Deploying Enterprise Applications on RAMCloud

Christian Tinnefeld, Hasso Plattner Institute,
University of Potsdam, Germany

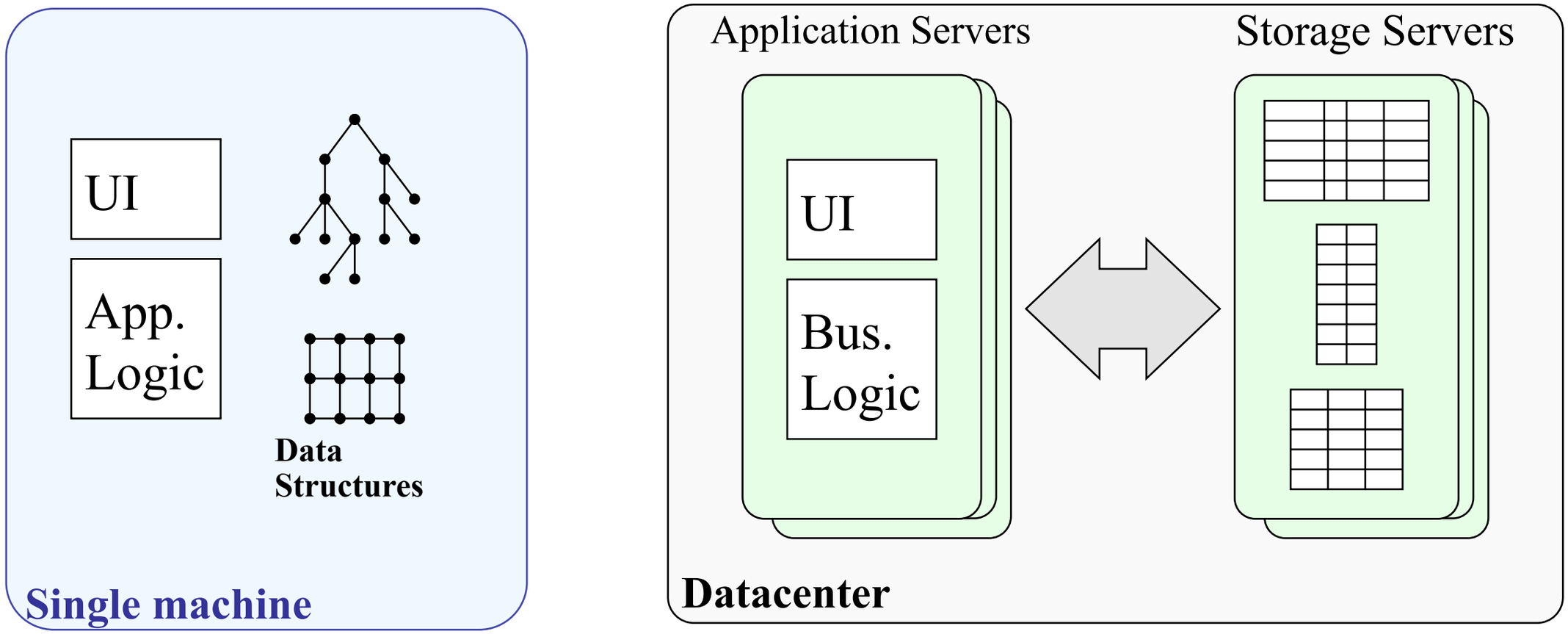
Agenda

- What is RAMCloud?
- Why is it interesting for Enterprise In-Memory Computing?
- Why is it interesting for Enterprise Applications?

What is RAMCloud?

- Research project at Stanford University
- Storage for datacenters
- 1.000-10.000 commodity servers
- ~64 GB DRAM/server
- All data always in RAM
- Throughput: One million ops/sec/server
- Latency: 5-10 µs RTT using InfiniBand
- Key-value data model

