RAMCloud the new frontier of DRAM

Abdulaziz A. Alamri

George Washington University

Table of Contents

Abstract

In this paper, we will discuss a new Technology of DRAM called RAMCloud. We will Analysis all the features of this RAMCloud and its applicability. The paper will start with a complete background. Next, the paper will demonstrate how DRAM’s leading companies think about RAMCloud. This is followed by a comparison of RAMCloud strengths and weaknesses. Then we analyze the cost effective to have RAMCloud for the companies. Finally, the conclusion displayed the cost effect of applying RAMCloud.

Keywords- RAMCloud; DRAM; RAMCloud Architecture; RAMCloud Latency.

Introduction

Over the past few decades, performance of accessing data has become the main concern for the computer's manufacture and the computer's user. Both concerned about accessing and processing data efficiently. Also, Scalability of the data centers is serious issue for companies that have a data hosting, providing and storing. Moreover, these companies various from each other because of the how many restore points we have and it depend completely on the validity of the backups. One more issue with a huge data centers is collecting the data that need to be process from many data bases which lead to a noticeable delay in the whole process.

As we can see, many issues regarding storing, accessing and processing the data. From that concerns RAMCloud came to build a solution to handle the performance and fast access of a data. RAMCloud now can achieve inside a very high speed network that connect to many datacenters 5 ms of response time of accessing the data. In addition, RAMCloud handle the data centers scale issue completely.

RAMCloud can store 64TB if the data center have 1000 servers each have 64GB. As same as, RAMCloud can store 1Petabyte if the data center have 4000 servers each have 256GB. Furthermore, RAMCloud can handle the backups in desks with a high confidence that backup is restorable and the backup write operation cost 15 ms without effecting the performance of any other processes. Likewise, RAMCloud can collect independent segments of data from to many database instances without effecting the response time. Finally, RAMCloud could help the processing time of dependent requests that can't be process in parallel.

In this paper, we will study in detail one major technique that RAMCloud clam use to handle all mentioned issues. This technique uses DRAMs as a whole server instead of just a memory. Furthermore, This technique will handle mentioned issues.

Background

The concept of Dynamic Random Access Memory (DRAM)

Recently, most of main memory production line produce a Dynamic Random Access Memory(DRAM). The architecture of this type of memory consists of cells. Each cell have a capacitor to make it able to store 1bit of data when the electricity charged. However, the capacitor loss the charge over the time that's make the capacitor loss the data. DRAM as it standard use a refresh functionality periodically to help it keep the electricity charge and then keep the data stored.

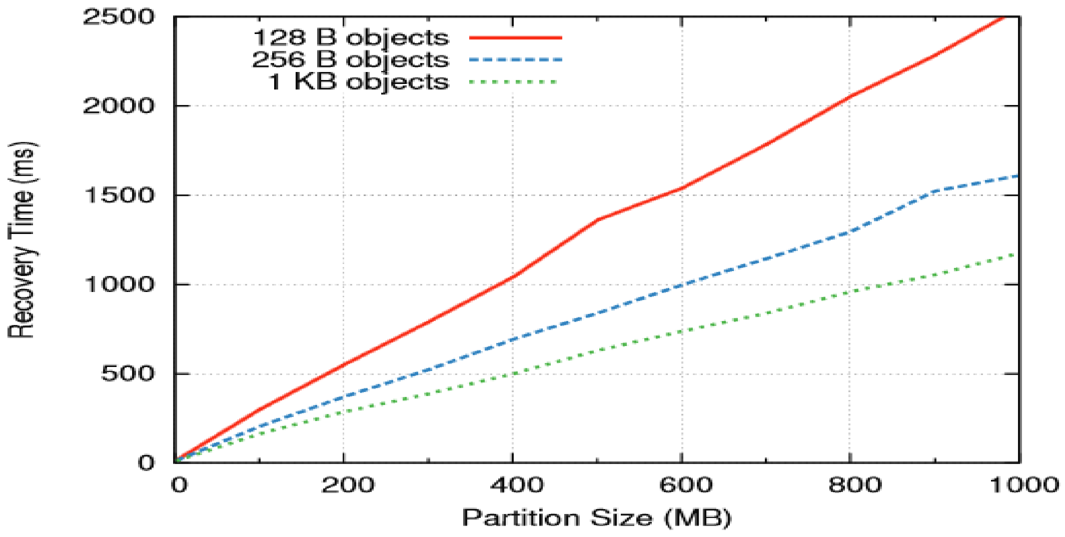
During that refresh time, memory can't response to any access request and that call refresh latency. As many as refresh latency a system have as much as delay in performance and accessing data will have.

According to the research [1] the refresh latency cause a delay as 8.2% in 8 GB memory size and in 32 GB cause a delay of 19.9%. Also that refresh function make the DRAM one of the biggest power consumer in any device of computer.

What is RAMCloud?

RAMCloud is the a new storage category. RAMCloud project is based in the Department of Computer Science at Stanford University.

