RAMCloud Overview and Update

SEDCL Retreat
June, 2014

John Ousterhout

Stanford University

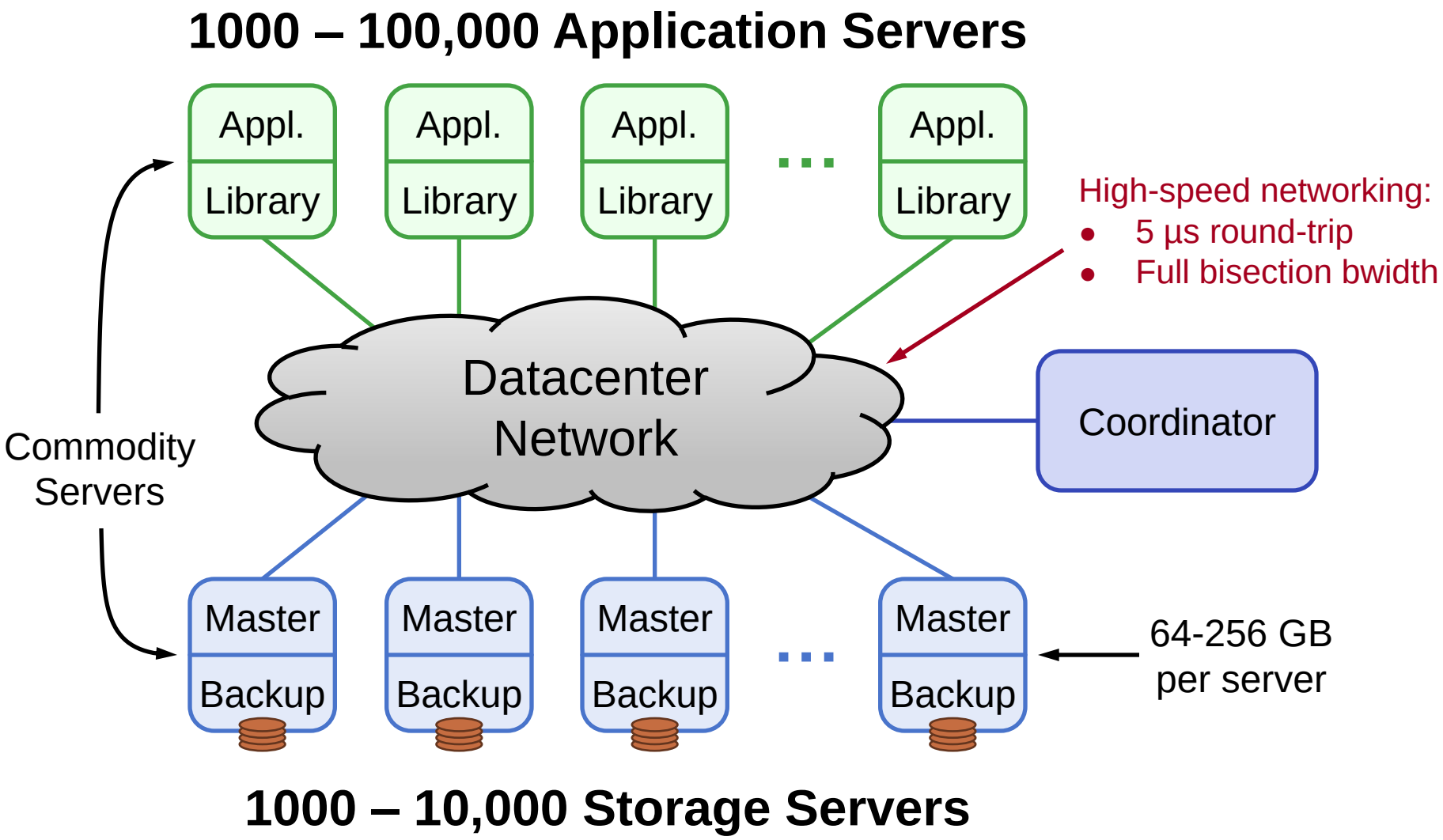
What is RAMCloud?

