Fast Crash Recovery in RAMCloud

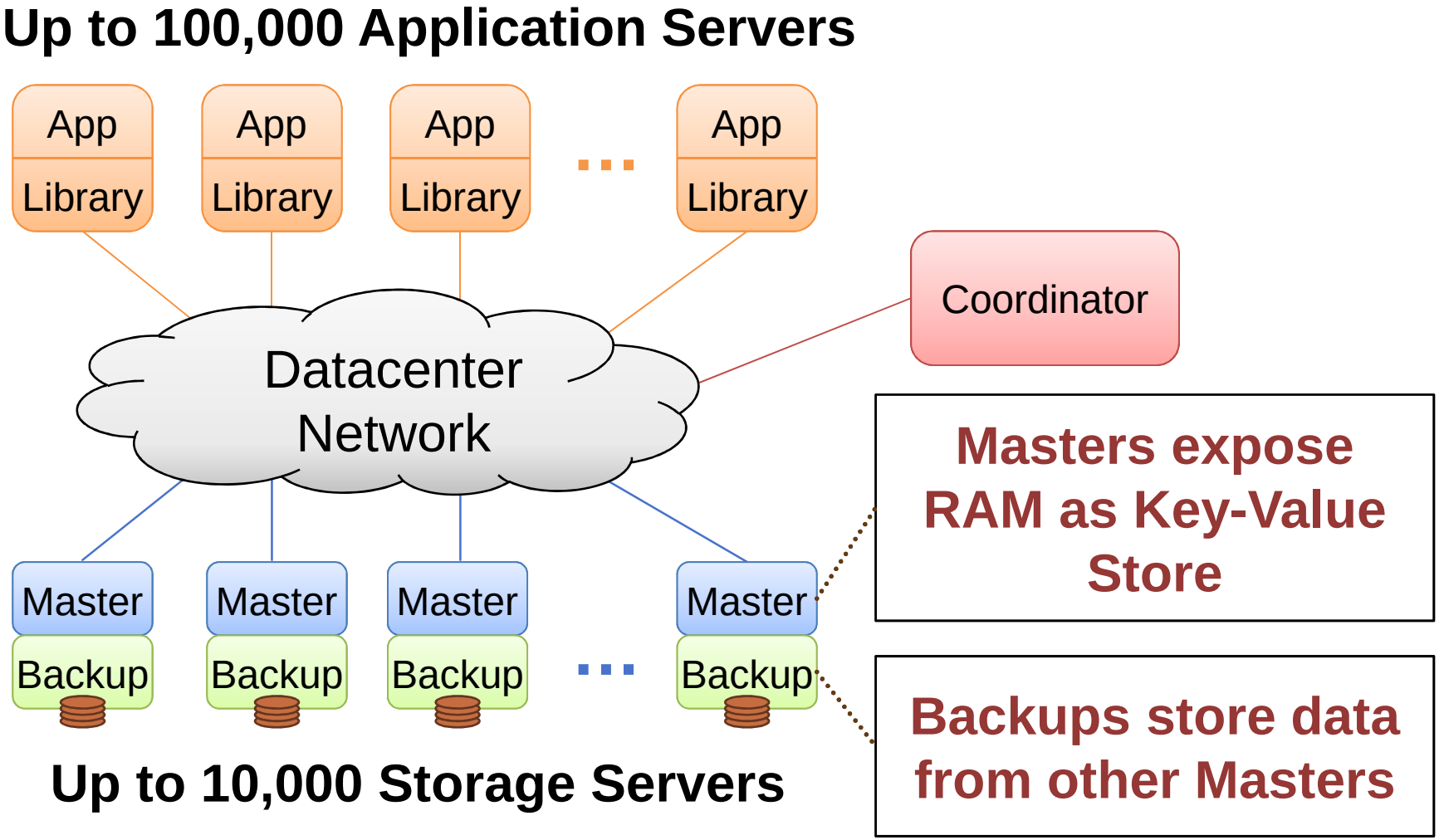
Diego Ongaro, Stephen M. Rumble,
Ryan Stutsman, John Ousterhout,
and Mendel Rosenblum

Stanford University

Overview

- **RAMCloud: General purpose storage in RAM**
  - Low latency: **5-10 µs** remote access
  - Large scale: 10,000 nodes, 100 TB to 1 PB
- **Key Problem: RAM's lack of durability**
- **Durability: Pervasive log structure, even in RAM**
  - Uses inexpensive disk-based replication
  - RAM performance by eliminating synchronous disk writes
- **Availability: Fast crash recovery in 1 to 2 s**
  - Recovers **35 GB to RAM in 1.6 s** using 60 nodes
  - Leverages the scale of the cluster
    - Balances work evenly across hosts
    - Avoids centralized control

RAMCloud Architecture



Durability & Availability

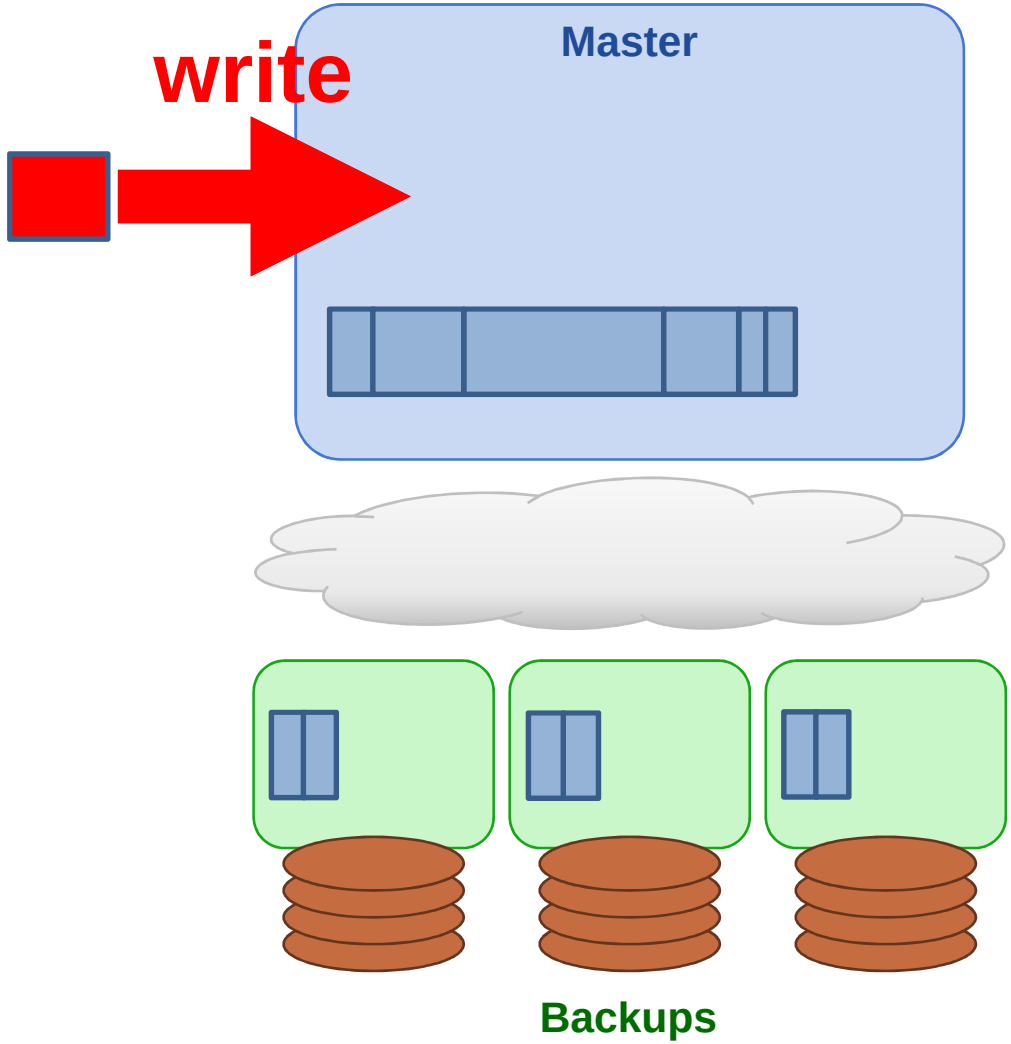
- **Requirements**
  - Retain high performance
  - Minimum cost, energy

- **Replicate in RAM of other masters?**
  - 3x system cost, energy
  - Still have to handle power failures

