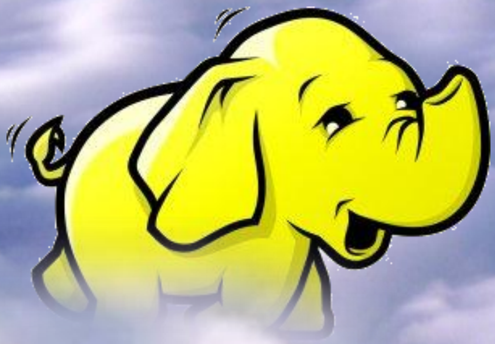
Department of Computer Science and Engineering

The Chinese University of Hong Kong



2010-2011 Final Year Project Report (1st Term)

Cloud computing technologies and applications

Supervised by Prof. Michael R. Lyu

Group: LYU1005

Chu Yan Shing (1008618841)

Wu Bing Chuan (1008612783)

This is a blank page

Abstract

This report is introducing our group's final year project. We will describe our project's:

- Project objective,
- Experiment we did ,
- Benefits and trade off of using cloud and Hadoop,
- Problems we solved.

The objective of our project is using a private cloud and Hadoop to handle massive data processing like video searching.

In this final year project, we choose to use two technologies. The first one is Cloud which allows us to allocate the resource with an easy and efficient way. The cloud we choose is Ubuntu Enterprise private Cloud. The second technology we use is Hadoop, it is a software framework to implement data-intensive program with thousand of machines. Eucalyptus provides the resource in form of individual virtual machine, and Hadoop manage all the resource of the virtual machine. Hadoop evenly allocate the storage, computation power and divide large jobs to separate machines with many tiny jobs.

Since we are using two 'new' technologies (cloud and Hadoop) to handle a large scale job, we must fully understand the characteristic of these systems. Aim to fully understand two technologies, we did many experiment on testing the performance, especially the scalability of two technologies and optimize the environment.

We find private cloud powerful but with much limitation, especially the performance trades off. Hadoop allow us to easily process large amount of data in parallel which is difficult to handle in traditional ways.

At the middle of this report, we will describe the advantages and disadvantages of using Cloud and Hadoop

Table of content

Chapter 1: Introduction

1.1 Motivation

File size keeps increase

Although the computing power of machines is keeping increase in a very high speed. Almost every 3 years, CPU's computing power increase twice. However size of the files keeps increasing also in an amazing rate.

20 years ago, the common format is only text file. Later, computer can handle the graphics well, and play some low quality movies. In recent year, we are not satisfied in the quality of DVD, and introduce the Blu-ray disk. The file is changed form a few KBs to now nearly 1 TB.

Limitation of CPU core

CPU is the most important part of a computer. However, with current technology, CPU speed has reached the upper bound of speed. Directly increase the frequency of a CPU core is not the solution to improve the computation power as the temperature of such core may increase to thousand of degree. The current solution is putting more cores into a single CPU, but the best solution is running the application in multi cores and also in multi machine in parallel. Which are the ultimate aims of most of the application?

Parallel system can make computation faster

File size is increasing but the algorithms don't improve so much. The improvement of using faster CPU but similar algorithms is limited. Moreover, newer format always require more complicated decode algorithms and make the processing time longer. The only way to greatly speed up the process is make the job parallel running on different machines.

Cloud reduce the cost

Since buying expensive facilities for only a single purpose is unreasonable but using the cloud charge only how long has we use. It enables many people to run large scale project with an acceptable cost.

We hope using cloud and some related technologies to implement application which are difficult to implement.

1.2 Project Objective

This project aims to implement large applications with Hadoop (a parallel programming software framework) on a cloud. Instead of programming very fast applications on a few machines, we are focusing on programming applications that has high scalability from a few machine to thousands of machines.

1.3 Idea of our project

Our idea is very simple. Assume there is a very large video data base. Giving a set of video or frames, we hope to find it from that database, and tell the position of that video.