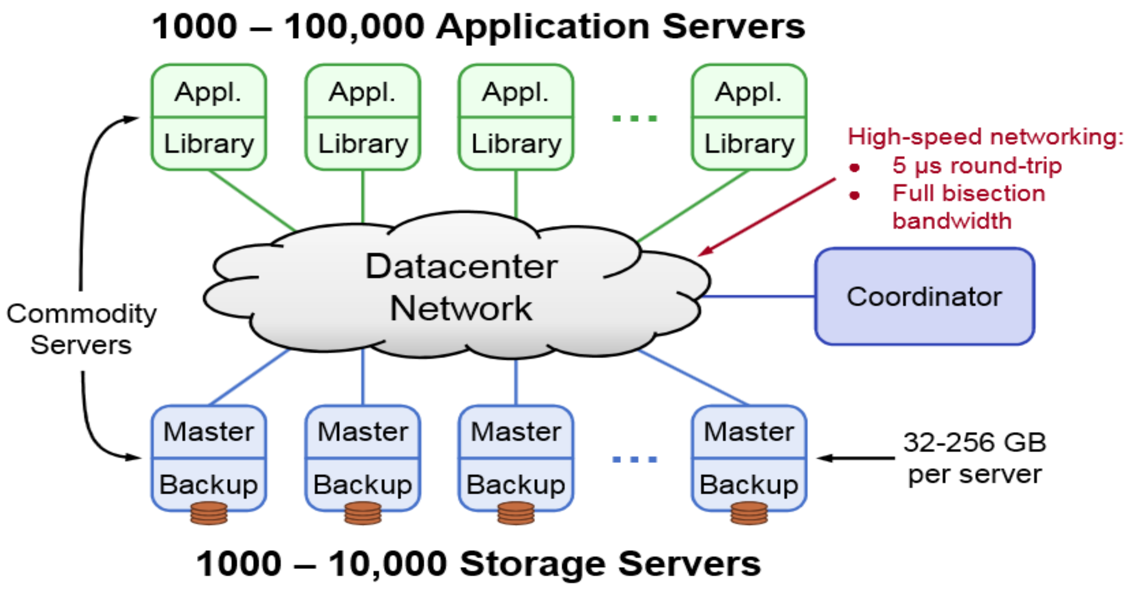
According to [3] RAMCloud designed to deal with a huge scale datacenter for applications. RAMCloud scalability could contain hundreds and thousands of servers reaching up to few petabytes of data. Since the principal of RAMCloud is to have a complete DRAM storge, there is no cache to handle the performance. RAMCloud reaching an outstanding performance if the datacenter support a high speed intercommunications between storage servers. It reached a very high speed data access with 5 ms read operation and 15 ms write operation and that makes RAMCloud 1000 times faster than regular disk storage servers.



RAMCloud is reliable system can recovered and backup easily. It can be replicated on nonvolatile storage like disks. Finally, RAMCloud availability is outstand because it can recover to the normal situation after a crash happed in less than 2 seconds. Since RAMCloud support the new category of storage it also support a new applications that collect data from different data storage and make it seems to be from a one data storage which support a real time system, So that will make the data it very fast to access.

According to the [3] “We believe that RAMCloud, or something like it, will become the primary storage system for structured data in cloud computing environments such as Amazon's AWS or Microsoft's Azure. We have built the system not as a research prototype, but as a production-quality software system, suitable for use by real applications”.

The RAMCloud Architecture can be shown as in Figure 2 below



Two main concerns for the RAMCloud project which is the latency and the scalability. The difficulty is how to improve the salability without affecting the latency. The latency is very important that keep RAMCloud have advantage of the DRAM and make high speed end to end latency.

RAMCloud support a self-domain management for the whole severs inside its data centers, Which make RAMCloud a valuable solution for management for complex distribution systems. RAMCloud now can support up to 10,000 storage servers.

Log-Structured Memory for DRAM-based Storage

Since the RAMCloud consist of thousands of DRAM storages, and DRAM cost too much RAMCloud should use that DRAM storage with complete efficiency. However, the tradition DRAM use allocators for a general purpose storage, and these types of allocators almost waste half of the memory with efficiently.

According to [4] they try to handle this issue by implementing a log structure memory in the RAMCloud storage servers. As a result, the system became as a file system log structure with three difference keys:

- Since the metadata of the structured memory consist of two values log digest and tombstons which recollect the log after a crash happed and avoid deleting instance by resurrection. Because of that memory log structure become simpler than file system log structure.
- Maximize RAMCloud utilization by applying two different policies for clear the memory.
- Minimalize the cost of cleaning objects by running parallel cleaning that run on a many threads simultaneously.

By applying these three keys and according to [4] the memory utilization reached 80-90%. The two clearing policies enhanced the performance up to 6 times.

One of the suggested solution was using Malloc function that exist in the C-Library. However, changing the access patterns will affect the memory utilization. According to [4] “We measured variety of all locators under synthetic workloads and found that all of them waste at least50% of memory under conditions that seem plausible for a storage system”.