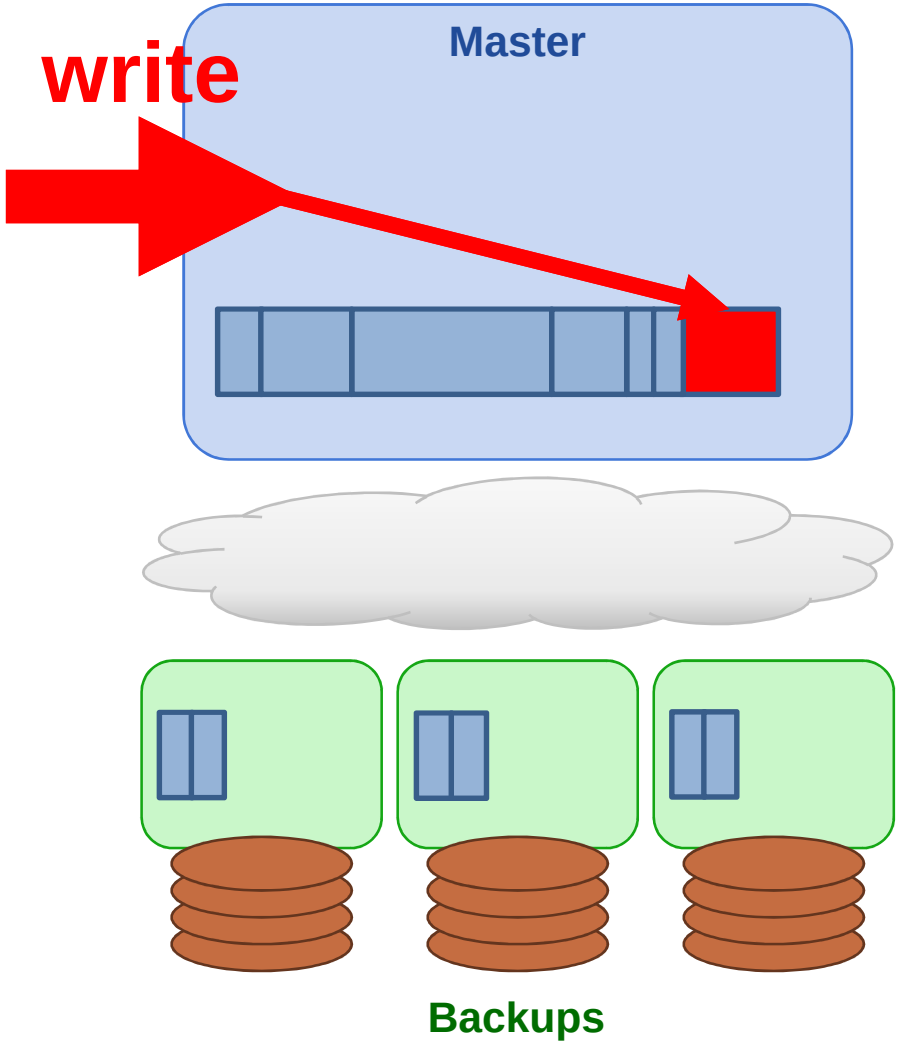
The RAMCloud Approach

- **1 copy in RAM**
- **Backup copies on disk/flash: durability ~ free!**
- **Problem: Synchronous disk writes too slow**
  - **Pervasive log structure, even in RAM**
- **Problem: Data is unavailable on crash**
  - **Fast Crash Recovery in 1 to 2 s**
    - Fast enough that applications won't notice