Why Latency matters

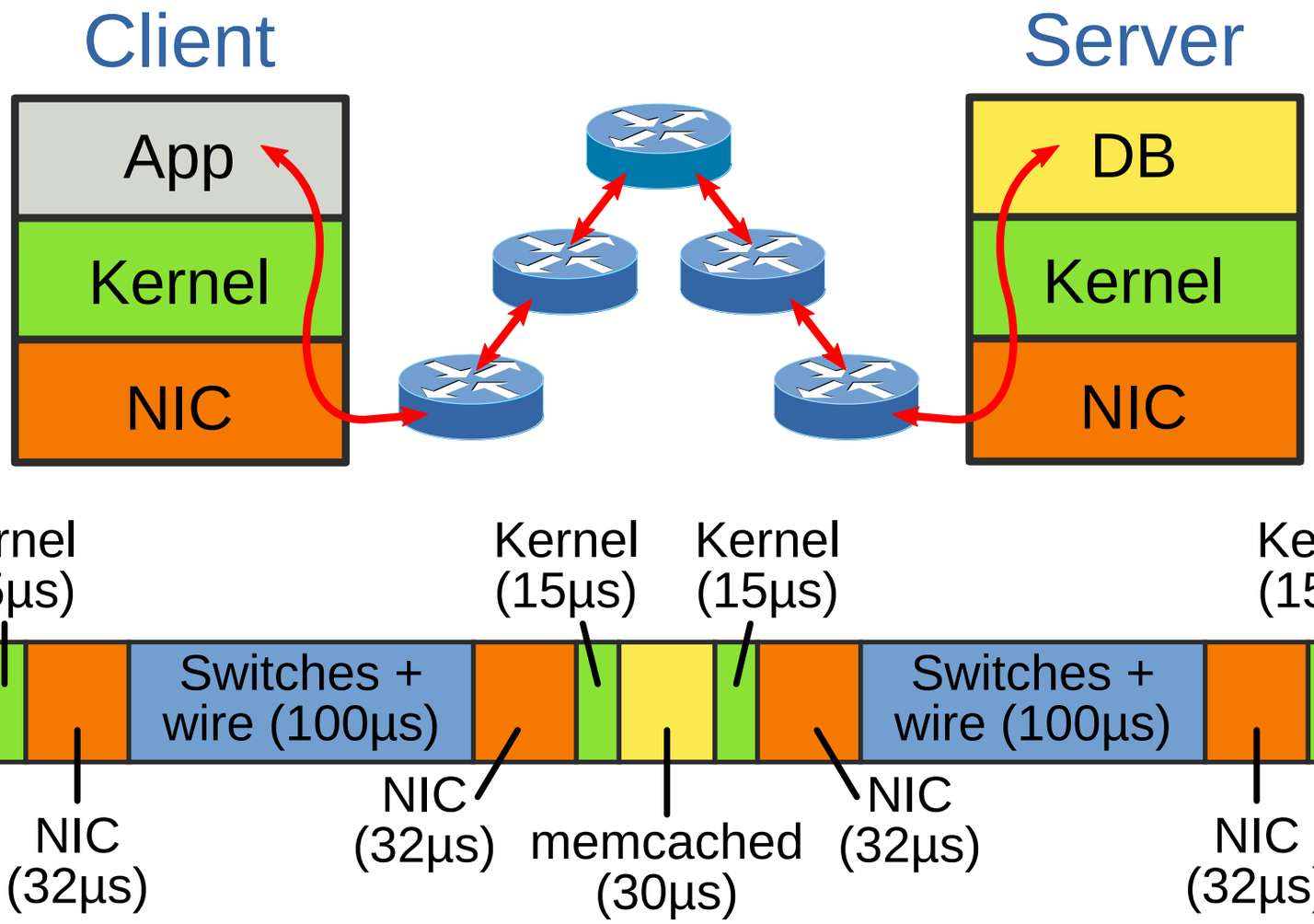


<1µs latency

1-10 ms latency

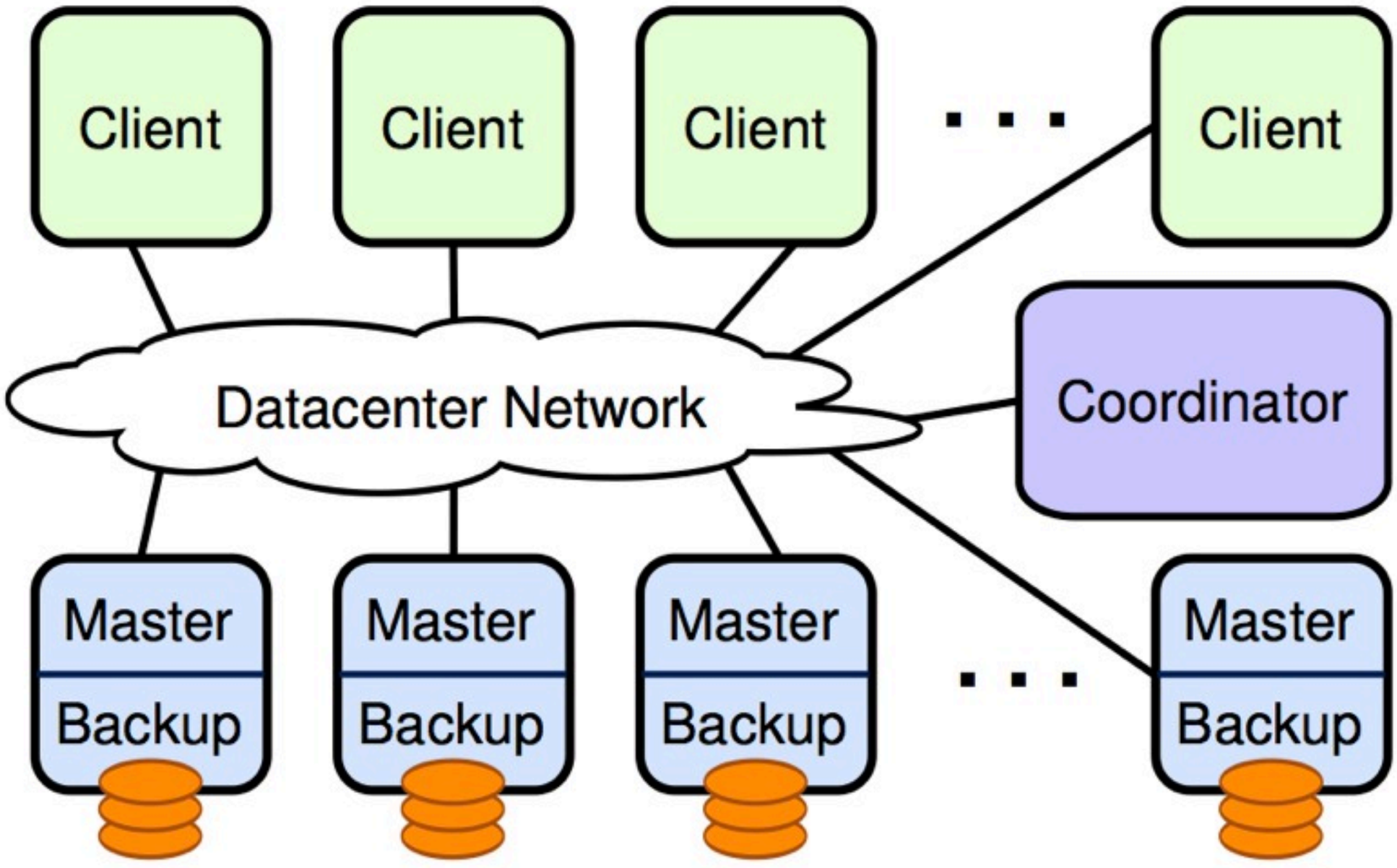
- RAMCloud goal: large scale and low latency

