V-LAB REPORT FOR FALL 2014

REPORTER: CHUN-JEN CHUNG

ADVISOR: DIJIANG HUANG

SECURE NETWORKING AND COMPUTING (SNAC) RESEARCH GROUP

SCHOOL OF COMPUTING INFORMATICS & DECISION SYSTEMS ENGINEERING (CIDSE)

12/19/2014

ARIZONA STATE UNIVERSITY

Table of Contents

List of Tables

List of Figures

1. Executive Summary

During the Fall2014 semester, the V-Lab system has served 445 unique students (76% graduate and 24% undergraduate) in 10 laboratories of 4 courses. More than 326 virtual machines (80% Linux and 20% Windows) are created to allow students to perform experiments for their laboratories in a virtualized environment, and help instructors and teaching assistants evaluate and grade the experiment results.

At the beginning of Fall2014, the development team of V-Lab incorporated several new technologies and designs to the system.

- Upgrade the OpenStack cloud operating system.
- Deploy the system with MAAS/JUJU
- Multiple cloud cluster architecture
- Ganglia monitoring system

New V-Lab system also adapted and improved features from previous version of the system:

- Faster project lookup and VM access in the V-Lab web portal
- High availability cluster architecture for RabbitMQ and MySQL

During the Fall2014 semester, V-Lab has encountered a few problems as follows.

- Lack of a single sign on (SSO) mechanism
- Bad image support for VM creation
- MTU issue for a VM to be accessed from external
- noVNC timeout issue

For the Spring2015 semester, V-Lab plans to implement and enhance the following new features to provide better services for students and public users:

- Upgrade to a new version of OpenStack
- Redesign the web portal of V-Lab System

More details are described in the rest of the report.

2. V-Lab Usage Statistics

This section presents the V-Lab system's usage information.

2.1. Supported Courses and Student Counts

There are four major CSE courses used V-Lab system in Fall2014. They are CSE 468/598, CSE545, CSE445/598, and CSE591. Two courses combines the undergraduate and graduate level classes (468/598 and 445/598) and one course includes three different class type (545 in-class, 545 hybrid, and 545 online). We consider them as different courses to show the student's distribution. During this semester, there are 445 students use V-Lab system for their homework, experiments, and course projects. 76% of students are graduate students (337) and 24% of students are under-graduate students (108). Table 2-1 shows the detail information of each course, the number of students and the student's distributions.

Table 2-1: Supported course and student counts

Course	Total Students	Grad Students	Under-graduate Students	% of graduate	% of Under-graduate	Instructor	Course Name
468/598	62	20	42	32%	68%	Dijiang Huang	Computer Network Security
545 (in-person)	57	57	0	100%	0%	Stephen Yau	Software Security
545 (hybrid)	82	82	0	100%	0%	Stephen Yau	Software Security
545 (online)	5	5	0	100%	0%	Stephen Yau	Software Security
445/598	120	54	66	45%	55%	Janaka Balasooriya	Distributed Software Development
591	119	119	0	100%	0%	Dijiang Huang	Cloud Computing
Total	**445**	**337**	**108**	**76%**	**24%**		

2.2. Supported Laboratories and Allocated Resource

Table 2-2 shows the information of each supported laboratories and the allocated resources from V-Lab system during Fall2014. The table shows the number of projects, the number of virtual machines (VMs), number of virtual CPU (vCPU), the total amount of disk space usage, and the total amount of memory used for each course. During this semester, we created 180 projects in V-Lab system to support students' experiments and allocated 326 virtual machines (80% Linux and 20% Windows) for these projects. The required resources to support these virtual machines are 460 virtual CPUs, 6730GB disk space (6.57TB), and 402GB memory in total.

Table 2-2: Supported Laboratories and Allocated Resource

Course	Lab Name	# of Projects	# of Linux VMs	# of Window VMs	# of vCPU	Total Disk (GB)	Total Memory (GB)
468/598	1). Basic Network and Services Setup 2). Firewall Setup 3). IDS/IPS and syslog setup	62	186	0	186	2790	186
545 (in-person)	1). Course Project	10	2	8	10	370	20
545 (hybrid)	1). Course Project	14	2	12	148	510	28
545 (online)	1). Course Project	5	2	3	5	180	8
445/598	1). Web-based Application Development and State Management Lab 2). Service-Oriented Software Development Lab 3). XML and Related Technologies Lab	45	0	45	45	1800	90
591	1). Cloud computing research term project	44	66	0	66	1080	70
Total		**180**	**258 (80%)**	**68 (20%)**	**460**	**6730 (GB)**	**402 (GB)**

2.3. Resource Usage

This section presents the resource usage statistics from V-lab system during Fall2014. We measured the resource usage based on 3 major resources in a cloud system: RAM, vCPU and Disk. Table 2-3 shows the memory usage for each course during this semester from September to early December (12/1~12/10). The memory resource usage is measured by RAM MB-Hours to represent the total MB-hours of using assigned RAM. Figure 2-1 shows the chart of collecting data from Table 2-3. Table 2-4 shows the CPU usage by each course in this semester and it is measured by the occupied vCPU-Hours in the cloud resources. Figure 2-2 shows the chart of collecting data from Table 2.4. Table 2-5 shows the disk space used by each course in this semester and it is measured with GB-Hours in disk resource they used. Figure 2-3 shows the chart of collecting data from Table 2.5.

Table 2-3: Memory Usage measured with MB-Hours

Course	Course Name	Sept.	Oct.	Nov.	Dec.
CSE 468/598	Computer Network Security	55728224.12	66355170.42	66425766.69	18481152.00
CSE 498/598	Distributed Software Development	167464709.12	166376548.96	156595194.88	43450368.00
CSE 545	Software Security	17479211.53	44974080.00	42792837.12	11796480.00
CSE 591	Cloud Computing	39015541.48	74299682.13	71983104.00	19857408.00
Subtotal		279687686.25	352005481.51	337796902.69	93585408.00
Total		1063075478.45			