Durability with RAM performance

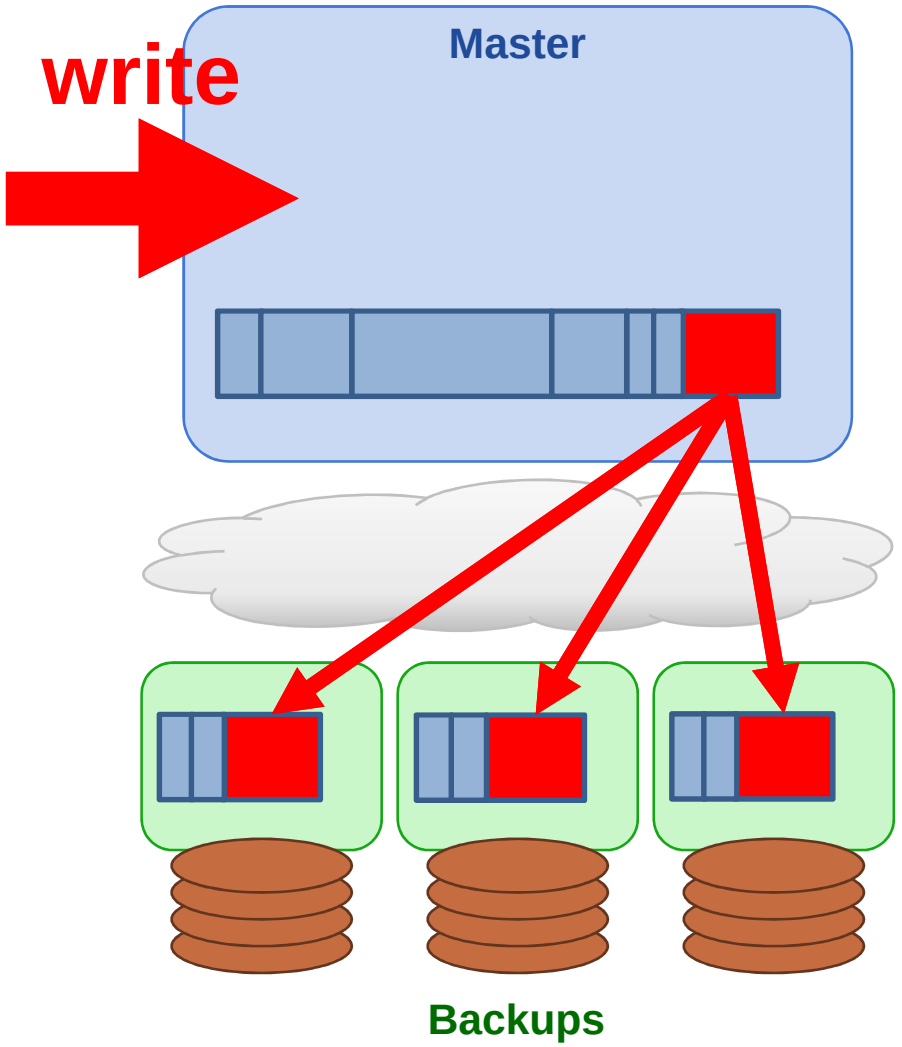


Durability with RAM performance



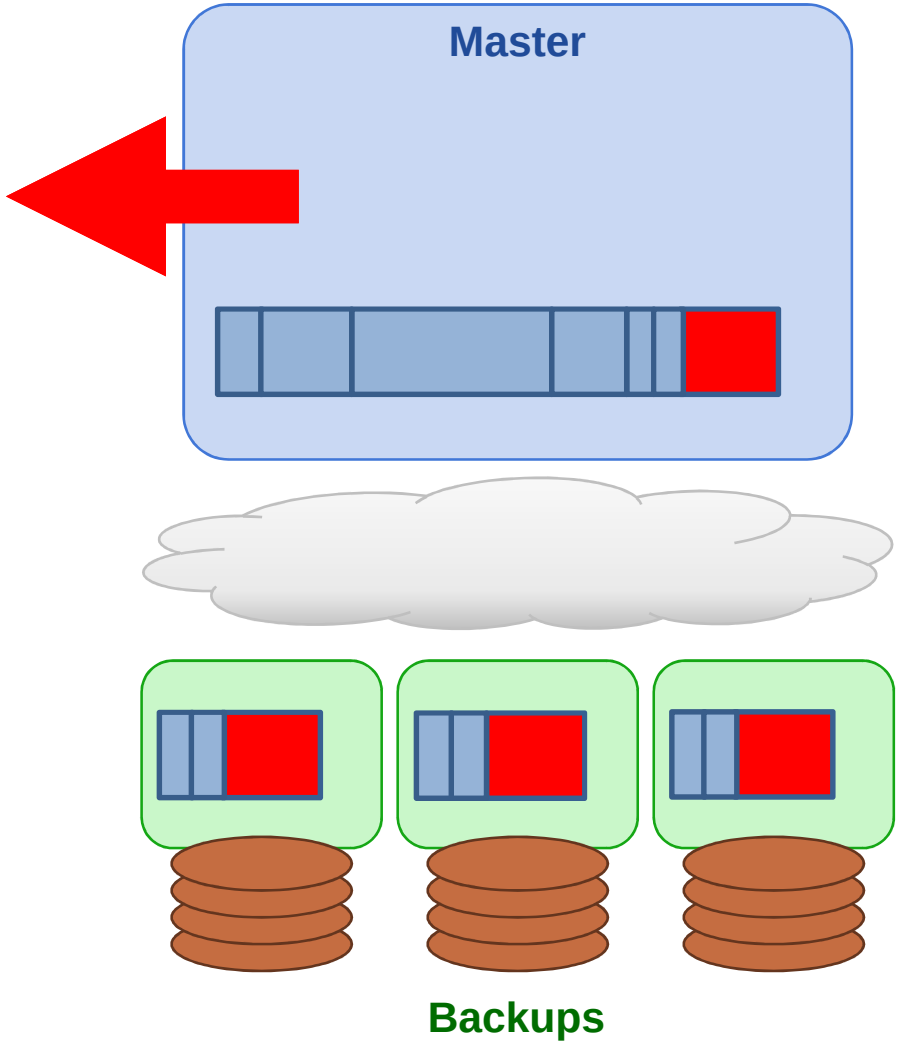
Durability with RAM performance

- **Backups buffer update**
  - No synchronous disk write



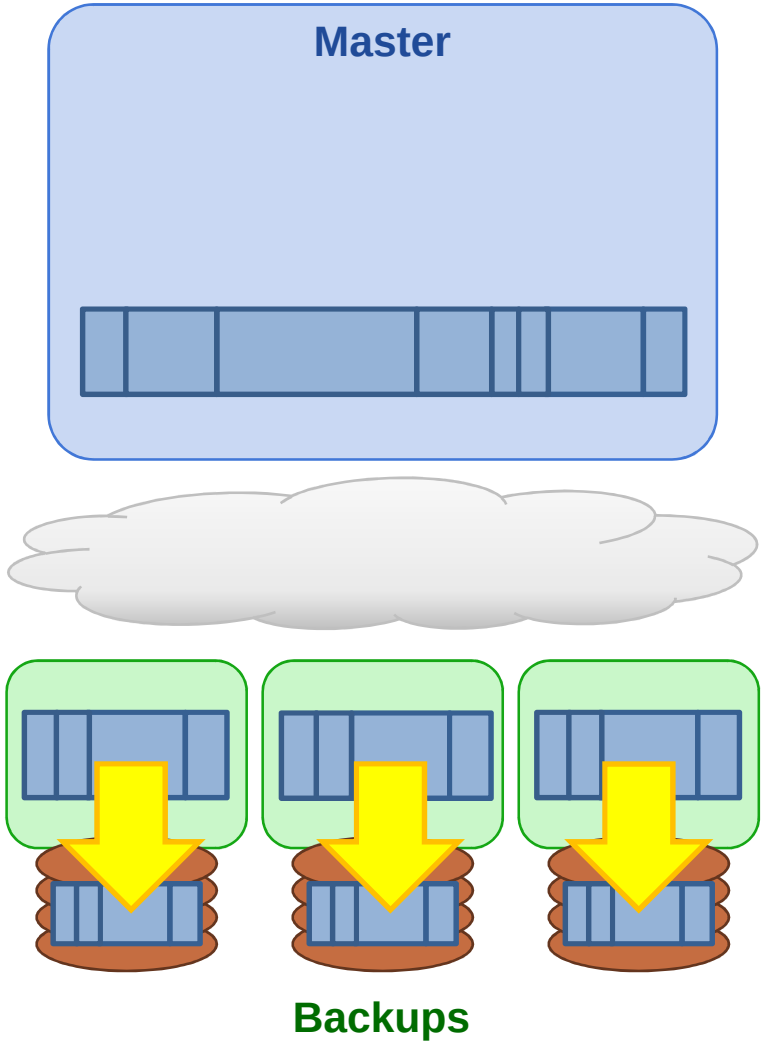
Durability with RAM performance

- **Backups buffer update**
  - No synchronous disk write



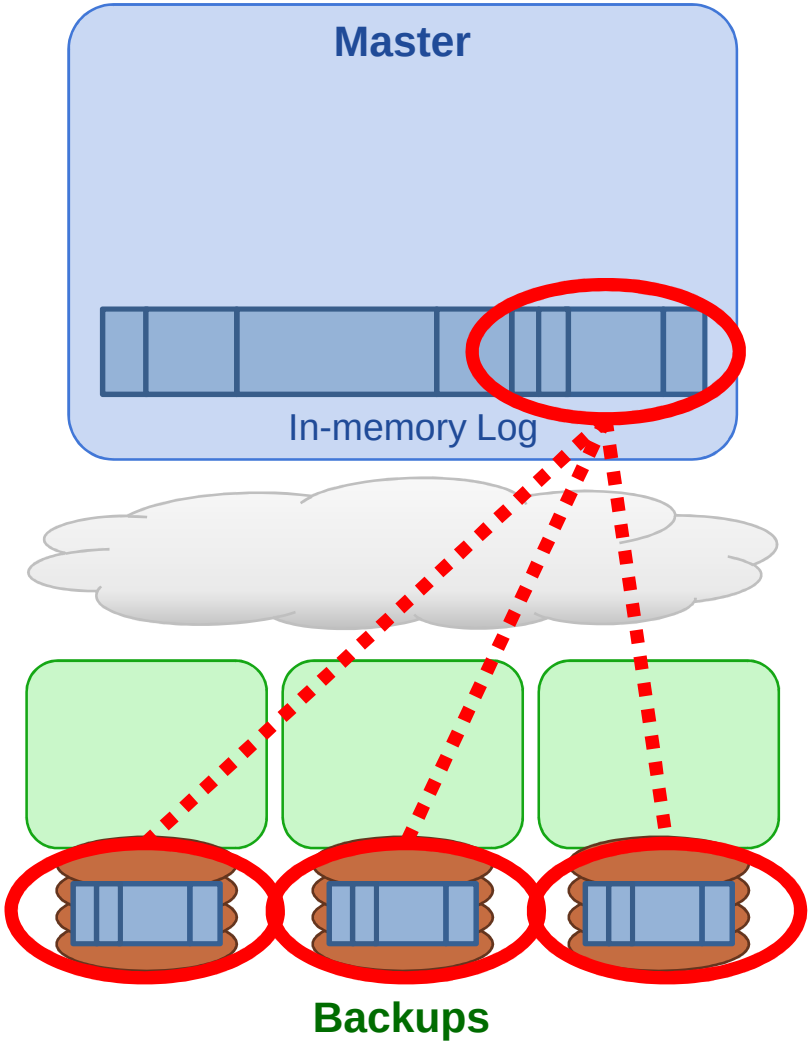
Durability with RAM performance

- **Backups buffer update**
  - No synchronous disk write
- **Bulk writes in background**
  - Must flush on power loss



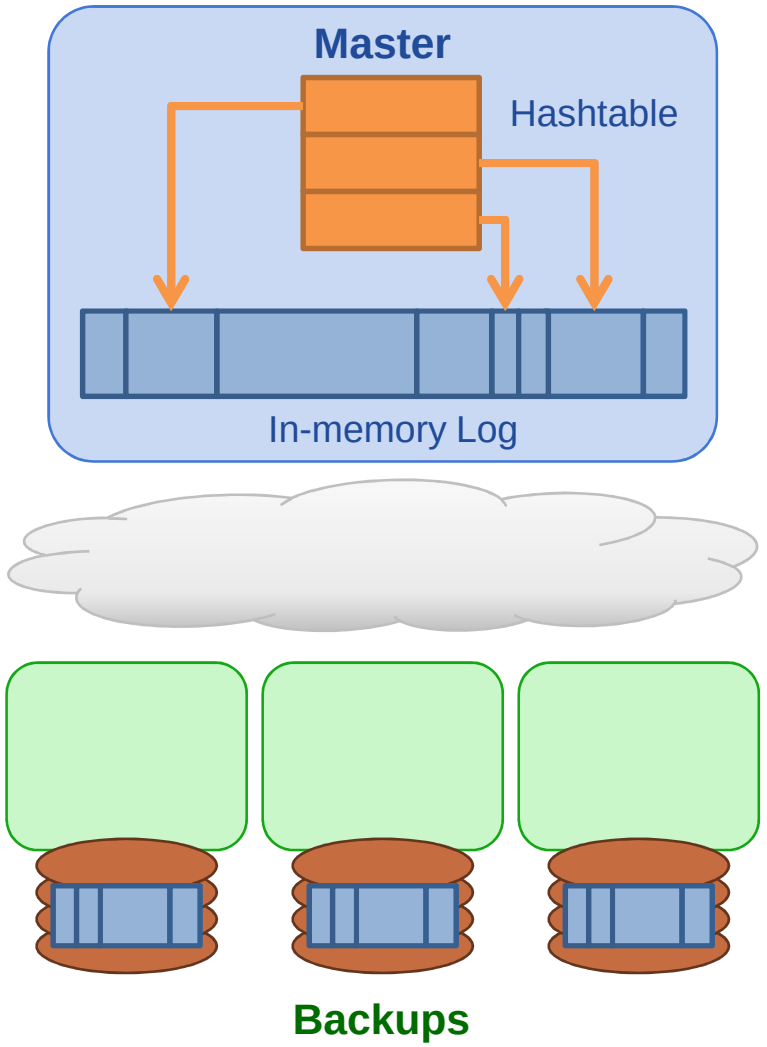
Durability with RAM performance

- **Backups buffer update**
  - No synchronous disk write
- **Bulk writes in background**
  - Must flush on power loss
- **Pervasive log structure**
  - Even RAM is a log
  - Log cleaner



Durability with RAM performance

- **Backups buffer update**
  - No synchronous disk write
- **Bulk writes in background**
  - Must flush on power loss
- **Pervasive log structure**
  - Even RAM is a log
  - Log cleaner
- **Hashtable, key → location**