RPC Latency

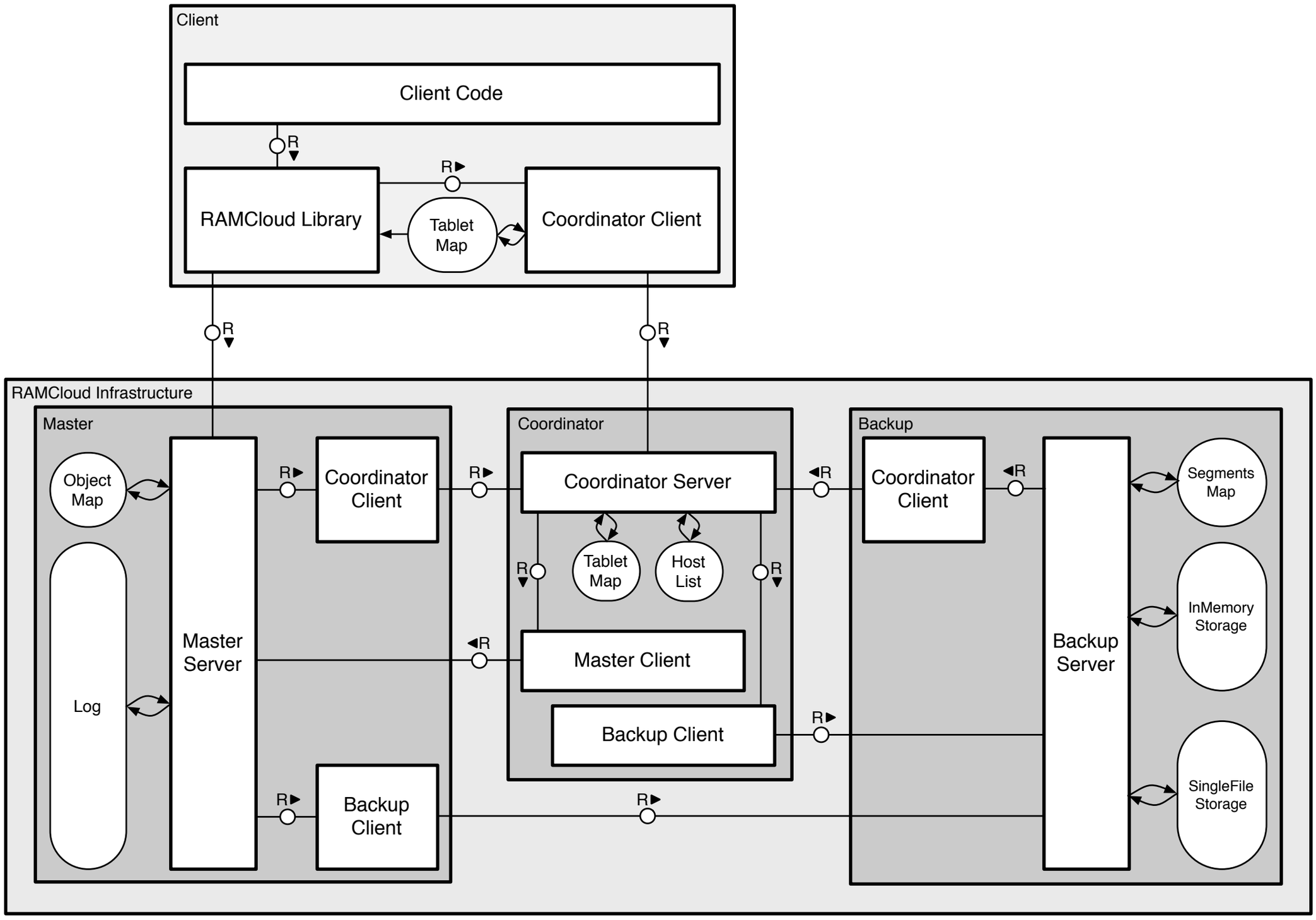


- Above: Typical large datacenter today
- RAMCloud: ~7µs RTT (InfiniBand)