Figure 2-1: Memory usage from each course

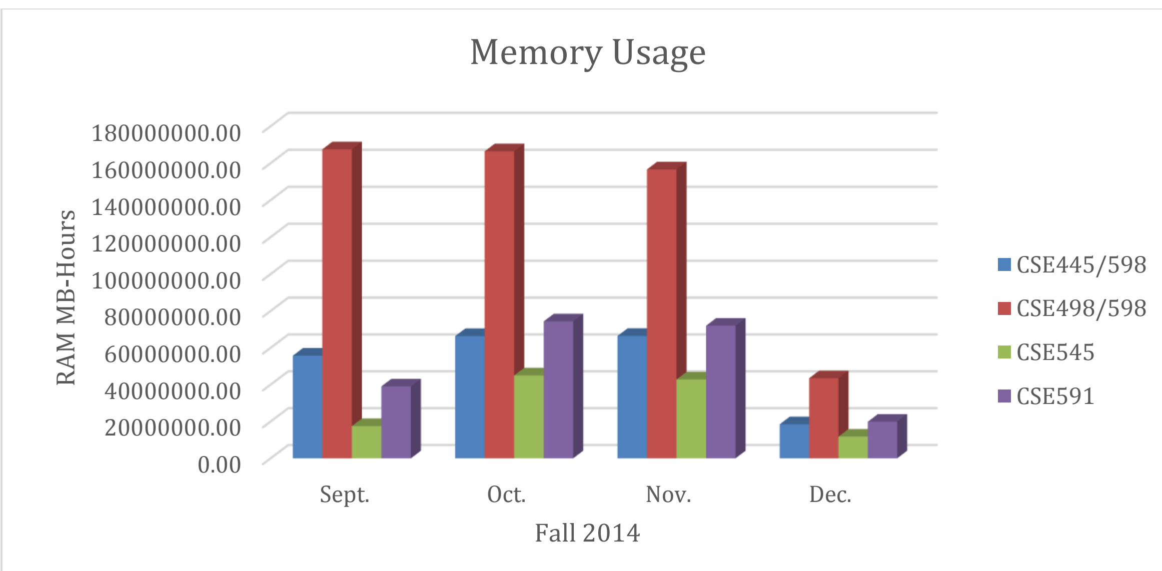


Table 2-4: Memory Usage measured with MB-Hours

Course	Course Name	Sept.	Oct.	Nov.	Dec.
CSE 468/598	Computer Network Security	40770.58	32432.76	32434.46	9024.00
CSE 498/598	Distributed Software Development	162894.99	162477.11	152925.00	42432.00
CSE 545	Software Security	8810.42	23040.00	21605.54	42432.00
CSE 591	Cloud Computing	36775.79	66887.42	64728.00	17856.00
Subtotal		249251.78	284837.29	271693.00	111744.00
Total		917526.07			

Figure 2-2: Memory usage from each course

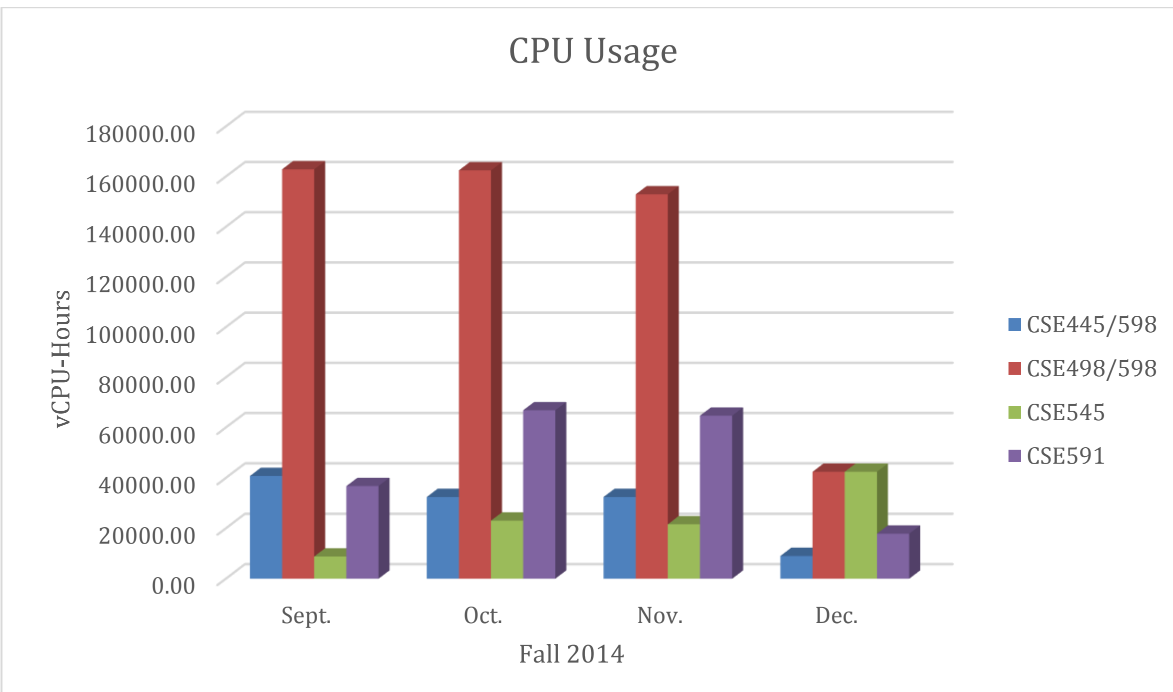
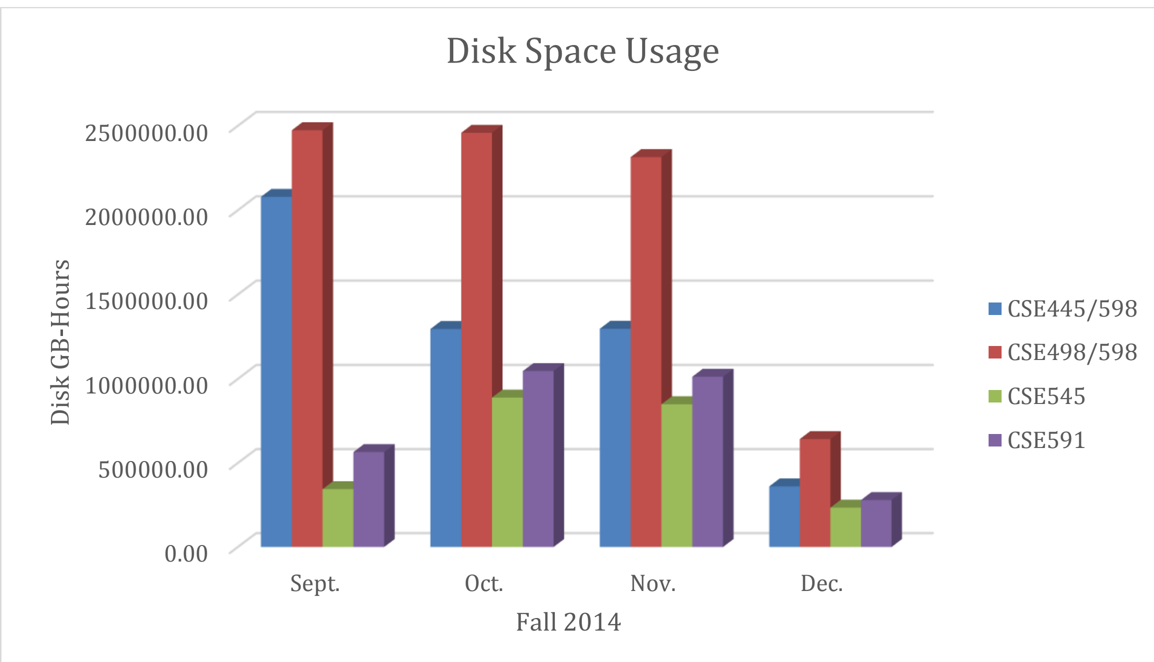


Table 2-5: Disk space usage measured with GB-Hours

Course	Course Name	Sept.	Oct.	Nov.	Dec.
CSE 468/598	Computer Network Security	2078933.90	1295999.42	1297378.25	360960.00
CSE 498/598	Distributed Software Development	2473976.33	2458756.47	2314770.85	642240.00
CSE 545	Software Security	344475.84	889200.00	849575.00	234240.00
CSE 591	Cloud Computing	564956.74	1046065.57	1012680.00	279360.00
Subtotal		5462342.81	5690021.46	5474404.10	1516800.00
Total		18143568.37			

Figure 2-3: Disk space usage from each course



3. New Features

This section presents the new features that have been developed and integrated with V-Lab system during the Fall2014 semester.

3.1. New version of OpenStack Cloud Operating System (Icehouse)

OpenStack provides an easy-to-use web front-end to plan, manage, and use resources in a cloud system. We have been used OpenStack as cloud operating system since Fall2013. The previous version of OpenStack (Grizzly) has been used 1 year in our V-Lab system. For Fall2014, we upgrade the system to the new OpenStack version – Icehouse.