Fast Crash Recovery

- **What is left when a Master crashes?**
  - Log data stored on disk on backups

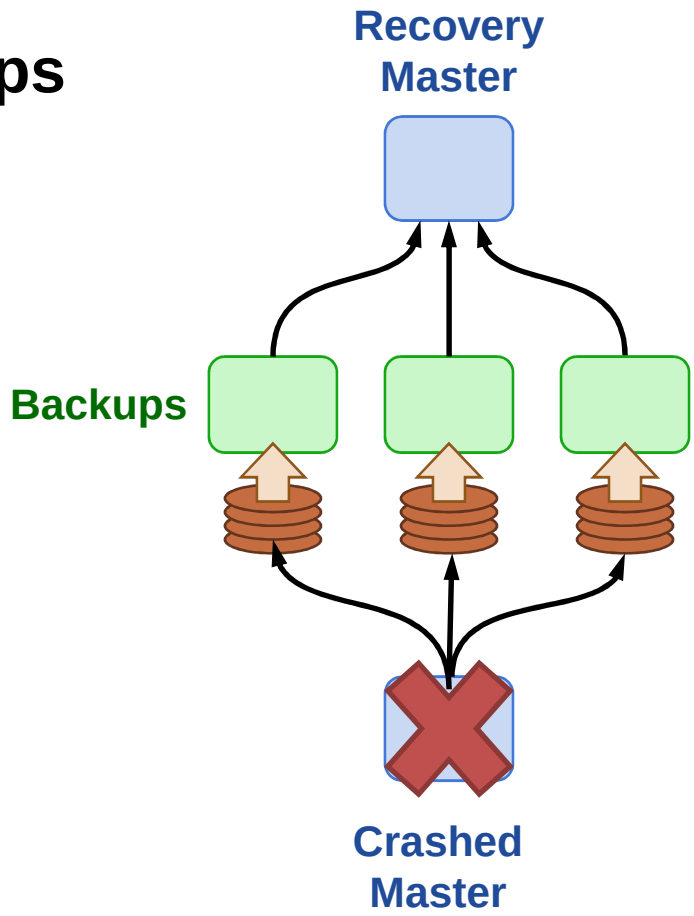
- **What must be done to restart servicing requests?**
  - Replay log data into RAM
  - Reconstruct the hashtable

- **Recover fast: 64 GB in 1-2 seconds**
- **Key to fast recovery: use system scale**

Recovery without Scale

- **Masters backed up to 3 Backups**
  - Each backup stores entire log
- **Problem: Disk bandwidth**
- **64 GB / 300 MB/sec**
  ≈ 210 seconds

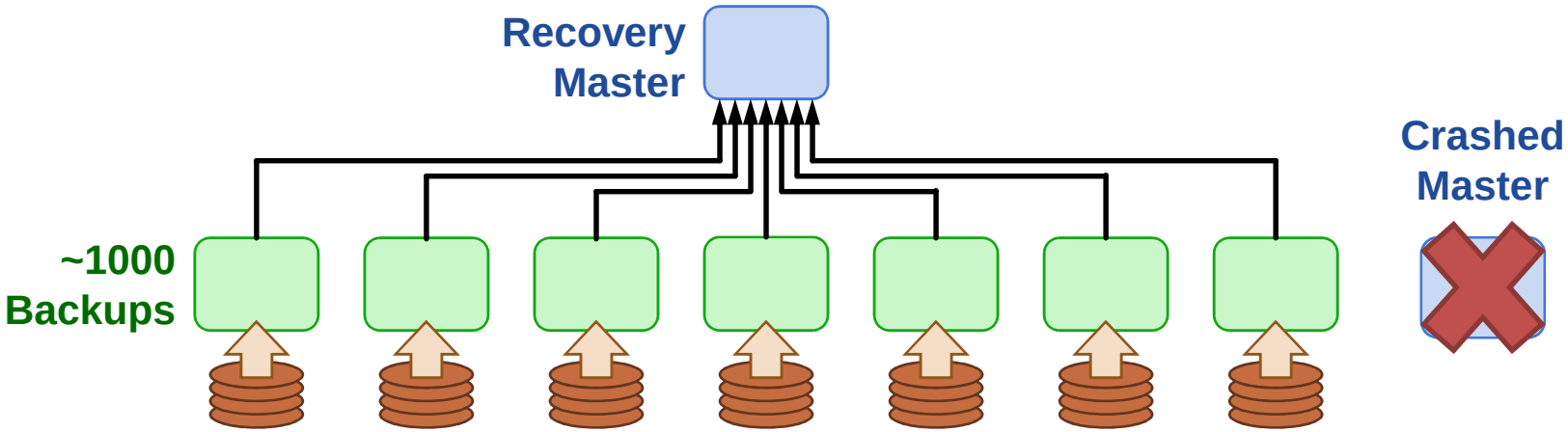
- **Solution: more disks**
  (more backups)



Solution: Scatter Log Data

- **Each log divided into 8MB segments**
- **Master chooses different backups for each segment (randomly)**
- **Segments scattered across all servers in the cluster**

- **Crash recovery:**
  - All backups read from disk in parallel
  - 64 GB / (1000 backups * 100 MB/s/backup) = **0.6 seconds**



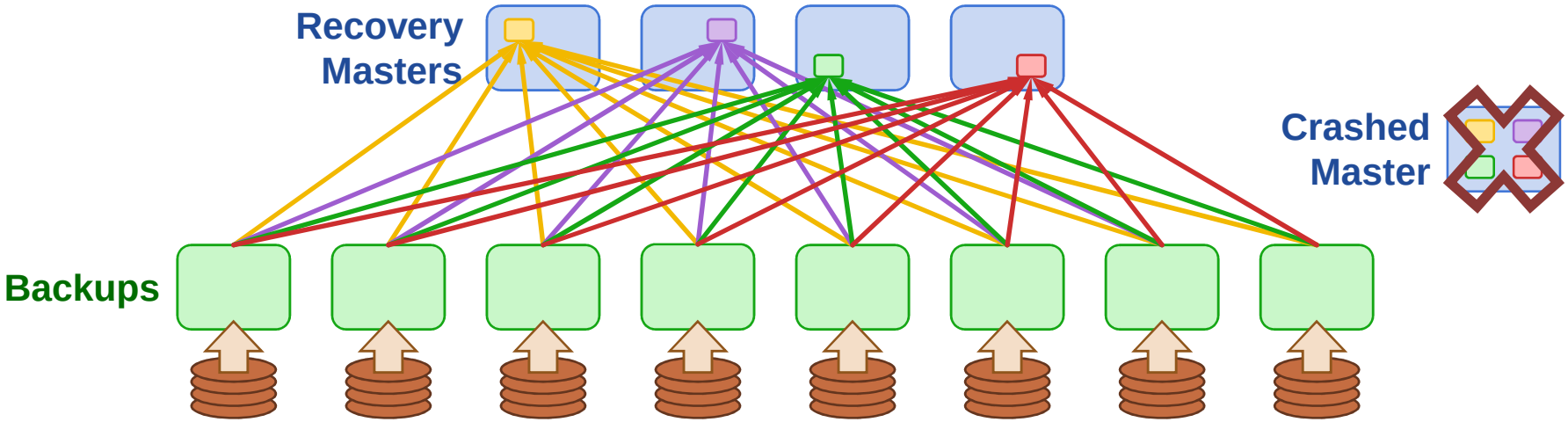
Problem: Network bandwidth

- **Second bottleneck: NIC on recovery master**
  - 64 GB / 10 Gbits/second ≈ 60 seconds
  - CPU and memory bandwidth a limitation

- **Solution: more recovery masters**
  - Spread work over 100 recovery masters
  - 60 seconds / 100 masters ≈ 0.6 seconds

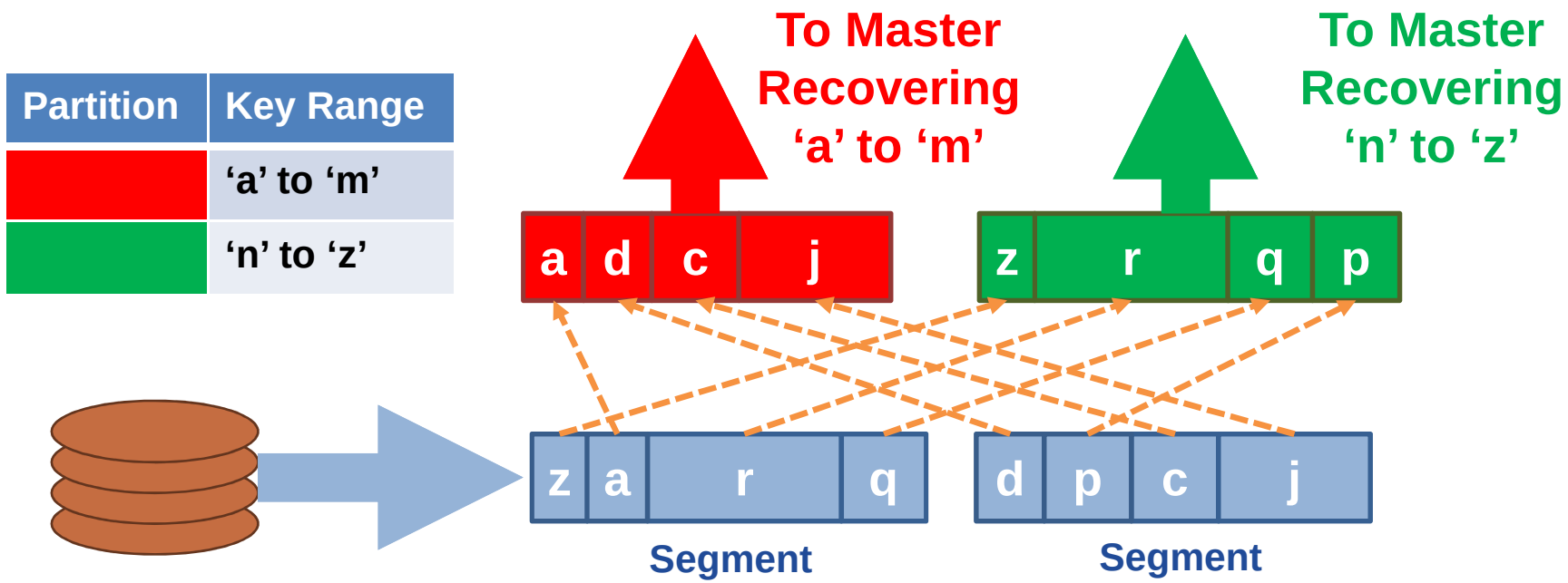
Solution: Partitioned Recovery

- **Divide each master's data into partitions**
  - Recover each partition on separate Recovery Master
  - Partitions based on key ranges, *not log segment*
  - Eliminates need for idle, empty Recovery Masters