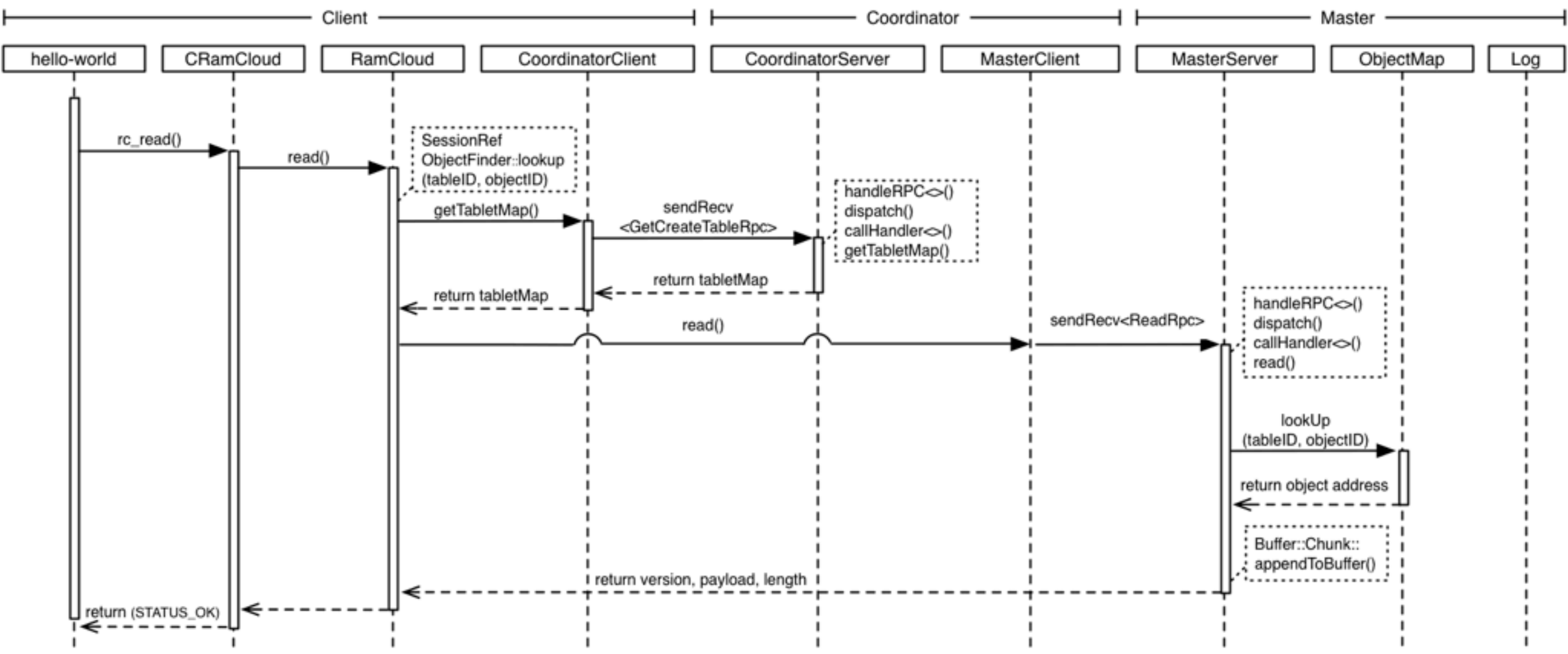
Architecture (1/2)