General-purpose storage system for large-scale applications:

- **All data is stored in DRAM at all times**
- **As durable and available as disk**
- **Simple key-value data model**
- **Large scale: 1000+ servers, 100+ TB**
- **Low latency: 5-10 µs remote access time**

Project goal: enable a new class of data-intensive applications

RAMCloud Architecture

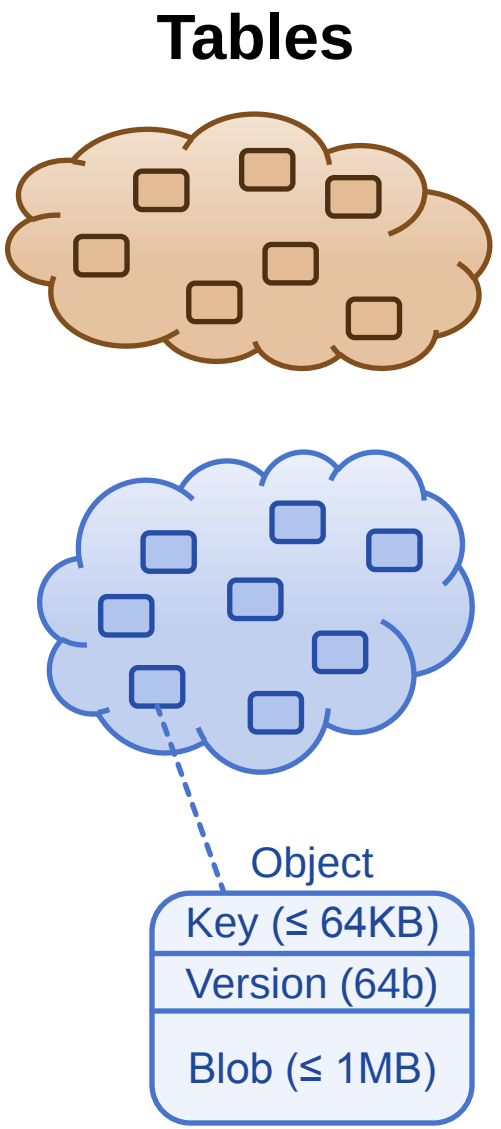


Data Model: Key-Value Store

- **Basic operations:**
  - `read(tableId, key)`
    `=> blob, version`
  - `write(tableId, key, blob)`
    `=> version`
  - `delete(tableId, key)`

(Only overwrite if version matches)

- **Other operations:**
  - `cwrite(tableId, key, blob, version)`
    `=> version`
  - Enumerate objects in table
  - Efficient multi-read, multi-write
  - Atomic increment
- **Not in RAMCloud 1.0:**
  - Atomic updates of multiple objects
  - Secondary indexes



Status at June 2013 Retreat

- **Close to 1.0 release:**
  - Core system becoming stable
  - Coordinator not yet fault-tolerant
- **Original students working on dissertations**
- **New students staring to think about new projects**

Progress Since June 2013

- **RAMCloud 1.0, January 2014:**
  - Key-value store
  - Low-latency RPC system (4.9 µs reads, 15.3 µs durable writes)
  - Log-structured storage management
  - 1-2 second recovery from storage server crashes
  - Coordinator crash recovery
- **New projects (see below)**
- **Application experiments/interest:**
  - Graph processing: Jonathan Ellithorpe
  - ONOS (operating system for software-defined networks) Open Networking Laboratory
  - Various projects/experiments at Huawei
  - High-energy physics(CERN): Jakob Blomer visiting for summer
  - Port to NEC Atom cluster: Satoshi Matsushita

Progress, cont'd

- **PhD dissertations:**
  - Ryan Stutsman: "Durability and Crash Recovery in Distributed In-memory Storage Systems"
    Now at Microsoft Research
  - Steve Rumble: "Memory and Object Management in RAMCloud"
    Now at Google Zurich
  - Diego Ongaro: "Consensus: Bridging Theory and Practice"
    ETA summer 2014
- **Papers published:**
  - "Log-Structured Memory for DRAM-Based Storage"
    Best Paper Award, FAST
  - "In Search of an Understandable Consensus Algorithm"
    USENIX ATC

