ISSN 2515-8260 Volume 07, Issue 08, 2020

Live Migration Of Virtual Machines With Software Defined Storage

V.A.Prabha[1] and Dr. P. Varalakshmi[2]

[1,2]*Department of Computer Technology, MIT Anna University, Chennai, India*

shrudhavishal@gmail.com

Abstract

Cloud providers takes the upper hand of server virtualization technology for efficient resource utilization, scalable architecture, on-demand provisioning and reduced energy consumption, etc. The key feature of server virtualization is the live migration of virtual machines (VMs).Data Centre Administrators highly depend on the most powerful technique to migrate or evacuate VMs across physical hosts. Our research work provides an insight into the intricacies of the optimized method of live migration using Software Defined Storage solution using Ceph,KVM hypervisor and the Openstack platform. The Openstack deployment of Ceph based storage solution based on Filestore storage backend has many issues related to overhead of journaling traffic and metadata. In this paper we perform a heuristic approach in choosing the VMs for migration using dynamic threshold algorithm. We also perform a detailed analysis with benchmarks and heavy workload to compare the impact of Filestore and Bluestore software defined storage during live migration of virtual machines. We include the results of the total migration time by each VM flavors and the total downtime. The results indicate which storage backend of the SDS helps in the long run to a cloud provider

Keywords: Virtualization, Cloud computing, Live Migration, Software Defined Storage & Ceph.

1. INTRODUCTION

Cloud being the most sizzling popular expression captures a paradigmatic transference in the business industry as the Datacenters are more powerful in hosting computing resources, application development platforms and softwares. Cloud administrator uses the migration techniques when the resources are overloaded and load-balanced by migrating VMs to underutilized physical machines and VMs evacuation from a server which is in imminent failure and requires maintenance, hardware or software upgrades. In this paper we consolidate the idle VMs and optimize the resource utilization by performing live migrations based on our dynamic threshold algorithm. Optimum VM placement means placing the VM close to the storage cluster to reduce network and I/O latency. Our research on VM live migration focuses mostly on optimizing live migration by using our algorithm to reduce resource usage or by reducing the VM downtime.

Openstack middleware controls compute, storage and network resources in a datacenter, that are accessible through a self-service web portal which enables the cloud administrators/users to resource provisioning/de-provisioning through a dashboard. Openstack provides persistent storage in two ways: Cinder - a block storage service and Swift-an object storage service. Non-shared mode of VM migration is also provided by Openstack. Live-migration techniques can be performed both using shared storage and non-shared storage in Openstack.

Compute nodes sharing a common shared storage for the ephemeral disk of the VMs performs live migration with the help of libvirt API and nova components. Block live migration is where VMs ephemeral disk can be migrated without any shared storage using Nova but it takes more time as the entire disk has to be copied from source to destination over network thereby increasing load over network and storage. Volume-backed live migration is boot from volume VMs in Cinder without any shared storage.

Based on the metrics collected, the algorithm initiates the live migration of VMs which includes the resource discovery by the nova conductor and analyze the compute nodes with shared storage solution. When the resource well suited for migration is selected, the state of VM is stored in the destination nova conductor.VM is transferred to the destination nova compute node and is resumed from its new location with zero downtime. This is impeded with the shared software defined storage system, Ceph a unified storage system that offers a highly scalable, self-healing, deduplication feature with high fault tolerance. It also provides location independence with specific physical paths. Ceph offers splendid scalability from petabytes to exabytes. But there are some challenges with the Ceph Filestore backend as it uses journaling that writes the data whose performance is masked for larger writes and hence the throughput gets decreased.

The live migration process in Openstack is uncontrollable and it prevents from tuning the migration parameters to meet the specific performance needs. There are user specific requirements to have control over the migration downtime. There are specific problems like Migration Convergence in pre-copy phase as it merely depends on network bandwidth of the cloud. In this paper we focus on a heuristic approach to choose the VM to be migrated based on a dynamic threshold algorithm considering RAM/CPU utilization, VM migration network traffic along with the unified scalable low cost storage solution that benefits the cloud administrators to have a controllable auto-scaling environment.

2. Related Work

Dong-Yun Lee [1] presented an analysis on the write behaviors of the storage backends of Ceph using the Write Amplification Factor and have identified that the write overhead increases by 13x factor in Filestore, Bluestore and Kstore. They finally conclude that Bluestore out shows a stable performance in Solid State Drives(SSD) and Hard disk Drives(HDD) whereas FileStore outperforms on SSDs.

Yaser Mansouri et al [2] presented an optimal offline and online algorithm that provides an optimal cost analysis method of Dynamic data replication and migration in cloud. This algorithm uses dynamic and linear programming techniques (Integer Linear Programming(ILP)) and a randomized algorithm stationed on Fixed Receding Horizon Control(FRHC) to accomplish dynamic VM migrations. Migration is initiated when the cost saving is estimated from the current time slot and the lost cost savings aggregated from the previous migration that is presumed to be equal/large to the migration cost was the perceptive idea behind this algorithm.

Anita Choudhary et al [3] presented a critical survey on the live migration techniques in which they classify the types, VM migration techniques using duplication mechanisms and performance metrics like data transferred, total migration time. They also discuss on the security threats in live migration and the mitigation solution.