Icehouse includes approximately 350 new features, the major features are described as follows:

OpenStack Compute (Nova): New support for rolling upgrades minimizes the impact to running workloads during the upgrade process. Testing requirements for third-party drivers have become more stringent, and scheduler performance is improved. Other enhancements include improved boot process reliability across platform services, new features exposed to end users via API updates (e.g., target machines by affinity) and more efficient access to the data layer to improve performance, especially at scale.

OpenStack Object Storage (Swift): A major new feature is discoverability, which dramatically improves workflows and saves time by allowing users to ask any Object Storage cloud what capabilities are available via API call. A new replication process significantly improves performance, with the introduction of s-sync to more efficiently transport data.

OpenStack Block Storage (Cinder): Enhancements have been added for backend migration with tiered storage environments, allowing for performance management in heterogeneous environments. Mandatory testing for external drivers now ensures a consistent user experience across storage platforms, and fully distributed services improve scalability.

OpenStack Networking (Neutron): Tighter integration with OpenStack Compute improves performance of provisioning actions as well as consistency with bulk instance creation. Better functional testing for actions that require coordination between multiple services and third-party driver testing ensure consistency and reliability across network implementations.

OpenStack Identity Service (Keystone): First iteration of federated authentication is now supported allowing users to access private and public OpenStack clouds with the same credentials.

OpenStack Orchestration (Heat): Automated scaling of additional resources across the platform, including compute, storage and networking is now available. A new configuration API brings more lifecycle management for applications, and new capabilities are available to end-users that were previously limited to cloud administrators. Collaboration with OASIS resulted in the TOSCA Simple Profile in YAML v1.0, demonstrating how the feedback and expertise of hands-on OpenStack developers can dramatically improve the applicability of standards.

OpenStack Telemetry (Ceilometer): Improved access to metering data used for automated actions or billing / chargeback purposes.

OpenStack Dashboard (Horizon): Design is updated with new navigation and user experience improvements (e.g., in-line editing). The Dashboard is now available in 16 languages, including German, Serbian and Hindi added during this release cycle.

OpenStack Database Service (Trove): A new capability included in the integrated release allows users to manage relational database services in an OpenStack environment.

Currently, we only allow system administrator, instructor, TAs, and advanced users accessing the cloud resources from the web front-end [1].

3.2. New Deployment strategy using MAAS and JUJU

V-Lab system is built on top of several physical servers and network devices. It's hard to deploy and manage the system and application in the system manually. We applied Ubuntu's MAAS and JUJU tools to help us for the deployment the system.

MAAS (Metal As A Service) is software which allows the administrator to deal with physical hardware just as easily as virtual nodes. Rather than having to manage each server individually, MAAS turns the bare metal resource into an elastic cloud-like resource. Specifically, MAAS allows for services to be deployed to bare metal via Juju.

Juju is a powerful service orchestration tool from Ubuntu that helps the administrator defines, configures and deploys services to any cloud quickly and easily. Juju provides both a command-line interface and an intuitive web application to design, build, configure, deploy and manage your infrastructure. Juju automates the mundane tasks allowing you to focus on creating amazing applications.

3.3. Multiple Cloud Cluster Architecture

To extend the V-Lab system into different geographical sites, we created a multiple cloud cluster architecture with the help from MAAS. We created three separate cloud site in this architecture: Mobicloud, Mobisphere, and V-Lab. Each cloud site has their own hardware and software resources, as well as the management system.

Mobicloud site provides the cloud resources for researcher or personal on-line experiment environment. Mobisphere provides the similar resources and usage, but with the smaller scale. V-Lab system is dedicated for the courses and students to create virtualized environment for their course and project. We built up a central identity, authentication, and authorization scheme using LDAP to manage the access for these three different sites.

3.4. Git-based Version Control Repository

As an extension of the MobiCloud research project, a self-developed Git-based version control repository called MobiCloud Source Forge [3] has integrated into the V-Lab system at the beginning of Fall2014 semester.

A Version Control Repository keeps track of all work and all changes in a set of source code, and allows members of a software development team to work together on the same set of code. By using our version control repository in V-Lab, students can upload their team software project's source code into our system anytime they've made a change, then they are able to keep the track of changes to the codes. The system will provide a single point of access so that all source codes are available from one location by all the students and instructors. It also allows roll back function so that users can get old versions of codes or other files any time. Git, which is the backbone of this system, is a free open source distributed version control repository initially designed for Linux kernel development in 2005. It is distributed under the term of the GNU General Public License.

A Web UI for the version control repository is also introduced into the system. This web interface provides various options like user account management and access control, project wiki site with tutorials, source code viewing and downloads, etc. Compare to the traditional command line based interface, this UI not only provides students with a more user-friendly environment but also makes the management of source code much easier and clearer. The system is currently being used by all the student team of CSE 591 Cloud Computing Class, which includes about 20 teams.

3.5. Ganglia Monitoring System

In order to monitor the overall performance of V-Lab system and health status of each server in the cloud architecture, V-Lab system has built a real-time online monitoring system using Ganglia. Administrators are able to check many system performance metrics online [6], such as CPU load average, disk usage, various parts of memory utilization, byte-level and packet-level traffic statistics, and system information from each sever.

Ganglia [7] is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids. It is based on a hierarchical design targeted at federations of clusters. It leverages widely used technologies such as XML for data representation, XDR for compact, portable data transport, and RRDtool for data storage and visualization. It uses carefully engineered data structures and algorithms to achieve very low per-node overheads and high concurrency. The implementation is robust, has been ported to an extensive set of operating systems and processor architectures, and is currently in use on thousands of clusters around the world. It has been used to link clusters across university campuses and around the world and can scale to handle clusters with 2000 nodes.

3.6. iSCSI SAN-based Storage and Backup

V-Lab system has been migrated from NFS-based storage solution to iSCSI-based storage and backup system at the beginning of Fall 2012. iSCSI is in many ways dependent on the operating system, it allows for multi-path to be configured, effectively using multiple data-links as individual paths that data may take to reach the target. By using iSCSI, the overall V-Lab system performance has been improved.

In Fall2014, V-Lab system has installed and configured iSCSI SAN-based storage and backup system with two Dell PowerVault MD3600i iSCSCI SAN Storage Array to support the storage for VMs, images, snapshots, and backups. The current storage space is around 24 terabytes and can be extended by installing additional hard drive.

4. Improved Features

This section presents improved features migrating from the previous version of the V-Lab system.

4.1. Faster Project list lookup and VM access in the V-Lab web portal

In Fall2014, we re-created a new VM resource access module in the V-Lab web portal. The new V-Lab web portal still based on Drupal to provide the web based user interface for users [2]. In the old version of V-Lab web port, the Resource Hub module was slow to retrieve the project and VM information from the backend cloud server. We re-create a new module using new APIs from OpenStack. It's faster than the old approach. The user is no longer to wait for the entire list of project displayed if they have many project.

4.2. High Availability Cluster for RabbitMq and MySQL

A crucial aspect of high availability is the elimination of single points of failure (SPOFs). In order to eliminate SPOFs, check that mechanisms exist for redundancy of (1) network components, such as switches and routers, (2) applications and automatic service migration, and (3) storage components.

V-Lab system has built a high availability (HA) architecture by clustering major components in OpenStack to provide non-SPOF services since Fall2013. The HA architecture of the V-Lab system is implemented with Active/Passive mode to clustering MySQL and RabbitMQ in the cloud controller and quantum service in the network controller.

During the heavy usage in the cloud system, we found that the Active/Passive mode of clustering is not able to process the continually incoming request message efficiently. In the Fall2014, we modify the RabbitMq service to an Active/Active mode using RabbitMq's native support and increase the node in the cluster to three nodes for increasing the capacity.

