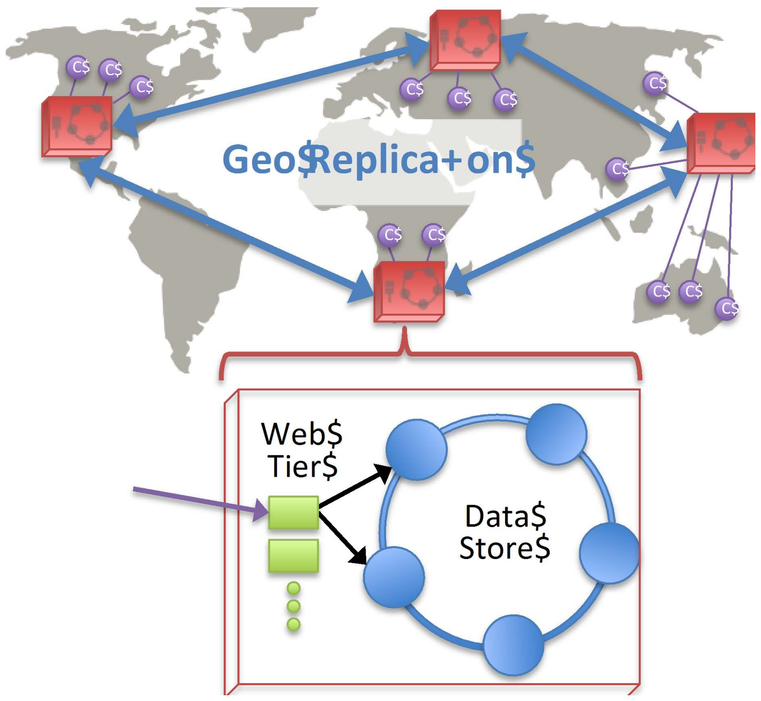
Eiger: Stronger Semantics for Low-Latency Geo-Replicated Storage

Wyatt Lloyd (Princeton), Michael J. Freedman (Princeton), Michael Kaminsky (Intel Labs), David G. Andersen (CMU)

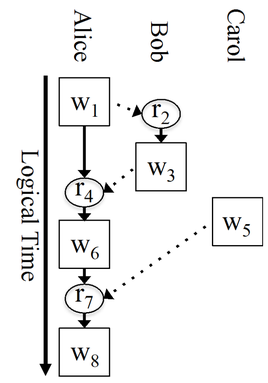
Motivation

- Distributed data stores support complex online applications
  - e.g. social networks
- Response times affect revenue
  - Low latency storage (<10ms) needed
- Massive scalability needed
  - Data is "big"
- Make system more useful
  - Causal Consistency
  - Richer Data Model
  - Read-only Transactions
  - Write-only Transactions

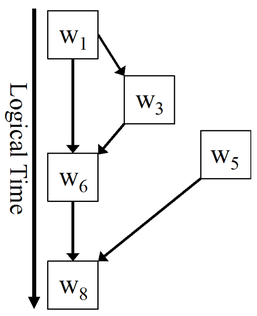


Causal Consistency

- Related ops appear in the correct order
- Track with dependencies on previous ops



Causality Graph



Dependency Graph

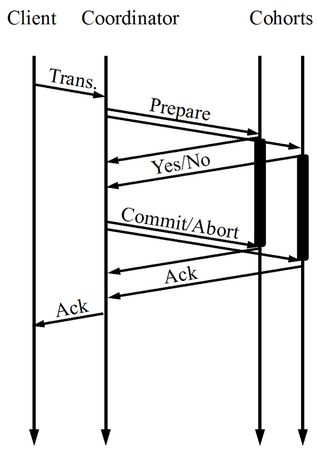
Richer Data Model

	Col Family 1		Col Family 2				
			Super Col 1			Super Col 2	
	Col 1	Col 2	Col 7	Col 8	Col 9	Col 1	Col 2
Key 1	A	-	D	E	F	-	H
Key 2	B	C	-	-	G	-	I
⋮							

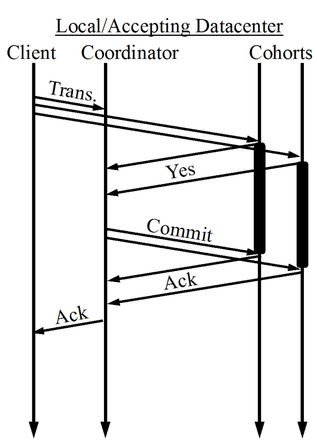
- Column-family data model
  - Map of maps of maps of columns
- Batch mutate / Multiget Slice
  - Writes / reads columns from many keys
  - Key technique for causal consistency: Deps on operations not values
- Counter Columns keep a commutatively updated integer
  - e.g. Like count

Write-Only Transactions

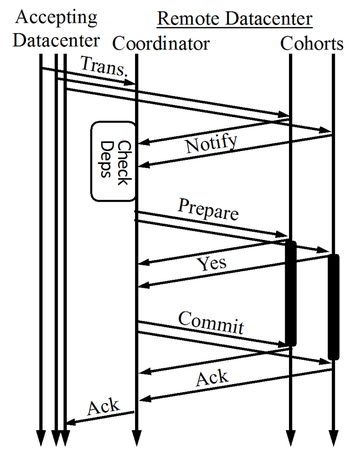
- Atomically write many columns across many keys in local datacenter
- Appear atomically (in causal order) in remote datacenters
- Guarantees low latency
  - At most 2.5 local RTTs to complete
  - No locks



Traditional 2PC



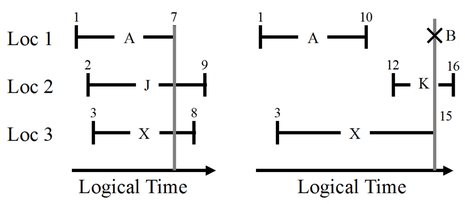
Local Write-Txn



Replicated Write-Txn

Read-Only Transactions

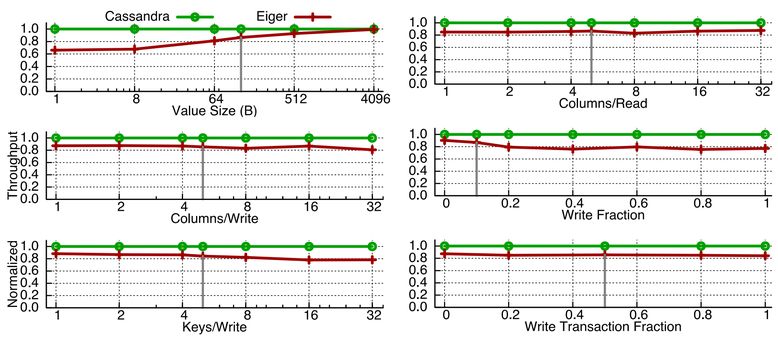
- Consistent view of many columns across many keys

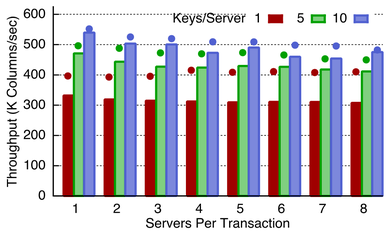


- Guarantees low latency
  - At most 3 local RTTs to complete
  - Normally only 1 local RTT to complete
- Use logical time metadata to ensure consistency

Evaluation

- Competitive with eventually-consistent and transaction-free Cassandra





- Write transactions are competitive with non-transactional batch updates

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · Princeton University · UC Berkeley