Progress, cont'd

- **New papers submitted to OSDI:**
  - "SLIK: Scalable Low-Latency Indexes for a Key-Value Store" (Ankita, Arjun, Ashish, Zhihao)
  - "Experience with Rules-Based Programming for Distributed, Concurrent, Fault-Tolerant Code" (Ryan, Collin)

Changing of the Guard

Ryan Stutsman	Graduated (PhD)
Steve Rumble	Graduated (PhD)
Diego Ongaro	Graduating soon (PhD)
Ankita Kejriwal	
Arjun Gopalan	Graduating soon (MS)
Behnam Montazeri	
Collin Lee	New
Henry Qin	New
Ashish Gupta	New (but leaving with MS)
Seo Jin Park	New
Zhihao Jia	New (rotation only)
Stephen Yang	Rejoining Fall 2014

New Projects

RAMCloud 1.0

- First-generation RPC (based on Infiniband)
- Key-value store
- Log-structured storage management
- Crash recovery

Higher-Level Data Model

- Secondary indexes
- Linearizability
- Multi-object transactions
- Graph support?

Networking Infrastructure

- Analyze RPC latency
- Driver(s) for 10 GigE
- Clean-slate RPC redesign

Phase I: 2009 – 2013

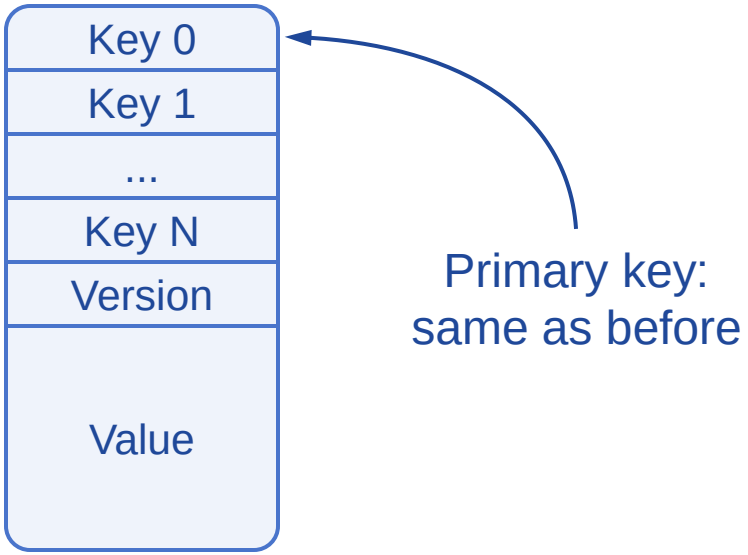
Phase II: 2014 – ?

