RAMCloud Overview and Update

SEDCL Retreat
June, 2013

John Ousterhout

Stanford University

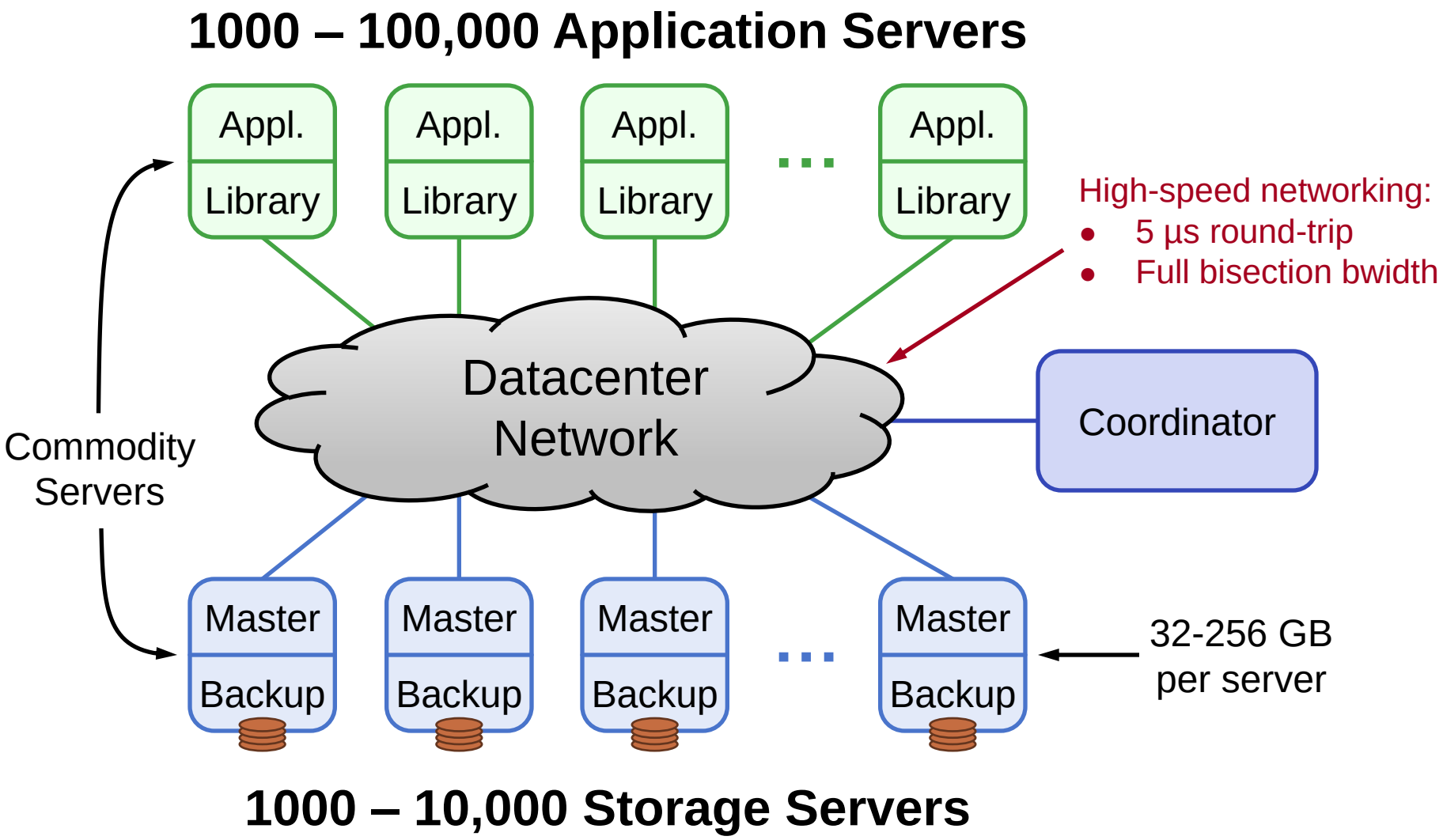
What is RAMCloud?