Konstantinos et al [4] studied about Migration in an IaaS Cloud in a shared-nothing scenario using time-specific windows that accomplishes the task on time without affecting SLAs. A special file system called MigrateFS was used to develop a scalable distributed broker that performs replication and syncing virtual disks during live migration. Policies on Resource allocation are performed be a coordinating Migration scheduler and network brokers that are distributed.

Tin-Yu Wua el al[5] proposed a solution of prediction using dirty memory in cloud that can solve the problem on long duration in migration in all methods - pre-copy, time-series based pre-copy and post-copy. They proposed the pre-copy approach that uses Markov prediction model to define the dirty rate as Writable Working Set(WWS) .This approach credited the appropriate time by stopping the source VM and performed the WWS migration at the downtime.

Ryan el al[6] presented a Rapid Asymmetric Maximum(RAM),a hash-less chunking method that provides high-throughput that declares the cut points using bytes values instead of hash values. This algorithm formalizes the cut point that utilizes a fixed-size/variable-sized window to find the maximum-valued byte. The cut point is included in the chunk and is located at the chunk boundary.

Nikos el al[7] proposed an partitioning algorithm using hyper-graph for placement of VMs in edge computing systems that minimizes the network overhead and constraints on account capacity. It resulted in a network overhead that equals the minimum cut weight by partitioning the graph into two.

Zheng Yan el al[8] proposed a management scheme of heterogenous data storage that offers both role based access control and de-duplication for heterogenous multi-cloud service providers. Attribute-based encryption offers a set of attributes that provides user identity and performs data encryption based on access structure that has two divisions like cipher text policy ABE(CP-ABE) and key-policy ABE(KP-ABE).

Keitaro el al[9] presented an adaptive software defined storage named SuperCell for cloud Storage workloads that is least bothered on the design and configuration but focuses on the continuous measurement driven process. Hippodrome does not scale as it is not based on Cloud solution.

Samer el al[10] proposed VMFlockMS which is a pioneer migration service, specifically optimized to transfer groups of correlated VMs in cross-cloud platform, executing multi-tier applications. VMFlockMS uses the high-throughput data de-duplication algorithm to reduce the migration time by limiting the data volume transferred over the network. Wang and Yogi projected 3 topics live-migration, Planning on migration and improving quality and resilience of services.

Mehiar el al[11] proposed a framework for over-committed clouds that offers an integrated, energy-efficient, resource allocation. A heuristic approach using Integer Linear Program (ILP) and a resource predictor module is used that outperforms existing overload avoidance techniques. It also reduces the number of overloads that are unpredicted, migration overheads using prior VM migration strategies , thereby the resource utilization is increased and cloud energy consumption is minimized.

3. Technology Overview

The core service model of cloud includes Infrastructure as a Service(IaaS),Platform as a Service(PaaS) and Software as a Service(SaaS). IaaS offers users the computing resources that includes network, storage and also provides a base platform to the PaaS model that runs framework for application development and deployment. Virtualization being the base of cloud middleware enables multiple virtual machines (VMs) creation in a physical host. The software hypervisor achieves this by abstraction of the underlying host's physical resources into isolated virtualized environments, thereby enabling concurrent execution of guests on a single host. Live migration imposes transfer of the memory, CPU, network and disk state. Shared Storage between the hosts circumvents the transfer of the disk state in the live migration process. The 3 phases involved in transferring memory state: push phase, pull phase and stop and copy phase. The pure stop-and-copy migration technique is used by the hot and cold VM migration approaches.

Kernel Virtual Machine(KVM),a full virtualization hypervisor that is implemented as kernel modules by transforming the Linux kernel into a bare-metal hypervisor thereby allowing unmodified OSes to run as VMs. The motherboard processors Intel and AMD has different processor-specific loadable kernel module (kvm_amd.ko and kvm_intel.ko) that provides the core virtualization infrastructure. The KVM kernel component is exposed as the virtualization extensions to the user-space as a character device /dev/kvm. QEMU(Quick Emulator) is an emulator that provides hardware virtualization through dynamic binary translation. It is combined with KVM to provide near-native performance with the help of virtualization extension in Intel/AMD. Libvirt is the API support library for KVM management that enables these tools or applications to create/destroy guests, automate management tasks and configure virtual storage and networks. Overall, the entire virtualization capability is the aggregation of KVM,QEMU and libvirt.

Openstack in general supports 3 types of migration: Non-Live Migration (Cold Migration),Hot Migration, Live Migration. Operators use live migration mostly to avoid VM downtime but the Nova compute service in Openstack does not use the libvirt live migration functionality by default, as guests are suspended before migration and experiences several minutes of downtime. The downtime duration can be estimated from the RAM size of the disk to be migrated and is calculated using exponential backoff between each migration step. The maximum downtime duration starts off small and later increases gradually as the nova compute service wait for migration to complete. This gives the guest a chance to complete the migration successfully, with a minimum amount of downtime.

Nova Live Migration consists of the following stages:

- Pre-live migration
- Live-migration operation
- Post-live migration

Pre-live migration actions

The source node compute1 requests the destination node compute2 to perform pre-live migration actions with a synchronous RPC call. The neutron REST API provides VM's ports information to compute2. Then, the vif driver is called to create the VM's port (VIF) using plug_vifs().Since we use Open switch hybrid plug, the network agent running in compute2 will detect this new VIF based on the information from neutron server and configure it accordingly. However, port's status won't change, since this port is not bound to compute2.

