
Background : Column-Family Data Model

- Pioneered by Google's BigTable
- A "map of maps of maps" of named columns

	User Data		Associations				
			Friends			Likes	
	ID	Town	Alice	Bob	Carol	Cats	Dogs
Alice	1337	NYC	-	3/2/11	9/2/12	9/1/12	-
Bob	2664	LA	3/2/11	-	-	-	-
⋮							

Background: Causal Consistency





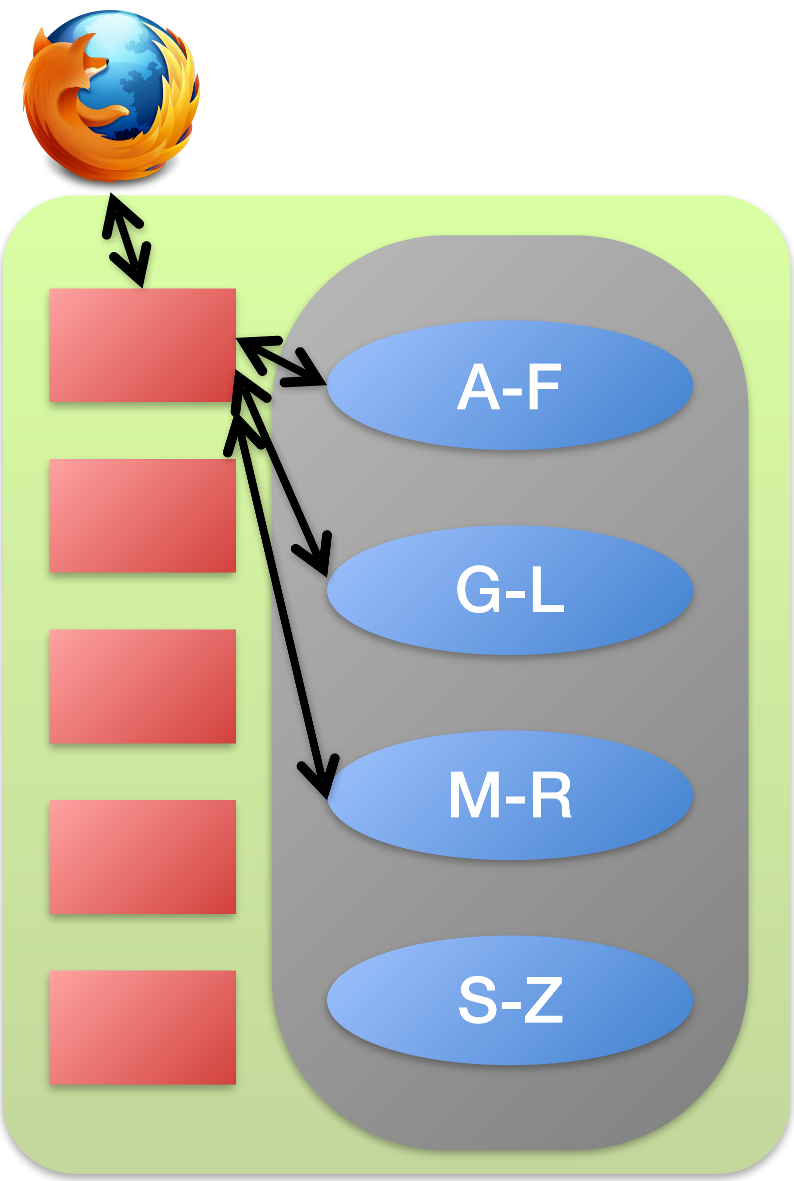
Op	Dependencies
w_1	-
w_3	$\underline{\boldsymbol{w}_1}$
w_5	-
w_6	$\underline{\boldsymbol{w}_3}$ $\underline{w_1}$
w_8	$\underline{\boldsymbol{w}_6}$ $\underline{\boldsymbol{w}_5}$ w_3 w_1

Eiger - Assumptions

- Each data center should have:
    - Partitioned key-space across *logical* servers
    - Linearizability
    - Logical servers that are available unless the whole data center fails

Client Library

- Mediates access to the servers
  - Create sub-requests based how the keys are partitioned
- Tracks causality and attaches dependencies to writes:



Basic Operations

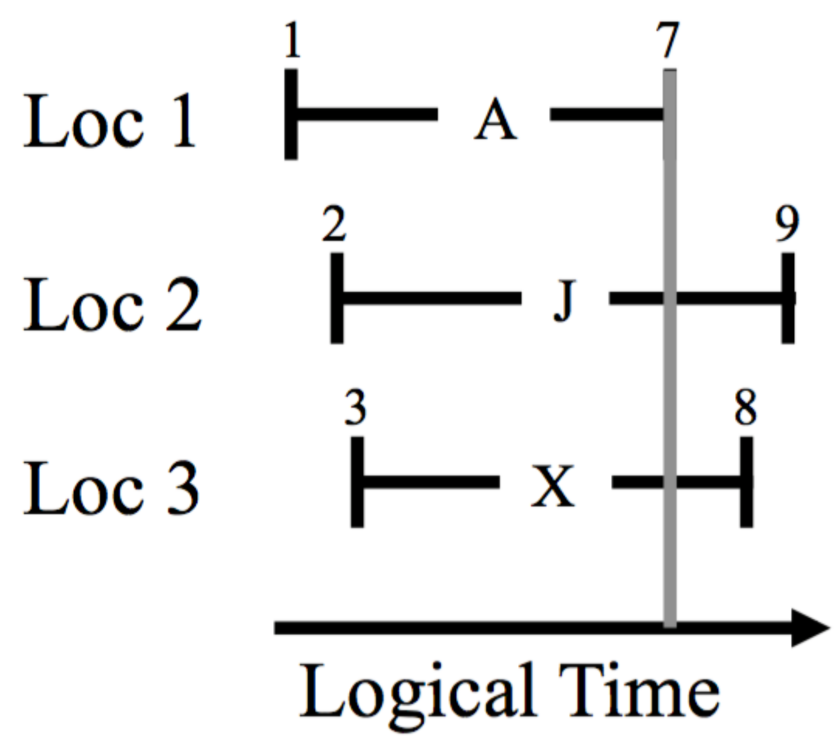
- Logical time
  - based on Lamport clocks
  - provide global timestamps:
    - stored with the data
- Read Operations
  - return the data and timestamp
  - timestamp used for tracking dependencies

Basic Operations

- Local writes
  - updates the value
  - records timestamp (with the server id)
- Replication
  - The remote server discards if it has a newer version (based on timestamp)
  - Handles writes conflicts!
  - last writer wins