Partitioning During Recovery

- **Backups receive a partition list at the start of recovery**
- **Backups load segments from disk and partition log entries**
- **Each recovery master replays only relevant log entries**



Issues Harnessing Scale

- **Balancing work evenly**
  - Parallel work is only as fast as the slowest unit
- **Avoiding centralized control**
  - Centralized control eventually becomes a bottleneck
  - Nodes often work without perfect/global knowledge

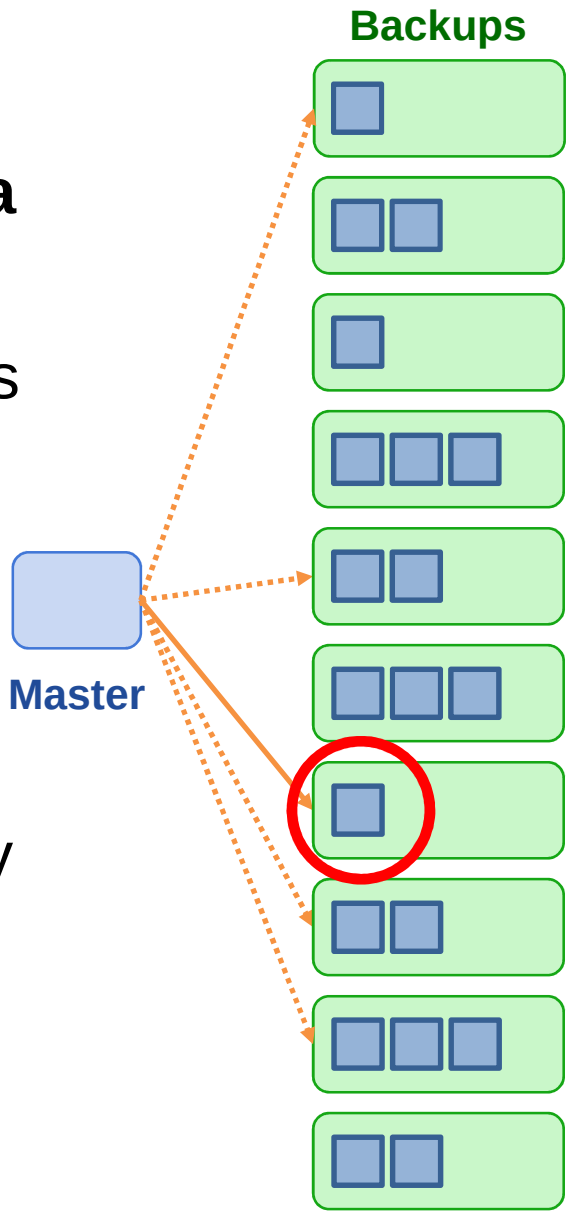
Balancing Partitions

- **Problem: Balancing work of each recovery master**
  - Recovery will be slow if a single Master is given
    - Too much data
    - Too many objects

- **Solution: Profiler tracks density of key ranges**
  - Done locally on each master
  - Balance size and number of objects per partition

Segment Scattering

- **Problem: Balancing time reading data across disks**
  - Recovery is slow if just one Backup is slow
- **Solution: Use similar approach to [Mitzenmacher 1996]**
  - Choose candidate Backups randomly
  - Select the “best”
  - Minimize worst-case disk read time



Detecting Incomplete Logs

- **Problem: Ensure entire log is found during recovery**
  - Centrally cataloging segments for each log expensive

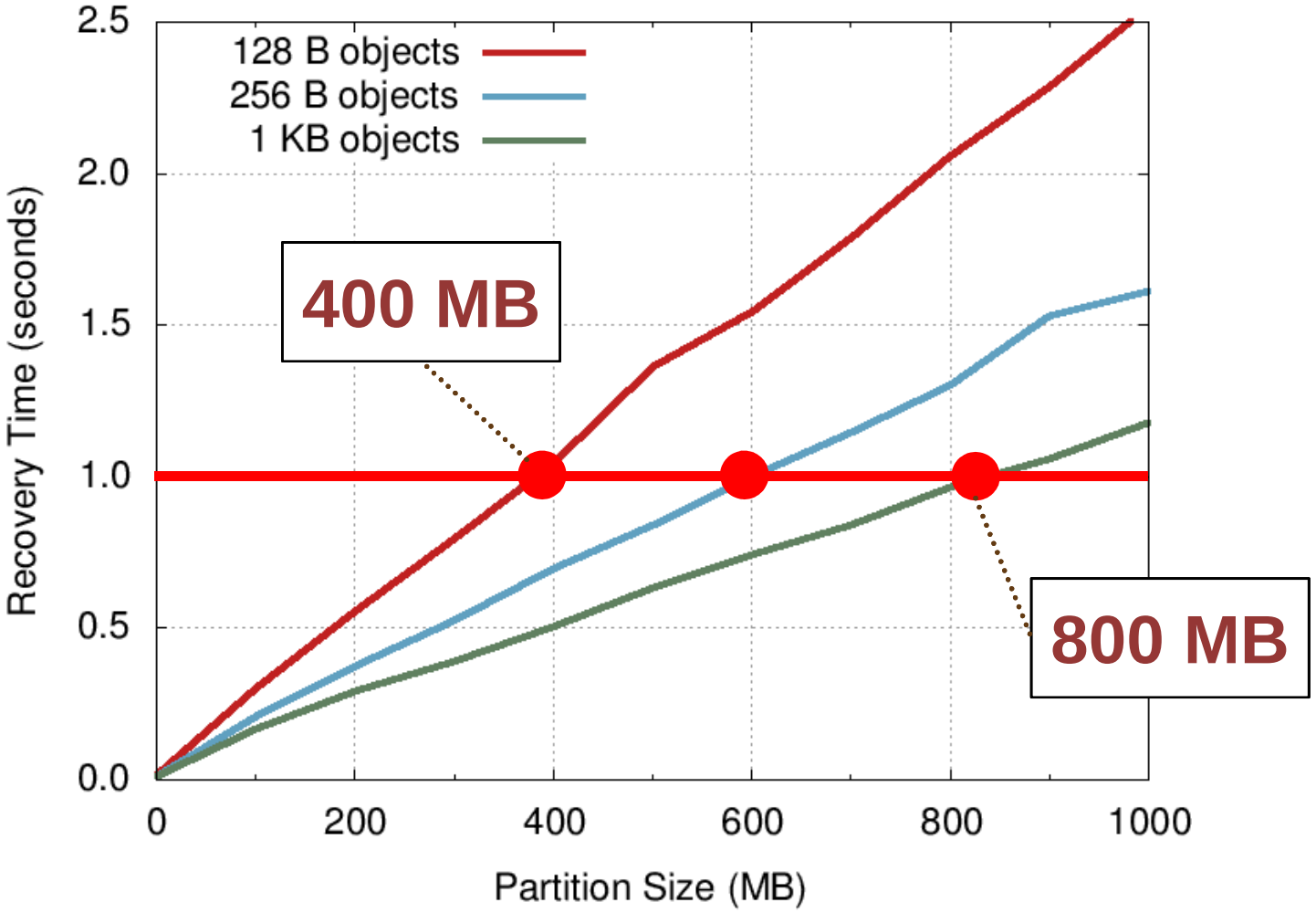
- **Solution: Self-describing log**
  - Masters record catalog of log segments in segments
  - Coordinator talks to each Backup at start of recovery
    - Finds most recent catalog
  - Can detect if all copies of most recent catalog are lost

