RAMCloud Overview and Update

SEDCL Forum
January, 2015

John Ousterhout

Stanford University

Outline

- **Quick overview of RAMCloud**
- **Progress since June retreat**
- **Current projects:**
  - Secondary indexes
  - Multi-object transactions
  - New transport architecture

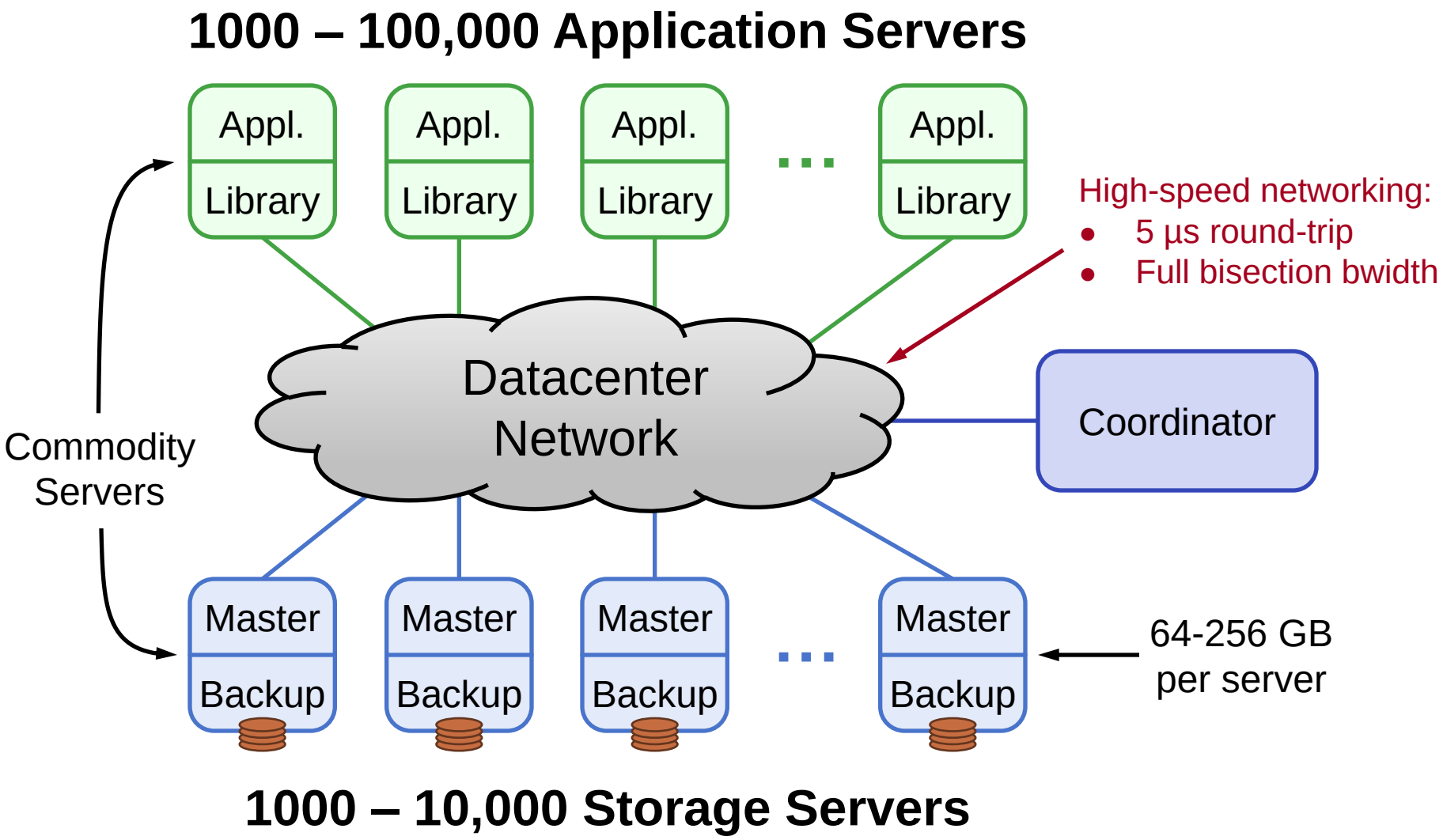
What is RAMCloud?