Secondary Indexes

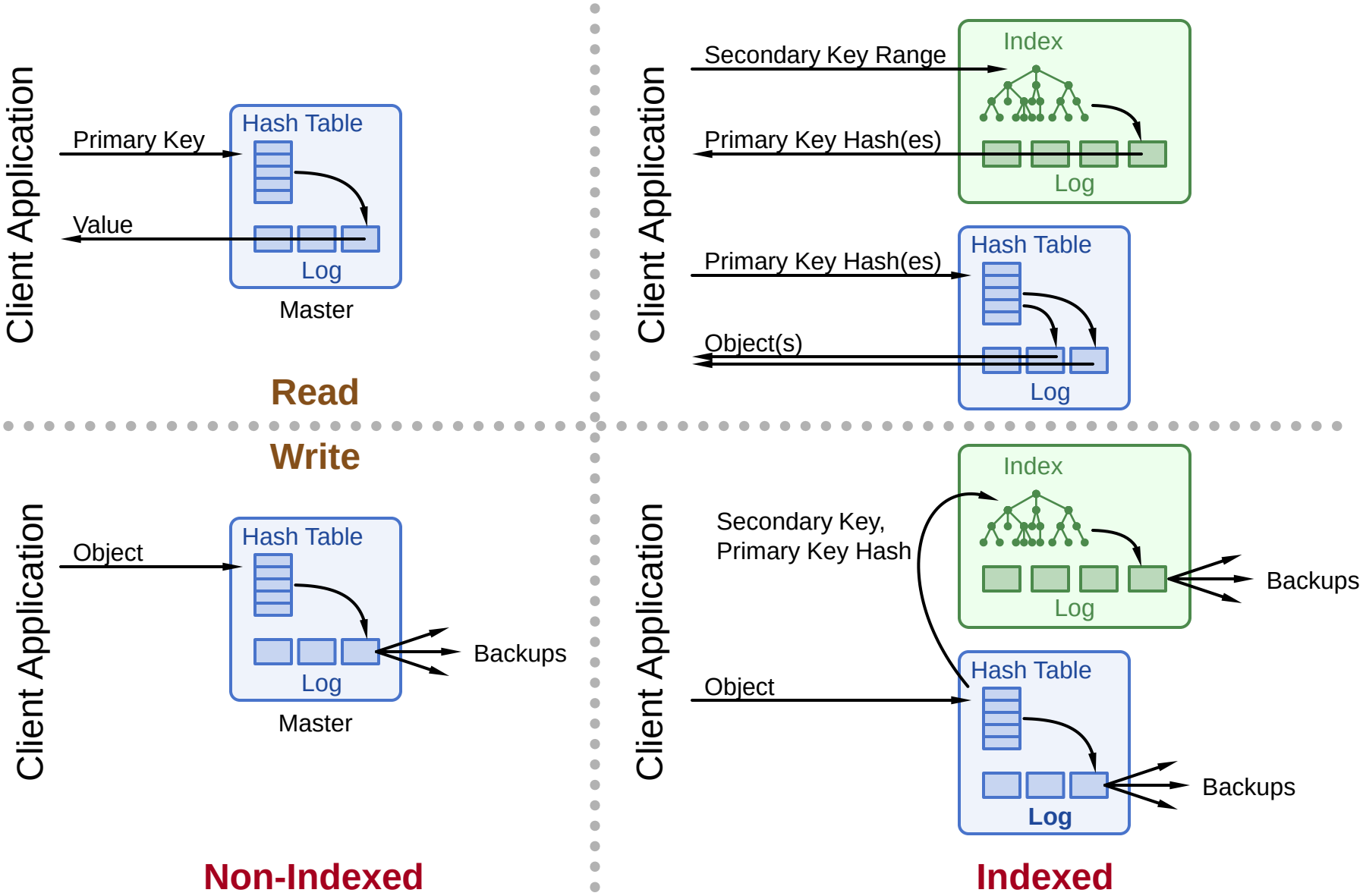
- **SLIK: Scalable, Low-latency Indexes for a Key-value Store**
- **Requires new object format:**