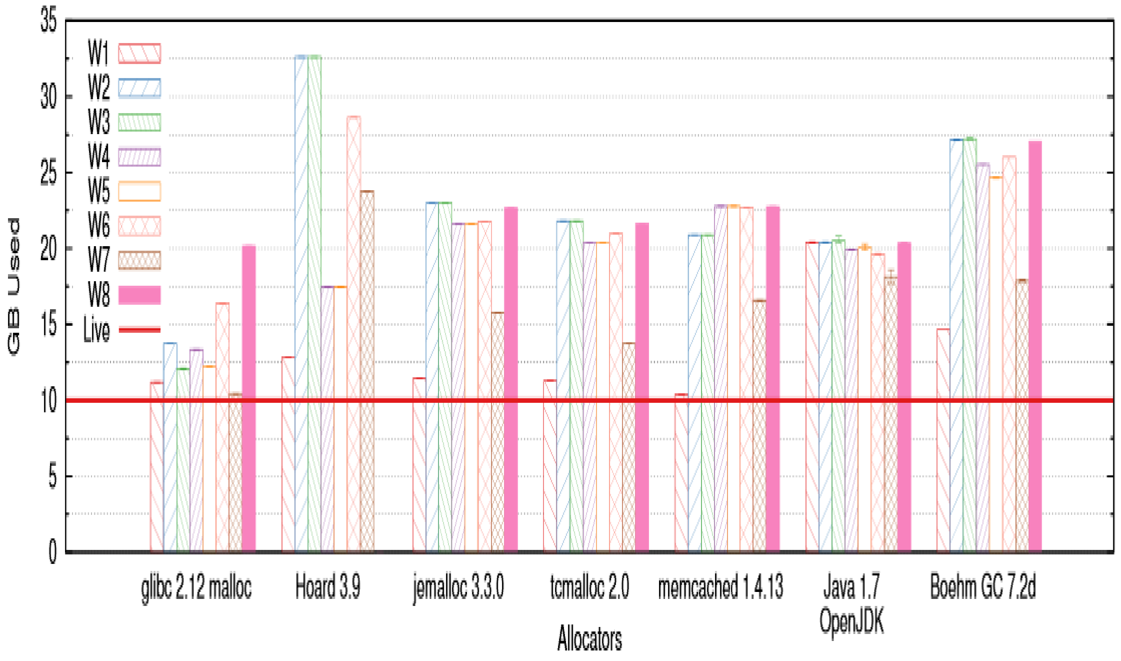
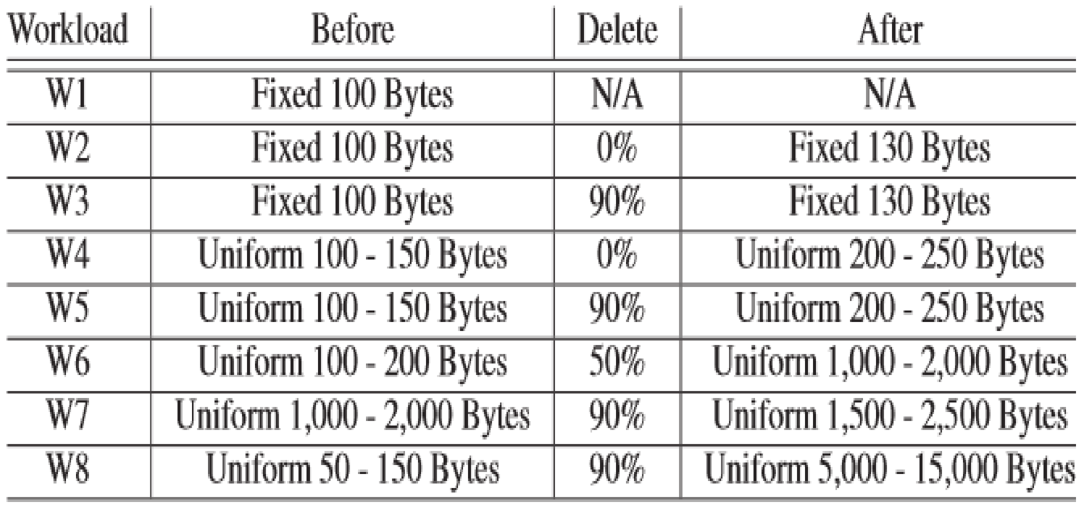


Figure 2: shows the total memory needed by allocators that support 10GB of data changing workload as explained in the Table 1 below



Workload	Before	Delete	After
W1	Fixed 100 Bytes	N/A	N/A
W2	Fixed 100 Bytes	0%	Fixed 130 Bytes
W3	Fixed 100 Bytes	90%	Fixed 130 Bytes
W4	Uniform 100 - 150 Bytes	0%	Uniform 200 - 250 Bytes
W5	Uniform 100 - 150 Bytes	90%	Uniform 200 - 250 Bytes
W6	Uniform 100 - 200 Bytes	50%	Uniform 1,000 - 2,000 Bytes
W7	Uniform 1,000 - 2,000 Bytes	90%	Uniform 1,500 - 2,500 Bytes
W8	Uniform 50 - 150 Bytes	90%	Uniform 5,000 - 15,000 Bytes