Compute1 updates the Neutron ports binding: profile with the information of the target host. The port update RPC message sent out by Neutron server will be received by neutron-l3-agent2, which proactively sets up the DVR router. If pre-live migration fails, nova rollbacks and port is removed from compute2. If pre-live migration succeeds, nova proceeds with live-migration-operation.

Live-migration operation

After the pre-live migration actions, compute1 starts the live-migration by creating the VM and its corresponding tap interface on compute2.The neutron-l2-agent2 will detect this new tap device and configure it. As soon as the VM is active on compute2, the original VM and its respective tap-device on compute1 gets removed. There is no rollback if failure happens in live-migration operation stage.

Post-live-migration actions

Once live-migration succeeded, both compute1 and compute2 perform post-live migration actions and send a RPC cast to compute2 to tell it to perform post-live migration actions.Compute2 sends a REST call "update_port(binding=host2, profile={})" to the neutron server to update the port's binding which clears the port binding information and move the port's status to DOWN. The modular layer (ML2) plug-in which is a framework for layer 2 technologies then try to rebind the port according to its new host. This update_port REST call always triggers a port-update RPC fanout message to every neutron-l2-agent that takes this message into account and re-synchronize the port by asking the neutron server details about it through RPC messages. This will move the port from DOWN status to BUILD, and then back to ACTIVE.

Internals of Ceph Storage

Ceph provides different modes of storage services

- Ceph object storage
- Ceph block storage
- Ceph file system

RADOS (Reliable Autonomic Distributed Object Store) provides an exceptional high-scalable storage service for data in the form of objects and consists of several daemons with specific task. Ceph consists of two types of daemons: Ceph Monitor(MON) and Ceph Object Storage(OSD).The key design of RADOS is that the OSD's are able to operate with relative autonomy during recovery from failures. The Ceph monitor (MON) daemon is responsible for monitoring, managing the cluster map which provides cluster-wide node information. Ceph object and block storage communicates directly with the OSD daemon where the primary data copy resides. CRUSH(Controlled Replication Under Scalable Hashing) is a pseudo random hash-based algorithm that decides the placement of objects at different locations in a distributed storage cluster thereby eliminating the need for a centralized server.

Ceph provides 3 different storage backends: Filestore, Bluestore and KStore. Bluestore is the latest storage engine for Ceph which is represented as a block with a new store that has a raw block device and a small key-value database(RocksDB) to store the metadata. There are policies defined by pluggable block allocator and different compression methods. The limitation with Filestore is the XFS filesystem that does not support the atomicity property/transactions that is mandatory for the write sync operation. Filestore does double-write thereby decreasing the performance and it requires fast performing SSD to put the journal.

4. Implementation

We have implemented live migration of VMs with shared Ceph storage on the KVM hypervisor and Openstack and have identified the sequence of events in the VM migration process in both the scenarios. We have analyzed the performance of the migration phases with varying VM factors, workloads and analyzed the performance. Migration of a VM is influenced by 3 factors-running specific application such as read-intensive, write-intensive and memory-intensive.

Performance degradation of a VM that is running memory-intensive applications is due to network traffic, downtime, and latency.VM executing read-intensive applications can reduce the downtime and adverse effects on application performance using the pre-copy technique.VM running Write-intensive application does not compete well with the pre-copy technique as a large number of memory pages are frequently dirtied.

This paper focuses on a heuristic approach in VM selection to perform live migration based on a dynamic threshold algorithm in the auto-scaling procedure. The weightage is assigned based on the different states. We have collected the metrics from the monitoring tool Hyperic and FIO command line utility which focuses on the CPU/RAM usage and VM network migration traffic

Pseudocode

- Rmin = Average % of RAM for t interval
- Cmin = Average % of CPU for t interval
- RAM = Current RAM usage in %
- CPU = Current CPU usage in %

  **Note : Eg (t=10 Minutes)

State Definition

State 0 if (current RAM & CPU value < Minimal RAM & CPU) → Destroy VM as it is idle for long time

State 1 if (current state of RAM||CPU < 25)&& (Network Traffic <VM_{net})→ Assign a weightage W1 that
allows migration

State 2 if (25 < current RAM||CPU < 70) → Assign a weightage W2 that prevents migration

State 3 if (current RAM||CPU > 70) → Create a New VM

RAM/CPU Monitoring algorithm

Compute the current RAM||CPU % (for t interval)

case (State 0): Inform Load Balancer to TERMINATE – (TERM IP)

case (State 1): Inform Load Balancer to Change Weight as W1 – (WEIGHT W1) and initiate migration

case (State 2): Inform Load Balancer to Change Weight as W2 – (WEIGHT W2) and prevent migration

case (State 3): Inform Load Balancer to Create VM – CREATE (CREATE)

Load Balancing Algorithm

```
Begin
if socket message = CREATE VMj
for each Vms of all nodes do
    report all RAM||CPU
```

```
if ((26<current(RAM||CPU) <70))
        Initialize VM = NO
if Creation Condition = YES
        Initialize a new VM
end
        end
end for
if socket message = TERMINATE
          Destroy VM
end
if socket message = WEIGHT W2
        Construct vm with new weight and reload
end
if socket message = WEIGHT W1
for each all Vms except VMj do
         Get Minimum RAM||CPU of VMi
end for
end
if host machine(RAM||CPU(VMi)+RAM|| CPU(VMj) is greater than 70
           Construct  migration from VMj to VMi

Initialize weight W2 to VMi and reload
else
            Migrate Vmi to VMj
reinitialize weight W2 to VMj and reload
 end if
 else
reinitialize Weight of the VM and reload
 end
```