General-purpose storage system for large-scale applications:

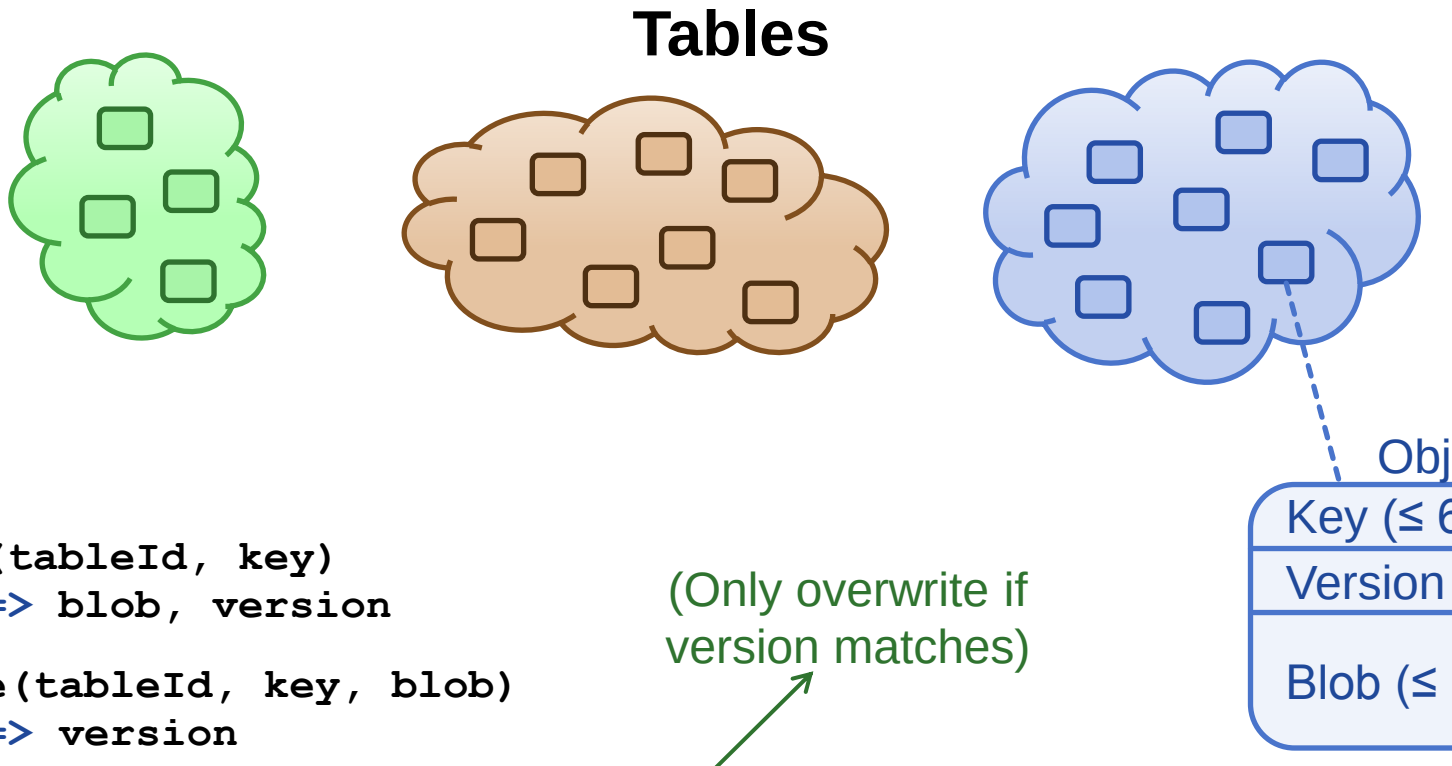
- **All data is stored in DRAM at all times**
- **Large scale: 1000+ servers, 100+ TB**
- **Low latency: 5-10 µs remote access time**
- **As durable and available as disk**
- **Simple key-value data model (for now)**

