

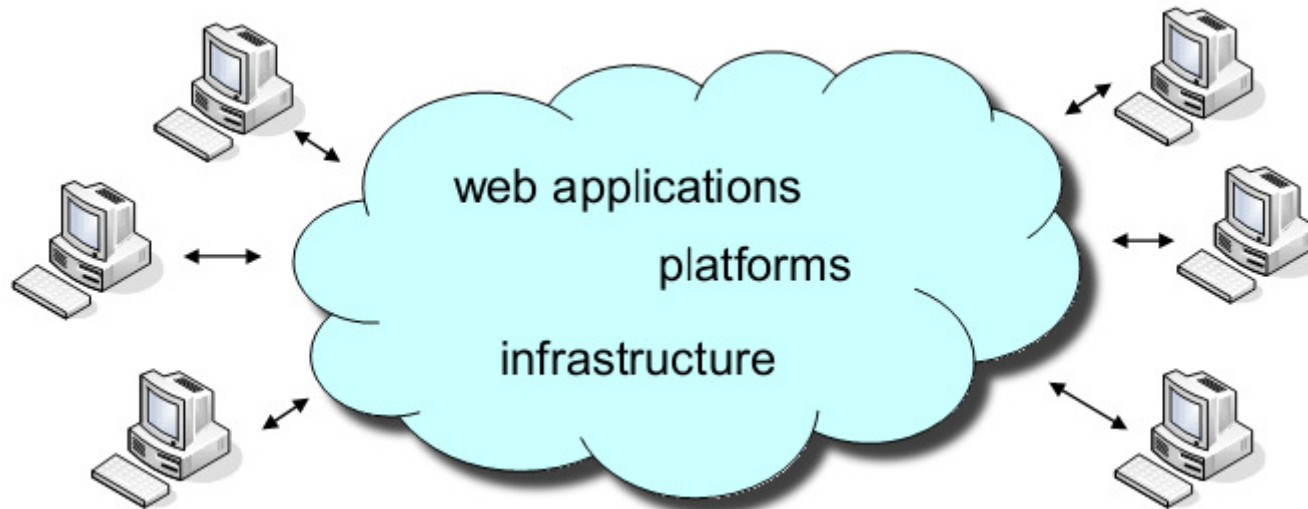
MyCloudLab: An Interactive Web-based Management System for Cloud Computing Administration

Hoi-Wan Chan¹, Min Xu², Chung-Pan Tang¹,
Patrick P. C. Lee¹ and Tsz-Yeung Wong¹

¹Department of Computer Science and Engineering

²Department of Information Engineering

Cloud computing



- **Cloud computing** has been an emerging topic in Information Technology
- It provides a new computing paradigm for enterprises and individuals to manage computational resources in an on-demand manner

CSCI4180: Introduction to Cloud Computing

- Department of Computer Sci. & Eng. offers a new course “CSCI4180: Introduction to Cloud Computing” for senior undergraduate students since Spring 2012
- It aims to enable students
 - to understand the fundamental concepts of cloud computing
 - to develop hands-on skills of building and programming cloud computing applications (e.g., MapReduce programming)

Motivation

- We provide **virtual machines (VMs)** for students to develop cloud computing applications on our cloud testbed.
- It is also important for students to learn
 - how to manage a cloud testbed by playing the role of a system administrator
- But not feasible to provide full access privileges
 - any configuration errors can potentially compromise the stability of the entire cloud testbed

Our work

MyCloudLab, an interactive, web-based management system for use in cloud computing administration.