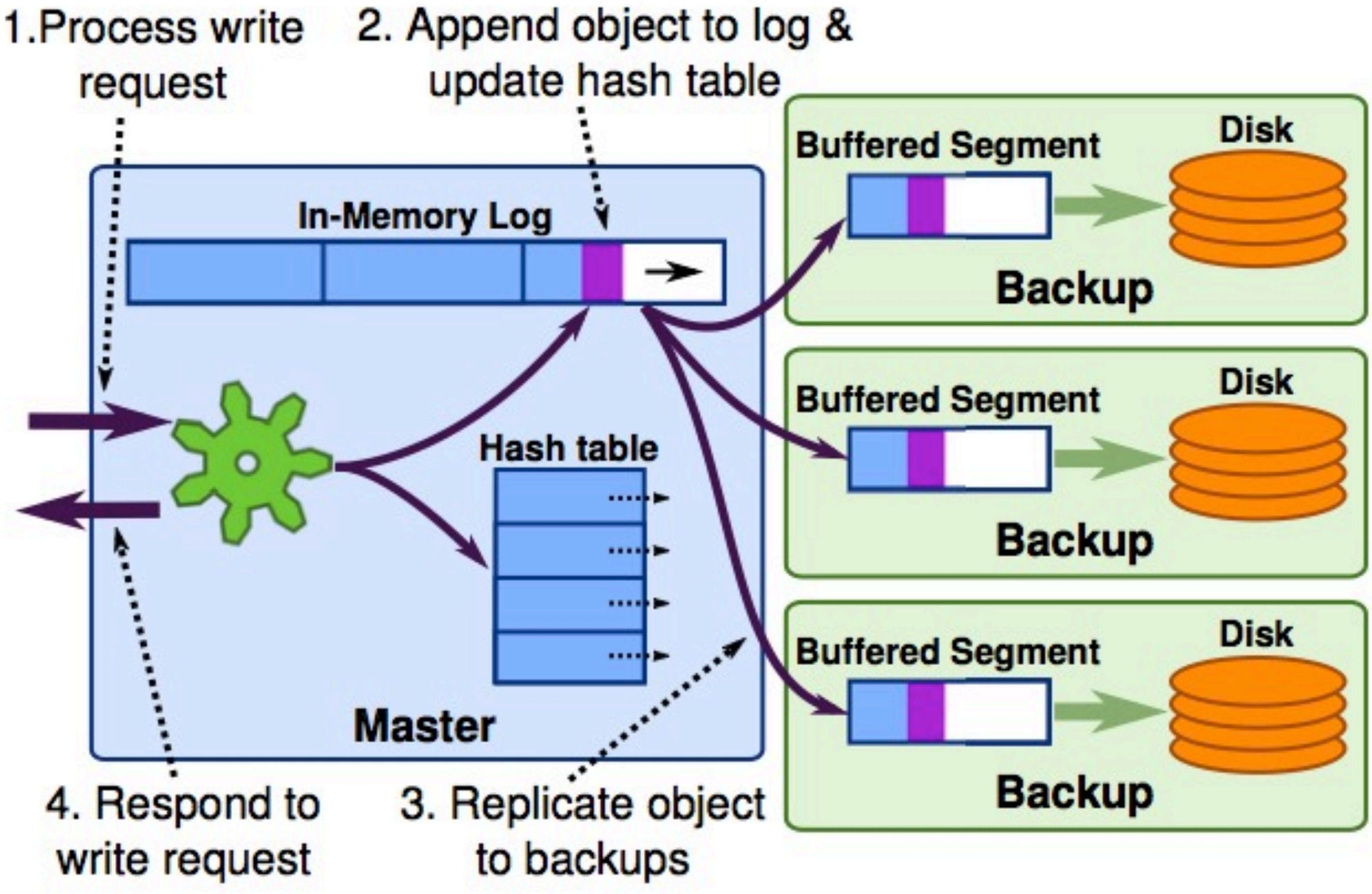
Architecture (2/2)