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

RAMCloud Architecture



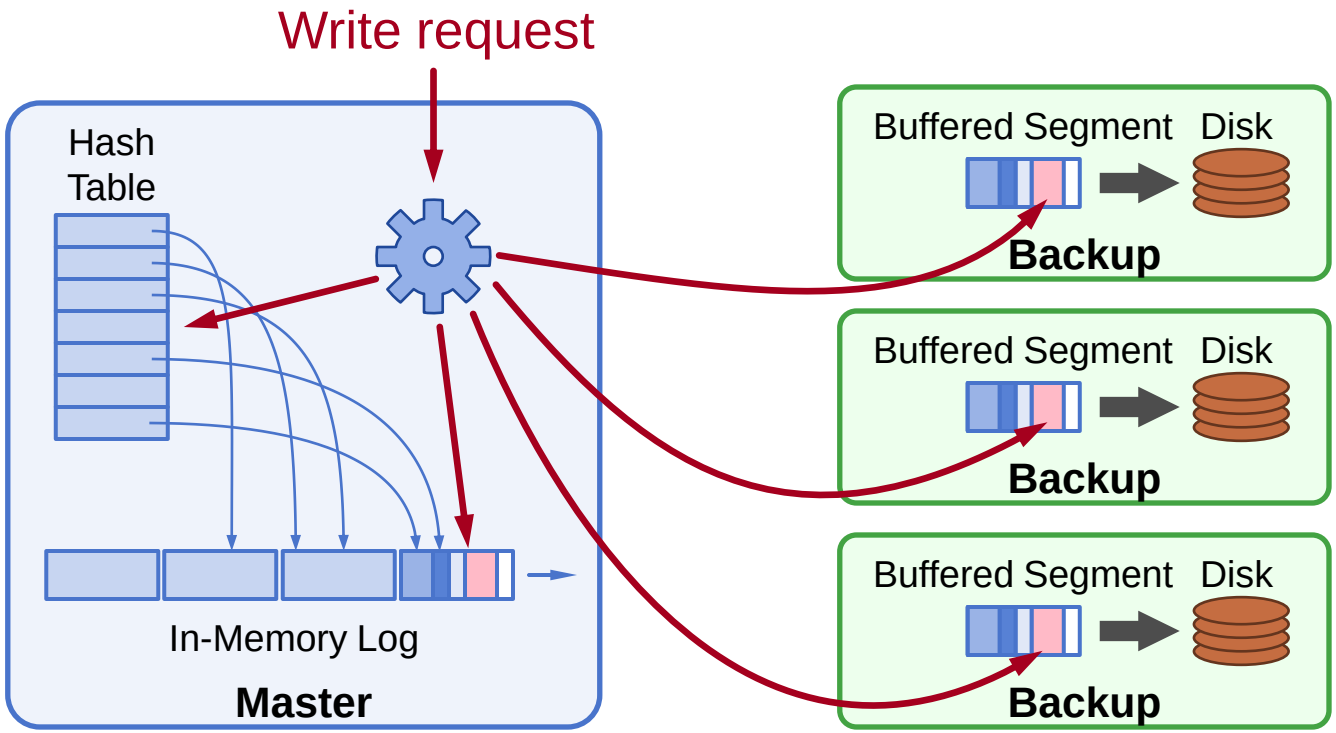
Data Model: Key-Value Store




```
read(tableId, key)
    => blob, version

write(tableId, key, blob)
    => version

cwrite(tableId, key, blob, version)
    => version

delete(tableId, key)
```

(Only overwrite if version matches)