Old:
key-value store

New:
multikey-value store



RAMCloud Operations

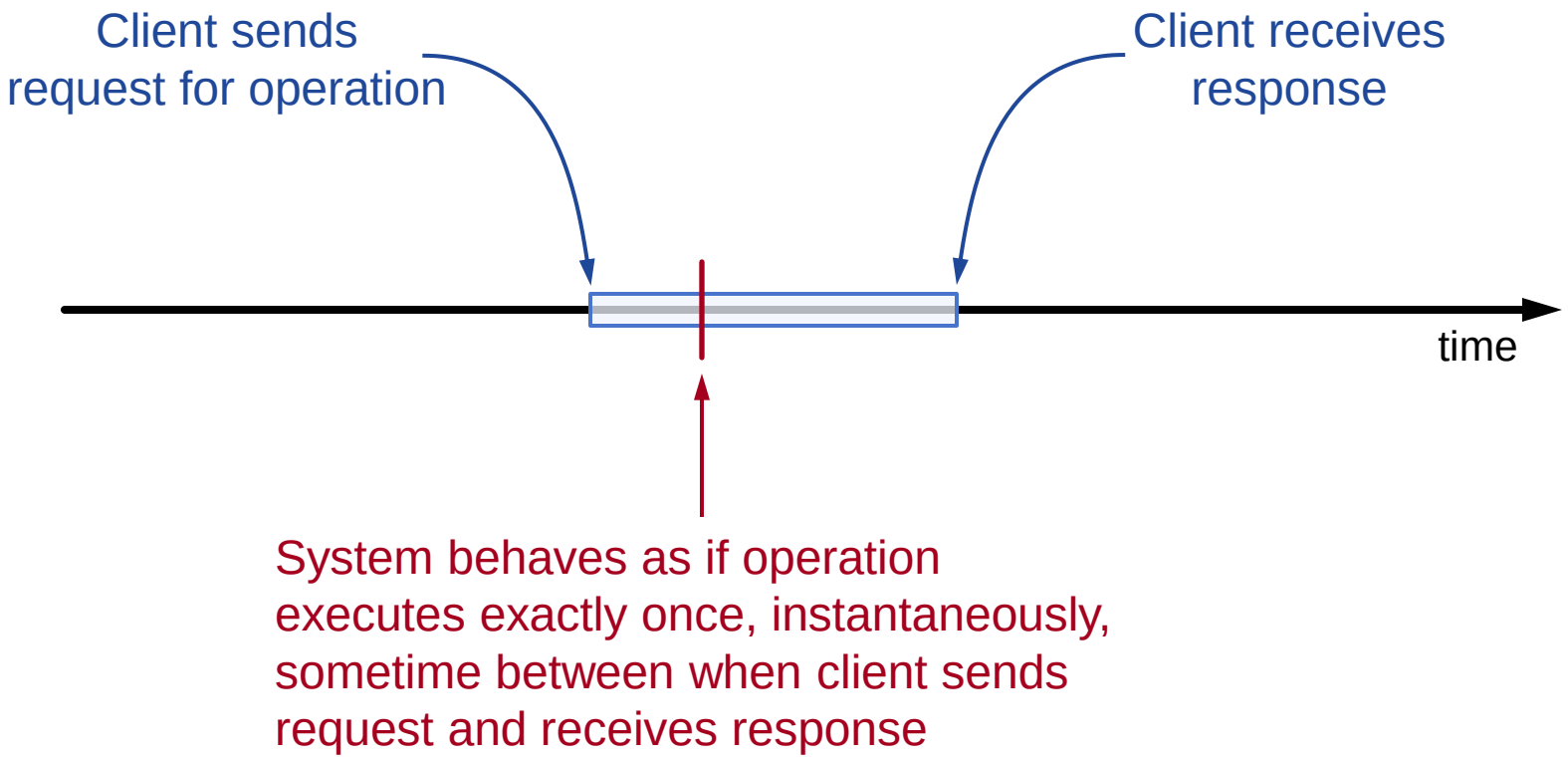


SLIK, cont'd

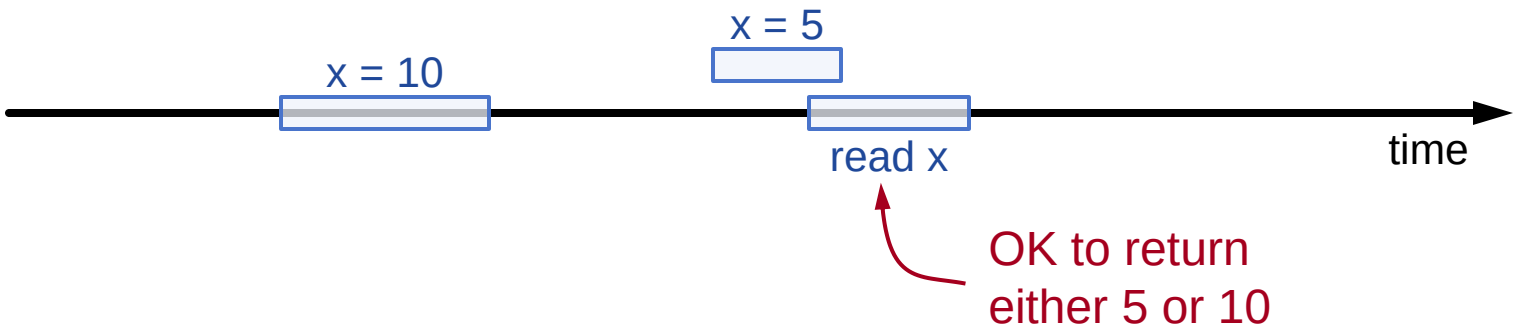
- **Status:**
  - Preliminary limitations of most mechanism
  - Initial performance measurements
- **Students involved:**
  - Ankita Kejriwal (talk later today)
  - Arjun Gopalan
  - Ashish Gupta
  - Zhihao Jia