Experimental Setup

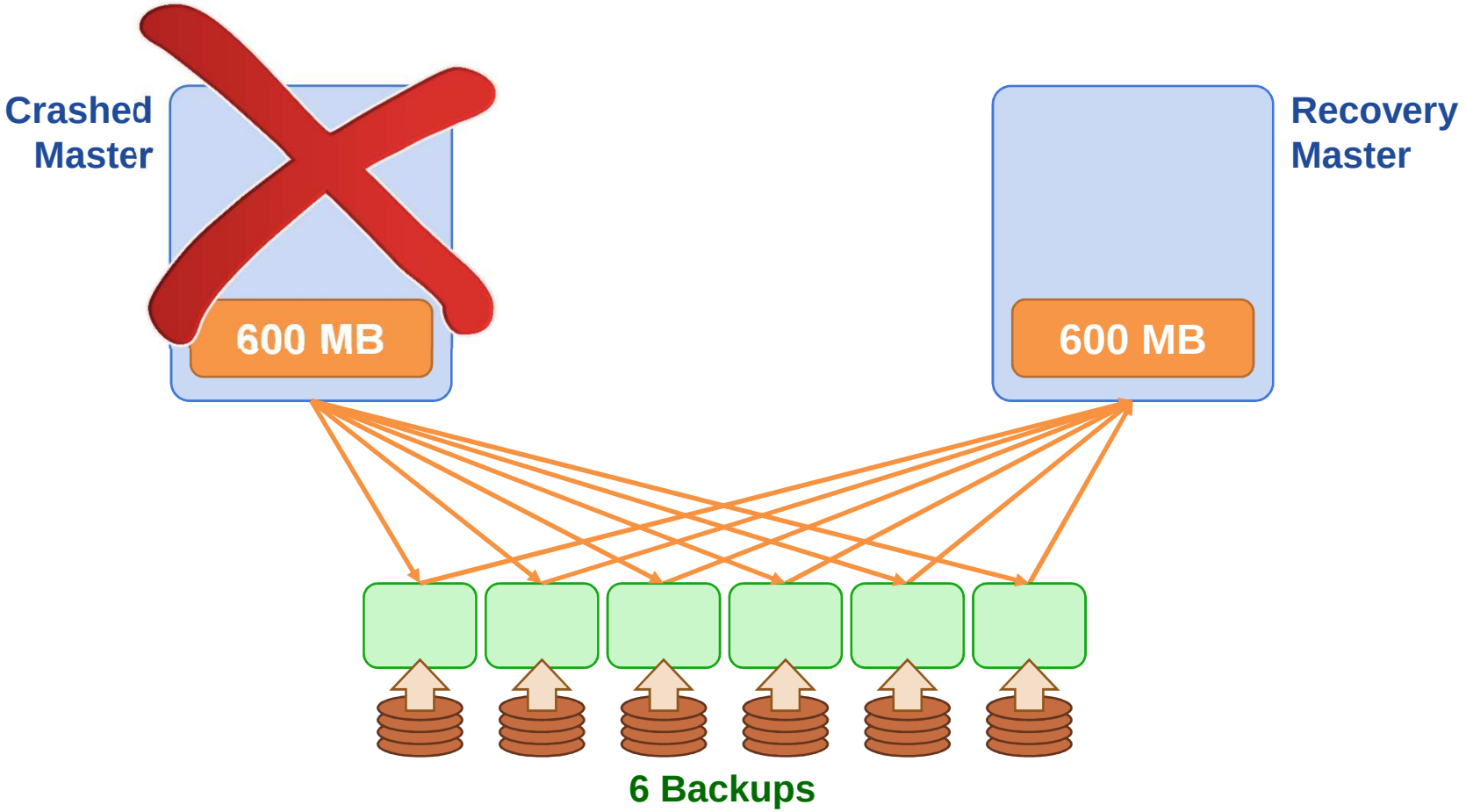
Cluster Configuration
60 Machines
2 Disks per Machine (100 MB/s/disk)
Mellanox Infiniband HCAs (25 Gbps, PCI Express limited)
5 Mellanox Infiniband Switches Two layer topology Nearly full bisection bandwidth

- **Approx. for datacenter networks in 3-5 years**
- **5.2 µs round trip from 100 B read operations**

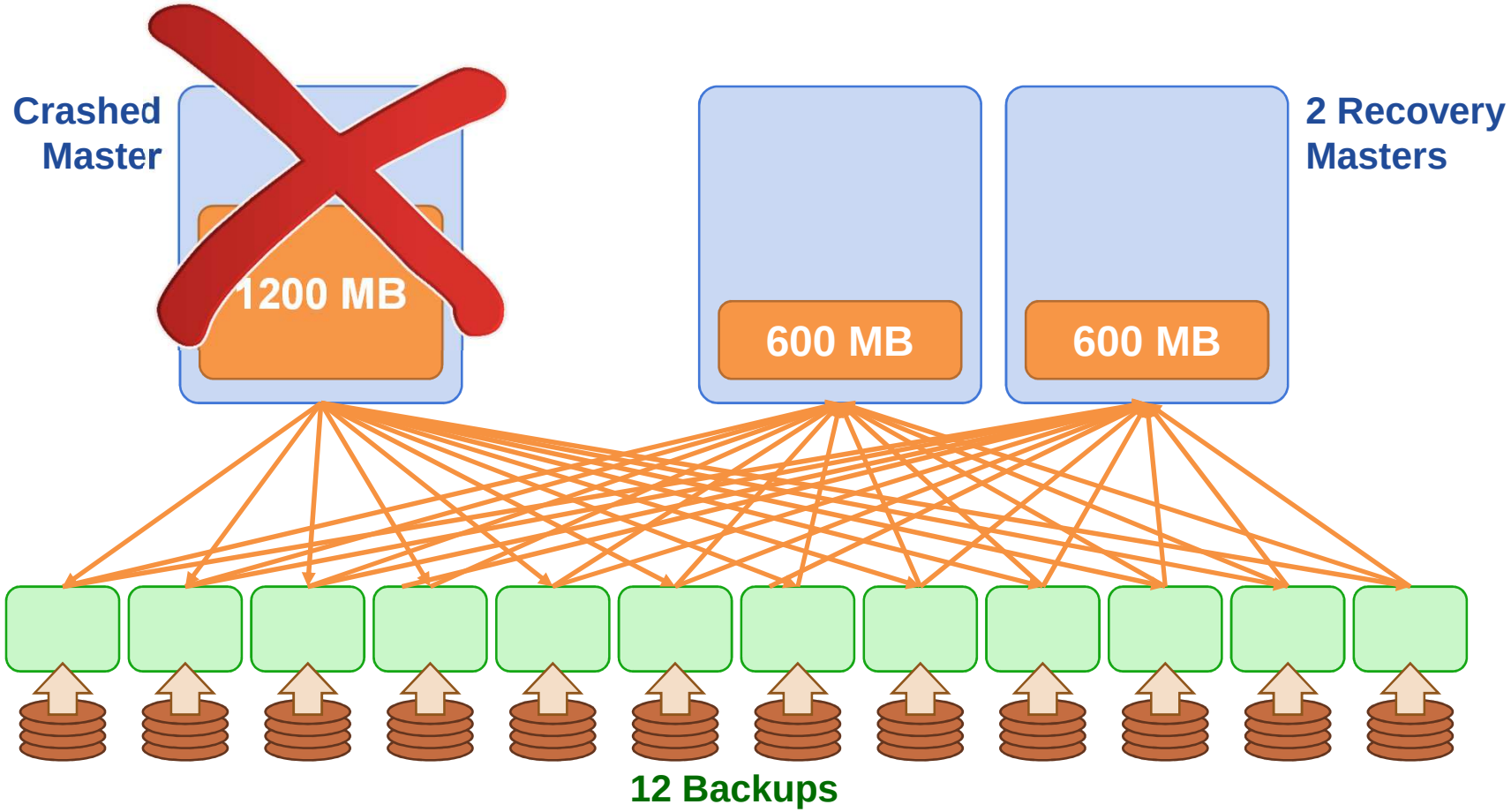
How much can a Master recover in 1s?



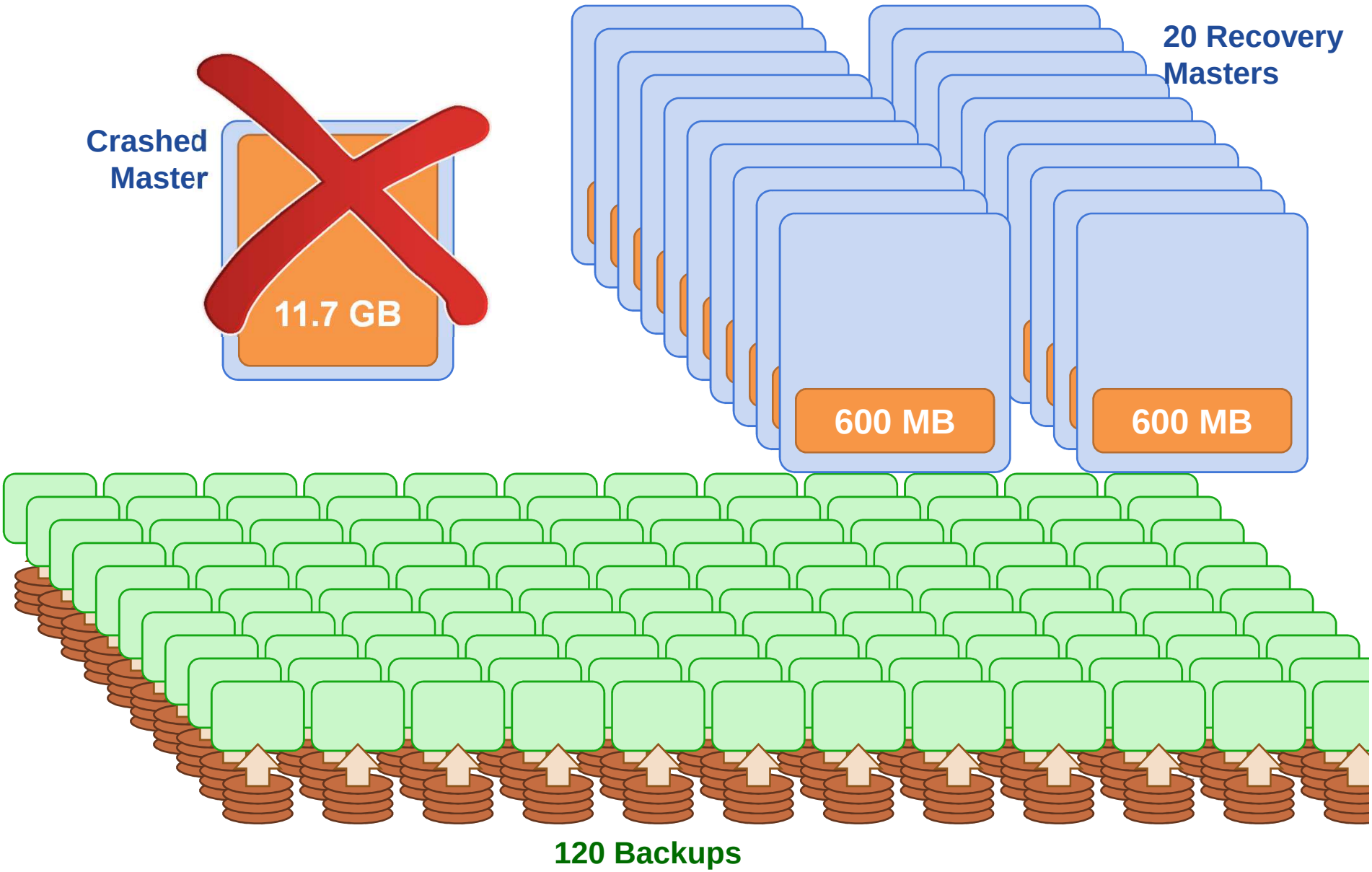
How well does recovery scale?