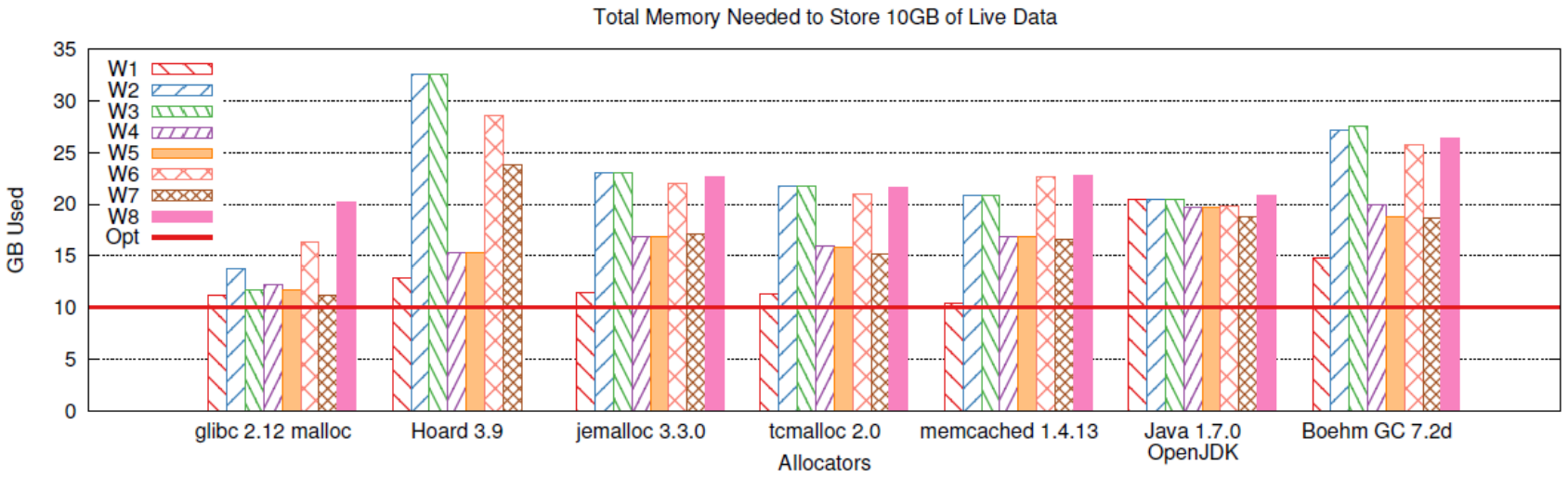
Buffered Logging



- **Log-structured: backup disk and master's memory**
- **No disk I/O during write requests**
- **Log cleaning ~ generational garbage collection**

Log-Structured Memory

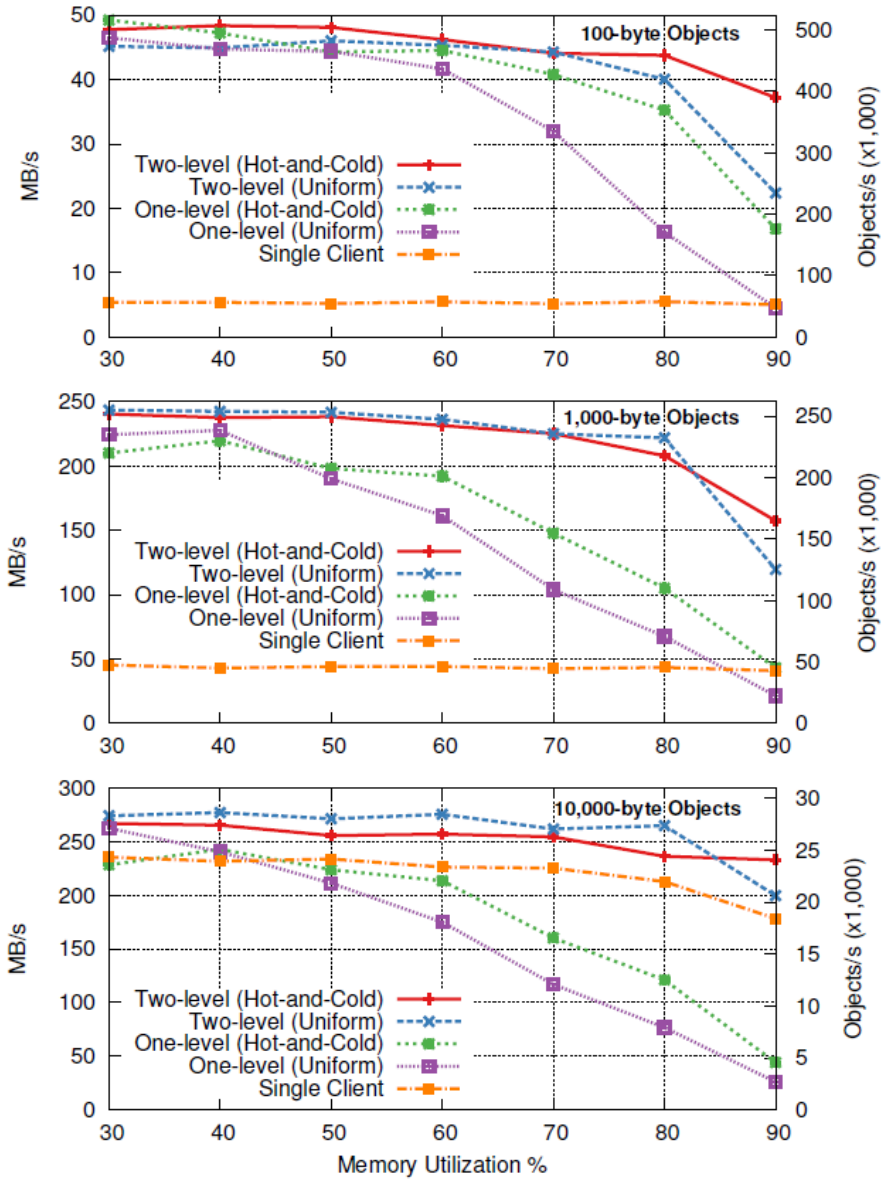
- **Don't use malloc for memory management**
  - Wastes 50% of memory