➤ Design goals of MyCloudLab

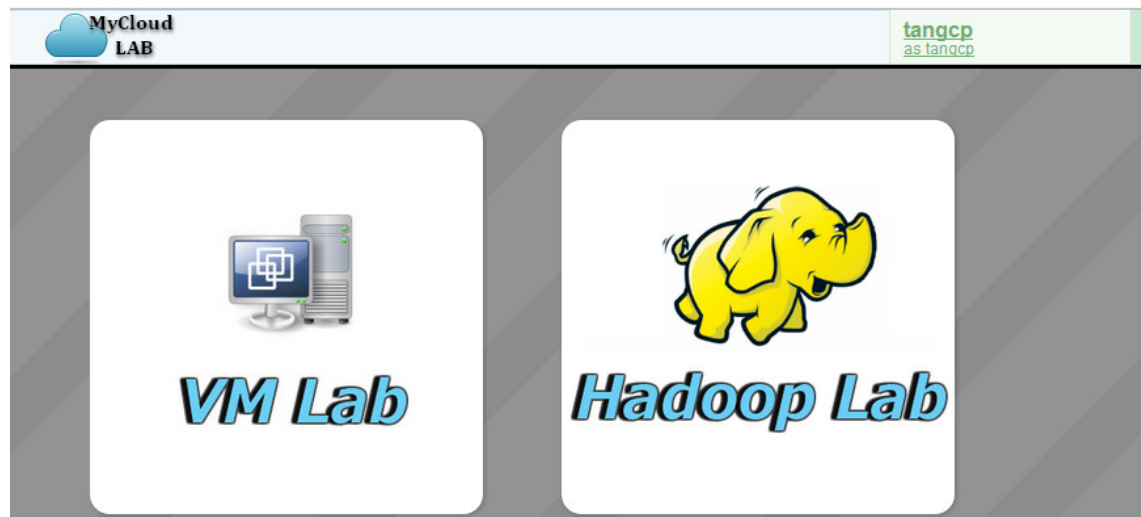
- Provide an interactive platform for **students** to learn the essential administration skills of a cloud computing platform.
- Provide a centralized, fully-controlled platform for **teaching staff** to limit the privileges of students to use our cloud computing platform.

Our work

- Design features of MyCloudLab:
 - **Isolation:** uses a sandbox approach to isolate different groups of VMs
 - **Abstraction:** puts all the management functionalities of VMs into a web-based interface
 - **Simplicity:** provides only the basic functionalities (and hides the advanced features)
 - **Extensibility:** provides interface to add new lab modules
 - **Well-documented:** includes detailed instructions that guide students in performing cloud administration tasks

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, namely
 - JavaScript
 - PHP



MyCloudLab Design

- MyCloudLab is realized with two components
 - **VM Lab** provides a web interface for the students to manage their own VM instances hosted in a cloud
 - **Hadoop Lab** provides a simplified interface to run MapReduce program
 - It requires neither full knowledge of the underlying infrastructure nor complex Hadoop cluster setup procedures
 - Students can focus on their MapReduce programming

VM Lab

- Most of the management functions supported in VM Lab
 - retrieval of VM instance list
 - resource configuration
 - status of VMs
 - reboot and terminate a VM instance

Hadoop Lab AMI	<ul style="list-style-type: none">• 1 VCPUS• 512 MB RAM• 10 GB DISK	10.10.11.2	ACTIVE	<ul style="list-style-type: none">• Terminate• Reboot• Log• Snapshot• VNCConsole
Hadoop Lab AMI	<ul style="list-style-type: none">• 1 VCPUS• 512 MB RAM• 10 GB DISK	10.10.11.4	ACTIVE	<ul style="list-style-type: none">• Terminate• Reboot• Log• Snapshot• VNCConsole

- launch a VM instance
 - different virtual machine (OS) templates
 - different resource configuration (virtual CPU, memory, and storage space)

Name	Created
Ubuntu Server 12.04 (i386)	2012-09-25
Hadoop Lab AMI	2012-09-05
Ubuntu Server 11.10 (i386)	2012-08-30
Ubuntu Server 12.04 (amd64)	2012-08-29
Demo Cloud image	2012-08-29

Flavor:


- m1.tiny (1 vcpus/0GB Disk/512MB Ram)
- m1.tiny (1 vcpus/0GB Disk/512MB Ram)**
- flavor.tiny (1 vcpus/10GB Disk/512MB Ram)
- m1.small (1 vcpus/10GB Disk/2048MB Ram)
- flavor.small (2 vcpus/10GB Disk/512MB Ram)
- m1.medium (2 vcpus/10GB Disk/4096MB Ram)
- m1.large (4 vcpus/10GB Disk/8192MB Ram)
- m1.xlarge (8 vcpus/10GB Disk/16384MB Ram)

Description:

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

Quota Name	Limit
VCPUs	20
RAM (MB)	10240MB
Floating IPs	10
Instances	20
Gigabytes	30GB