The MySQL service has also been modified into active/active mode using Percona XTRADB Cluster. Percona XtraDB Cluster is an active/active high availability and high scalability open source solution for MySQL clustering. It integrates Percona Server and Percona XtraBackup with the Galera library of MySQL high availability solutions in a single package which enables you to create a cost-effective MySQL high availability cluster [8].

5. V-Lab System Capacity

5.1. Overview

The current V-Lab system consists of 14 servers and 2 iSCSI SAN-based storage array with four major hardware models:

1. Dell PowerEdge 410 1U Rack Server
   CPU: Dual Quad-core Intel Xeon processor E5640@2.66GHz
   Total 8 cores & 16 threads
   Memory: 32G
2. Dell PowerEdge 620 1U Rack Server
   CPU: Dual Six-core Intel Xeon processor E5-2640@2.50GHz
   Total 12 cores & 24 threads
   Memory: 64G
3. Dell PowerVault MD3600i iSCSCI SAN Storage Array
4. Dell Old Server (unknown model)
   CPU: Dual core Intel Pentium 4 @3.00GHZ

5.2. Computation and Storage Resources

The computation and storage resources are two important components of a cloud system. Table 5-1 shows the physical servers providing the resource to support these two services in the V-Lab system.

Table 5-1: The list of physical servers for computation and storage resources

Role	Hostname	Model	Cores	Disk	RAM	Purpose
Nova	compute-1	R620	24	2T	128G	VM provisioning
Nova	compute-2	R620	24	1.5T	64G	VM provisioning
Nova	compute-3	R620	24	1.5T	64G	VM provisioning
Nova	compute-4	R620	24	1.5T	64G	VM provisioning
Nova	compute-5	R620	32	2T	128G	VM provisioning, VPN server
Nova	compute-6	R620	32	2T	128G	VM provisioning
Nova	compute-7	R620	24	2T	64G	VM provisioning
Nova	compute-8	R620	24	2T	64G	VM provisioning
storage1		md3600i		12.6T		iscsi-storage
storage2		md3600i		10.8T		iscsi-storage
Total physical resources			208	34.4T	704G	
Total allocatable physical resources			166.4	34.4T	563.2	
Virtual resource allocation ratio			1:16		1:1.5	
Total allocatable virtual resources			**2662.4**	**37.9T**	**844.8G**	

The number of CPU cores and the amount of memory space are two major elements to determine the capacity of the V-Lab system. In Fall2014, the V-Lab system supports 208 physical CPU cores and 704G physical RAM in total. Based on the default allocation ratio from OpenStack (CPU allocation ratio 1:16 and RAM allocation ratio 1:1.5), the V-Lab system is able to support at most **844 VMs with 1G RAM**.

5.3. Controller, Network, and other Resources

In addition to the computation and storage resources, V-Lab system also requires servers to host the control function for the system administrator and the network services for users. Table 5-2 shows the list of physical servers have installed in the V-Lab system at the beginning of Fall2014.

Table 5-2: The list of physical servers for cloud control and network services.

Role	Hostname	Model	Cores	Disk	RAM	Purpose
Cloud Controller	controller	R620	24	0.5T	64G	Cloud Controller
Networking	neutron	R620	24	0.5T	64G	Networking
Webserver	vlabwebserver	Old server	2	500G	8G	vlab web portal
RabbitMq/ MySQL	rabbitmq-mysql-1	R410	16	2T	32G	DB and Queue Cluster
RabbitMq/ MySQL	rabbitmq-mysql-2	R410	16	2T	32G	DB and Queue Cluster
RabbitMq/ MySQL	rabbitmq-mysql-3	R410	16	2T	32G	DB and Queue Cluster
Gateway	vlabgateway	R620	24	2T	64G	Gateway
Monitoring and Log Server	monitor	Old server	2	500G	8G	Monitor/Auditing

5.4. VM Templates and Resource Allocation Flavor

Each laboratory in V-Lab system consists of one or multiple virtual machines (VMs). The VMs are built based on pre-configured templates. Current system configures templates with qcow2 format (OpenStack default image format) that are available to be created for laboratories:

- Ubuntu 14.04 64-bit: 1GB Memory and 15GB Storage
- Ubuntu 13.10 64-bit: 1GB Memory and 8GB Storage
- Centos 6.0 64-bit: 1GB Memory and 15GB Storage
- Windows 7 Professional SP1 64-bit: 1GB Memory and 40GB Storage
- Windows 2008 Server R2SP1 64-bit: 2GB Memory and 40GB Storage
- Windows Server 2012 64-bit: 2GB Memory and 4GB Storage

In addition to the VM templates, the virtual CPU, disk storage, and RAM also need to be assigned by selecting the flavor in the V-Lab system. Current system defines the following flavors for allocating resources to users:

- vlab-small, 1 vCPU, 1GB RAM, 15G Disk
- m1.s-small, 1 vCPU, 1GB RAM, 20G Disk
- vlab-medium, 1 vCPU, 1G RAM, 30G Disk
- m1.s-medium, 1 vCPY, 2G RAM, 40G Disk
- m1.medium, 2 vCPU, 4G RAM, 40 Disk

5.5. Estimated Capacity for Resource Allocation

We have improved the memory issues from Fall2013 by increasing the memory capacity in each server. In Fall2014, we established the servers for computing service with 704GB RAM. Based on the information from previous sections, the current capacity of V-Lab system is determined by the amount of RAM requested by users. The total amount of RAM in the current system is 704G. If we take 20% of RAM resources for the operating system control and administrative purpose, the total allocatable RAM is around 844.8G. Based on this capacity of free RAM, the current V-Lab system is able to support at most 840 VMs with Ubuntu's basic requirement, or at most 420 VMs with Windows_64bits environment requirement to install various types of Windows OS.

6. Computation Resource Performance Analysis

In Fall2014, the V-Lab system keep using the Ganglia Cluster Monitoring System [6][7] to collect and monitor the resource utilization on each monitored server. This section presents the utilization and health status of each nova-compute servers which provide the VM provisioning and computation services for students during the Fall2014 semester. This section also discusses the performance analysis based on the information from the monitoring system.