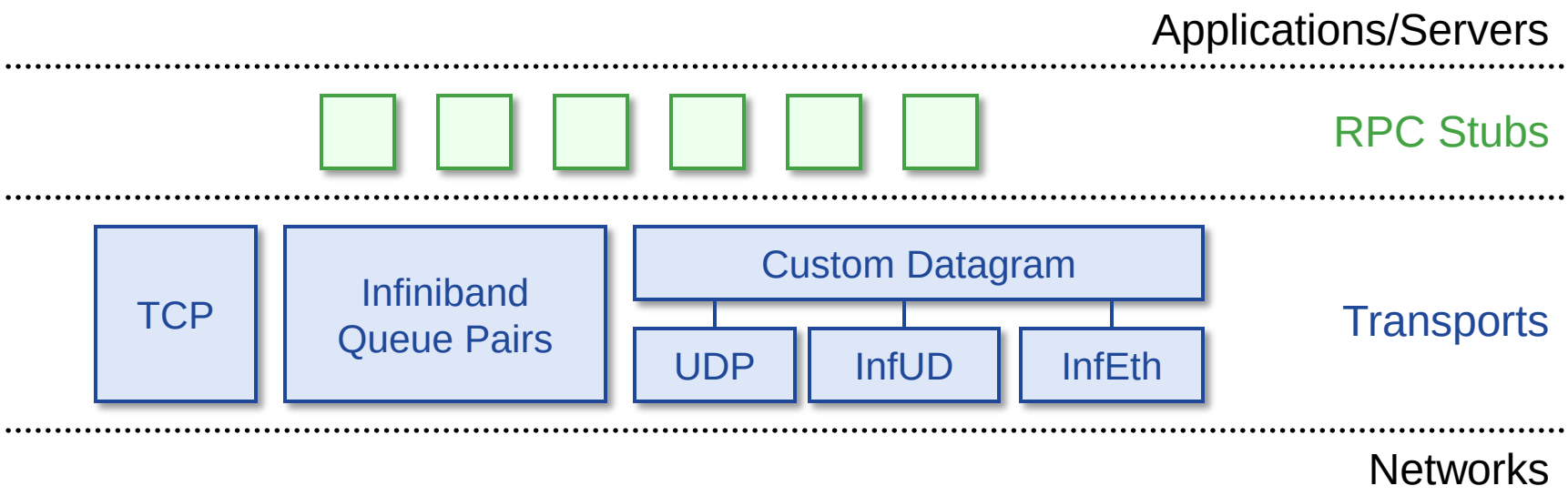
- **Instead, structured memory as a log**
  - Allocate by appending
  - Log cleaning to reclaim free space
  - Control over pointers allows incremental cleaning

Log-Structured Memory, cont'd

- **Creates tradeoff:**
  - Performance vs. utilization
- **Two-level cleaner:**
  - Disk and memory
  - Memory only (compaction)
- **Concurrent cleaning**
- **Multiple cleaner threads**
- **80-90% memory utilization feasible**
- **Paper under submission**



