This is a post-print version of the following article: J. Riley, J. Noss, J. Cuff, I. M. Llorente, “A High-Availability Cloud for Research Computing”, IEEE Computer, pp: 91-95, Issue No. 06 - June (2017 vol. 50)

A High-Availability Cloud for Research Computing

June 2017

Justin Riley, John Noss, Wes Dillingham, James Cuff, Harvard University

Ignacio M. Llorente, OpenNebula

Abstract

The design of a common reference architecture for all cloud instances, their integration with the existing infrastructure and processes, and the automation of the operations and maintenance processes were the three critical components for the success of the private cloud strategy at Harvard FAS Research Computing. This article describes the lessons learned, challenges faced and innovations made in the design and implementation of the cloud architecture and operations to guide those organizations that are transitioning to private cloud and automated infrastructure.

1. Introduction

As part of the Faculty of Arts & Sciences (FAS) Division of Science at Harvard University, Research Computing (RC) facilitates the advancement of complex research by providing leading edge computing services for high performance and scientific computing, bioinformatics analysis, visualization, and data storage. Since 2008, Harvard FAS RC has undertaken a significant scaling challenge increasing their available High Performance Computing (HPC) and storage from 200 cores and 20TB to over 70,000 cores and 35PB.

In addition to providing bare-metal access to large amounts of compute, FAS RC also builds and fully maintains custom virtual machines (VM) tailored to faculty and researchers needs including lab websites, portals, databases and project development environments. Recently FAS RC defined a private cloud computing strategy to convert its legacy internal KVM based infrastructure from a completely home-made virtualized cluster based on scripted tools to a more robust, reliable and automated private cloud system. They then configured the system with public cloud integration, to improve agility, increase resource utilization, and enable implementation of further advanced services.

The first step was to design a cloud reference architecture with high availability, multi-tenancy, orchestration and provisioning features to provide a common frame of reference for all the private cloud instances. This provided a foundation for further development and innovation,

and helped make well-founded strategic and technical decisions as we adopted private cloud computing. The architecture design provides APIs and features that both help us serve users more efficiently and improve our internal processes for testing new system configurations and dynamically spinning up resources for continuous integration and deployment.

In order to ensure the success of the cloud strategy, it was also necessary to plan and automate the operations and maintenance processes of the clouds and integrate them with existing datacenter and configuration management tools. The cloud infrastructure deployment is fully automated and has been used to provision three fault-tolerant cloud infrastructures with a multi-tiered backup system and robust VM and virtual disk monitoring: a “core” cloud infrastructure to simplify the internal management and enhance resilience of hosted services; a “science” cloud to allow members of the scientific community at Harvard to lease resources in a manner similar to existing public clouds; and a “testing” cloud provisioned the same way to allow testing and staging changes and to experiment with IaaS-style cloud computing.

2. Architecture Design for High Availability

High Availability (HA) is the ability for the cloud to continue functioning after failure of one or more of the hardware or software components [NaTK16]. This ultimately means that there is no single point of failure and the cloud service does not go down and continues serving requests even if a system component fails. HA has been achieved by incorporating specific features, such as redundancy for failover and replication for load balancing, in each of the system components without the need for costly specialized hardware and software.

Our reference architecture design adopts a classical cluster-like organization [MoML12] with a controller host, a set of hypervisors where VMs will be hosted, a storage cluster and at least one physical network joining all of the hosts. The architecture has been implemented with OpenNebula [SMLF10] as the orchestration manager, Ceph [WBML06] as the storage cluster, KVM as the hypervisor, and Microsoft Azure and Amazon Web Services as two public clouds for cloud-bursting. The architecture was designed to be functional across two datacenters to enable live-migration of running VMs between them for load balancing, or in case of maintenance or issues.

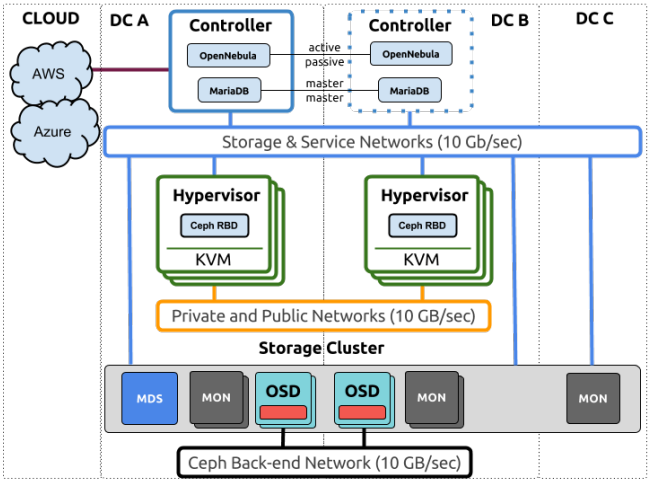

Figure 1. Internal private cloud architecture across two datacenters with access to remote public clouds.