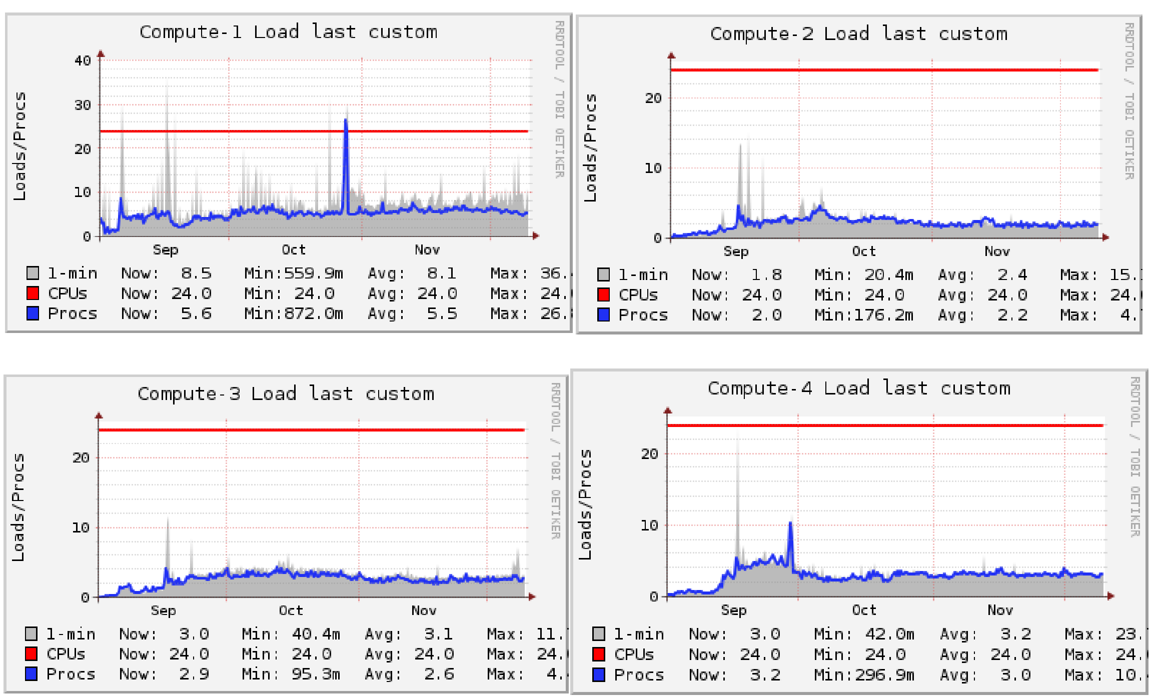
Since the monitoring system collects metrics from each server between during Sept.1.2014 ~ Dec.10.2014, the following figures show various resource utilization on the following servers during this period of time.

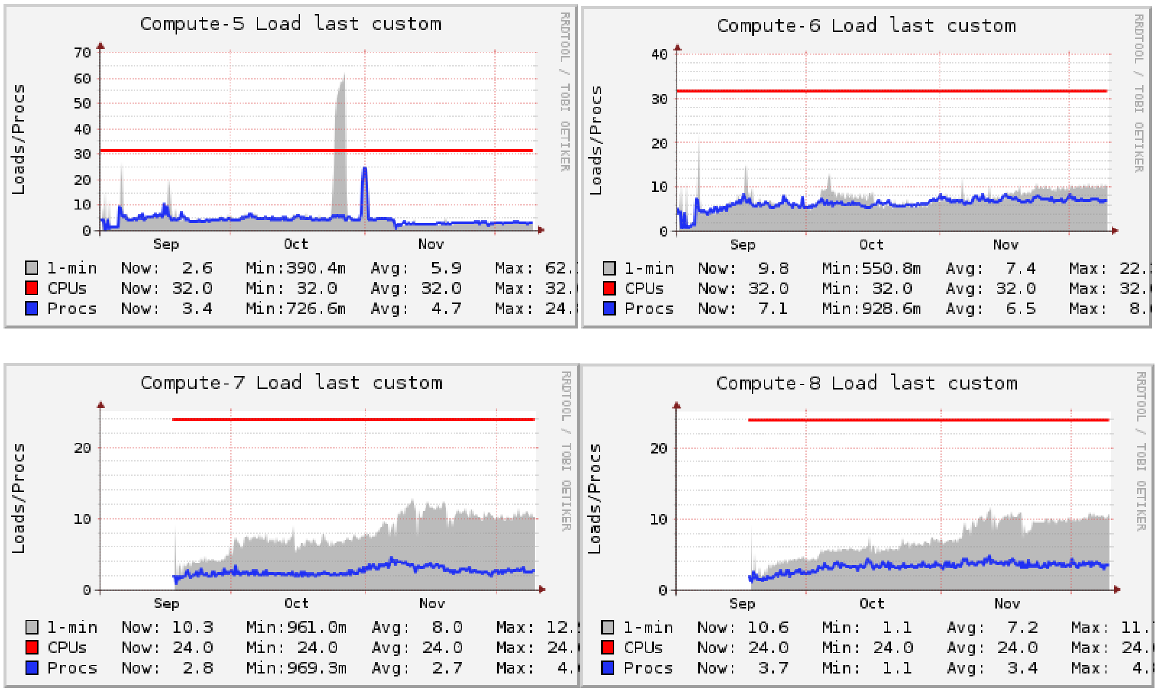
6.1. System Load Average

The system load average represents the average system load over a period of time. It conventionally appears in the form of three numbers which represent the system load during the last one-, five-, and fifteen-minute periods. Technically speaking, the load average represents the average number of processes that have to wait for CPU time during the last 1, 5 or 15 minutes.

Figure 6-1 shows the last one-min system load average of each nova-compute server during the monitoring time. Higher numbers represent an overloaded machine, and the load should not exceed the number of cores available on the system (the red line in the figure).