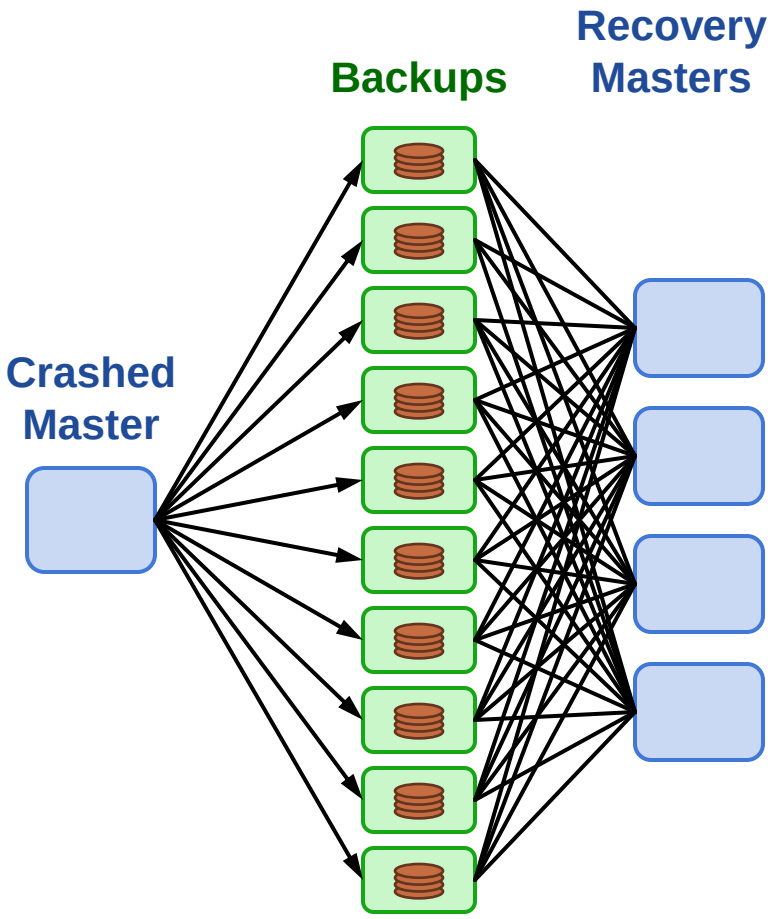
RAMCloud RPC



- **Transport layer enables experimentation with different networking protocols/technologies**
- **Basic Infiniband performance (one switch):**
  - 100-byte reads: 4.9 µs
  - 100-byte writes (3x replication): 15.3 µs
  - Read throughput (100 bytes, 1 server): 700 Kops/sec

RAMCloud Crash Recovery

- **Each master scatters segment replicas across entire cluster**
- **On crash:**
  - Coordinator partitions dead master's tablets.
  - Partitions assigned to different recovery masters
  - Log data shuffled from backups to recovery masters
  - Recovery masters replay log entries
- **Total recovery time: 1-2s**



Status at June 2012 Retreat

- **Major system components (barely) working:**
  - RPC transports (timeout mechanism new)
  - Basic key-value store (variable-length keys new)
  - Log-structured memory management (log cleaner new)
  - Crash recovery (backup recovery new)
  - Coordinator overhaul just starting
- **Overall project goal: push towards a 1.0 release**
  - “Least usable system” for real applications

Current Status

- **Core system becoming stable**
  - Not quite at 1.0, but close!
- **First PhDs coming soon:**
  - Ryan Stutsman: crash recovery
  - Steve Rumble: log-structured memory
- **Many research opportunities still left**
- **About to start new projects:**
  - Higher-level data model
  - New RPC mechanism
  - Cluster management

Progress Towards RAMCloud 1.0

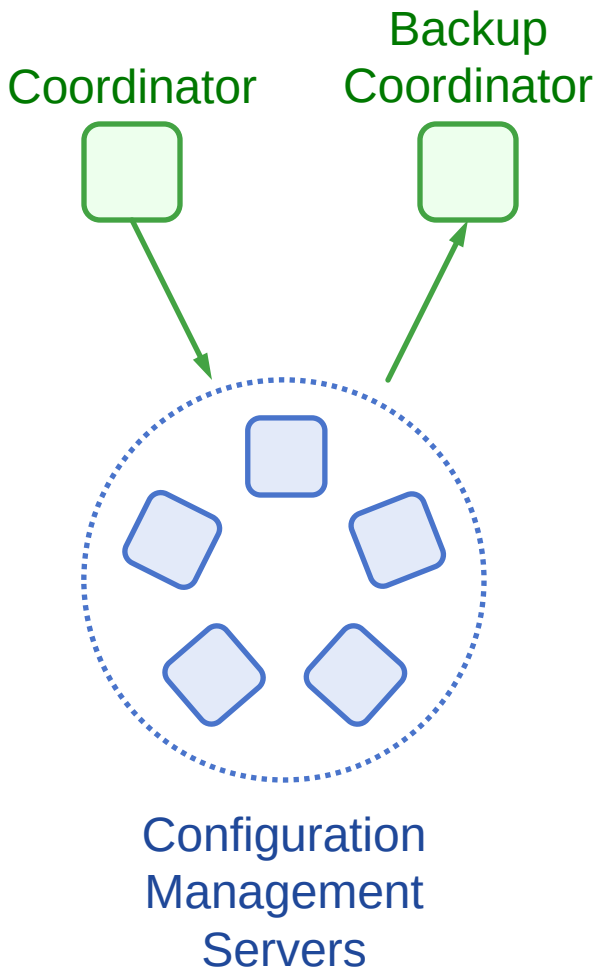
- **Fault-tolerant coordinator (Ankita Kejriwal):**
  - Server and tablet configuration info now durable
- **LogCabin configuration manager (Diego Ongaro)**
- **Additional recovery mechanisms (Ryan Stutsman):**
  - Simultaneous server failures
  - Cold start
  - Overhaul of backup storage management
  - RPC retry (John Ousterhout)
- **Overhaul of cluster membership management (Stephen Yang)**
  - More robust
  - Better performance