5. Experimental Setup

Two experimental test-beds are implemented to investigate the live migration process with our dynamic threshold migration algorithm carried out between the 2 storage backend engines of Ceph-Filestore and Bluestore using KVM/QEMU hypervisors deployed in an Openstack environment. There are two aspects common to both these test-beds. These are the shared storage implemented to support live migration and a measurement setup for migration traffic analysis. Each node configured with 4 OSDs (HDDs), 3 monitors and 3 managers. The configurations of the hardware and software are as follows

6. Experimentation Results

Ceph version (12.2.1) is configured in one setup with Ceph Filestore as the backend comprising of 2 OSDs and a 5G journaling filesystem for each OSD. Ceph Luminous version (12.2.4) is used in another setup with Bluestore as the backend engine with 2 OSDs, a database RocksDB to manage internal data such as metadata and a WAL device that is used for Bluestore's internal journaling and write-ahead log in the same partition. Storage pool was created with 4096 placement groups and 3x replication. Performance is tested with 70 RBD images with different flavors, on a 3x replicated pool, 27TB of total data.FIO tool was used to measure the 4KB random block performance against the Rados Block Driver.

- Small (2 VCPUs, 4GB RAM, 40GB DISK)
- Medium (4 VCPUs, 8GB RAM, 80GB DISK)
- Large(8 VCPUs, 16GB RAM, 160GB DISK)

- XLarge(8 VCPUs, 16GB RAM, 250GB DISK)

We have analyzed 4 flavors of different configurations with Ceph Bluestore and Filestore with load and without load where the X-axis depicts the different flavors and Y-axis depicts the migration time in seconds. It clearly states that Bluestore outperforms Filestore and the Migration Time is considerably less using our optimized migration algorithm.

The figure 8 shows that Bluestore migration time is 5 seconds with XL flavor VM whereas Filestore is 18 seconds..The test is conducted 20 times and the average migration time is estimated .The downtime of Ceph Filestore rapidly increases with the increase in size of the VM.

Figure 10 and 11 depicted a comparison of 4KB random write throughput and IOPS block performance measured using FIO and it showed significant increase in throughput and latency using Bluestore and Filestore. There is a 18% increase in throughput and 5%average latency in Bluestore.