Figure 6-1: One-min Process Load Average



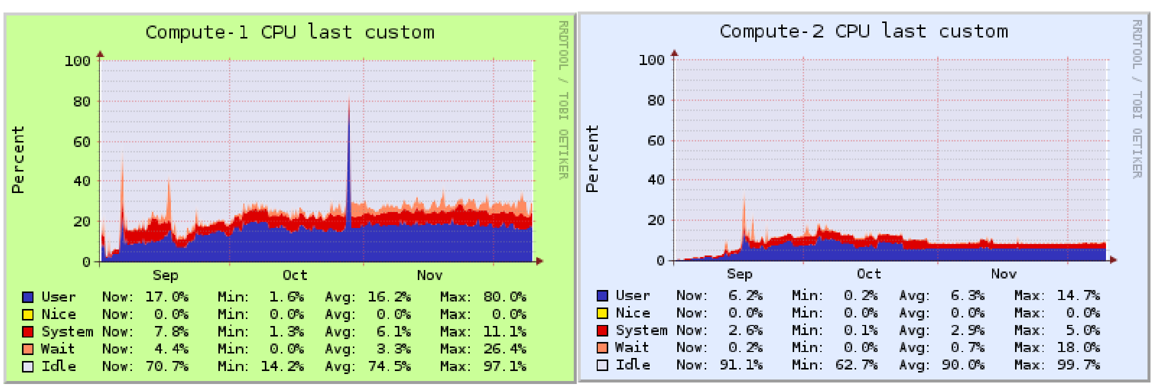


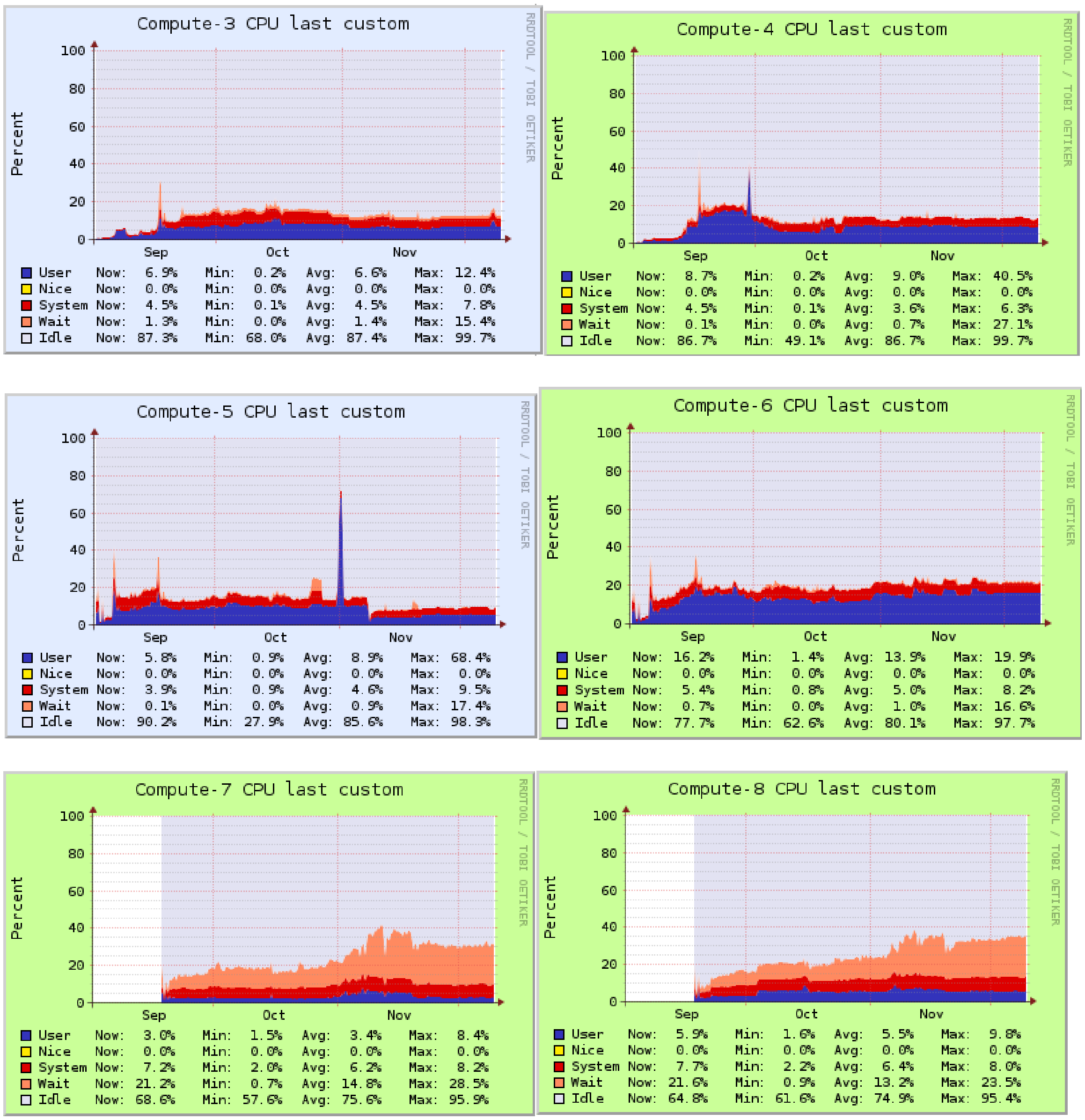
Based on the information from figure 6-1, we know that all compute servers are working in a good condition which they didn't overloaded. We have sufficient vCPU in each server to provide the computing service for VMs.

6.2. CPU Utilization

CPU utilization monitoring shows the workload of a given physical processor for real machines or of virtual processors for virtual machines. Figure 6-2 shows the CPU utilization of each nova-compute server during the monitoring time.

Figure 6-2: CPU Utilization



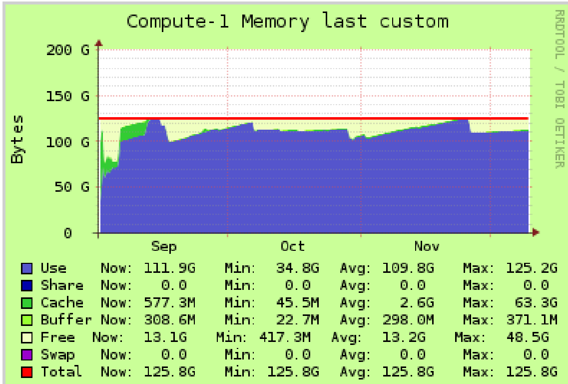


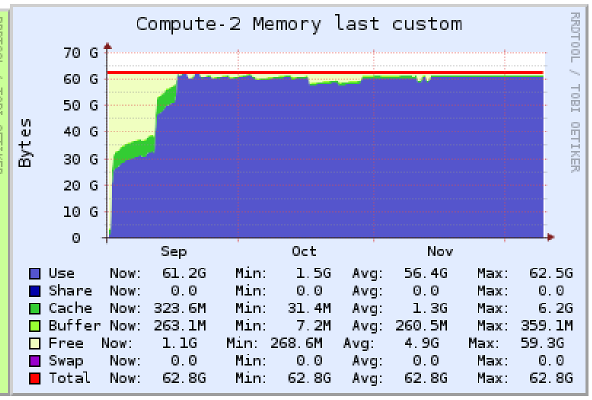
Based on the information from figure 6-2, we know that the computer1 and computer6 are busier than other compute nodes, since these two servers with more powerful in computation and memory resources. However, most of servers are operating in a good condition, since the CPU utilization are all under the threshold (70%) on average. There are only one abnormal event happened in the mid of Oct in compute5, the CPU utilization reaching to 70%.

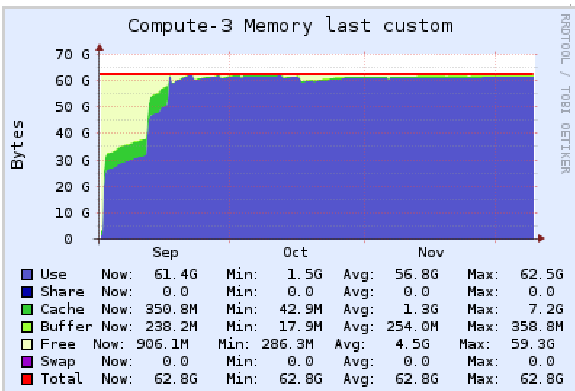
6.3. Memory Usage

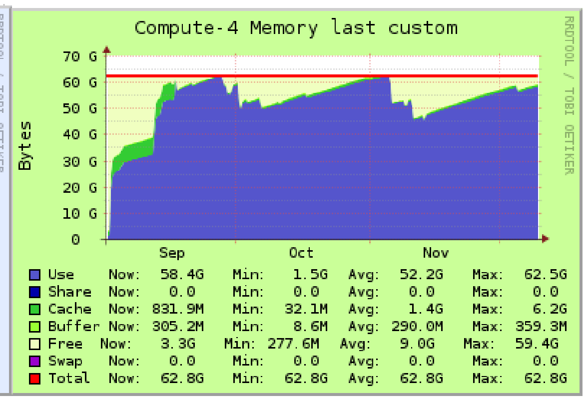
Memory usage shows how users' processes use system memory. Figure 6-3 shows the memory usage on each nova-compute server during the monitoring time. The red line in each figure represents the total memory size on the server.