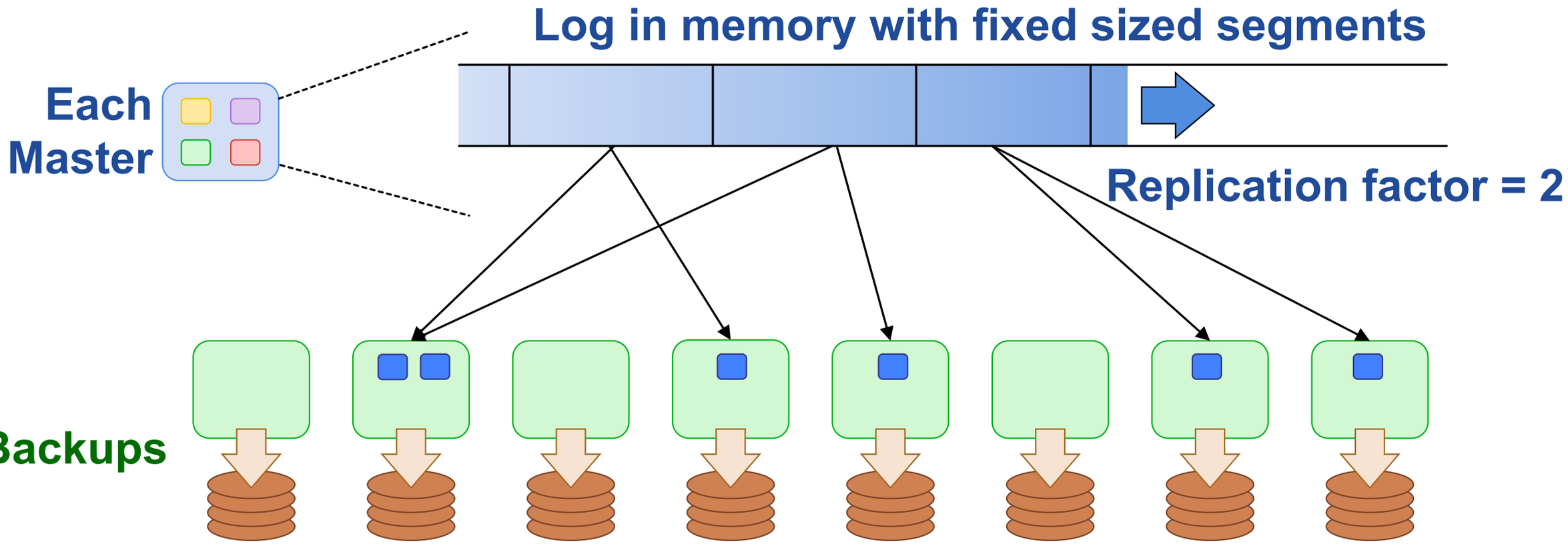
Read request and response flow



Durability

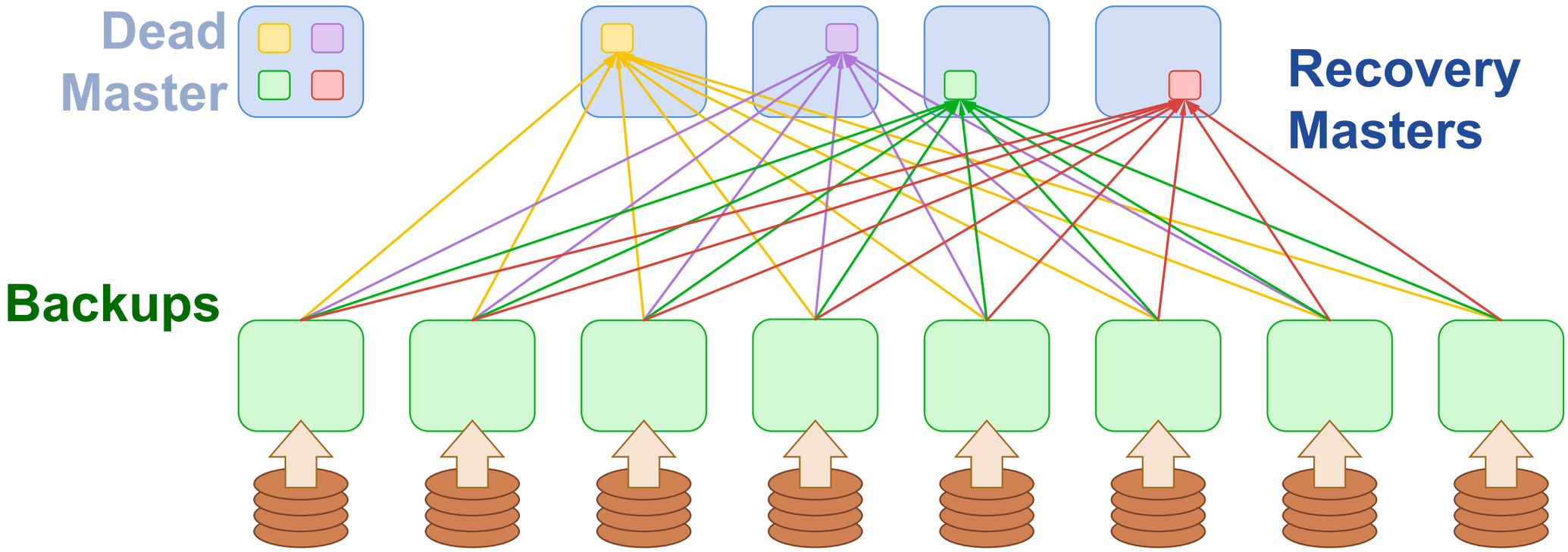

1.Process write request
2. Append object to log & update hash table
In-Memory Log
Hash table
Master
Buffered Segment
Disk
Backup
Buffered Segment
Disk
Backup
Buffered Segment
Disk
Backup
4. Respond to write request
3. Replicate object to backups

Log-structured Memory



- One log per master divided into segments
- Results in optimal Disk/SSD writes

Availability



- Fast recovery on failure for high availability
- Recovery time of 1 second for node failures

(Non-Enterprise) Killer Use Cases

- Facebook
  - FB's biggest problem: abuse of the platform
  - Every user interaction is registered as an event and evaluated against 30+ criteria
  - 1500 servers running Memcached to track the results
  - Goal: detect patterns among the events
- Morgan Stanley
  - All trading applications need low-latency storage for temporary results and coordination
  - Data must be instantly available after a hardware crash

Why is RAMCloud interesting for Enterprise In-Memory Computing?

- Synchronous data replication into the DRAM of other nodes
- Recovery of crashed MasterServers ~1s
- Memory sizing per node
- Bandwidth between nodes vs local memory
  - Xeon E5450 3Ghz Random Read/Write: 1.4/0.8 GB/s
  - InfiniBand Mellanox ConnectX-2: 3.4 GB/s