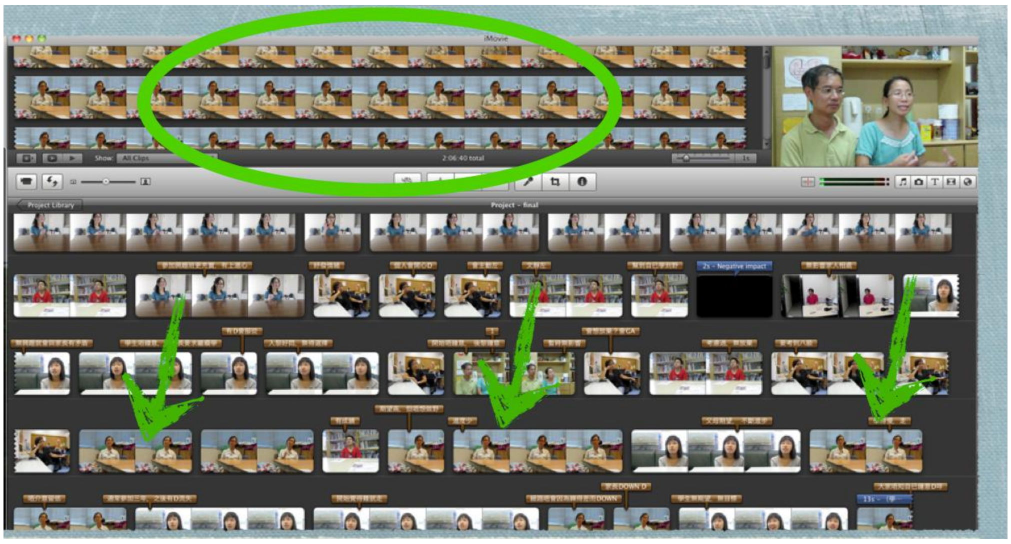


Figure 1 Our idea to search frames from a video database

The idea is simple but it is very useful in different aspects. If we change the algorithm to an object detection algorithm, it can be used in a surveillance video application. If we put face detection algorithm, it become a video diary.

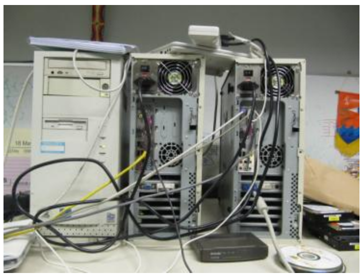
The key point of this project it building application with high scalability. When database is increased, the application can still handle it.

1.3 Project resource

Our project is going to introduce cloud and its performance. View Lab gives us machines to set up a cloud.

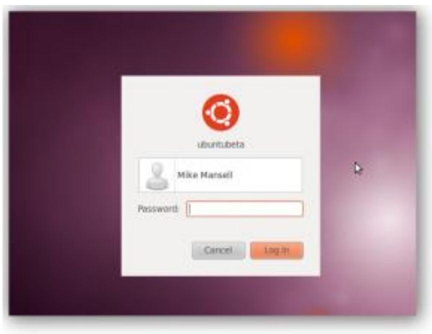
1.3.1Hardware

3 X	Dell Desktop PC: CPU: Intel Pentium D 3Ghz (support VT-x) Memory:1GB Storage:160GB	Be the node server of the cloud. Kvm require the physical hardware support VT-x or AMD-v, these are the

		only machine we can use to build the cloud
5 X	Desktop PC CPU: Pentium 4 2.8Ghz Memory: 2GB Storage: 160-240GB 	One machine plays the roles as cloud controller and all machines has installed Hadoop and FFmpeg. So we can evaluate the performance of cloud and MapReduce, especially the video processing performance.
1 X	D-link wireless router 	It plays the roles as DHCP sever and switch and help setting up a local network
1 X	Asus 1000M switch 	Most of the machines are connected to this switch in order to achieve 1000M bandwidth and eliminate the bottle neck of the network

1 X	Netbook	This is our own netbook. It is connected to the local network and allows us to control the cloud and all other machines from anywhere within the CSE network.
-	CAT 5E cables	

1.3.2Software and framework



Ubuntu Server Edition 10.10 (64Bits)

Ubuntu Enterprise Cloud (UEC) is the most common platform to set up a private cloud. There are more support and discussion on the internet.

Hadoop

MapReduce is a theory appears in recent year. Hadoop (software framework) is the famous open source project; it is very suitable to implement large scale data-intensive applications.

Figure 2 Logo of Hadoop

Chapter 2: Cloud Computing

2.1 Introduction to cloud's concept

Cloud Computing is a web processing with large amount of resource. The user of the cloud can obtain the service thought network (in both internet and intranet). In other words, users are using or buying computing service from others. The resource of the cloud can be anything IT related. In general, cloud provides application, computation power, storage, bandwidth, database and some technologies like MapReduce. As the resource pool is very large, user can increase the application on cloud to any scale. It is fully under users' control.

2.2 From point of views of resource

2.2.1 Types of cloud

There are four main type of cloud:

1) **Public cloud**: The cloud computing resource is shared outside, anyone can use it and some payment maybe need.
2) **Private cloud**: It is opposite to public cloud, private cloud's resource is limit to a group of people, like a staff of a company etc.
3) **Hybrid cloud**: this is a mixture of previous two clouds, some cloud computing resource is shared outside but some don't.
4) **Community cloud**: this is a special cloud to make use of cloud computing's features. More than one community shares a cloud to share and reduce the cost of computing system. This will be discussing later.

