MyCloudLab: An Interactive Web-based Management System for Cloud Computing Administration

Hoi-Wan Chan[1], Min Xu[2], Chung-Pan Tang[1], Patrick P. C. Lee[1] & Tsz-Yeung Wong[1],
[1]Department of Computer Science and Engineering
[2]Department of Information Engineering
pclee@cse.cuhk.edu.hk

We would like to present MyCloudLab, a web-based management system for the cloud computing administration. It provides a platform for students to easily experience the environment of cloud computing platform and learn essential techniques of administering a cloud computing system. We deployed MyCloudLab in the Introduction to the Cloud Computing course in autumn 2012.

Keywords - Cloud computing, MapReduce, Hadoop

Introduction

Cloud computing has been an emerging topic in Information Technology for some time. It provides a new computing paradigm for enterprises and individuals to manage computational resources in an on-demand manner. In view of the emergence of cloud computing, the Department of Computer Science and Engineering (CSE) now offers a new course 'CSCI4180: Introduction to Cloud Computing' for senior undergraduate students since Spring 2012. This course aims to enable students to understand the fundamental concepts of cloud computing and to develop hands-on skills of building and programming cloud computing applications (e.g., MapReduce programming [1]). To make cloud computing programming feasible, we have set up a cloud testbed with around 20 servers, using the open-source cloud software OpenStack [3]. We provide virtual machines (VMs) for students to develop cloud computing applications on our cloud testbed.

In addition to building cloud computing applications, it is also important for students to learn how to manage a cloud testbed by playing the role of a system administrator. Learning the cloud administration skills is necessary, especially with the increasing trend for many organisations (e.g., CUHK) to set up private clouds for internal use. However, it is not currently feasible for us to provide full access privileges for students to manage our cloud testbed, which is shared by all students in the class. Given that students are inexperienced in managing a cloud, any configuration errors can potentially compromise the stability of the entire cloud testbed. Thus, the motivation of this project is to provide a learning environment where students can learn cloud computing administration skills, while the administration tasks carried out by students are under the proper control of the teaching staff and thus the stability of our cloud testbed will be preserved.

Objectives

In this project, we propose *MyCloudLab*, an interactive, web-based management system for use in cloud computing administration. MyCloudLab uses a sandbox approach to isolate different groups of VMs, and abstracts all the management functionalities of each group of VMs into a web-based interface. Thus, students only have the full privileges to administer the VMs that are assigned to them through our web-based interface, but not the VMs assigned to other students. Any administration errors can be controlled and recovered through

MyCloudLab. In particular, MyCloudLab teaches students different administration techniques:

- Students can monitor and fine-tune the resource usage (e.g., CPU and storage) of their assigned VMs, so that they can understand the performance of VMs under different workload patterns;
- Students can configure a MapReduce computing cluster, so that they can run their own MapReduce program with a different configuration, in order to learn programming skills within a MapReduce framework;

We would emphasise that similar web management systems have been available for some time, such as Amazon EC2 Web Management Console [5] and OpenStack Dashboard [4]. On the other hand, MyCloudLab places a strong emphasis on teaching, with the following unique features:

- It provides only the basic functionalities (and hides the advanced features) for students to learn the essential cloud administration skills
- It includes detailed instructions that guide students in performing cloud administration tasks.

The project aims for the following outcomes:

- It provides an interactive platform for students to learn the essential administration skills of a cloud computing platform.
- It provides a centralised, fully-controlled platform for teaching staff to limit the privileges of students to use our cloud computing platform.

It provides a standardised platform for teaching staff to manage the usage of VMs by students, design laboratories, and grade students' work.

MyCloudLab Design

We implement MyCloudLab as a web interface via which students can perform all the administrative tasks of their VMs which are hosted on our cloud testbed. The web interface is implemented via standard web programming, such as JavaScript, PHP or AJAX programming.

MyCloudLab is realised with two components (as shown in Fig. 1):

- *VM Lab*: OpenStack allows each student to manage their own VM instance hosted in a cloud using the command line interface. VM Lab provides a web interface as an alternative for the students.
- *Hadoop Lab*: Hadoop is an implementation of MapReduce, which is a prevalent parallel computing framework for large amounts of data. Hadoop Lab provides a simplified interface to run MapReduce, without the full knowledge of the underlying infrastructure. The complex Hadoop cluster setup procedures can be avoided. Hence, students can focus on their MapReduce programming.