Table 1: shows a summary of the workload that represented in Figure 1

The RAMCloud Storage System

Storage management

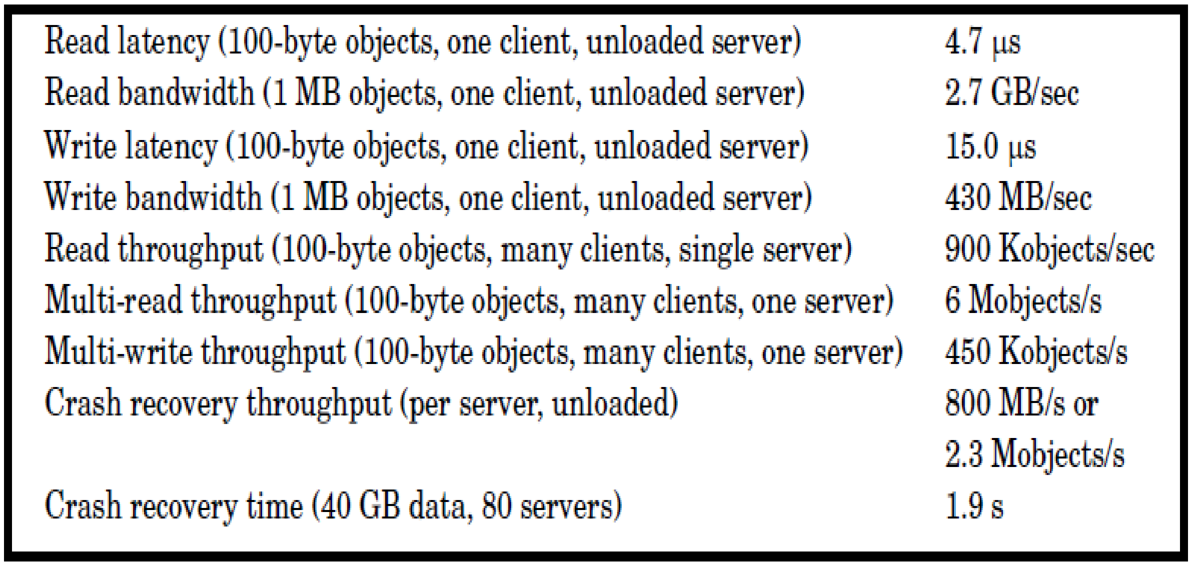
RAMCloud follow the approach of the “unified log structure” which manage all data either in the DRAM memory or in the disk backup storage. That makes the backup smoothly access either by read or by write. Observed point that the unified log structure restoration and efficiently utilize the DRAM twice as efficiently as the disk storage systems. With this management we can minimize the bandwidth of INPUT/OUTPUT in the disk storage system.

Latency

To have a flexible system we are going to have some latency. And what make that latency value increase is the interrupt calls from the system to the system resources especially the Network Interface Card(NIC). This is almost the biggest challenge of the DRAM architecture.

Crash recovery

According to [5] RAMCloud have a great level of availability and that came because of the scalability of the whole system. While the DRAM main server working there is hundreds of concurrently working storage working servers at the same time making many backups. So, as soon as the crash happen there is a ready to recover backup in less than 2 seconds.



Read latency (100-byte objects, one client, unloaded server)	4.7 μs
Read bandwidth (1 MB objects, one client, unloaded server)	2.7 GB/sec
Write latency (100-byte objects, one client, unloaded server)	15.0 μs
Write bandwidth (1 MB objects, one client, unloaded server)	430 MB/sec
Read throughput (100-byte objects, many clients, single server)	900 Kobjects/sec
Multi-read throughput (100-byte objects, many clients, one server)	6 Mobjects/s
Multi-write throughput (100-byte objects, many clients, one server)	450 Kobjects/s
Crash recovery throughput (per server, unloaded)	800 MB/s or 2.3 Mobjects/s
Crash recovery time (40 GB data, 80 servers)	1.9 s

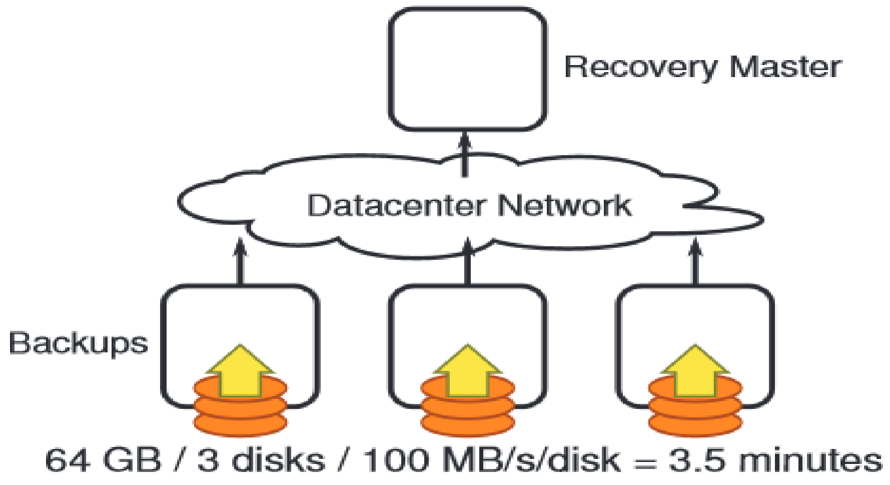
Table 2: summarizes a few key performance measurements conducted in September 2014