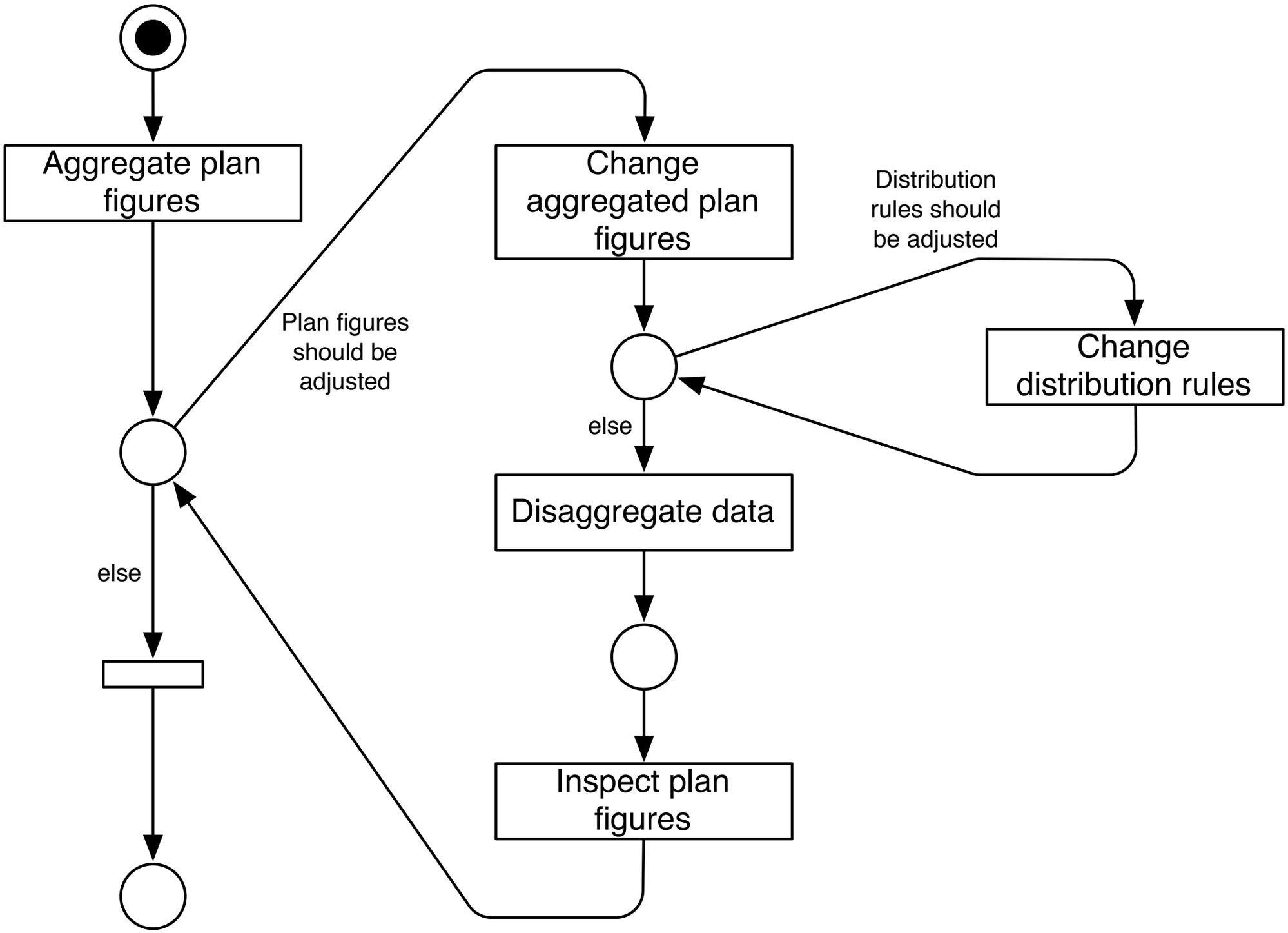
Why is RAMCloud interesting for Enterprise Applications?

- Low latency reduces the time each transactions stays in the system
  - Reduction of aborted transactions (optimistic cc)
  - Reducing lock wait times (pessimistic cc)
- High bandwidth for read AND write access
  - Planning and What-If Analytics
  - Large amounts of data are manipulated and must be immediately available for analysis