Progress Towards 1.0, cont'd

- **New in-memory log architecture (Steve Rumble)**
- **Automated crash tester (Arjun Gopalan):**
  - Synthetic workload with consistency checks
  - Force servers to crash randomly
  - Multiple simultaneous failures, coordinator failures
- **Goal: run crash tester for a few weeks with no loss of data**
- **Current status:**
  - Can survive some coordinator and master failures
  - Others causing crashes
  - Working through bugs

Configuration Management

- **External system for durable storage of top-level configuration information:**
  - Cluster membership
  - Tablet configuration
- **Typically consensus-based:**
  - Chubby (Google)
  - ZooKeeper (Yahoo/Apache)
- **Unhappy with ZooKeeper, so decided to build our own (Diego Ongaro):**
  - Development started before last year's retreat
  - Initial plan: use Paxos protocol

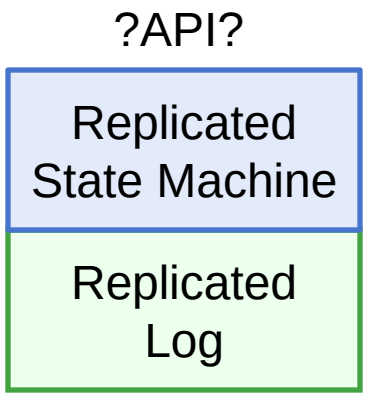


New Consensus Algorithm: Raft

- **Paxos is "industry standard", but:**
  - Very hard to understand
  - Not a good starting point for real implementations
- **Our new algorithm: Raft**
  - Primary design goal: understandability
  - Also must be practical and complete
  - Result: new approach to consensus
    - Design for replicated log from start
    - Strong leader
- **User study shows that Raft is significantly easier to understand than Paxos (stay tuned ...)**
- **Paper under submission**

API for Consensus

- **Key-value store (Chubby, ZooKeeper)?**
  - Applications really want a log?
  - Why build a log on a key-value-store on a log?
- **Collection of logs (LogCabin)?**
  - First approach for RAMCloud, based on Raft
  - Used in current coordinator implementation
  - However, log turned out not to be convenient after all
- **TreeHouse: key-value store on Raft?**
- **Export API for replicated state machine?**



New Work: Data Model

- **Goal: higher-level data model than just key-value store:**
  - Secondary indexes?
  - Transactions spanning multiple objects and servers?
  - Graph-processing primitives (sets)?
- **Can RAMCloud support these without sacrificing**
  - Latency
  - Scalability
- **Design work just getting underway (Ankita Kejriwal)**

New Work: Datacenter RPC

Complete redesign of RAMCloud RPC

- **General purpose (not just RAMCloud)**
- **Latency:**
  - Even lower latency?
  - Explore alternative threading strategies
- **Scale:**
  - Support 1M clients/server (minimal state/connection)
- **Network protocol/API:**
  - Optimize for kernel bypass
  - Minimize buffering
  - Congestion control: reservation based?

Current Status

- **Core system becoming stable**
  - Not quite at 1.0, but close!
- **First PhDs coming soon:**
  - Ryan Stutsman: crash recovery
  - Steve Rumble: log-structured memory
- **Many research opportunities still left**
- **About to start new projects:**
  - Higher-level data model
  - New RPC mechanism
  - Cluster management