We only focus on the private and public cloud. The typical example of the public cloud is Amazon Web Service (AWS EC2) and WSO2. We will introduce Eucalyptus and AWS later.

Figure 3 WSO2 and Amazon Web Service are common public cloud

2.2.2 Types of cloud service

There three main types of cloud service provided:

1) **Software as a Service (SaaS)** :

Clients can use the software provide by provider. Which usually need not to install and it is usually a one to many service. Like Gmail, search engine.

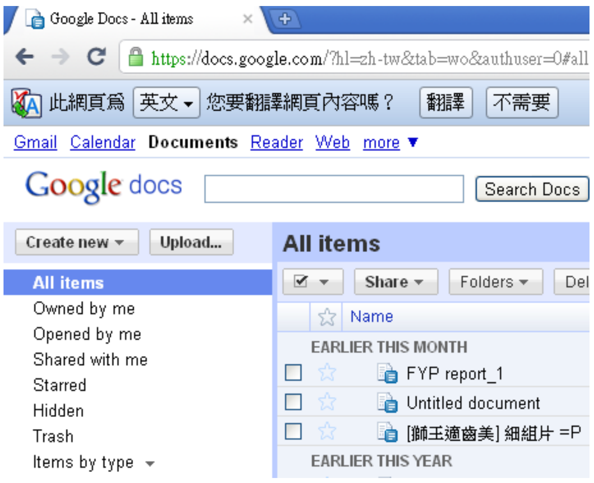

Google Docs - All items
https://docs.google.com/?hl=zh-tw&tab=wo&authuser=0#all
此網頁為 英文 您要翻譯網頁內容嗎？ 翻譯 不需要
Gmail Calendar Documents Reader Web more ▼
Google docs Search Docs
Create new Upload...
All items
All items
Owned by me
Opened by me
Shared with me
Starred
Hidden
Trash
Items by type
Share Folders Del
Name
EARLIER THIS MONTH
FYP report_1
Untitled document
[獅王適齒美] 細組片 =P
EARLIER THIS YEAR

Figure 4 Google doc is one of SaaS cloud

The typical cloud application we used is Google Doc, a powerful web application similar to Microsoft office. It is free for most users and has paid version for the company who want more features.

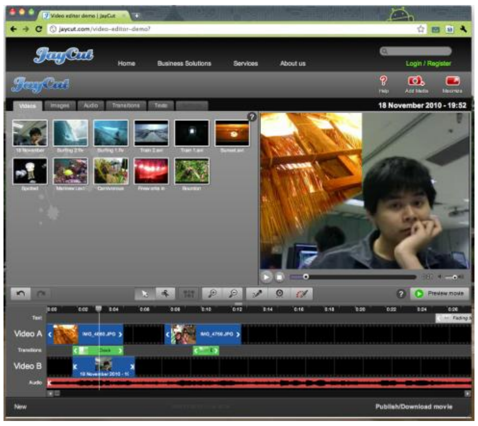


Figure 5 Jaycut, web video editor

Apart from the software we often use in office, there are some more powerful Cloud service like Jaycut and Pixlr.

Jaycut is a free online application implemented with Flash; you can upload movies and edit it. Jaycut is

Picture 1 Jaycut, online movie editor

a very powerful and can do almost all basic effects same as other desktop application. After finishing your work, it can compile the final product with a magical high speed and give you a download link. You can deliver the movies by passing the links to others and need not to worry about the storage problem.

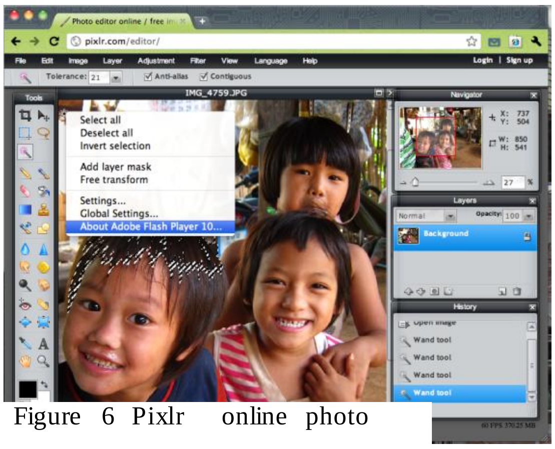
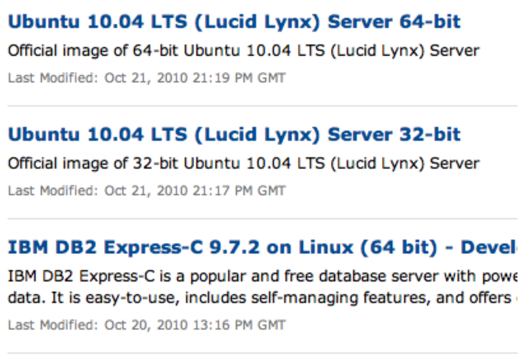


Figure 6 Pixlr online photo editor

Pixlr is an online version of Photoshop. It is also free and has many features similar to Photoshop e.g. layers and steps management. Actually, it is more user friendly than the Photoshop.