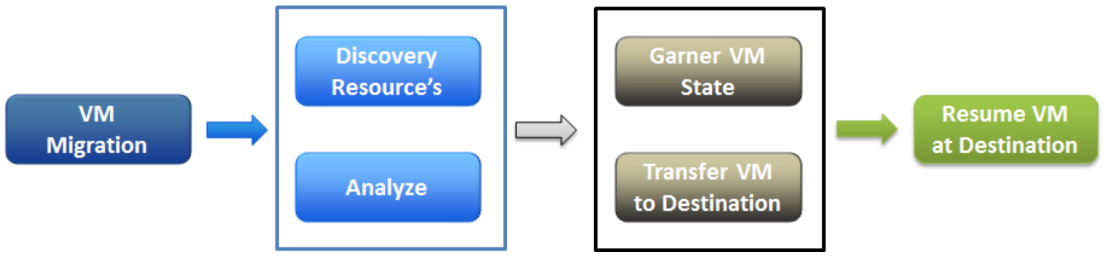

Figure 1: Logical steps of VM Migration

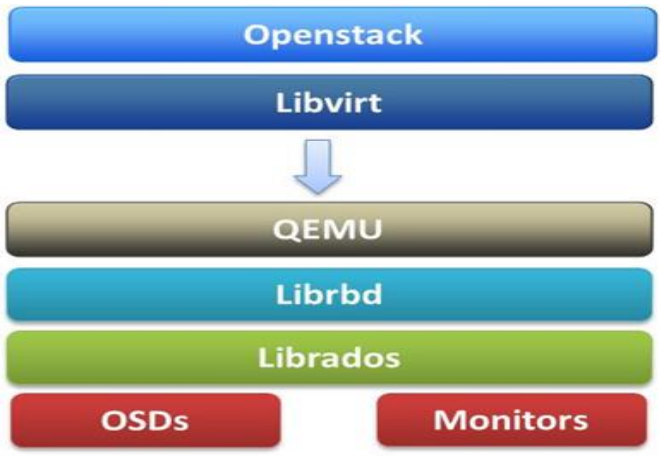

Figure 2: Integration of Ceph with Openstack, QEMU and Libvirt

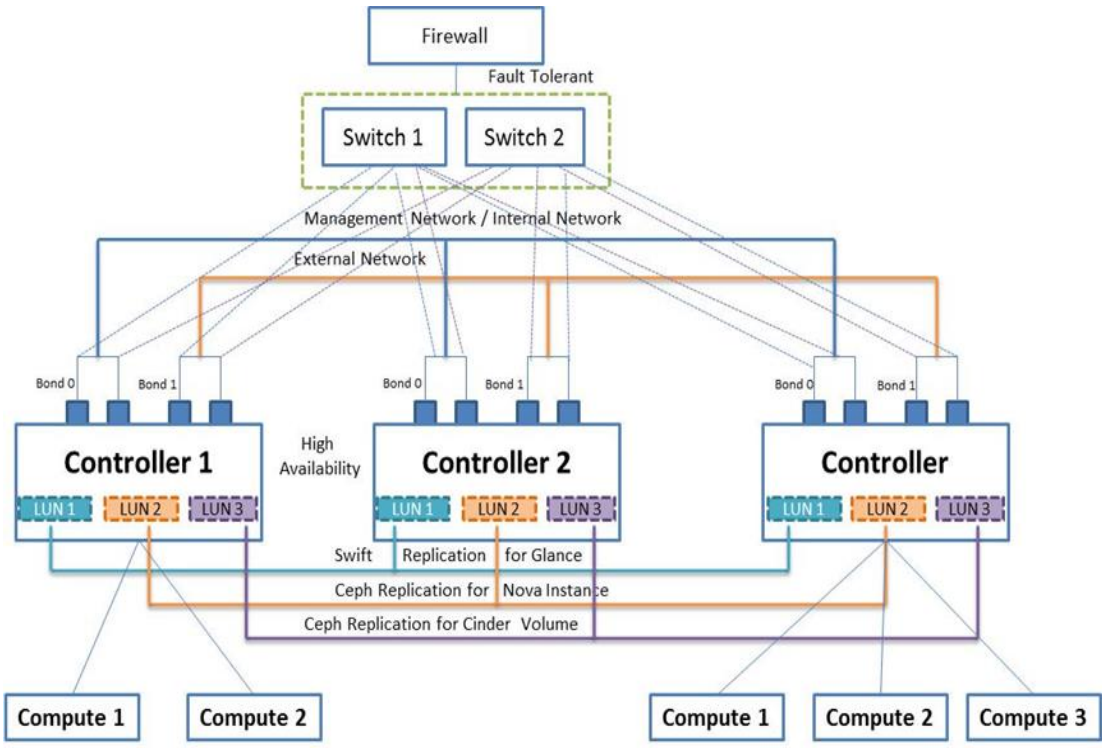

Figure 3: The cloud deployment architecture

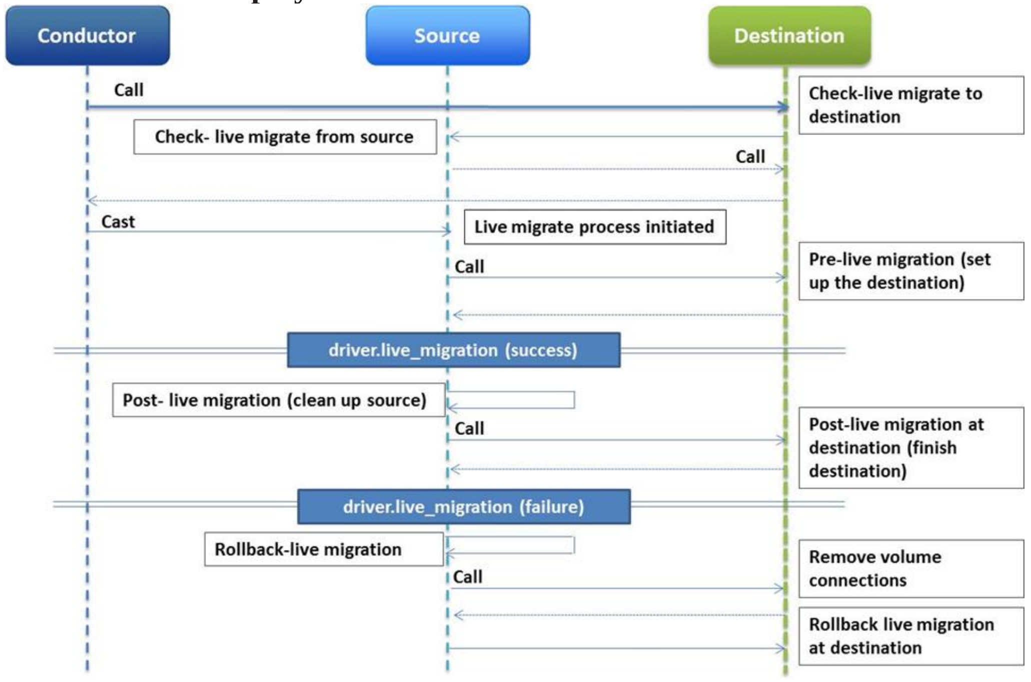
Conductor
Source
Destination
Call
Check-live migrate to destination
Check- live migrate from source
Call
Cast
Live migrate process initiated
Call
Pre-live migration (set up the destination)
driver.live_migration (success)
Post- live migration (clean up source)
Call
Post-live migration at destination (finish destination)
driver.live_migration (failure)
Rollback-live migration
Call
Remove volume connections
Rollback live migration at destination

