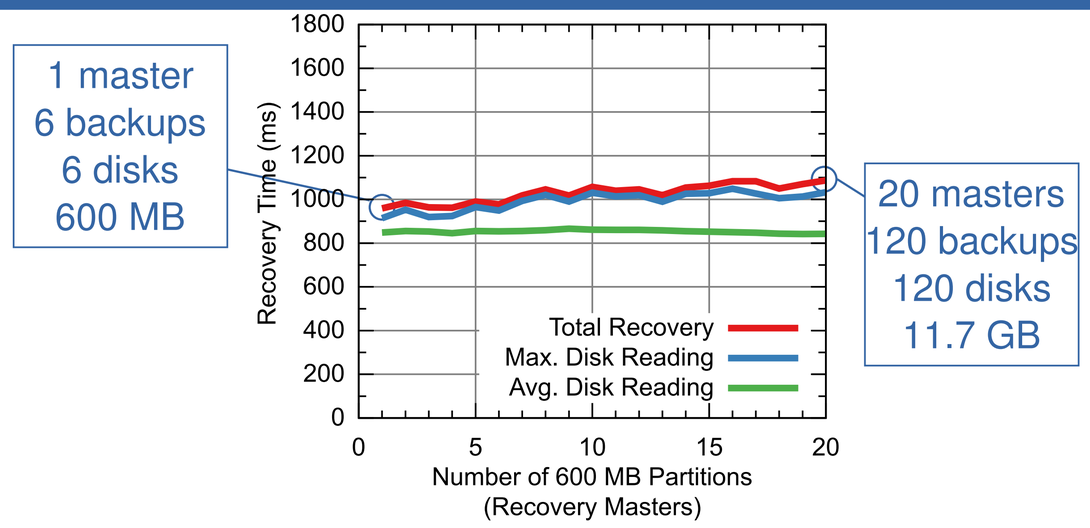
Fast Crash Recovery in RAMCloud

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

Motivation

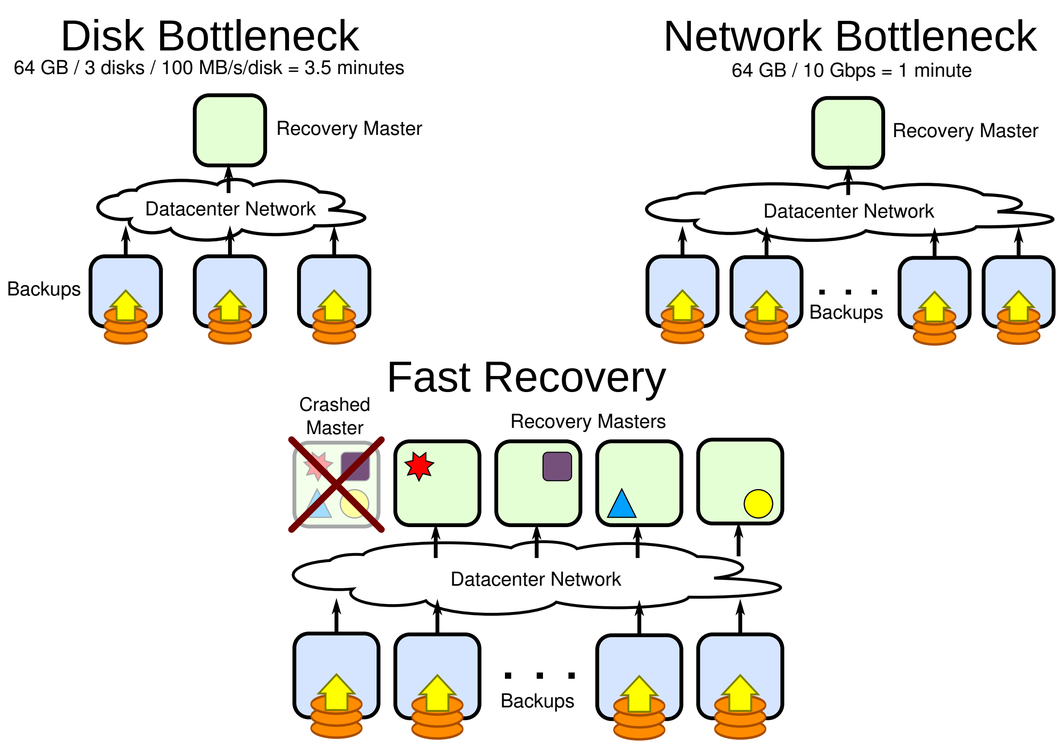
- All data always in RAM
  - 1,000 - 10,000 commodity servers
  - 64 GB DRAM/server or more
- Durability goals:
  - Small impact on performance
  - Minimum cost and energy
- Keep replicas in DRAM of other servers?
  - Triples cost and energy usage
  - Power failures are still a problem
- RAMCloud's approach: fast recovery
  - 1 copy in DRAM, backup copies on disk/flash
  - Hypothesis: failures will not be noticed

