FAST CRASH RECOVERY IN RAMCLOUD

Presentation by: Wesley Coomber
AGENDA

- Motivation
  - What is the problem?
  - Why is it important?

- Technical Contribution
  - How do they solve the problem?

- Evaluation
  - How well did they accomplish their goals?
Motivation

- Large-scale web applications are becoming more common
- These applications manipulate data intensively
- Data manipulation is very slow on disks
- What if we could store all the data in DRAM??
Motivation

- DRAM is much faster to manipulate, but there are issues
- It is difficult for developers to effectively leverage DRAM
- Eg. Cache Consistency, Cache misses, and backing store overheads
- Enter Ramcloud.
What is Ramcloud?

- All Data in DRAM
- Log-structured Memory and Storage
- Key-value distributed storage system
- Replicas stored on Disks
- Leverages cluster system’s large scale
- **Excellent Availability from fast crash recovery**
Data Model

• Simple Key-value store
• Ramcloud is made up of tables that contain objects
• An Object is composed of 3 things:
  • 64-bit identifier key
  • Upto 1MB byte array value
  • 64-bit version number
RAMCLOUD CLUSTER ARCHITECTURE
SYSTEM STRUCTURE

• Master
  • Stores objects in DRAM and serves clients
• Backups
  • Stores replicas from masters in disk memory
SYSTEM STRUCTURE

• Coordinator
  • Manages configuration info such as network addresses of servers
  • Also stores location of objects
  • Usually transparent to client
**SYSTEM STRUCTURE**

- Coordinator places objects in masters using tablets
- Tablets are a consecutive range of keys in 1 table
- Small tables stored entirely on 1 server
- Larger tables split across multiple servers
- Tablet config transparent to Client apps
Log-structured Storage

1. Process write request
2. Append object to log and update hash table
3. Replicate object to backups
4. Respond to write request

In-Memory Log
Hash table
Master
Buffered Segment
Backup
Disk
**Log Structure**

- Logs are divided into 8MB segments
- Segment is unit of buffering + I/O
- Background log cleaner
- Log-structure for both Disk and DRAM
- Master maintains hash table mapping to most recent version of object
- Caveat: Backups must use SRAM to ensure durability
Recovery Approach
RECOVERY

• Taking Advantage of Cluster Scale
• Hundreds of Recovery Masters working with thousands of Backups
• Significantly speed up recovery times
**Steps to Recovery**

- Pre-failure, scatter log segments across backups
  - Uses independent refined random selection

- Failure Detection
  - Detectable by Client and Ramcloud System

- Recovery Flow
  - Setup into Replay into Clean-up
SMARTLY SCATTERING LOG SEGS

• Segment’s master and backups must be in different racks

• Distributed to ensure equal backup recovery time

• Masters should work together to not overwork any 1 backup

• Storage servers are constantly being added/removed
Failure Detection

• Clients notice if server fails to respond to a RPC
• Ramcloud periodically self-checks servers using suspicion pings
• Either way, failures reported to coordinator
• Coordinator verifies server failure and starts Recovery
Recovery Flow

- First Setup and find log segment replicas
- Detect incomplete logs
  - Log Digests, active and inactive
- Start partition recoveries
  - Master itself divides recovery work
  - Dead Master’s will
Recovery Flow (2)

- Second Step: Replay
- Dominates recovery time
- Fast replay through harnessing concurrency in 2 dimensions
  - Data Parallelism
  - Pipelining
RECOVERY REPLAY PROCESS
Recovery Flow (3)

• Finally, clean-up after masters have completed recovery, talk to coordinator
• Tablet Profiling
• Abstractly represented by dynamic tree of bucket arrays

• Consistency (full description outside of paper scope)
  • Ram cloud will disable a sick master’s backup operations
  • Coordinator Failures handled using ZooKeeper
TABLET PROFILING

Bucket Key
- Not in Partition
- In Partition
- Unknown

Partition
Evaluation

- Performance and Scalability of Crash Recovery on 60-node cluster
- Cluster made up of traditional off-shelf components
- Except for the high-end Infiniband-based networking hardware
- Default config was one backup server per machine, with one disk.
**Experimental Cluster Config**

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td>Xeon X3470 (4x2.93 GHz cores, 3.6 GHz Turbo)</td>
</tr>
<tr>
<td><strong>RAM</strong></td>
<td>16 GB DDR3 at 1333 MHz</td>
</tr>
<tr>
<td><strong>Disk 1</strong></td>
<td>WD 2503ABYX (7200 RPM, 250 GB)</td>
</tr>
<tr>
<td></td>
<td>Effective read/write: 105/110 MB/s</td>
</tr>
<tr>
<td><strong>Disk 2</strong></td>
<td>Seagate ST3500418AS (7200 RPM, 500 GB)</td>
</tr>
<tr>
<td></td>
<td>Effective read/write: 108/87 MB/s</td>
</tr>
<tr>
<td><strong>Flash</strong></td>
<td>Crucial M4 CT128M4SSD2 (128GB)</td>
</tr>
<tr>
<td><strong>Disks</strong></td>
<td>Effective read/write: 269/182 MB/s</td>
</tr>
<tr>
<td><strong>NIC</strong></td>
<td>Mellanox ConnectX-2 Infiniband HCA</td>
</tr>
<tr>
<td><strong>Switches</strong></td>
<td>5x 36-port Mellanox InfiniScale IV (4X QDR)</td>
</tr>
</tbody>
</table>
Ideal size of partitions?
HOW MANY DISKS PER RECOVERY MASTER?
Ideal count of partitions?
How well does recovery scale? (1)
How well does recovery scale?
HOW WELL SEG SCATTERING WORKS?

![Cumulative Percent vs Recovery Time Graphs](image)

- **Cumulative Percent**
  - 100%
  - 75%
  - 50%
  - 25%
  - 0%

- **Recovery Time (seconds)**
  - 0
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8

- **Even Read Time**
- **Even Segments**
- **Uniform Random**
- **Fans Normal**
- **Fans High**
Will scattering cause data loss?


**Evaluation Conclusion**

- Demonstrated we can harness cluster resources to quickly recover from crashes
- RAMCloud distributes backup data across many machines
- System also uses data parallelism + pipelining to achieve 1-2 second recovery times
- Evaluation only performed on a small cluster (60 machines)
The End

Questions?