Figure 4: Live migration process in Nova

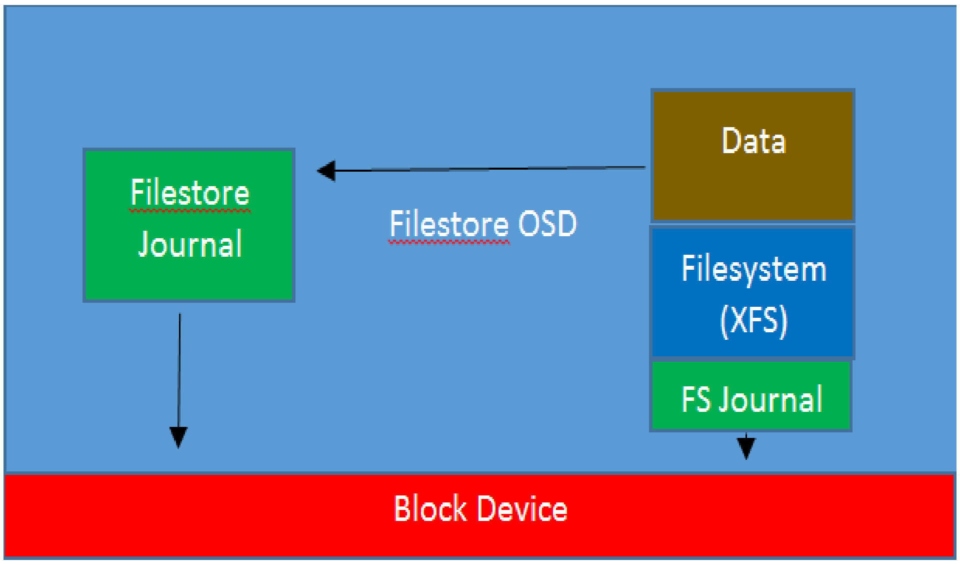

Figure 5: Ceph FileStore Backend

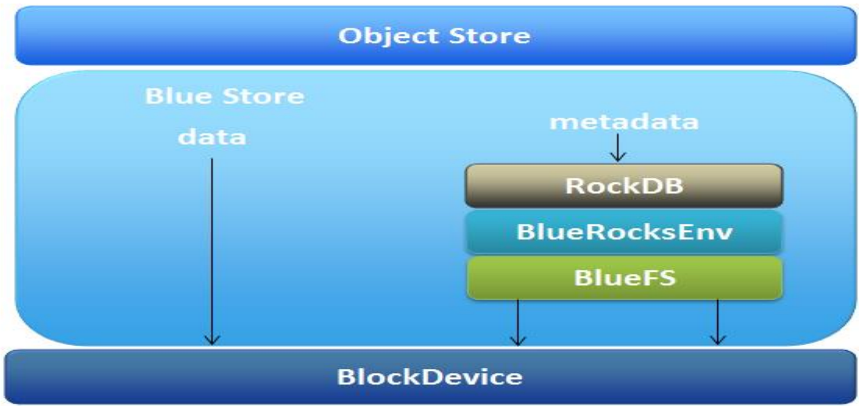

Figure 6: Ceph Bluestore Backend

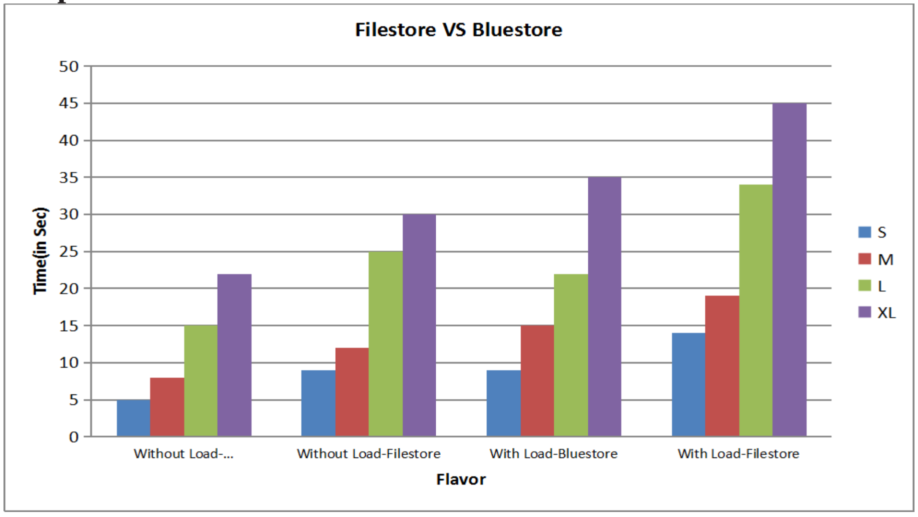

Figure 7: Ceph Bluestore vs Filestore with/without load

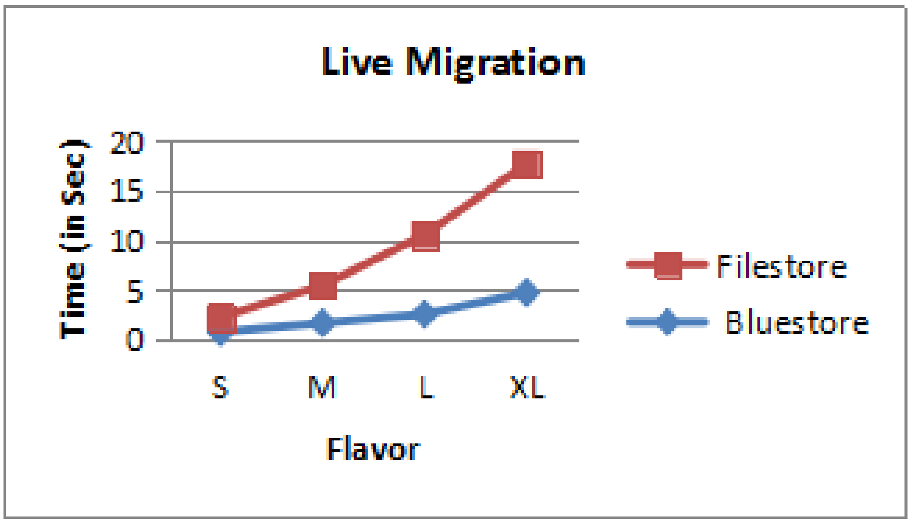

Figure 8: Live Migration Time

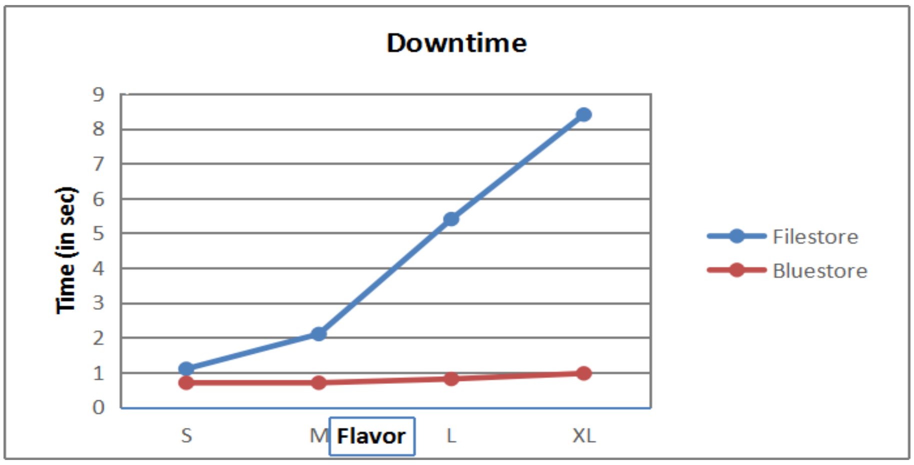