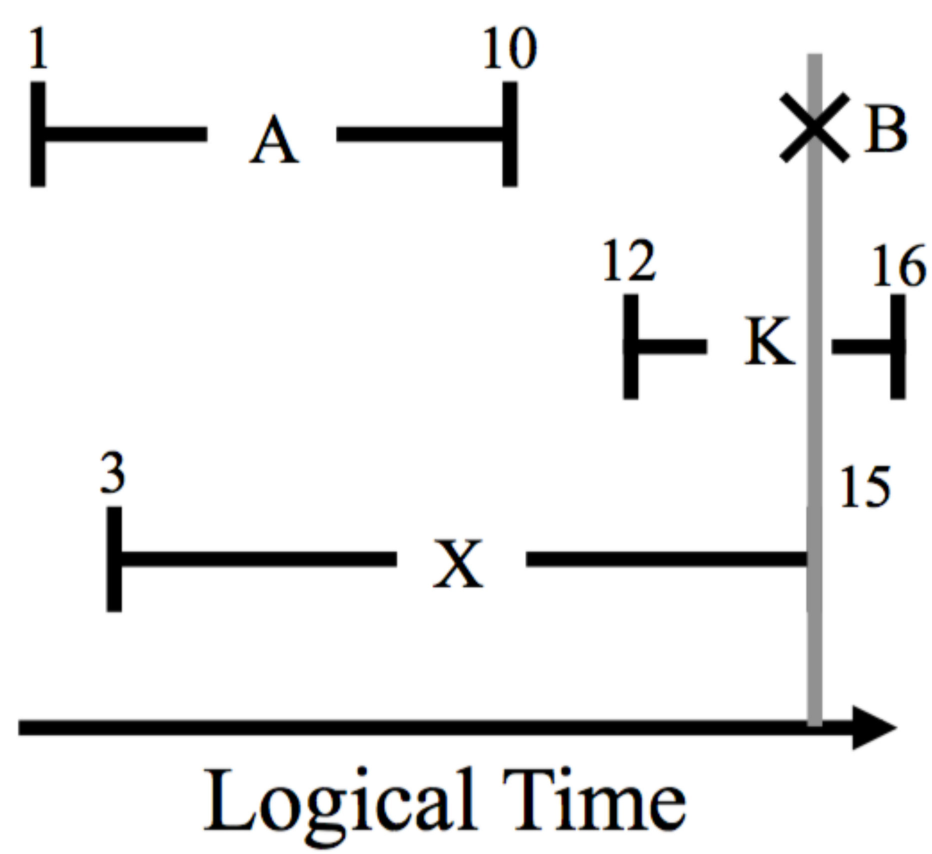
Read-Only Transactions

- First round:
  - receive *earliest valid time* (EVT) and *latest valid time* (LVT) from each server
  - If minimum LVT >= maximum EVT, there is a time where all the values are valid (*effective time*)



Read-Only Transactions

- Second round:
  - Ask the server for the location value at the *effective time*



Write-Only Transactions

- **Two-phase commit** with positive cohorts and indirection (2PC-PCI)
- **The client library**
  - chooses one key as the coordinator
  - sends sub-requests to corresponding servers with the coordinator key
- **Each server**
  - writes the value with “pending” status
  - sends a “YESVOTE” to coordinator
- **Coordinator**
  - timestamps the transaction
  - sends “COMMIT” to participants

Write-Only Transactions

- Each transaction sub-request is replicated
- Each remote server
  - sends a “NOTIFTY” to the remote coordinator
- The remote coordinator
  - checks dependencies
  - sends “PREPARE” messages
  - the rest continues similar to the local datacenter