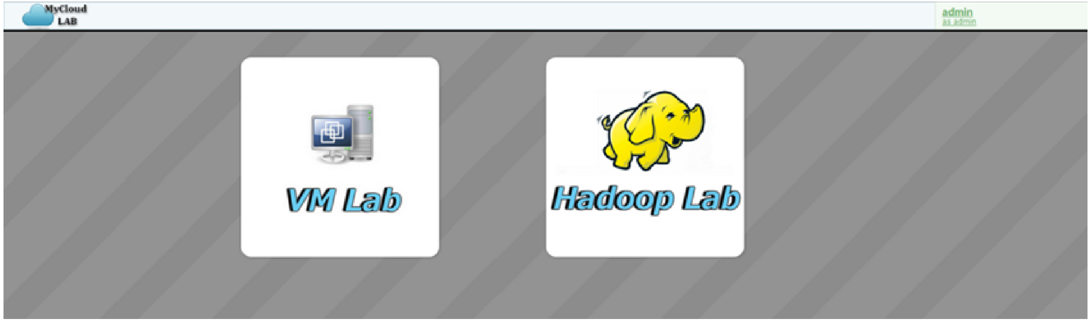


Figure 1: Two components in the MyCloudLab main interface

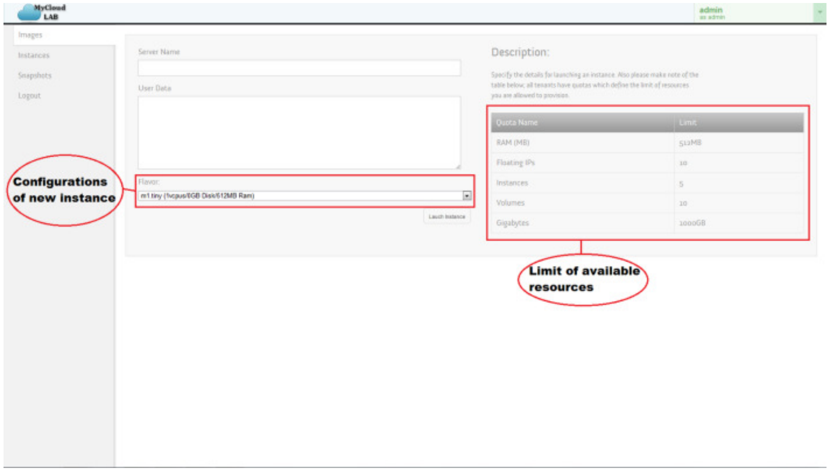
VM Lab

Most of the management functions supported in VM Lab are enabled by the OpenStack API [6]. Functions like retrieval of a VM image list, an instance list and a snapshot list can be categorised as sending a GET request to API. Operational functions like launching and rebooting a VM instance are implemented by sending POST request to API. We send DETELTE requests to the API to terminate a VM instance. We incorporate this useful API with our user interface so that all these functions can be achieved by simple mouse clicking and limited text input.

Creating a VM instance

Users can select a VM image, with which they can launch a VM instance, from the list of available images in the 'images' section. Then they will be directed to a form inquiring about some details of the VM instance that will be launched as shown in Fig. 2.

The users can then decide on the name and the resource configuration of the VM instance. The resource configuration includes virtual CPU, memory, and storage space. The user can select a different combination of resource allocations from the list. We provide each user with a certain quota in terms of resource usage of the cloud platform, in order to avoid overloading the platform. A description table showing those quota limits is displayed in the VM launching interface. This can provide some hints to help users allocate their resources.



Flavor: m1.tiny (1vcpus/0GB Disk/512MB Ram)

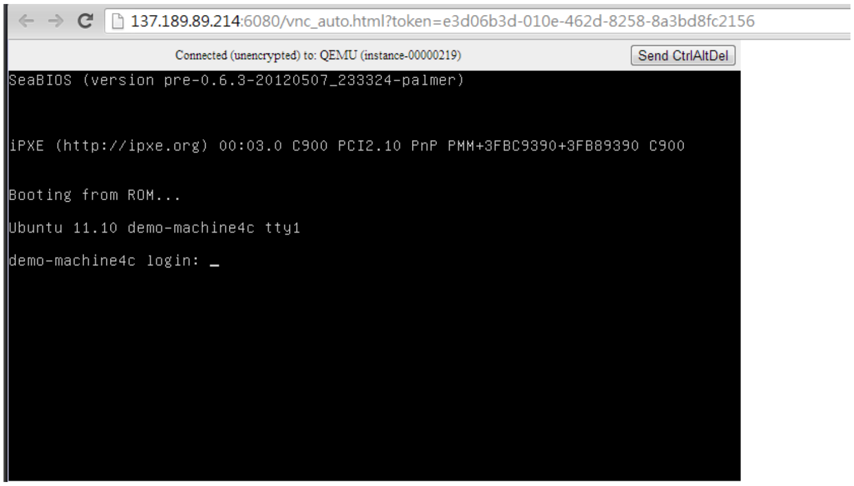
Description:

Specify the details for launching an instance. Also please make note of the table below; all tenants have quotas which define the limit of resources you are allowed to provision.