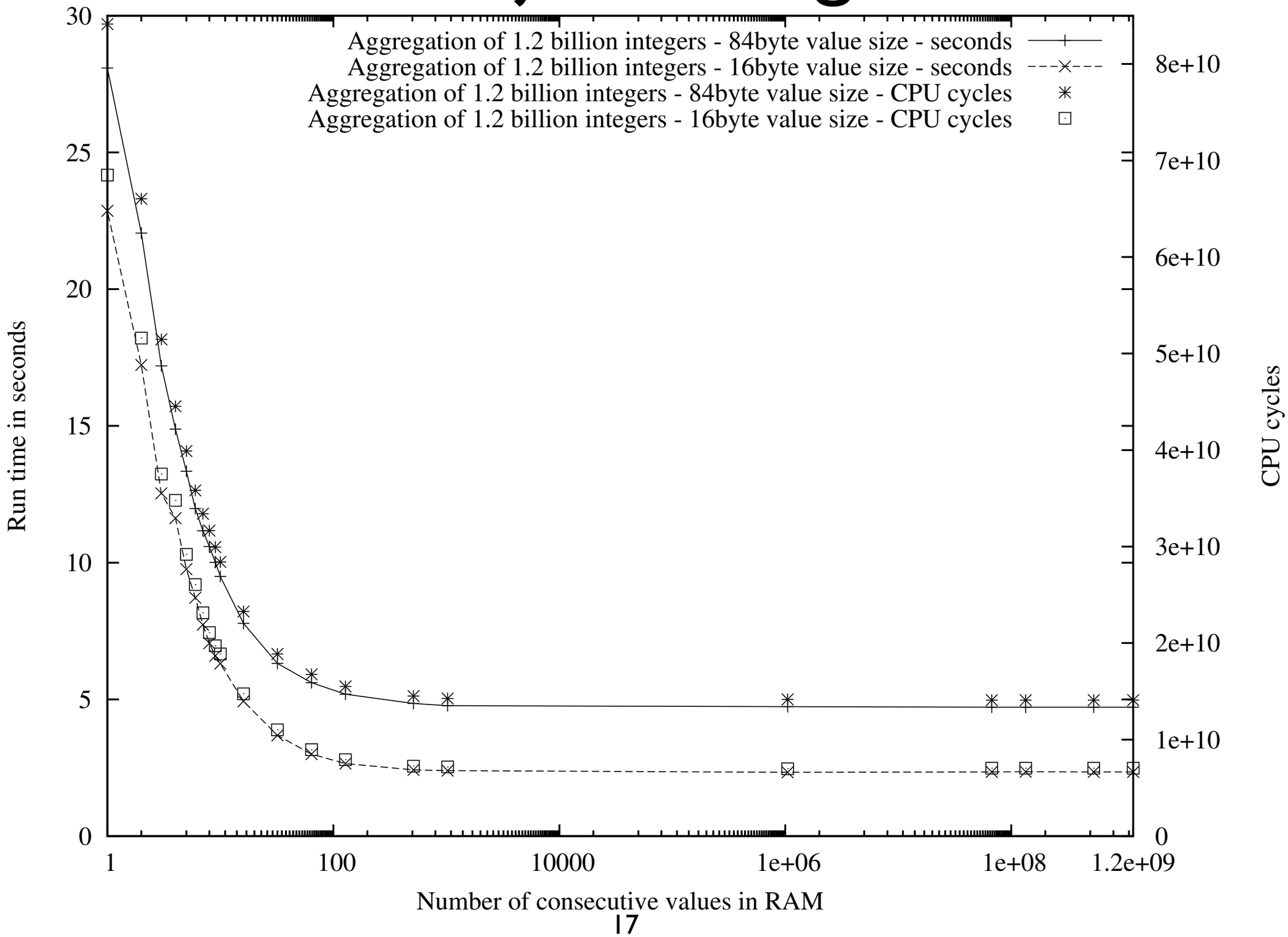
Deploying a planning application on top of RAMCloud

- What are the most fundamental data operations?
- What is a good object sizing strategy?
- Where to execute the data operations?
- How to interpret the data inside objects?
- How to perform (relational) queries?

Data operations in planning



Object sizing



Aggregation operator throughput with increasing number of objects

# number of objects	client-side aggregation	server-side aggregation		
		hash table lookup	hash table iteration	log traversal
1.000.000	4790 ms	127 ms	168 ms	21 ms
10.000.000	48127 ms	1378 ms	781 ms	142 ms
100.000.000	485091 ms	19854 ms	6245 ms	1422 ms

Aggregation operator throughput with decreasing selectivity

selectivity	client-side aggregation	server-side aggregation		
		hash table lookup	hash table iteration	log traversal
100%	206 objects/ms	5036 objects/ms	16012 objects/ms	70323 objects/ms
10%	203 objects/ms	4030 objects/ms	1322 objects/ms	6825 objects/ms
0.5%	203 objects/ms	3650 objects/ms	63 objects/ms	345 objects/ms

Disaggregation operator throughput with increasing number of objects

# number of objects	client-side aggregation	server-side aggregation		
		hash table lookup	hash table iteration	log traversal
1.000.000	12360 ms	515 ms	152 ms	33 ms
10.000.000	128432 ms	5411 ms	1677 ms	378 ms
100.000.000	1918644 ms	80278 ms	23982 ms	5589 ms

Design decisions for planning application on RAMCloud

- Implement (dis)aggregation operators
- Execution on server-side via hash-table look-up or log (segment) traversal
- Size objects in a way that they contain ~ 100 values or more
- Support native data types and arrays thereof and range queries
- Use relational DB for (complex) queries

Disadvantages of RAMCloud

- High cost per bit and high energy usage per bit
- Requires more floorspace in the datacenter
- No replication across data centers
- Needs critical amount of nodes (e.g. 20+)
- No native support for data queries

Conclusions (so far)

- RAMCloud anticipates the standard server hardware of the coming 2-3 years
- Demonstrates that high bandwidth and low latency are achievable at scale
- Provides a recovery mechanism for in-memory resident data that is as fast as hot-standby

Thank you for your attention!

- Questions?
- Feedback?
- We can also...
  - ...talk about other RAMCloud applications
  - ...walk through the code
  - ...discuss experiments you're interested in

BACKUP

Slides

Memory bandwidth on Xeon E5450 3Ghz Hapertown 1333 Mhz FSB