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

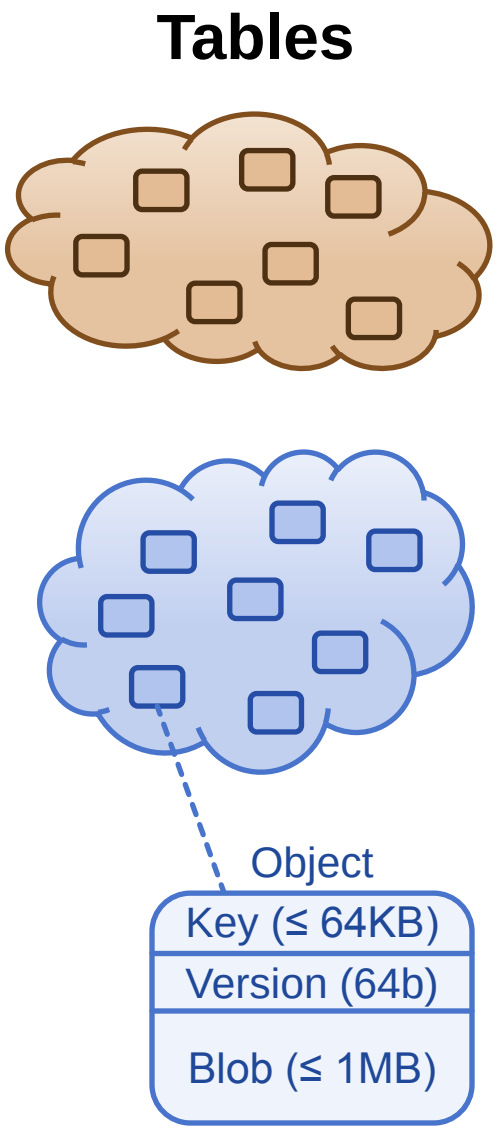


Data Model: Key-Value Store

- **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



Updates

- **Raft project (new consensus algorithm) finished:**
  - Best Paper Award at USENIX ATC
  - Diego Ongaro graduated in September
  - Usage continues to grow
- **Incremental performance improvements:**
  - Small random reads: 5.0µs → 4.7µs
  - Small durable writes: 15µs → 13.5µs
- **Ongoing experiments with potential applications (more details in upcoming talk)**

Secondary Indexes

- **Participants: Ankita Kejriwal, Stephen Yang**
- **Many interesting issues:**
  - Representation (objects, indexes)
  - Scalability
  - Consistency
  - Crash recovery
- **Status in June:**
  - Skeletal implementation running
  - Many restrictions, missing features (e.g. fixed-size keys)
- **Progress:**
  - Implemented indexlet split/migrate for reconfiguration
  - Reworked B-tree implementation to eliminate restrictions
  - Working towards SOSP paper submission

Multi-Object Transactions

- **Participants: Collin Lee, Seojin Park, Ankita Kejriwal**
- **Based on general-purpose linearizability support**
  - Discussed at June retreat
  - Completed this fall
- **Commit protocol designed:**
  - Client-driven (similar to Sinfonia)
  - Capitalizes on linearizability infrastructure
  - Detailed talk coming next
- **Implementation underway**
- **Targeting SOSP paper (March)**

Clean-Slate Transport Redesign

- **Participants: Behnam Montazeri, Henry Qin, Mohammad Alizadeh**
- **New network protocol for datacenter RPC:**
  - Replace TCP/IP: better latency, scalability
  - Receiver-driven congestion/flow control
  - Design & simulation just getting started
- **New threading architecture:**
  - Goals:
    - Minimize thread crossings
    - Reduce latency (polling-based)
    - Improve throughput
  - Design work just starting

Conclusion

- **Lots of work in progress**
- **Should have more results by June retreat**