Fast Recovery in RAMCloud

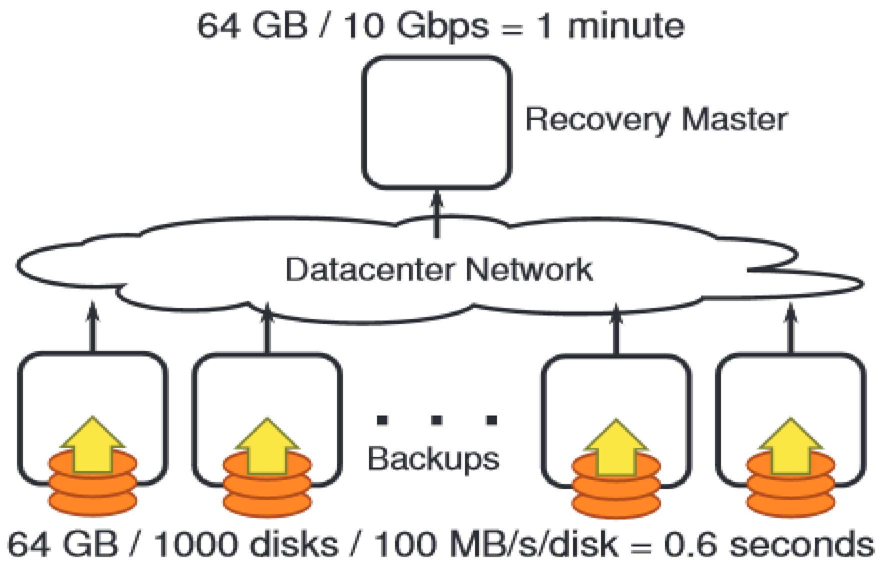
Recovery

When RAMCloud have a crash it directly will try to reconstruct the crashed object. This will need to contact the updated version of the backup and have the object and the hash table for comparison the correct version of the object to be restored. During this whole life cycle the crash still and the server not recovered yet. The greatest thing of the RAMCloud architecture is it can minimize this whole life cycle of the crash recovery period and make if not noticeable. Three situations we can have in the RAMCloud recovery methodology.

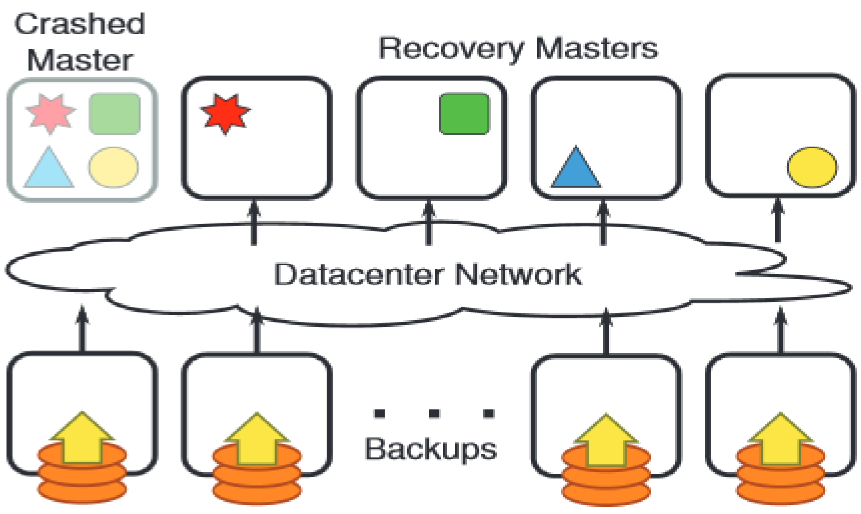
- Methodology 1 : Recovery Bottleneck is shown when a few backups servers try to share the same disk bandwidth



- Methodology 2 : Scattering log segments across many backups to eliminate the disk storage bottleneck, but this will lead to the CPU bottleneck since the NIC have limited the resources.


64 GB / 10 Gbps = 1 minute
Recovery Master
Datacenter Network
Backups
64 GB / 1000 disks / 100 MB/s/disk = 0.6 seconds

- Methodology 3 : Fast recovery is gained by recovering each partition on a separate recovery master.



Using Scale

RAMCloud's huge cluster give it a big advantage in the recovery after the crash happed. we can see in the methodology 1 with a very basic mirrored approach only 3 backups and the log segments on the three of them. The issue with this methodology is the bottleneck for the recovery because the read operation for a few disk will consume the time. According to [6] " it would take about 3.5 minutes to read 64 GB of data."

Methodology 2 try to fix the bottleneck issue by use more disks and that fix the bottleneck issue in the disks but unfortunately it create another bottleneck for the CPU. According to [6] "with a 10 Gbps network interface, it will take about 1 minute to read 64 GB of data"

Methodology 3 try to fix completely the recovery issue by terminating any bottleneck in the process. After the crash happed and during the recovery phase RAMCloud split the recovery data to partitions and assign each partition to a different recovery master. Accodring to [6] “With 100 recovery masters operating in parallel, 64 GB of data can be transferred over a 10 Gbps network in less than 1 second”.

Scattering Log Segments

According to [6] RAMCloud master recovery quickly if it distributed uniformly through all the backup cluster servers. Some will prevent this methodology:

- Segment placement should be in different racks to avoid the top rack switch failer that effect the whole rack.
- I/O different bandwidth because we have different backup servers in the data center. How many disks in each server, differ storage class and different storage speed. In the same period the data should split to all types of servers.
- Write operation should be done simultaneously without affecting any other storage because backups have a limited buffer for a storage.
- Since the Storage cluster is changing over time by read and write operations which change the pool of distribution segment. Bottleneck problem will be possible for the coordinator to make a decision.

Failure Detection

RAMCloud have a feature of detecting the failures of the storage server.