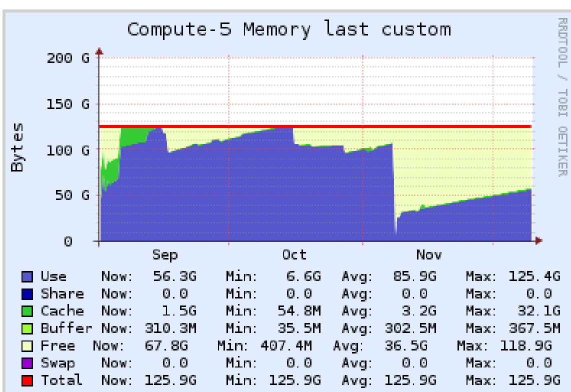
Figure 6-3: Memory Usage

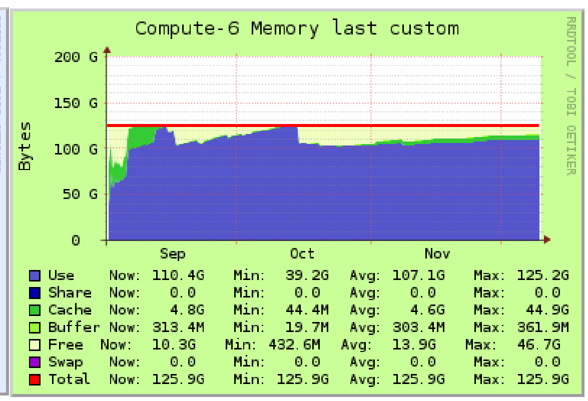

Compute-1 Memory last custom
Bytes
Sep Oct Nov
Use Now: 111.9G Min: 34.8G Avg: 109.8G Max: 125.2G
Share Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Cache Now: 577.3M Min: 45.5M Avg: 2.6G Max: 63.3G
Buffer Now: 308.6M Min: 22.7M Avg: 298.0M Max: 371.1M
Free Now: 13.1G Min: 417.3M Avg: 13.2G Max: 48.5G
Swap Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Total Now: 125.8G Min: 125.8G Avg: 125.8G Max: 125.8G
RRDTOOL / TOBI OETIKER

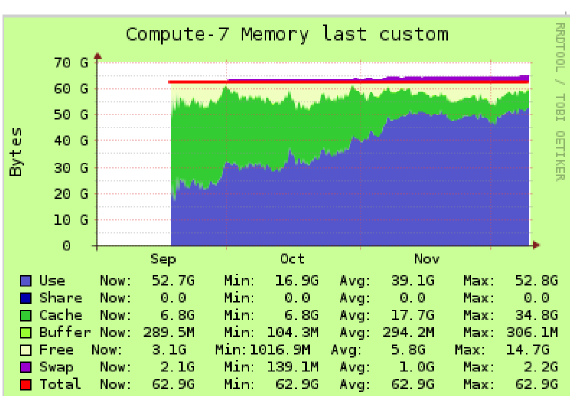

Compute-2 Memory last custom
Bytes
Sep Oct Nov
Use Now: 61.2G Min: 1.5G Avg: 56.4G Max: 62.5G
Share Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Cache Now: 323.6M Min: 31.4M Avg: 1.3G Max: 6.2G
Buffer Now: 263.1M Min: 7.2M Avg: 260.5M Max: 359.1M
Free Now: 1.1G Min: 268.6M Avg: 4.9G Max: 59.3G
Swap Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Total Now: 62.8G Min: 62.8G Avg: 62.8G Max: 62.8G
RRDTOOL / TOBI OETIKER

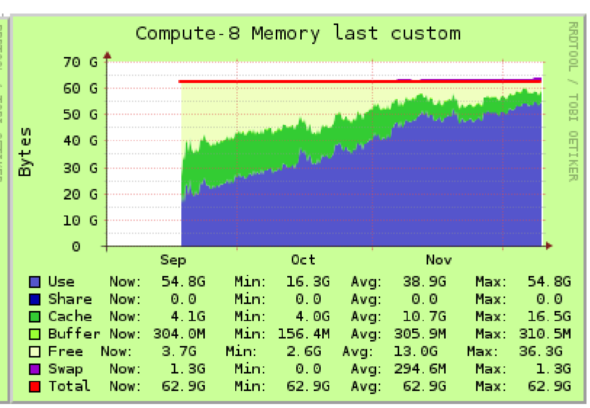

Compute-3 Memory last custom
Bytes
Sep Oct Nov
Use Now: 61.4G Min: 1.5G Avg: 56.8G Max: 62.5G
Share Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Cache Now: 350.8M Min: 42.9M Avg: 1.3G Max: 7.2G
Buffer Now: 238.2M Min: 17.9M Avg: 254.0M Max: 358.8M
Free Now: 906.1M Min: 286.3M Avg: 4.5G Max: 59.3G
Swap Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Total Now: 62.8G Min: 62.8G Avg: 62.8G Max: 62.8G
RRDTOOL / TOBI OETIKER


Compute-4 Memory last custom
Bytes
Sep Oct Nov
Use Now: 58.4G Min: 1.5G Avg: 52.2G Max: 62.5G
Share Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Cache Now: 831.9M Min: 32.1M Avg: 1.4G Max: 6.2G
Buffer Now: 305.2M Min: 8.6M Avg: 290.0M Max: 359.3M
Free Now: 3.3G Min: 277.6M Avg: 9.0G Max: 59.4G
Swap Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Total Now: 62.8G Min: 62.8G Avg: 62.8G Max: 62.8G
RRDTOOL / TOBI OETIKER


Compute-5 Memory last custom
Bytes
Sep Oct Nov
Use Now: 56.3G Min: 6.6G Avg: 85.9G Max: 125.4G
Share Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Cache Now: 1.5G Min: 54.8M Avg: 3.2G Max: 32.1G
Buffer Now: 310.3M Min: 35.5M Avg: 302.5M Max: 367.5M
Free Now: 67.8G Min: 407.4M Avg: 36.5G Max: 118.9G
Swap Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Total Now: 125.9G Min: 125.9G Avg: 125.9G Max: 125.9G
RRDTOOL / TOBI OETIKER


Compute-6 Memory last custom
Bytes
Sep Oct Nov
Use Now: 110.4G Min: 39.2G Avg: 107.1G Max: 125.2G
Share Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Cache Now: 4.8G Min: 44.4M Avg: 4.6G Max: 44.9G
Buffer Now: 313.4M Min: 19.7M Avg: 303.4M Max: 361.9M
Free Now: 10.3G Min: 432.6M Avg: 13.9G Max: 46.7G
Swap Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Total Now: 125.9G Min: 125.9G Avg: 125.9G Max: 125.9G
RRDTOOL / TOBI OETIKER


Compute-7 Memory last custom
Bytes
Sep Oct Nov
Use Now: 52.7G Min: 16.9G Avg: 39.1G Max: 52.8G
Share Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Cache Now: 6.8G Min: 6.8G Avg: 17.7G Max: 34.8G
Buffer Now: 289.5M Min: 104.3M Avg: 294.2M Max: 306.1M
Free Now: 3.1G Min: 1016.9M Avg: 5.8G Max: 14.7G
Swap Now: 2.1G Min: 139.1M Avg: 1.0G Max: 2.2G
Total Now: 62.9G Min: 62.9G Avg: 62.9G Max: 62.9G
RRDTOOL / TOBI OETIKER

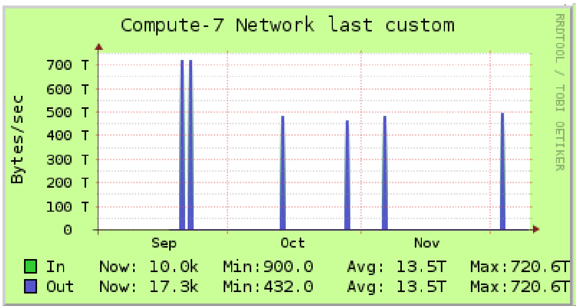

Compute-8 Memory last custom
Bytes
Sep Oct Nov
Use Now: 54.8G Min: 16.3G Avg: 38.9G Max: 54.8G
Share Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Cache Now: 4.1G Min: 4.0G Avg: 10.7G Max: 16.5G
Buffer Now: 304.0M Min: 156.4M Avg: 305.9M Max: 310.5M
Free Now: 3.7G Min: 2.6G Avg: 13.0G Max: 36.3G
Swap Now: 1.3G Min: 0.0 Avg: 294.6M Max: 1.3G
Total Now: 62.9G Min: 62.9G Avg: 62.9G Max: 62.9G
RRDTOOL / TOBI OETIKER

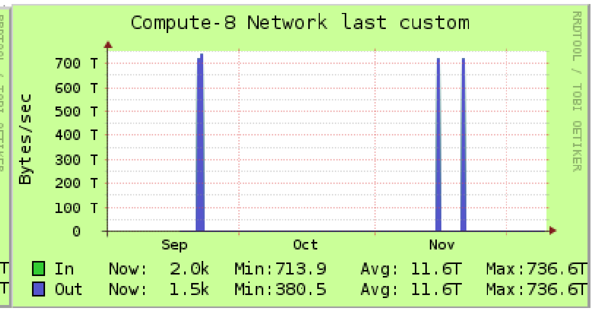
Based on the information from figure 6-2, we know that all servers are almost reaching to the memory threshold (the red-line in each figure). This mean that they almost run out of the memory, especially in the compute2, compute3, and compute4, since these three servers have less-powerful resources than the others. This is the performance bottleneck in both Nova clusters since the swapping causes the disk I/O and the CPU process delay, which in turn causes more processes waiting for the CPU processing time in the queue and the higher system load average.