Results



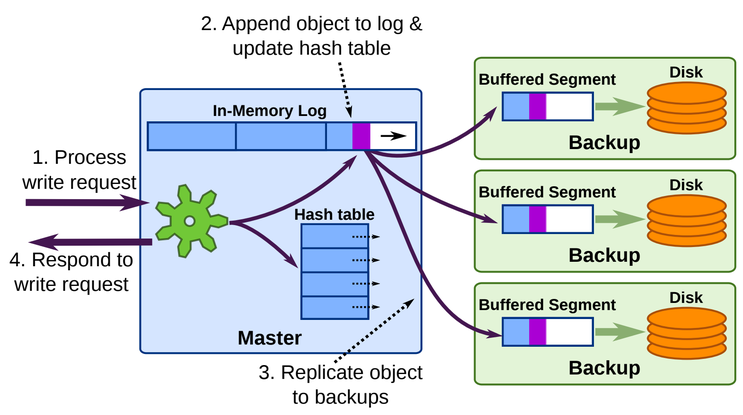
- 60-node cluster, 32 Gbps Infiniband network
- Recovered 11.7 GB in ~1 second
- Using flash improves to 35 GB in 1.6 seconds
- Time spent replicating is the current bottleneck
- Implementation hides disk speed variance well

Approach



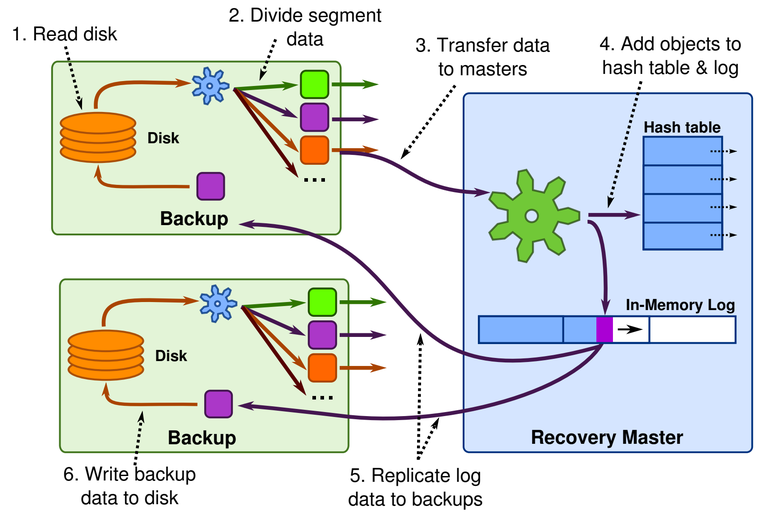
- Static set of backups is insufficient
  - Harness scale: Use many disks during recovery
    - From all 1,000+ machines
  - Scatter data throughout the cluster
  - 64 GB / 1000 disks / 100 MB/s/disk = 0.6 s
- Cannot reconstitute data quickly through a single NIC
  - Harness scale: Use many hosts (NICs)
    - About 100 recovery masters will do
    - Each recovery master can recover 400-800 MB/s
  - Need a ratio of about 6 disks to each recovery master

Data Scattering



- Masters replicate writes to backups immediately
- Backups buffer it and flush to disk/flash in batch
  - Need auxiliary power source for buffers for power failure
- Backup locations chosen randomly to scatter segments
  - Contraints on placment due to corrleated failures
  - Tweaked to balance expected read time
  - Provides the needed read bandwidth for recovery

Replay



- Every host is involved in recovery and they work in parallel
- Work on each host proceeds in parallel (steps are pipelined)
- Recovery masters make several parallel requests to backups
- Prevents pipeline stalls when backups are not ready with data
- New log segments are buffered until recovery is complete