- RAMCloud Client will try to connect to the server by sending a request to the server if the server doesn't response in amount of time the client will report to the coordinator to recover that storage server.
- RAMCloud will take the advantage if there is no activity for the RAMCloud client by making it's storage servers sending like a check RPC randomly to each other .If there is no response to that RPC the server will send a notice to the coordinator. Then the coordinator will send a RPC to that server if there is no response in the timeout period the coordinator will recovery that server because it is crashed already. According to [6] "The probability of detecting a crashed machine in a single round of pings is about 63% for clusters with 100 or more nodes; the odds are greater than 99% that a failed server will be detected within five rounds. In either case, server failures are reported to the coordinator".

Recovery Flow

RAMCloud coordinator is the responsible of handling the recovery process. The coordinator divide that process into three steps:

- Setup: in this step the coordinator will search about all log segment for that crashed server then it will decide which recovery master should be recovered

- Replay: in this step the log segment will be fetched and the partitions will be added to the recovery masters.
- Cleanup: in this step the recovered server will handle the request and response to it normally and the log segment will be cleared from the storage servers.

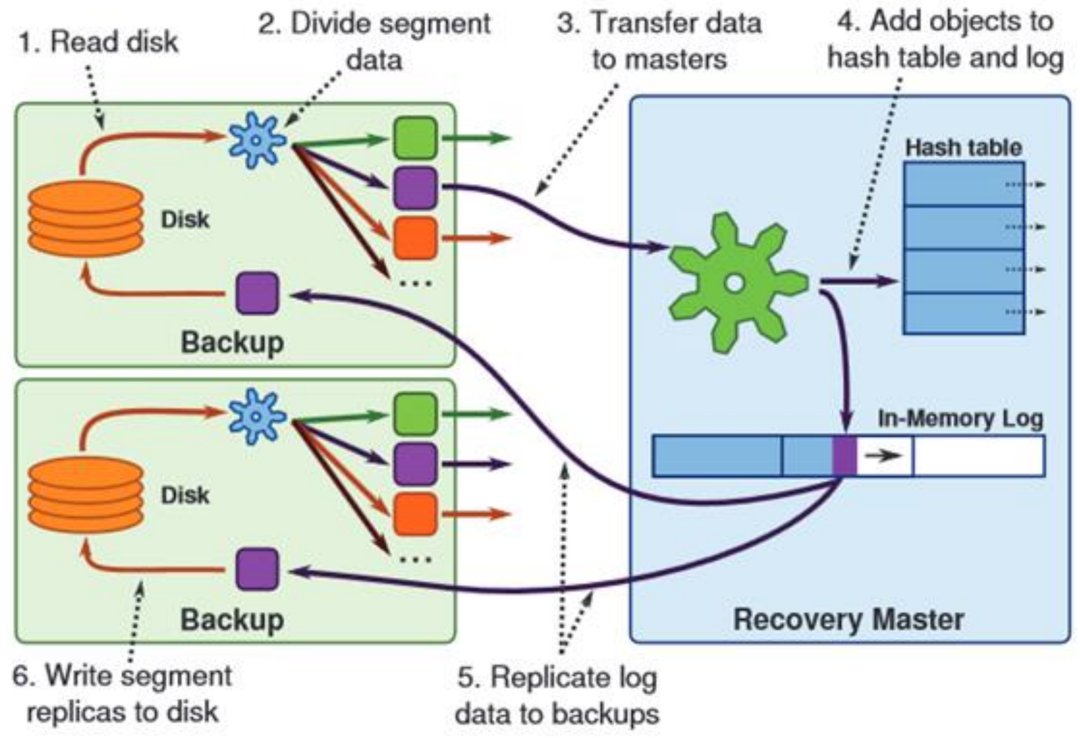

1. Read disk
2. Divide segment data
3. Transfer data to masters
4. Add objects to hash table and log
Hash table
Disk
Backup
Disk
Backup
In-Memory Log
Recovery Master
6. Write segment replicas to disk
5. Replicate log data to backups

Figure 4: During recovery, segment data flows from disk or flash on a backup over the network to a recovery master, then back to new backups as part of the recovery master's log

Toward Common Patterns for Distributed, Concurrent and Fault-Tolerant Code

According to [7] the concept of DCFT programming which mean "distributed, concurrent, and fault-tolerant" is one of the hardest programming mission for programmers. Responsibility is huge and the one code mistake can make a disaster before it can be resolved. That why it become hard to programmer to keep the availability and the consistency of the application during the system life.

Obviously, RAMCloud development team facing to many challenges and issues for example

- Conduct and figure out which content should be recovered when a crash or system failure happen.
- How many replicates need for a distribute log in case of one replica fail because of a system crash or failure.
- How to handle the algorithm of Publishing the membership of clusters that include response data, detecting the inconsistencies and log distribution.
- Asking for a storages and resources.

Since there is no standard for DCFT design patterns, development team almost lost and each programmer implementing his way without any convention.

However, in case of fault-tolerance, a developer need to think twice to handle the control flow for the unpredictable events. And the problem getting harder when the cluster in the real production environment keep changes it states. As a result, the development team implement a rule based code that can handle all major states of the RAMCloud clusters.