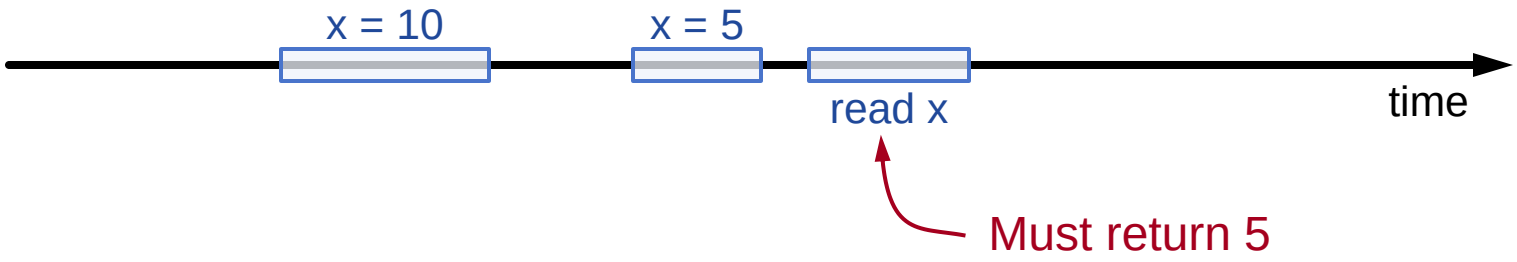
Linearizability

- **Holy Grail of consistency for large-scale apps**

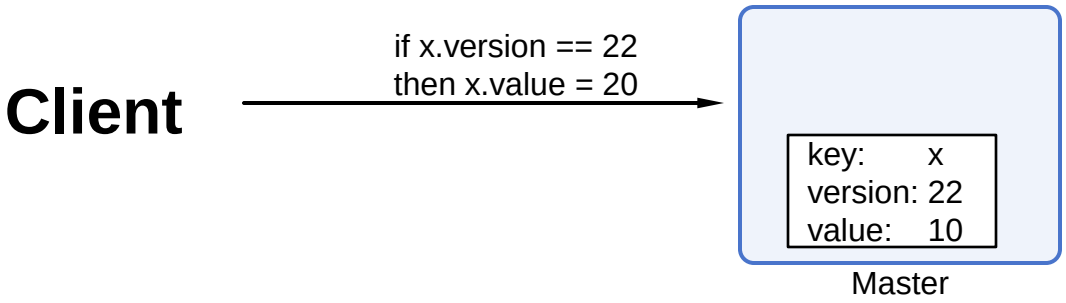


Linearizability

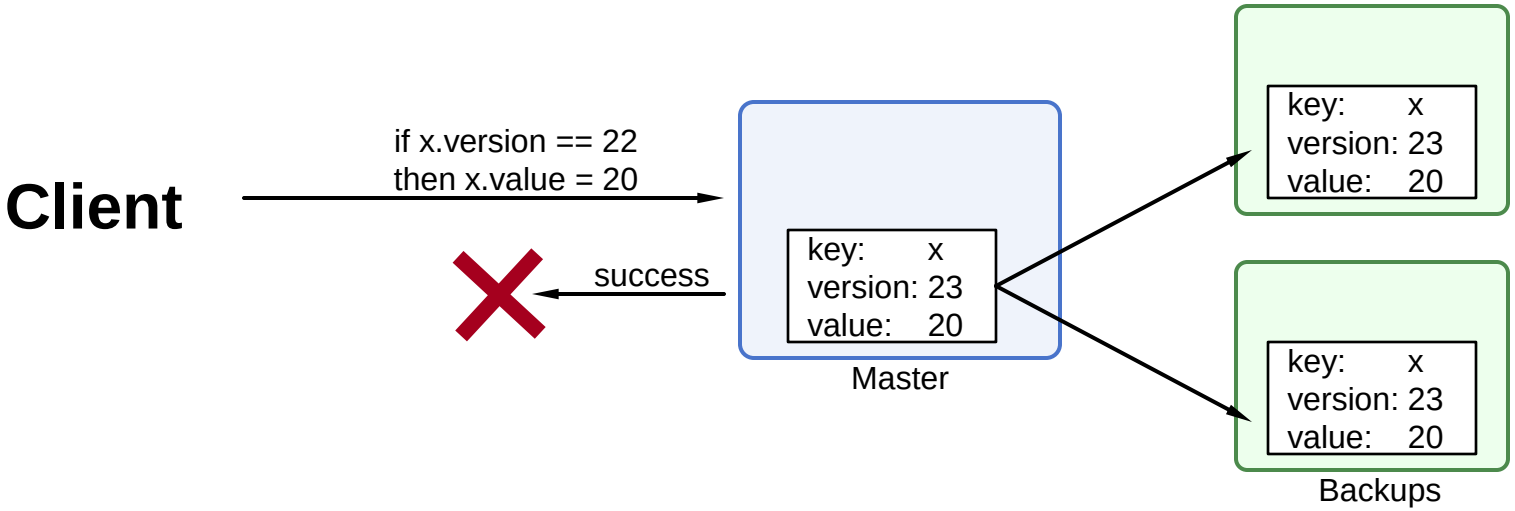




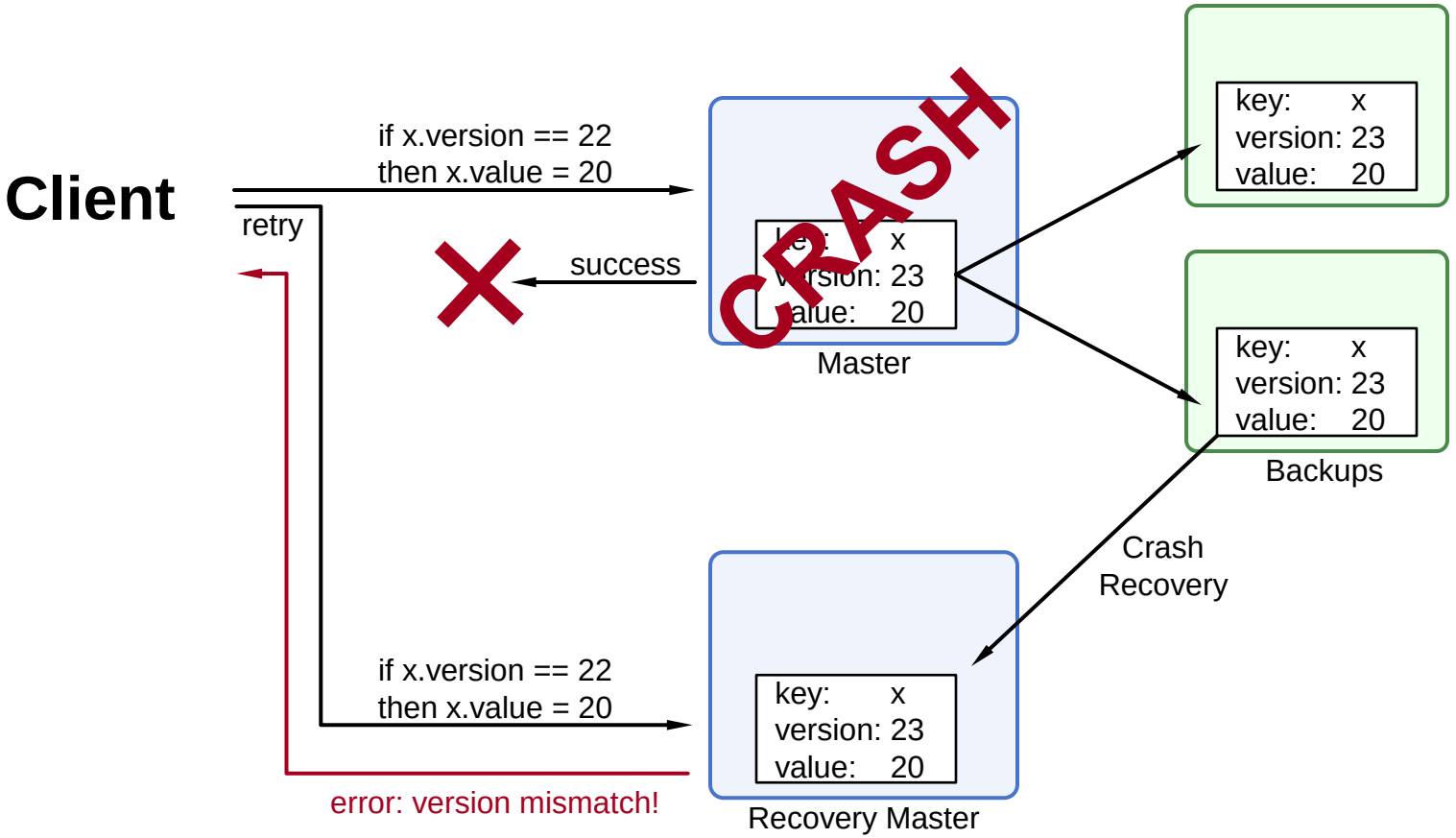
Linearizability Failure



Linearizability Failure



Linearizability Failure



Must remember old results, avoid re-executing requests

Linearizability Project

- **Create general-purpose infrastructure (use log to track RPC results)**
- **Use it to implement linearizable RPCs:**
  - Conditional write
  - Multi-object transactions
- **Students involved:**
  - Seo Jin Park (talk later today)
  - Collin Lee
  - Ankita Kejriwal

Latency Analysis