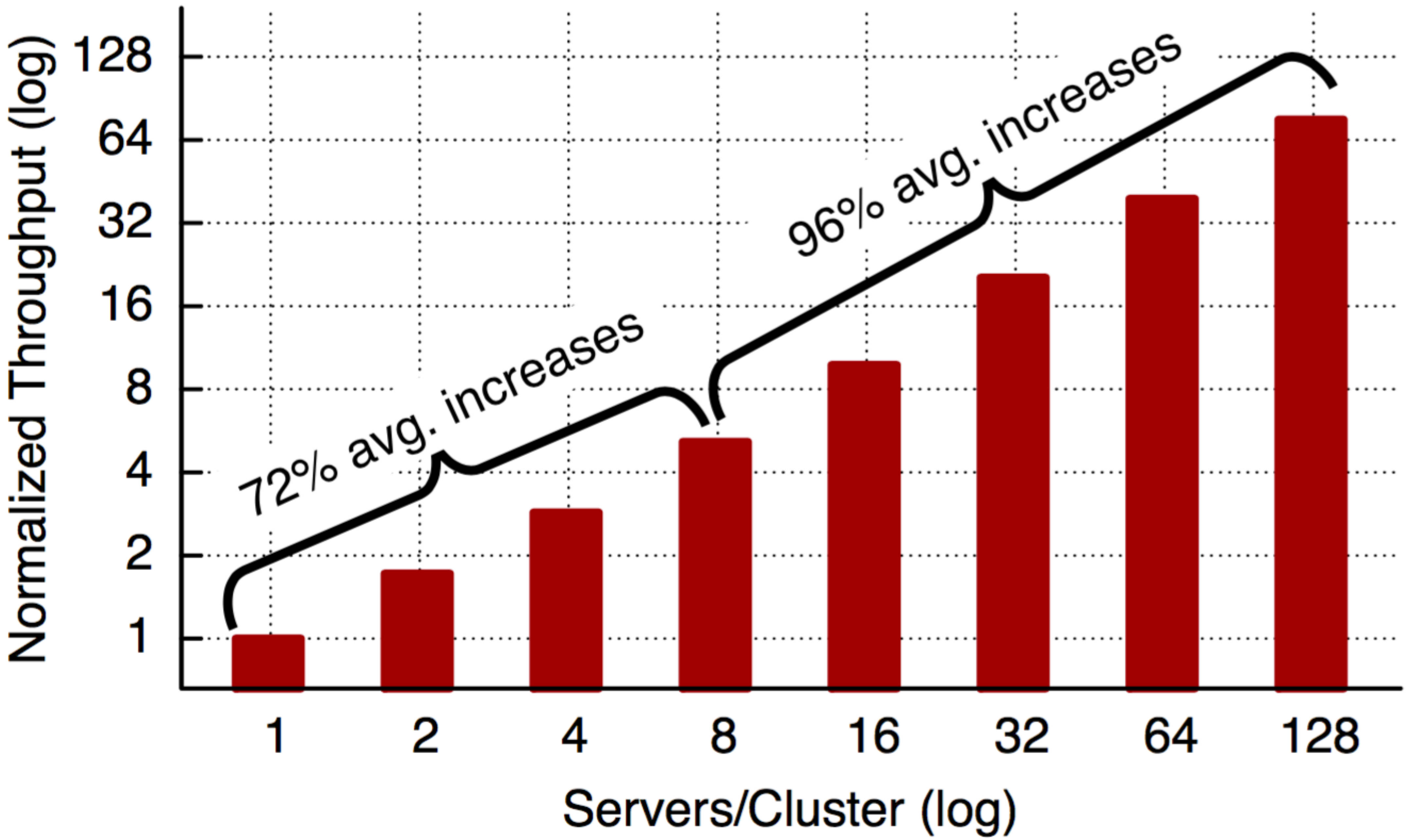
Failure

- Depends on the underlying building blocks assumptions for logical server's failure
- Transient datacenter failure : no ill effects
  - requires other datacenters to redirect the client to the original datacenter for configured period
- Long datacenter failures : causality loss
  - move to a new datacenter with empty context
- Permanent datacenter failure : data loss

Evaluation - Low Latency

	Latency (ms)			
	50%	**90%**	**95%**	**99%**
Reads				
Cassandra-Eventual	0.38	0.56	0.61	1.13
Eiger 1 Round	0.47	0.67	0.70	1.27
Eiger 2 Round	0.68	0.94	1.04	1.85
Eiger Indirected	0.78	1.11	1.18	2.28
Cassandra-Strong-A	85.21	85.72	85.96	86.77
Cassandra-Strong-B	21.89	22.28	22.39	22.92
Writes				
Cassandra-Eventual Cassandra-Strong-A	0.42	0.63	0.91	1.67
Eiger Normal	0.45	0.67	0.75	1.92
Eiger Normal (2)	0.51	0.79	1.38	4.05
Eiger Transaction (2)	0.73	2.28	2.94	4.39
Cassandra-Strong-B	21.65	21.85	21.93	22.29

Evaluation - Scalability



Related Work

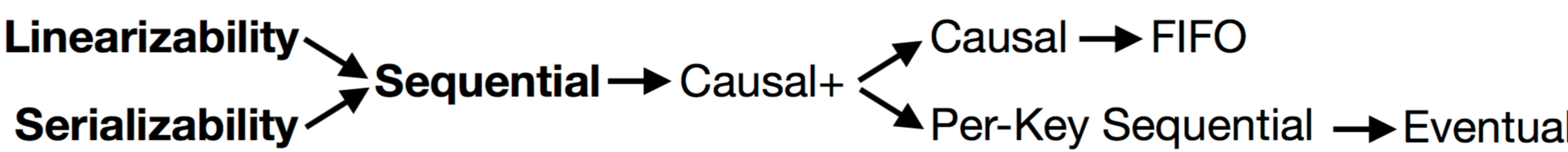
- Bayou
  - Requires single-machine replicas (datacenters)
- COPS
  - Also causal consistency, low latency, and read-only transactions
  - Eiger has richer data model, more powerful abstractions, and has less dependency overhead

Strengths

- Has low latency despite being geo-replicated
- Provides stronger consistency guarantees than previous work with negligible overhead
- Offers fast and non-blocking read-only and write-only transactions
- Scales almost linearly with #servers/datacenter
- Solid evaluation and comparison to previous work

Weaknesses

- Limited transactions
  - Read-only
  - Write-only
- Limited to causal consistency



* The figure is taken from Wyatt Lloyd's PhD thesis.

Questions?
Comments?

- References

  - Lloyd, Wyatt, et al. "Stronger Semantics for Low-Latency Geo-Replicated Storage." *NSDI*. 2013.
  - Chang, Fay, et al. "Bigtable: A distributed storage system for structured data." *ACM Transactions on Computer Systems (TOCS)* 26.2 (2008): 4.