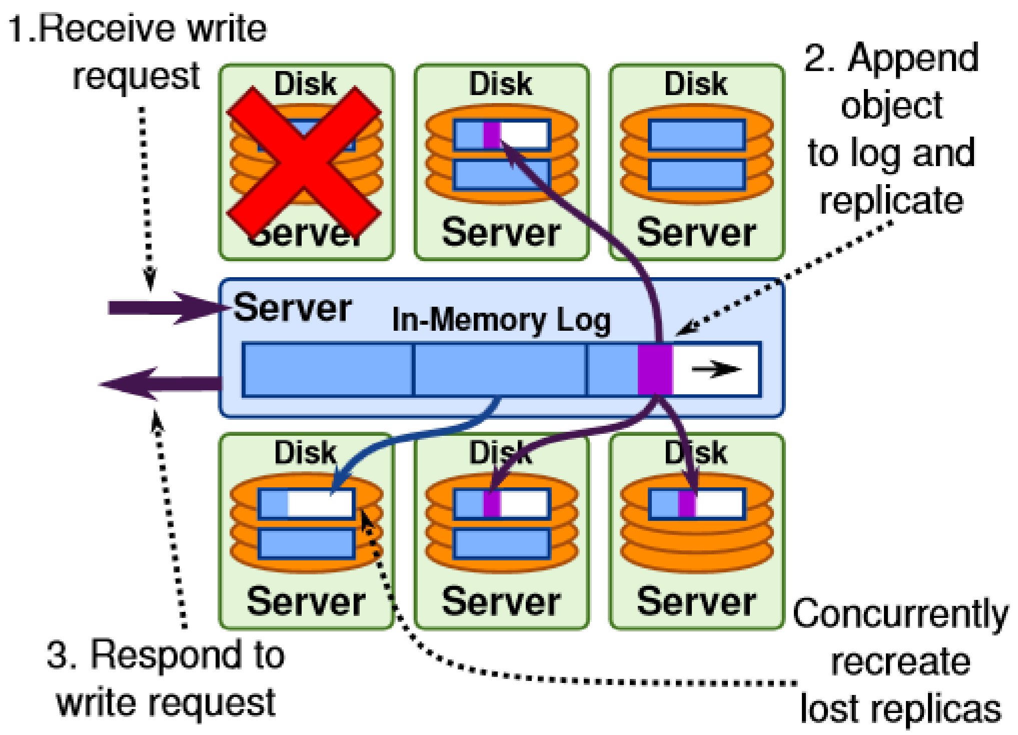


Figure 3: shows how to do a write operation by insert a data in the In-Memory log. And that shows how we lost a replica if the system have a crash or fail.

Following a rule based programming to handle the fault-tolerance raise some challenges like synchronize thousands of servers and instances which also are independent from each other. Ordering the tasks in a pool will support the rule based programming and make it effective. According to [7] “We believe a similarly prescriptive approach to DCFT programming will result in better code and higher developer productivity”.

Redis vs. RAMCloud

Data Model

According to [8] Redis is better than RAMCloud in the data model, because it provides the following services:

- Service1 : “Atomic increment“.
- Service2 : “Treat values as sets: add to set, remove from set, union, etc”.
- Service3 : “Treat values as sorted sets: which can be used to order the elements”.
- Service4 : “Treat values as lists: push, pop, index, range, etc”.
- Service5 : “Treat values as bit strings (e.g. count "on" bits)”.
- Service6 : “Treat values as hash tables”.
- Service7 : “Transactions involving multiple operations”.
- Service8 : “Publish-subscribe”
- Service9 : “Expiration times for objects”.

Performance

RAMCloud is much faster than the Redis epically in the very fast network. RAMCloud can achieve 5ms for read and 15 ms for write. While Redis can achieve almost 200 ms for only read operation which make the Redis slower than the RAMCloud and have more latency.

Scalability

Redis from the beginning it only support single server deployment. For multiple servers we need to split it first to partitions then deploy it to the servers with these conditions:

- Hash and the partition will be supported by Redis.
- Having a partitions with the Radis system mean that we need to deploy a proxy to redirect all requests to the correct servers which make some effect to the latency
- Redis doesn't support multi object transaction in case we have deploy the Redis servers in a partitions.
- Scaling Redis make it weak and bring many issues to the system like the data model and the performance.

However, RAMClould work perfectly with the salability, thousands of servers better than hunders in the RAMCloud data centers. No worries about the performance for the RAMCloud as a matter of fact if we increase the size of the data center the RAMCloud will be enhanced. Like the methodology 3 in the recovery section , as many servers we have as fast the recovery of a crashed server we will get.

Durability

Redis suffering from durability because:

- Even that Redis supports a snapshot operation and logging operation, the data still losing after any crash happened. And both operations will not help because logging need to be synchronized as soon as the fails happen. However, flushing the data like be in a flush mode will cause a huge performance penalty.

- Redis build like the master slave replications. It is not synchronized and therefore the data will be lost if there is a crash or system fail.
- Redis facing many issues with scaling like the management issue of how to manage a separately data objects.

RAMCloud enhanced with the durability:

- Automatic replication, crash recovery, and fail-over
- without effective the performance, data is always replicated and durable before requests return.
- Scalability doesn’t introduce any management issue.

Consistency

According to [8] RAMCloud doesn’t affect the client behavior visibility because it support the linerizable semantic for handling operations. Redis doesn’t seem to have any linearizability.