Quota Name	Limit
RAM (MB)	512MB
Floating IPs	10
Instances	5
Volumes	10
Gigabytes	1000GB

Figure 2: Interface to launch new VM. User can select different resource configuration

To enable users to access their VM instance easily, we can take advantage of noVNC, an open-sourced JavaScript library which enables remote access to the terminal through a browser. By simply clicking on the 'VNC Console' link of each VM instance, a new browser window, with a virtual terminal, will be prompted. This interface can support most the terminal features. With this feature, users can theoretically do anything as if they are in front of a physical machine.


137.189.89.214:6080/vnc_auto.html?token=e3d06b3d-010e-462d-8258-8a3bd8fc2156
Send CtrlAltDel

Connected (unencrypted) to: QEMU (instance-00000219)

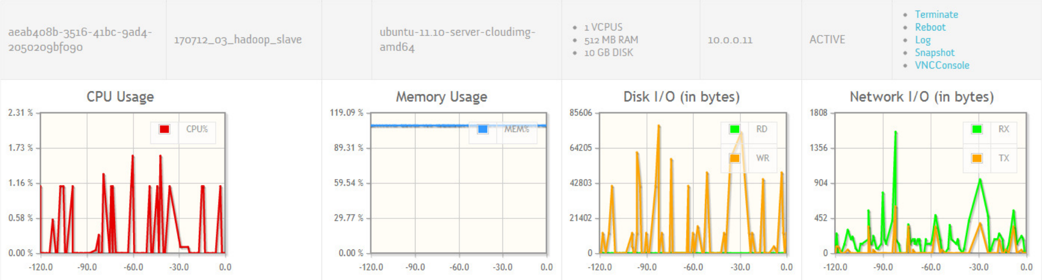
```
SeaBIOS (version pre-0.6.3-20120507_233324-palmer)


iPXE (http://ipxe.org) 00:03.0 C900 PCI2.10 PnP PMM+3FBC9390+3FB89390 C900

Booting from ROM...

Ubuntu 11.10 demo-machine4c tty1

demo-machine4c login: _
```

Figure 3: Interface to launch new VM. A user can select different resource configurations

Real-time monitoring of utilisation of resources

When a large job is running on the cluster, users may want to check the CPU usages of the VMs in the cluster in order to confirm that the workload is distributed to every VM in the cluster. In other cases, users may want to know the CPU, memory and hard disk usages of all available VMs to choose those with least workload when they plans to establish a Hadoop cluster. Therefore, monitoring the utilisation of resources in each instance is important.

In VM Lab, we provide this feature using a system developed on our own. It supports real-time monitoring of utilisations of resources, including CPU, memory, hard disk read/write rate and network transfer rate. It visualises the information using plots, as shown in Fig. 4.



aeab408b-3516-41bc-9ad4-2050209bf090	170712_03_hadoop_slave		ubuntu-11.10-server-cloudimg-amd64	1 VCPUS 512 MB RAM 10 GB DISK	10.0.0.11	ACTIVE	Terminate Reboot Log Snapshot VNCConsole

Figure 4: The real-time resource utilisation monitoring interface for a VM

Hadoop Lab

Establishing Hadoop Cluster on-demand

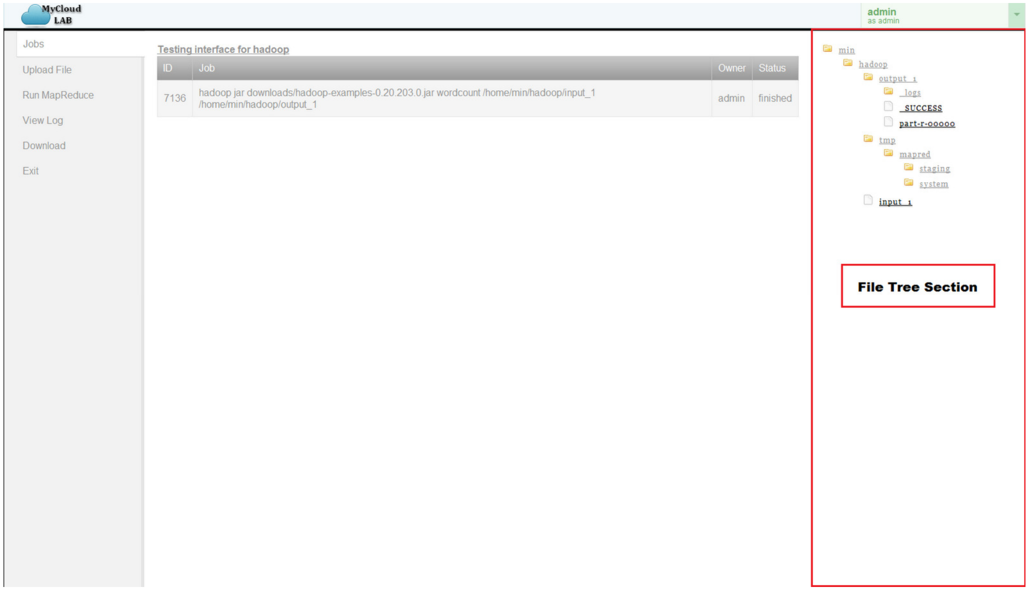
Through the Hadoop Lab, users can create a Hadoop cluster on their existing VMs in the cloud platform at any time. In the cluster setup process, users can specify the size of the cluster and select the corresponding number of VM created from VM Lab. This flexibility enables users to experience the discrepancies in capacity among clusters of different sizes. Apart from that, it is also possible for users to further customise their Hadoop cluster through direct access to the VM in the cluster.