- **After 4 years, still little understanding of RAMCloud latency!**
  - What accounts for current latency?
  - How much can it be improved?
  - What are the fundamental limits?
  - What is the right system structure to minimize latency?
- **Henry Qin starting to answer these questions**

RAMCloud Transports Today

- **Infiniband reliable queue pairs:**
  - Highest performance; our main workhorse
  - Reliable, in-order delivery implemented in hardware
  - Doesn't support Ethernet-style networks
  - Driver is old, thrown-together, warty ("temporary solution")
- **Kernel TCP:**
  - Easy to use
  - Too slow for real applications (50-150µs round-trips)
- **FastTransport:**
  - Custom transport for RAMCloud
  - Works with any underlying datagram protocol (e.g. kernel UDP)
  - Provides reliable, in-order, flow-controlled delivery
  - Not as fast as infrc, too complex, never fully debugged

Transport Redesign

- **Goal: clean-slate replacement for FastTransport:**
  - Better latency and scalability
  - Replace infrc as workhorse transport
  - Separable from RAMCloud
  - “RPC for future datacenters”
- **First steps (Behnam Montazeri):**
  - Build SolarFlare datagram driver for FastTransport
    - Kernel bypass for 10 GigE
  - Understand FastTransport weaknesses

Conclusion

- **Several new projects in early stages**
- **Talks this retreat: mostly work in progress**
- **Should have many interesting results over the next year**