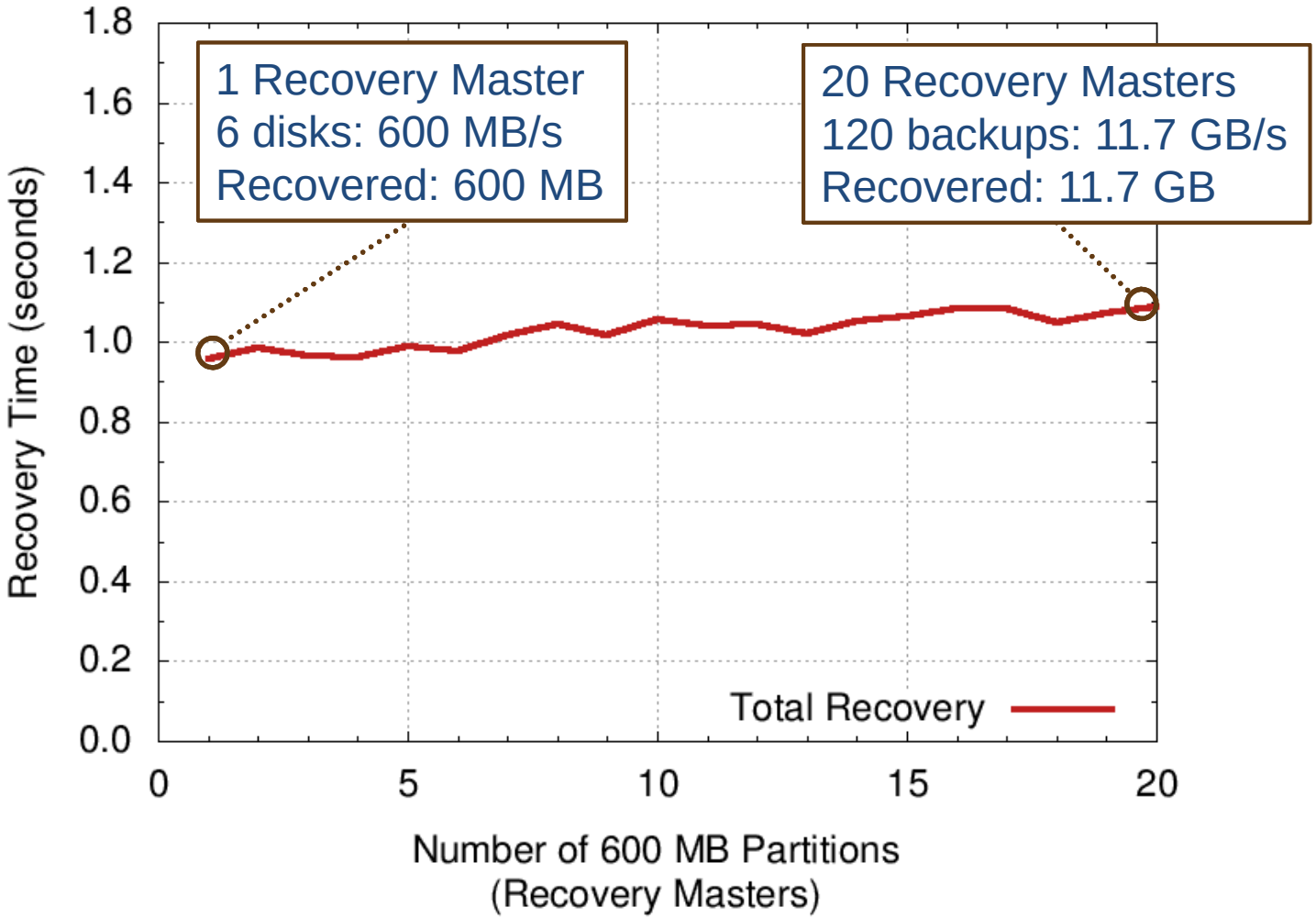
How well does recovery scale?



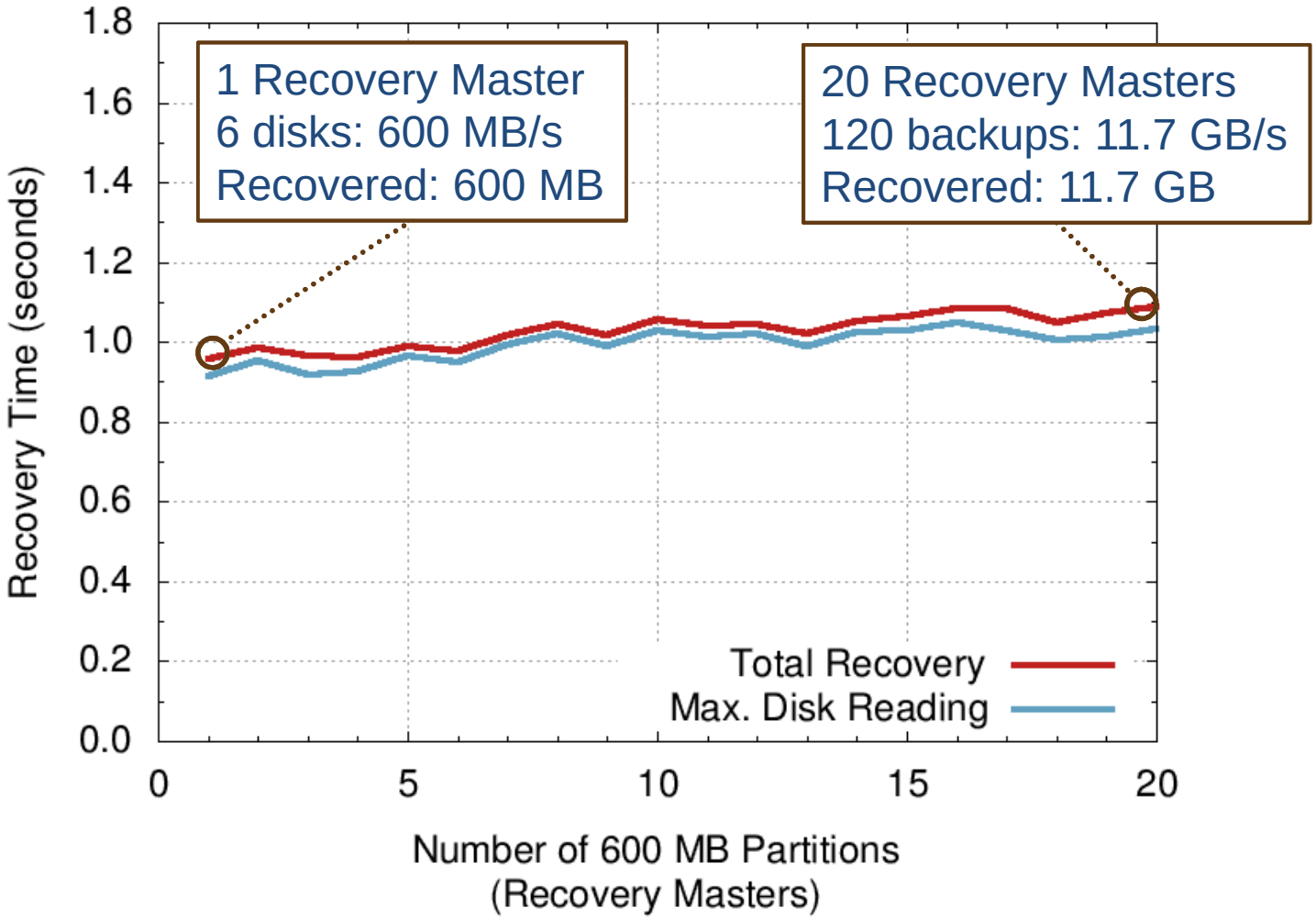
How well does recovery scale?