Figure 9: VM Downtime

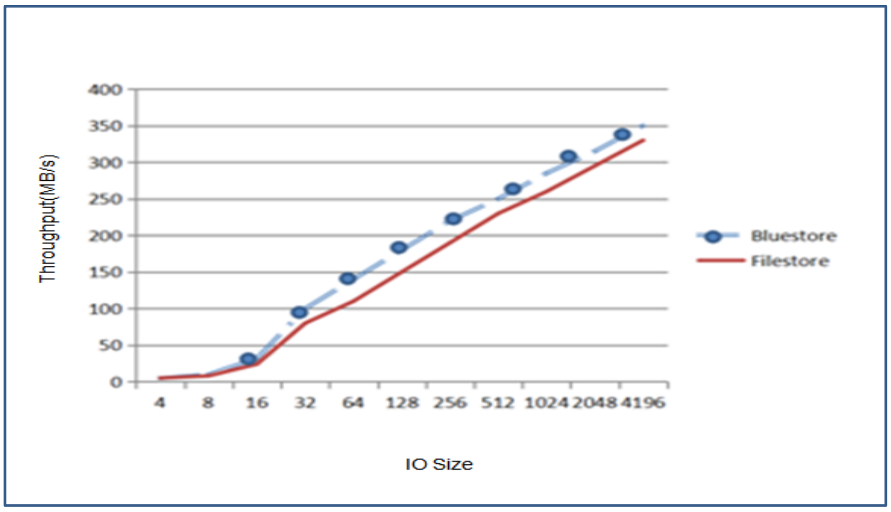

Figure 10: Bluestore vs Filestore HDD Random Write Throughput

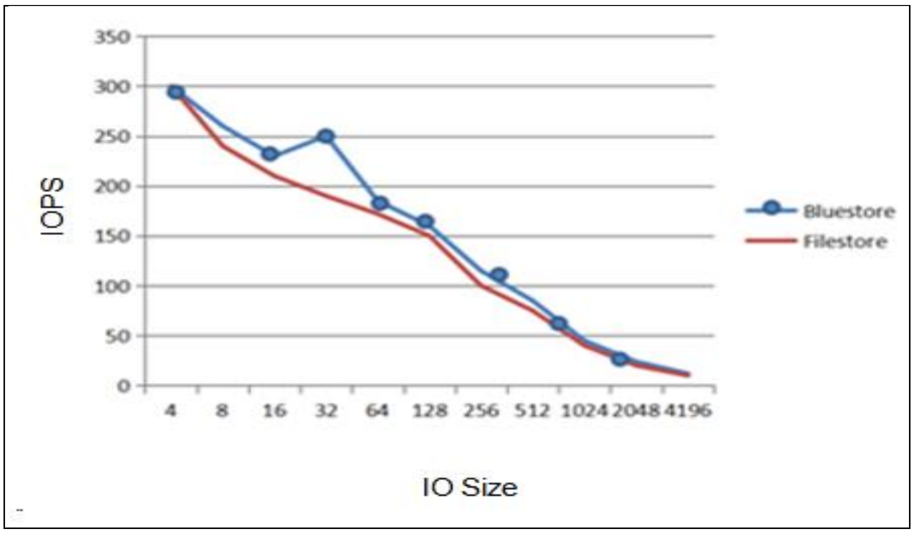

Figure 11: Bluestore vs Filestore HDD Random Write IOPS

Table 1: VM Network Traffic Analysis

VMmem	VM Memory Size
B	Network Bandwidth
d	Dirty rate
β	Dirty Rate/Bandwidth
VMnet	VM network traffic = VMmem * β