Two controller hosts run an HA active-passive setup of the OpenNebula cloud manager and an HA active-active configuration of a MariaDB database cluster. The OpenNebula front-ends are deployed with fencing mechanisms so that if one OpenNebula instance fails, the node is put offline. The IP address is acquired by the passive node so that the cloud service is not disrupted. The MariaDB cluster keeps all of the data related to the objects of the cloud, like hosts, networks, VMs, users, etc. synchronized between the controller nodes so that in the case of failure of one node all of the data is still reachable.

Hypervisor hosts are responsible for providing VMs with execution resources, such as CPU, memory, and network access. The hypervisor nodes, based on KVM, can be configured through OpenNebula with automated VM restart in order to prepare for failures in the VMs or hosts, and recover from them. If one of the physical hosts fails all of its VMs can be re-deployed on another host. If a VM crashes, the VM is restarted.

Cloud storage is provided by a dedicated Ceph cluster. The data placement algorithm CRUSH is utilized to ensure two replicas per object are placed on a unique host in each of the two datacenters containing Ceph object storage devices (OSD). This replication strategy ensures that the Ceph cluster can tolerate the loss of an entire datacenter without going into a read only state and can additionally tolerate the loss of a single host in the surviving datacenter before losing data. Ceph's monitor's (MONs), which form a quorum and negotiate consensus of cluster state are placed two-per-datacenter alongside the OSDs and a fifth in a third datacenter. This isolated MON enables the quorum to be maintained in case of link-loss between datacenters or single datacenter failure. In addition to providing block devices, Ceph's POSIX filesystem, CephFS, is utilized to provide storage for OpenNebula's files datastore. The CephFS metadata server (MDS) is made highly available via a standby node in each datacenter.

The storage system was thoroughly tested under various failure scenarios (failed disks, loss of network, etc.) and under high load before going into production. Moreover, enough spare capacity for maintenance events was configured on controllers, hypervisors, and storage hosts. This enables us to live-migrate VMs off of a hypervisor or take down Ceph OSDs/MONs for maintenance without service downtime.

Each host (storage nodes, hypervisors and controllers) has two LACP (802.3ad) bonded 10GbE NICs connected to two switches for HA. The cloud makes use of several VLANs (802.1q) for service, storage, public and private communication purposes. It is possible to enforce Security Groups over VMs. Using Security Groups, administrators can define the firewall rules and apply them to the Virtual Machines, defining permitted inbound/outbound traffic.

3. Cloud Deployment and Operation

The deployment and operation of our reference architecture involved several challenges related to the integration of the new private cloud environments into the existing infrastructure within FAS RC, the monitoring of real-time disk I/O from each running VM, the complete backup of the cloud, and the automation of the operation and maintenance processes.

The integration of the clouds with the existing infrastructure required writing both custom software and OpenNebula "hooks" in order to meet our internal requirements for backups,

monitoring, alerting, DNS, etc. For example, we developed “onedns” to provide a dynamic DNS server that is OpenNebula-aware, created a VM “hook” to prevent OpenNebula from clobbering existing physical systems on shared VLANs, and wrote tools such as “ceph-rsnapshot” for our multi-tiered back-up system. Along the way we discovered and documented procedures for supporting various types of non-trivial VMs such as license servers that require specific MAC addresses in order to function properly.

The Ceph cluster is monitored using the “Ceph-dash” open source utility, a flask-based API and web dashboard for monitoring the cluster’s overall status and Nagios alerting. Diamond collectors are also used for metric collection at the cluster and individual component level and Graphite/Grafana are used for metric storage and graphing. One of the pain points of our legacy KVM infrastructure was not having per-vm disk I/O statistics making it hard to identify VMs with high I/O load. The Diamond collector for Ceph was extended to query the OpenNebula frontend for VMs on the hypervisors and then ship metrics from Ceph’s admin socket to Graphite for each VM root disk. This provides a VM leaderboard to show real-time disk I/O from each running VM without relying on a client running inside of the VM. This enables FAS RC administrators to quickly detect VMs that are negatively impacting the storage cluster.