- save a VM instance as a snapshot
- restore a VM instance from a snapshot
 - Students can backup their VMs to avoid data loss due to any configuration errors


csci4180-0
as tangcp


Images

Instances

Snapshots

Logout

Name	Images	Size	IPs	State	Actions
dick-vm1	5070proj-networkx	<ul style="list-style-type: none"> 2 VCPUS 4096 MB RAM 10 GB DISK 	10.10.11.80	ACTIVE	<ul style="list-style-type: none"> Terminate Reboot Log Snapshot VNCConsole



Images

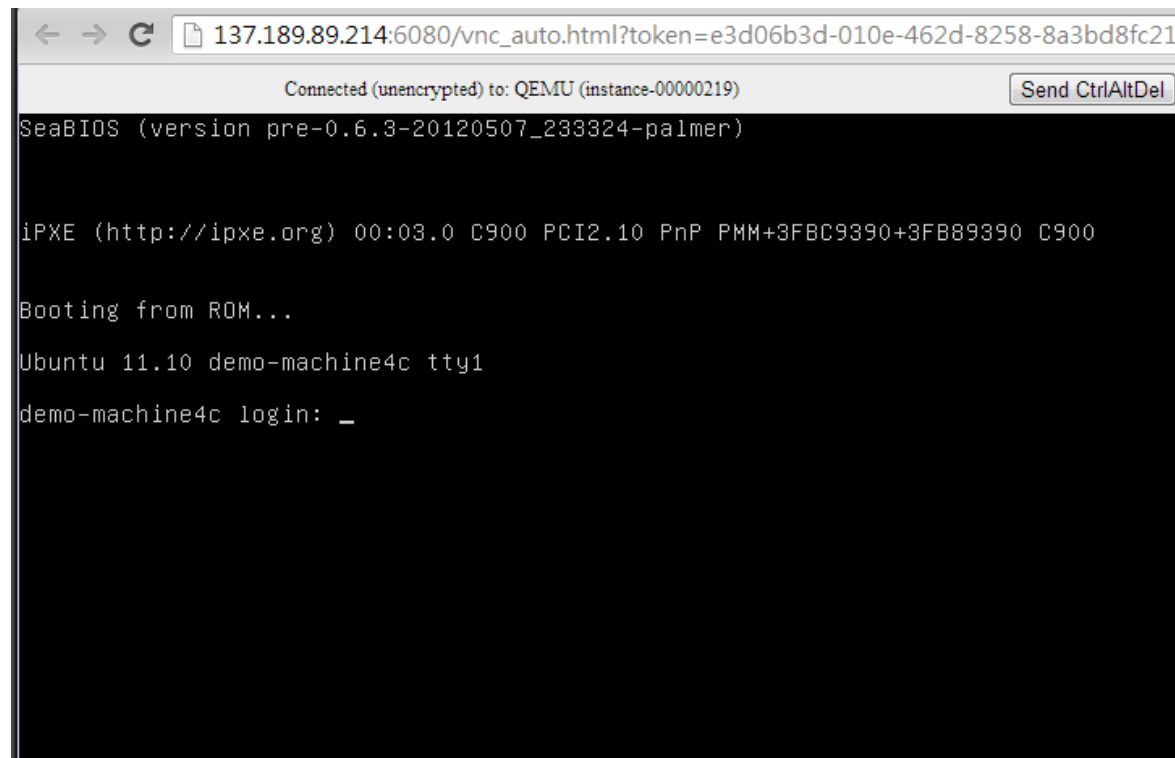
Instances

Snapshots

Name	Created	Updated
5070proj-networkx	2012-11-23 05:34:51	2012-11-23 05:42:05

Remote access to VM

- VM Lab provides students a simple remote access to their VM terminal through a browser, as if they are in front of a physical machine.



```
137.189.89.214:6080/vnc_auto.html?token=e3d06b3d-010e-462d-8258-8a3bd8fc21
Connected (unencrypted) to: QEMU (instance-00000219) Send CtrlAltDel
SeaBIOS (version pre-0.6.3-20120507_233324-palmer)

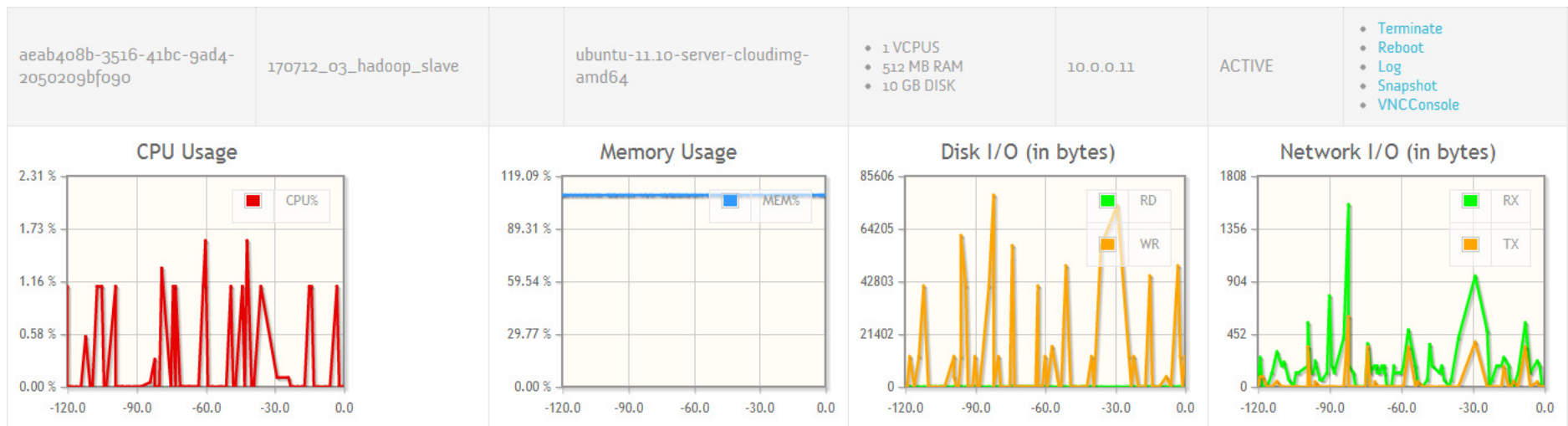
iPXE (http://ipxe.org) 00:03.0 C900 PCI2.10 PnP PMM+3FBC9390+3FB89390 C900

Booting from ROM...
Ubuntu 11.10 demo-machine4c tty1
demo-machine4c login: _
```

Real-time resource utilization monitoring

- Students may want to know the resource utilization of VMs
 - to choose the least workload VMs when establishing a cluster
 - to confirm that the workload is distributed to every VMs
- Thus, monitoring the utilization of resources is important.
- VM Lab supports real-time monitoring of utilizations of resources, including
 - CPU usage
 - Memory usage
 - hard disk read/write rate and
 - network transfer rate

Real-time resource utilisation monitoring



Hadoop Lab

- Hadoop is an implementation of MapReduce,
 - which is a prevalent parallel computing framework for large amounts of data.
- Hadoop Lab, as a simplified interface to run MapReduce program, abstracts the following complex setup procedures:



Hadoop Lab

- Students can establish a Hadoop cluster on their VMs launched in VM Lab at anytime
- Students can specify the number of VMs used in the cluster
 - enables students to experience the discrepancies in capacity among clusters of different sizes