Persistent cluster enabling concurrent access

Once a cluster is established, it remains until its owner explicitly destroys it by clicking the ‘exit’ button in Hadoop Lab. This enables two common use cases: i) concurrent accesses; ii) access later-on. When users are carrying out some operations in a cluster, their partners can concurrently enter and work in the same cluster as other browsers, as long as they log into MyCloudLab using the same account. In the case of access later-on, if the session of a user is over, either expiring or being invalidated by the user closing the browser, once the user logs in with the same account, he/she will enter the cluster directly without the need to create a new one.

Visualizing the hierarchy of HDFS

Once a cluster is established, the Hadoop Distributed File System (HDFS) [2] service will also be started. The HDFS stores all files in the cluster, including the user-uploaded input files for processing and the processed output from MapReduce program. Hence, it will be really helpful if a user could have an approach to viewing the structure of the HDFS without typing various commands into the terminal. In the Hadoop Lab, we can make this come true by showing the content as well as hierarchy of the file system in a tree, as shown in Fig. 5. With the help of this interface, users can easily locate a file in HDFS and the output directory of the MapReduce program.



Testing interface for hadoop

ID	Job	Owner	Status
7136	hadoop jar downloads/hadoop-examples-0.20.203.0.jar wordcount /home/min/hadoop/input_1 /home/min/hadoop/output_1	admin	finished

Figure 5: Tree hierarchy of the HDFS

Basic functions: upload dataset, run MapReduce program and view result

The primary target of Hadoop Lab is to run a MapReduce program, so it must support the fundamental operations runoff running such a program. Firstly, it supports file upload, which enables users to upload input data set to HDFS. Subsequently, when users plan to start a MapReduce program, the only need to provide information on the input data file, the output directory and the compiled MapReduce JAR program. After the job is done, the users can view the job log to find out the details of the whole job process, including total running time, task summary, task analysis, etc.

Tuning of MapReduce parameters

There are many configurable parameters provided by Hadoop in order to speed up a MapReduce job. For jobs with distinct features, such as size of the input data set, the number of task-trackers and the, quality of input data, different configurations are required to optimise the performance of these jobs. In our Hadoop Lab, we enable the customisation of three of the most useful parameters, which are JVM Reusing, Speculative Execution and Skipping Bad Records, so that users can have a deeper understanding about how and how much tuning a specific parameter can affect the performance of a job.

Export MapReduce output from HDFS

Hadoop Lab enables users to download any file or directory from HDFS as a ZIP file to their own computers. This feature is especially useful if users need to keep some files for their own later use, for tasks such as handing in assignments.

Conclusions and Future Work

We implemented the MyCloudLab and have deployed it in the course 'CSCI4180: Introduction to Cloud Computing' since autumn 2012. Students are now using this platform to learn both the administration skills and MapReduce programming skills. Our future work will include collecting students' feedbacks on MyCloudLab and adding new lab modules to MyCloudLab.

Acknowledgement

This work was supported by the CUHK Course Development Grant Scheme (CDGS) (Project number: 4621262).

References

[1] Dean, J. & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 51, 107-113.

[2] Shvachko, K. and Kuang, H. and Radia, S. and Chansler, R. (2010). The Hadoop distributed file system. In Proc. of ACM MSST, 1-10

[3] *OpenStack.* (n.d.). Retrieved October 17, 2012, from http://www.openstack.org /

[4] *OpenStack Dashboard.* (n.d.). Retrieved October 17, 2010, from http://www.openstack.org/software/openstack-dashboard/

[5] *AWS Management Console.* (n.d.). Retrieved October 17, 2010, from http://aws.amazon.com/console/

[6] *Specification for the Compute API 2.* (n.d.). Retrieved October 17, 2012, from http://docs.openstack.org/api/openstack-compute/2/content/