6.4. Network Usage

Network usage shows the inbound and outbound flow (bytes/second) passing through all interfaces (which include physical network interfaces and virtual network interfaces) on each server. Figure 6-4 shows the network usage of each nova-compute server during the monitoring time.

Figure 6-4: Network Usage





Although the average flow shown in each figure is a terabyte-level and the maximum flow reaches to a petabyte-level, the color shown in each figure still in gray or green which means the network utilization is below 50%, i.e., the network usage on each Nova compute server is normal. The reason for the network usage reaching to petabyte-level in compute1 is because we configured the compute1 as the collector of the Ganglia monitoring system, therefore, all the monitor information in the compute cluster will be picked up by the computer1 and then send them to the Ganglia's web portal for display.

Since Nova-compute node will create a lot of virtual interfaces for VMs and Ganglia's network usage will accumulate the traffic flow from all interfaces (both physical and virtual interfaces) on a server, therefore, we cannot tell the network throughput for a certain physical interface only. In order to monitor the interface-level network usage information, V-Lab system needs to use other monitoring tools especially for network analysis, for example, sFlow, NetFlow, or commercial tools.

7. Issues and Discussions

This section presents the issues found in V-Lab system with discussions.

7.1. Lack of Single Sign On mechanism

Currently, the V-Lab system provides user management on the Web-GUI and OpenVPN. But it still lacks of user management for each virtual machine. By default, each virtual machine is created based on a template, and thus they all have same login credentials. This could be a security issue when the default password may be used to login newly created virtual machines. To solve the problem, V-Lab system needs to extend the authentication and authorization mechanism to certificate-based and Kerberos token-based solutions, which allows single sign on (SSO) from a portal and requests the credentials or tokens to access the authorized resources or services based on users' privilege without checking the username and password everywhere.

7.2. Bad image support for VM creation

The new version of Ubuntu (14.04) has some issues to be installed as a VM. We spent lot of time to try the image creation with different approaches, but still got a problematic image. The performance of the VM created with this image cannot provide a good quality of service for users. We need to find out an optimal way to create a good quality VM effectively and efficiently in the future.

7.3. MTU issue for a VM to be accessed from external

In the new version of OpenStack, we have some issue for a VM to be accessed from external. After a deep inspection, we found that the GRE tunnel between servers limits the MTU size within the regular setting 1500. Each transmitted package passes through the GRE tunnel will be attached a GRE header, which causes the real data cannot beyond the normal size 1500, otherwise, some part of real data will be chopped off. The temporary solution is reduce the MTU size in each VM manually, such as "ifconfig mtu 1456". In the future, we need to find out the regular way to configure or extend the size of the external GRE tunnel.

7.4. noVNC timeout issue

In the beginning of Fall2014, we found that the VM access through web noVNC will get timeout in every minute. This causes the user unable to work on their VM using web browser. After deep investigation, we figured out the problem comes from the high availability.

The problem due to the connection timeout less than the default TCP_Keep_alive in the Ubuntu. The default value of TCP_Keep_alive in Ubuntu is 7200s (2hr). If the connection timeout shorter than this value, the connection will be dropped after 5 min if there is no any activity in NoVNC console.

Our Solution to solve this problem is: change the connection timeout, server timeout and client timeout in the haproxy to more than 7200s, such as:

timeout queue 7500s
timeout server 7500s
timeout client 7500s

8. Development and Enhancements of V-Lab for Spring2015

This section presents the new features of V-Lab system for the future courses. Especially for the coming Spring2015 semester, the development and enhancements described as follows.

8.1 Upgrade to New Version of OpenStack

Since the OpenStack is rapidly developing, the new version of OpenStack is available now. The new version includes two new modules that are Telemetry and Orchestration. The OpenStack Telemetry service aggregates usage and performance data across the services deployed in an OpenStack cloud. This powerful capability provides visibility and insight into the usage of the cloud across dozens of data points and allows cloud operators to view metrics globally or by individual deployed resources. OpenStack Orchestration is a template-driven engine that allows application developers to describe and automate the deployment of infrastructure. The flexible template language can specify compute, storage and networking configurations as well as detailed post-deployment activity to automate the full provisioning of infrastructure as well as services and applications. Through integration with the Telemetry service, the Orchestration engine can also perform auto-scaling of certain infrastructure elements.

To upgrade to the new version, a fresh installation is required. The system configuration in the current version could be used as the reference in the new installation. The difference will be the configuration of the two new components. The trial and tests are necessary to make sure the new modules are working properly.

8.2. Redesign the Web Portal of V-Lab System

The existing V-Lab Web-GUI has a self-developed PHP-based Web Portal that can be used for students to multiple services and resources provided by V-Lab. The portal was developed without native support for Web 2.0, and thus lacks the Ajax features for dynamic pulling and updating mechanism. As a result, the current web portal is not user-friendly and sometimes confusing for end users.

Our plan is to redesign the Web Portal from the sketch. The new Web Portal will be developed based on an open-source Drupal project and fully adopt the Web 2.0 standard, which comes with come with community support. The portal can work closely with the backend Web service and databases to provide a user-friendly Web-GUI to future V-Lab users. The redesign and development are in process.

9. Estimated Resource Requirement for Spring2015

Table 9-1 shows the possible courses and estimated number of students will use V-Lab system in Spring2015. It also shows the estimated resource, which includes the number of VMs, the number of vCPU, the amount of disk space, and the amount of RAM, to support these courses.

Table 9-1: Estimated Resource for Spring2015

Course	Total Students	# of VMs	# of vCPU	Disk Space (GB)	Memory (GB)	Instructor	Course Name
434	100	300	300	3000	300	Dijiang Huang	Computer Network
Total	**100**	**300**	**300**	**3000**	**300**		

The estimated Memory requirement for Spring2015 is 300G for 300 VM. Based on the current configuration of V-Lab system, we can support the resource requirement for Spring2015.

10. References

[1] OpenStack Web-front end in V-Lab system, http://openstack.mobicloud.asu.edu/horizon.
[2] The web portal of V-Lab system, http://vlab.asu.edu.
[3] MobiCloud Source Forge, http://mcsf.vlab.asu.edu/
[4] PaceMarker, http://clusterlabs.org/
[5] Corosync, http://en.wikipedia.org/wiki/Corosync_(project)
[6] V-Lab Monitoring System, http://management.vlab.asu.edu/ganglia.
[7] Ganglia Monitoring System, http://ganglia.sourceforge.net/.
[8] Percona XTRADB Cluster, http://www.percona.com/software/percona-xtradb-cluster