How well does recovery scale?

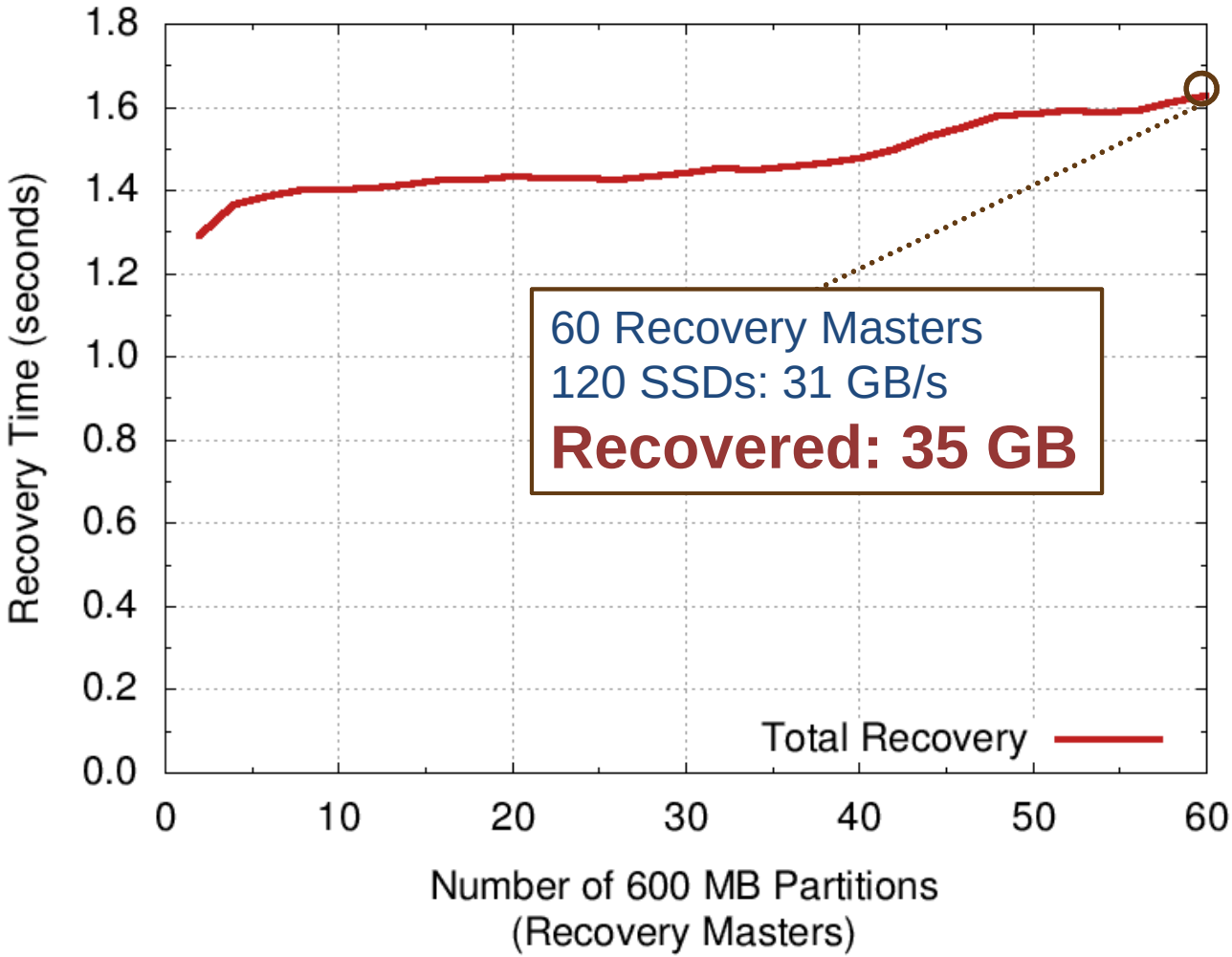


How well does recovery scale?



Total recovery time tightly tracks straggling disks

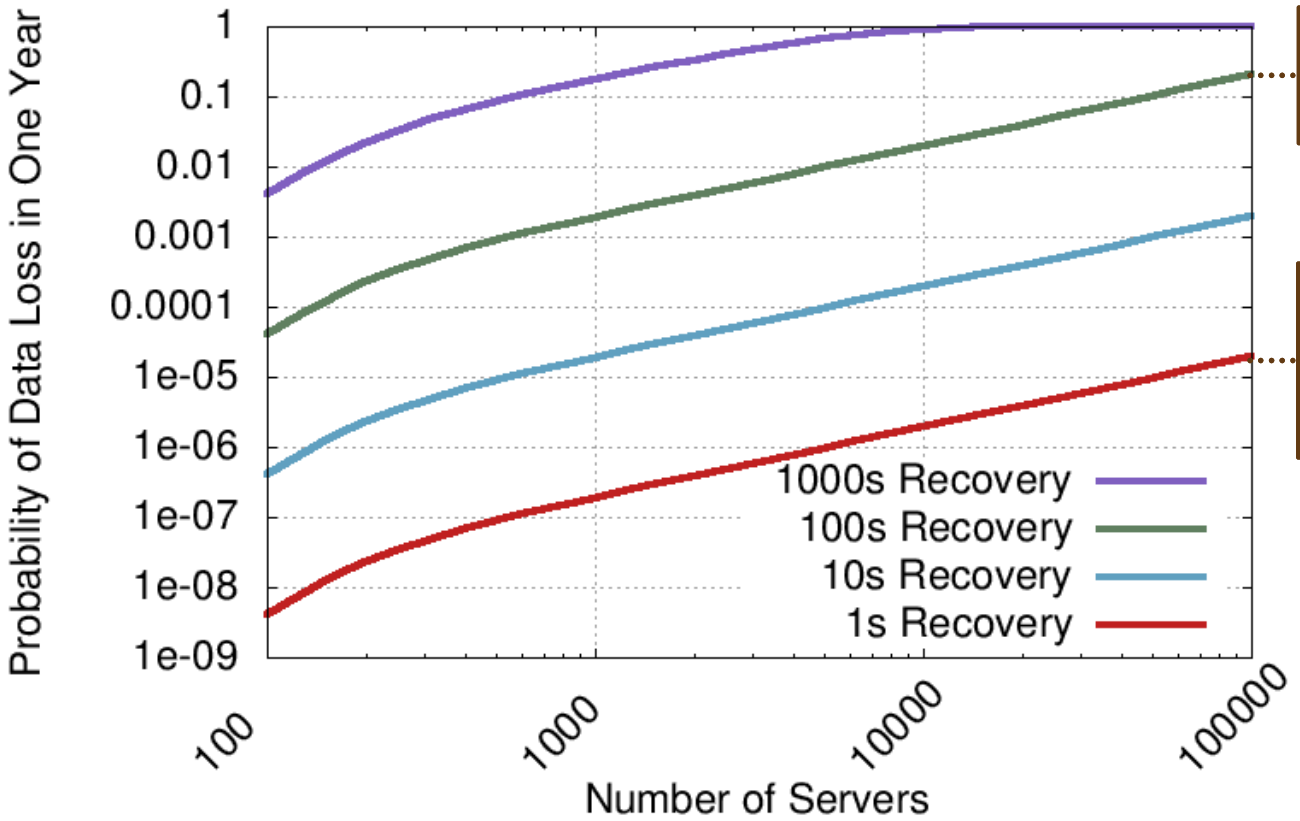
Flash Allows Higher Scalability



2x270 MB/s SSDs per recovery master

(vs. 6x100 MB/s disks per recovery master)

Fast Recovery Improves Durability



GFS (SOSP '03)
3-Copies on Disk

RAMCloud with
1-Copy in RAM
2-Copies on Disk

Assumptions:

- **2 failures/server/year**
- **Failures are independent**
- **Failures lose all contents of both RAM and disk**
- **Replication factor 3**

Related Work

- **Log-structured Filesystem (LFS)**
  - RAMCloud keeps log in-memory and on disk
  - More efficient cleaner; cleans from RAM instead of disk
- **memcached**
  - Apps must deal with backing store and consistency
  - Reduced performance from misses, cold caches
- **Bigtable + GFS**
  - Primarily disk based
  - Scatters across disks for durability
  - Bigtable uses a logging approach on GFS
  - Stores indexes, eliminates need for replay on recovery

Conclusions

- **Pervasive log structure**
  - Fast writes, inexpensive
- **Fast crash recovery in 1 to 2 s**
  - Recovers **35 GB to RAM in 1.6 s** using 60 nodes
  - Leverages the scale of the cluster
- **Potential Impact**
  - Easy to harness performance of RAM at scale
    - 5-10 µs access time
    - 100 TB to 1 PB
    - **As durable and available as disk**
  - Enable a new class data-intensive applications

Questions?

ramcloud.stanford.edu

Recovery Flow