The clouds utilize an in-house developed backup process utilizing Ceph’s per-block-device snapshot mechanism. Each day the current snapshot is compared to the previous day’s snapshot and all differing objects underlying the block device are exported to a separate backup cluster where they are applied to a sister block device, thereby providing point-in-time consistent incremental snapshots on an isolated cluster. A script exports Ceph RBD devices from this backup cluster to qcow and ships them to our backup filesystem in a separate datacenter. This helps to minimize the load on the production cluster when converting RBD devices to qcow2 backups. In addition to backing up the OpenNebula DB, the OpenNebula templates are also exported hourly for all VMs as xml files, so they can be referenced or used for a recovery.

We manage OpenNebula and Ceph with Puppet in order to integrate them with existing configuration management tools at FAS RC. We extended an existing OpenNebula Puppet module to automatically provision OpenNebula clusters with two front-end OpenNebula controller nodes and hypervisors, which can optionally be auto-added to the cluster via Puppet exported resources. This enables us to easily provision and manage three separate OpenNebula clouds with one toolset, as well as provides a common place for configurations with the rest of FAS RC infrastructure.

4. Science Cloud Resource Provisioning

The Science Cloud infrastructure will serve multiple user groups (such as project, department, lab or research areas) within Harvard. An on-premise private cloud in a large organization like Harvard requires powerful and flexible mechanisms to manage the access privileges to the virtual and physical infrastructure and to dynamically allocate the available resources.

The planned resource provisioning model defines a Virtual Data Center (VDC) for each user group, dynamically allocates resources according to its needs, and assigns resource quotas. The internal administration of the group will be delegated to an internal group administrator. VDCs will be completely isolated from each other. Each user group will have access to a private image

catalog specific for its research field that can be easily consumed through a self-service portal. This model will evolve and be improved incorporating additional features to meet user and FAS RC needs.

5. Towards a Data-centric Approach to Cloud Computing

Recently the New England Storage Exchange (NESE) project was funded by the National Science Foundation to build a multi-petabyte object store that will be designed, built and supported by the same University partners that founded the Massachusetts Green High Performance Computing Center (MGHPCC). This storage cloud infrastructure will also be tightly integrated with the cloud environments at FAS RC in order to provide research faculty with on-demand services for data-intensive applications supporting the new Harvard-wide Data Science Initiative.

As cloud infrastructures are increasingly used to process large volumes of data, we will have to rethink their architectures to focus on data storage and management. This means a paradigm shift in cloud design from computational to data exploration that requires infrastructures to be data-centric in order to minimize data movement by bringing compute closer to data within the architecture. The implementation and operation of data-driven cloud services present new challenges and opportunities for future cloud infrastructures.

The description of our private cloud strategy can serve as a framework to help understand and successfully address the architectural, operational and technical challenges when building and operating a new private cloud environment. The architecture design, the operation processes and the selection of the appropriate software components for a cloud environment depend on the user needs, current virtualization environment, the availability of skilled resources, and the budget. Our main recommendations are to completely integrate the new cloud environments into the existing data center infrastructure and processes, and to automate the operation and maintenance processes of the clouds in order to gain infrastructure efficiency and agility.

References

[MoML12] R. Moreno-Vozmediano, R. S. Montero, I. M. Llorente. “IaaS Cloud Architecture: From Virtualized Data Centers to Federated Cloud Infrastructures,” IEEE Computer, 45(12):65-72, December 2012.

[SMLF10] B. Sotomayor, R. S. Montero, I. M. Llorente, I. Foster. “Virtual Infrastructure Management in Private and Hybrid Clouds. Internet Computing,” IEEE Internet Computing, 13(5):14-22, September 2010.

[NaTK16] M. Nabi, M. Toeroe, F. Khendek. “Availability in the cloud: State of the art” Journal of Network and Computer Applications, Vol. 60, pp. 54-67, 2016.

[WBML06] S. A. WeilS, A. Brandt, E. L. Miller, D. D. E. Long, C. Maltzahn, “Ceph: a Scalable, High-performance Distributed File System”, OSDI '06 Proceedings of the 7th Symposium on Operating Systems Design and Implementation, 2006