RAMCloud vs. memcached

Recently, many DBAs use the memcached to improve their overall database access and performance. Memcache keep recent quires result which avoid the delay in every time it request. This technique and product are very important but have a four disadvantages:

- To have durability and availability in a place we update and edit the database, however this method is slow. According to [2] “RAMCloud performs writes 100-1000x faster than a database”.
- Memcache developers or programmers suffering to apply the consistency. The consistency mean when any data altered ,updated or inserted in the database, that mean every cached or temporary memory deal with that data should be updated too. This special area of programming have a lot of bugs and exceptions that happen during the runtime execution. However, RAMCloud manage all these consistency issue and developers don’t need to worry about it.
- Memcache is fast if it has 100% cache hit rate. If memcache have only 1% cache miss rate that will drop memcache by 10 times compared to RAMCloud. The reason is, when memcache does have a cache miss that mean the data will be in the database and searching and updating memcache with that data will cost much of latency penalty. In fact almost all large systems that running memcache have more than 1% miss rate. According to [2] “in a 2013 NSDI paper, Facebook reported miss rates of 0.053%, 1.76%, 6.35%, and 7.85% on four representative memcached servers”.
- One of the greatest principal of the RAMCloud is that enhancing and keep the latency as low as possible. Having the advantage of the high speed networking inside the datacenter is also one of the main characteristic of RAMCloud. Accodring to [2] “Facebook reported typical memcached

latencies of several hundred microseconds" which is in RAMCloud is only 5 ms.

Limitations of RAMCloud

RAMCloud is not recommended for every system, some cases RAMCloud is not the best solution for that particular case. Such that:

- If your system doesn't manipulate intensively with the data. That mean you cost your organization extra money to put the put the data in an expensive DRAM storage server without even used it efficiently.
- If your system is batch oriented type, that mean you need the performance at the begging when you read all the data. After that your system doesn't worry about the latency issue then Spark system may is the best choose for you because it concern about the throughput better than the RAMCloud which is fit for a high response time applications.
- Secondary indexes and multi-object transactions doesn't supported by RAMCloud yet. RAMCloud handle the indexing by a key value that have some extension. RAMCloud team project is working on add such a high level features and make it built in the RAMCloud product.

Conclusion

RAMCloud build based on DRAM storage servers. This will provide an amazing performance. The goal of the RAMCloud team is to achieve a large scaled storage servers with a very low latency.

According to [5] RAMCloud apply some special techniques to get these amazing result such as ” a uniform log structured mechanism for managing all storage, a networking layer that bypasses the kernel to communicate directly with the NIC using a polling approach, and an approach to availability that substitutes fast crash recovery for online replication”.

RAMCloud with all these capabilities become 1000 x faster than the disk storage. RAMCloud designed to support more than 10,000servers.

RAMCloud project team has a great unified goal that I believe in too, According to [5] “Our ultimate goal for RAMCloud is to enable new applications that could not existpreviously. We do not yet know what those applications will be, but history suggests that large performance improvements are always followed by exciting new applications that take advantage of the new capabilities. As RAMCloud and other low-latency storage systems become widely available, we look forward to seeing the applications that result”.

Bibliography

[1]K. Chang, D. Lee, Z. Chishti, A. Alameldeen, C. Wilkerson, Y. Kim and O. Mutlu, 'Improving DRAM performance by parallelizing refreshes with accesses', *2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*, 2014.

[2]J. Ousterhout, 'Deciding Whether to Use RAMCloud - RAMCloud Project - Confluence', *Ramcloud.atlassian.net*, 2015. [Online]. Available: https://ramcloud.atlassian.net/wiki/display/RAM/Deciding+Whether+to+Use+RAMCloud. [Accessed: 12- Mar- 2015].

[3]J. Ousterhout, 'RAMCloud - RAMCloud Project - Confluence', *Ramcloud.atlassian.net*, 2015. [Online]. Available: https://ramcloud.atlassian.net/wiki/display/RAM/RAMCloud. [Accessed: 12- Mar- 2015].

[4]S. M. Rumble, A. Kejriwal and J. Ousterhout, 'Log-structured Memory for DRAM-based Storage', in *12th USENIX Conference on File and Storage Technologies (FAST '14).*, Santa Clara, CA USA, 2015.

[5]J. OUSTERHOUT, A. GOPALAN, A. GUPTA, A. KEJRIWAL, C. LEE, B. MONTAZERI, D. ONGARO, S. JIN PARK, H. QIN, M. ROSENBLUM, S. RUMBLE and R. STUTSMAN, in *The RAMCloud Storage System*, 1st ed., 2014.

[6]D. Ongaro, S. M. Rumble, R. Stutsman, J. Ousterhout and M. Rosenblum, *Fast Crash Recovery in RAMCloud*, 1st ed. Stanford University, 2015.

[7]R. Stutsman and J. Ousterhout, Toward Common Patterns for Distributed, Concurrent, Fault-Tolerant Code, 1st ed. 2015.

[8]J. Ousterhout, 'Redis vs. RAMCloud - RAMCloud Project - Confluence', *Ramcloud.atlassian.net*, 2015. [Online]. Available: https://ramcloud.atlassian.net/wiki/display/RAM/Redis+vs.+RAMCloud. [Accessed: 14- Mar- 2015].

[9]S. M. Rumble and J. Ousterhout, 'Glossary of RAMCloud Terms - RAMCloud Project - Confluence', *Ramcloud.atlassian.net*, 2015. [Online]. Available: https://ramcloud.atlassian.net/wiki/display/RAM/Glossary+of+RAMCloud+Terms. [Accessed: 14- Mar- 2015].