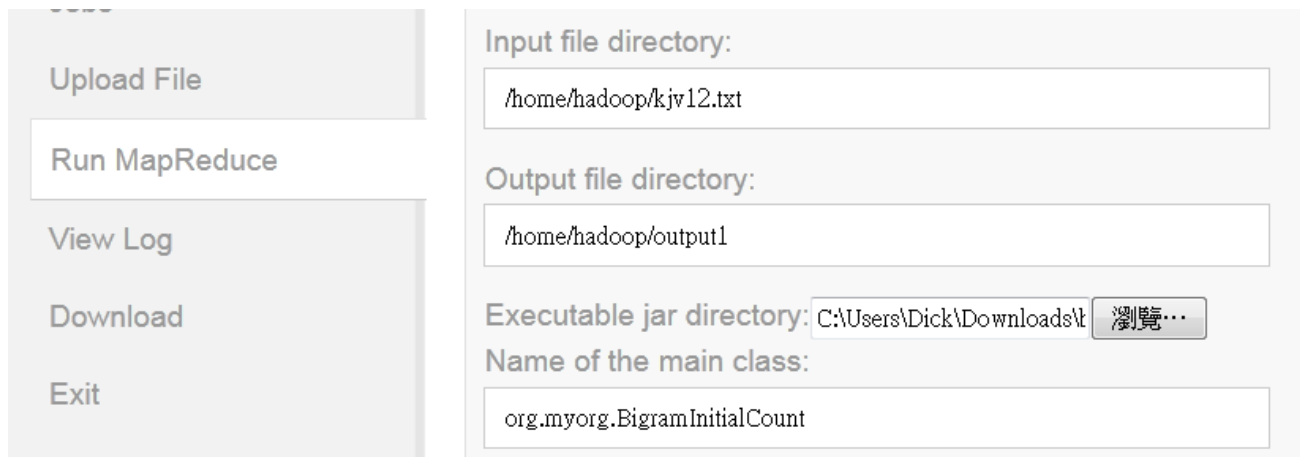
The screenshot displays the MyCloud LAB interface. On the left, there is a configuration panel for a Hadoop cluster. It includes sections for 'MasterNode', 'Cluster Size', and 'SlaveNodes'. The 'MasterNode' section has fields for 'NameNode/JobTracker IP:' (10.10.11.2), 'UserName:' (hadoop), and 'Passwd:' (masked with dots). The 'Cluster Size' section has a field for 'No. of DataNodes/TaskTrackers:' (1). The 'SlaveNodes' section has a table with one entry: '1' in the first column, 'IP:' (10.10.11.4) in the second, and 'Username:' (hadoop) in the third. An 'Enter' button is at the bottom left of this panel.

On the right, there is a file manager window titled 'MyCloud LAB' with a 'tangcp as tangcp' user indicator. It shows a 'Jobs' menu with options: 'Upload File', 'Run MapReduce', 'View Log', 'Download', and 'Exit'. Below the menu is a table with columns 'ID', 'Job', 'Owner', and 'Status'. To the right of the table is a file browser showing the '/home' directory.

- Hadoop Lab provides an intuitive web interface for students to prepare data file and manipulate file in HDFS (Datastore in Hadoop)
 - Direct upload data file
 - A tree structure visualizes the file hierarchy



- To run a MapReduce program, students only need to provide information on
- the input data file path
 - the output directory path
 - and the compiled MapReduce JAR program



The screenshot shows a web-based interface for configuring a MapReduce job. On the left is a vertical sidebar with five buttons: 'Upload File', 'Run MapReduce', 'View Log', 'Download', and 'Exit'. The 'Run MapReduce' button is highlighted. The main area contains four input fields:

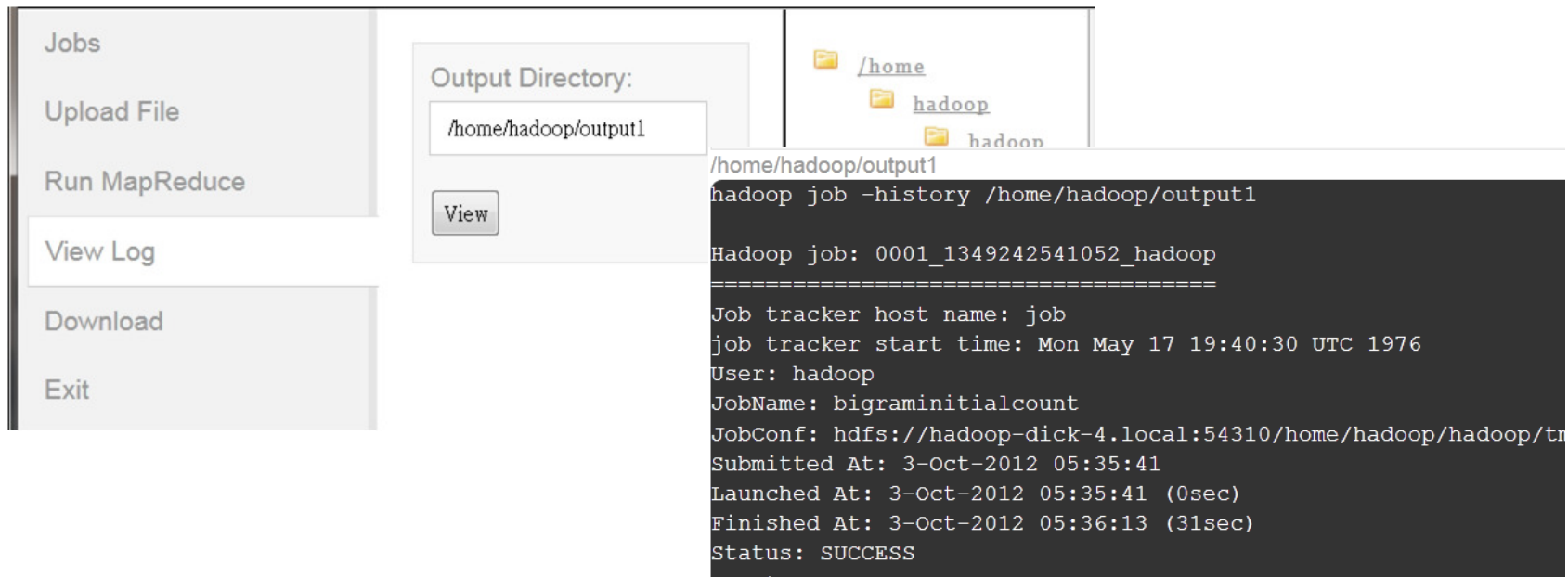
- Input file directory:** /home/hadoop/kjv12.txt
- Output file directory:** /home/hadoop/output1
- Executable jar directory:** C:\Users\Dick\Downloads\t (with a '瀏覽...' button)
- Name of the main class:** org.myorg.BigramInitialCount