Table II. Ceph Bluestore/Filestore Hardware Testbed Details

Openstack Controllers	3 VMS of 4 vCPU cores,16GB RAM,200GB HDD
KVM hosts	Dell PowerEdge Tower server Intel Xeon E3-1230 V2 @3.30GHz 256GB memory, 1TB disk
Ceph Storage	4 LUNS of 27 TB of Dell EMC Storage configured with Bluestore and Filestore with 2 LUNS each
Gigabit Ethernet Switches	L3 Blade switches with 1 Gbps connectivity between the devices

Table III. Ceph Bluestore/Filestore Software Testbed Details

Operating System	64-bit BOSS 7.0 version operating system
Hypervisor	Libvirt version -3.0.0 and QEMU version -2.8.1
Openstack	Queens version
Ceph	Luminous version
Monitoring Tool	Hyperic, FIO

Table IV. Ceph Bluestore and Filestore Migration Time

Flavor	**Using Ceph Filestore (in Seconds)**		**Using Ceph Bluestore (in Seconds)**	
	Without Load	**With Load**	**Without Load**	**With Load**

Small	9	14	5	9
Medium	12	19	8	15
Large	25	34	15	22
XLarge	30	45	22	35

7. CONCLUSION AND FUTURE WORK

In this paper we have analyzed the impact of Ceph storage backend (Bluestore and Filestore) during the live migration focusing on the different VM sizes and random read and write operations. Filestore triples the write traffic due to its external journaling file system. Bluestore stores the data in the raw device and avoids the external journaling file system and hence does not suffer IOPS issue. Hence we conclude that Bluestore provides a promising throughput and latency. The algorithm uses the threshold based technique to predict when to initiate the live migration. Our future work includes a machine learning algorithm to predict the load and when to initiate the VM migration over a WAN network.

REFERENCES

[1] Dong-Yun Lee, Kisik Jeong,Sang-Hoon Han,"Under standing Write behaviours of Storage backends in Ceph Object Store" in National Research Foundation of Korea

[2] Y. Mansouri, A. Nadjaran Toosi and R. Buyya, "Cost Optimization for Dynamic Replication and Migration of Data in Cloud Data Centers," in IEEE Transactions on Cloud Computing.doi: 10.1109/TCC.2017.2659728

[3] K. Tsakalozos, V. Verroios, M. Roussopoulos and A. Delis, "Live VM Migration Under Time-Constraints in Share-Nothing IaaS-Clouds," in IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 8, pp. 2285-2298, 1 Aug. 2017. doi: 10.1109/TPDS.2017.2658572

[4] Tin-Yu Wu, Nadra Guizani, and Jhih-Siang Huang. 2017. Live migration improvements by related dirty memory prediction in cloud computing. J. Netw. Comput. Appl. 90, C (July 2017), 83-89. DOI: https://doi.org/10.1016/j.jnca.2017.03.011

[5] Ryan N.S. Widodo, Hyotaek Lim, and Mohammed Atiquzzaman. 2017. A new content-defined chunking algorithm for data deduplication in cloud storage. Future Gener. Comput. Syst. 71, C (June 2017), 145-156. DOI: https://doi.org/10.1016/j.future.2017.02.013

[6] N. Tziritas et al., "Data Replication and Virtual Machine Migrations to Mitigate Network Overhead in Edge Computing Systems," in IEEE Transactions on Sustainable Computing, vol. 2, no. 4, pp. 320-332, 1 Oct.-Dec. 2017.doi: 10.1109/TSUSC.2017.2715662

[7] Z. Yan, L. Zhang, W. Ding and Q. Zheng, "Heterogeneous Data Storage Management with Deduplication in Cloud Computing," in IEEE Transactions on Big Data. doi: 10.1109/TBDATA.2017.2701352

[8] K. Uehara, Y. Xiang, Y. R. Chen, M. Hiltunen, K. Joshi and R. Schlichting, "SuperCell: Adaptive Software-Defined Storage for Cloud Storage Workloads," 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Washington, DC, 2018, pp. 103-112. doi: 10.1109/CCGRID.2018.00025

[9] Samer Al-Kiswany, Dinesh Subhraveti, Prasenjit Sarkar, and Matei Ripeanu. 2011. VMFlock: virtual machine co-migration for the cloud. In Proceedings of the 20th international symposium on High performance distributed computing (HPDC '11). ACM, New York, NY, USA, 159-170. DOI=http://dx.doi.org/10.1145/1996130.1996153

[10] M. Dabbagh, B. Hamdaoui, M. Guizani and A. Rayes, "An Energy-Efficient VM Prediction and Migration Framework for Overcommitted Clouds," in IEEE Transactions on Cloud Computing, vol. 6, no. 4, pp. 955-966, 1 Oct.-Dec. 2018. doi: 10.1109/TCC.2016.256440.