- Configurable parameters for MapReduce optimization
 - JVM Reusing
 - Speculative Execution
 - Skipping Bad Records

The screenshot shows a web-based configuration interface for a MapReduce job. On the left, there is a sidebar with buttons for 'view Log', 'Download', and 'Exit'. The main area contains the following configuration options:

- Executable jar directory:** C:\Users\Dick\Downloads\ [瀏覽...]
- Name of the main class:** org.myorg.BigramInitialCount
- Reuse JVM:** 1 times.
- Disable Speculative Execution on Map Tasks.
- Disable Speculative Execution on Reduce Tasks.
- Skip:** 0 Bad Records in Map Tasks.
- Skip:** 0 Bad Records in Reduce Tasks.
- Execute** button

- Students can view the job log to find out the details after the MapReduce job finished
 - total program running time, task summary, task analysis, etc.



The screenshot displays a web interface for viewing Hadoop job logs. On the left, a sidebar contains navigation options: Jobs, Upload File, Run MapReduce, View Log (highlighted), Download, and Exit. The main content area shows the 'Output Directory' as `/home/hadoop/output1` with a 'View' button. To the right, a file browser shows the directory structure: `/home` containing `hadoop` and `hadoop`. Below this, a terminal window shows the command `hadoop job -history /home/hadoop/output1` and its output:

```
hadoop job -history /home/hadoop/output1
Hadoop job: 0001_1349242541052_hadoop
=====
Job tracker host name: job
job tracker start time: Mon May 17 19:40:30 UTC 1976
User: hadoop
JobName: bigraminitialcount
JobConf: hdfs://hadoop-dick-4.local:54310/home/hadoop/hadoop/tr
Submitted At: 3-Oct-2012 05:35:41
Launched At: 3-Oct-2012 05:35:41 (0sec)
Finished At: 3-Oct-2012 05:36:13 (31sec)
Status: SUCCESS
```

Conclusions

- Implemented and deployed the MyCloudLab in the course CSCI4180 since autumn 2012.
- Students are now using this platform to learn both the administration skills and MapReduce programming skills.
- Future work
 - collecting students' feedbacks on MyCloudLab
 - adding new lab modules to MyCloudLab.

Acknowledgement

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

References

- Dean, J. & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51, 107-113.
- Shvachko, K. and Kuang, H. and Radia, S. and Chansler, R. (2010). The Hadoop distributed file system. In Proc. of ACM MSST, 1-10
- OpenStack.(n.d.). Retrieved October 17, 2012, from <http://www.openstack.org/>
- OpenStack Dashboard.(n.d.). Retrieved October 17, 2010, from <http://www.openstack.org/software/openstack-dashboard/>
- AWS Management Console.(n.d.). Retrieved October 17, 2010, from <http://aws.amazon.com/console/>
- Specification for the Compute API 2.(n.d.). Retrieved October 17, 2012, from <http://docs.openstack.org/api/openstack-compute/2/content/>