2) **Platform as a Service (PaaS)**: Clients can run their own applications on the platform provided; General platforms are Linux and Windows.

3) **Infrastructure as a Service (IaaS)**: Client can put their own operation system on cloud. For example, user can put an optimized Linux for networking ability.



Ubuntu 10.04 LTS (Lucid Lynx) Server 64-bit
Official image of 64-bit Ubuntu 10.04 LTS (Lucid Lynx) Server
Last Modified: Oct 21, 2010 21:19 PM GMT

Ubuntu 10.04 LTS (Lucid Lynx) Server 32-bit
Official image of 32-bit Ubuntu 10.04 LTS (Lucid Lynx) Server
Last Modified: Oct 21, 2010 21:17 PM GMT

IBM DB2 Express-C 9.7.2 on Linux (64 bit) - Devel
IBM DB2 Express-C is a popular and free database server with powe
data. It is easy-to-use, includes self-managing features, and offers
Last Modified: Oct 20, 2010 13:16 PM GMT

Picture 2 Pre-installed image provided by AWS

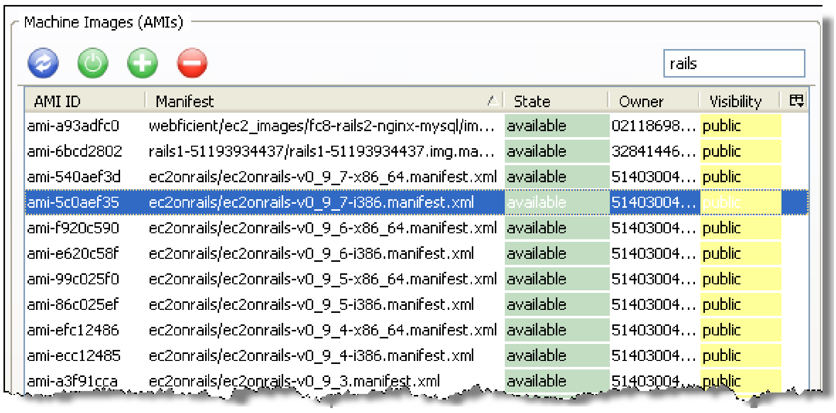
In practical usage, PaaS and IaaS are actually very similar with the difference that whether the image is provided by user or not. If you use the image provided, it is PaaS, otherwise, it is IaaS.

Cloud usually provides many images with pre-set-up environment like SQL server and PHP. Users treat it as online-shopping and buying the IT service.

2.2.3 User interface

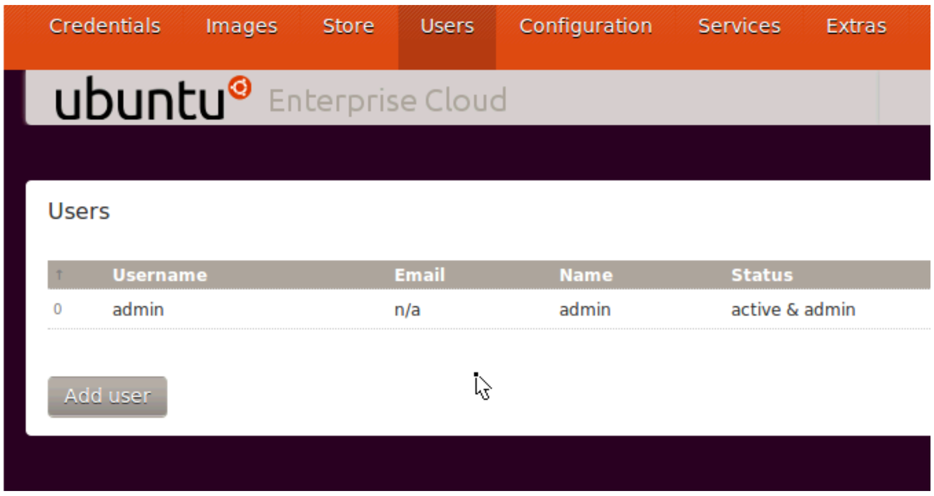
Apart from access the application with web browsers. There are many other tools are developed to provide user-friendly interface. For example, Eucalyptus and AWS EC2 can manage instance control with Elastic Fox with is developed by amazon. And Eucalyptus provides the web interface to access the Configuration of the cloud.

Terminal of course is a very useful interface to control the cloud. It is also the best way to handle much complex control like running an instance with other



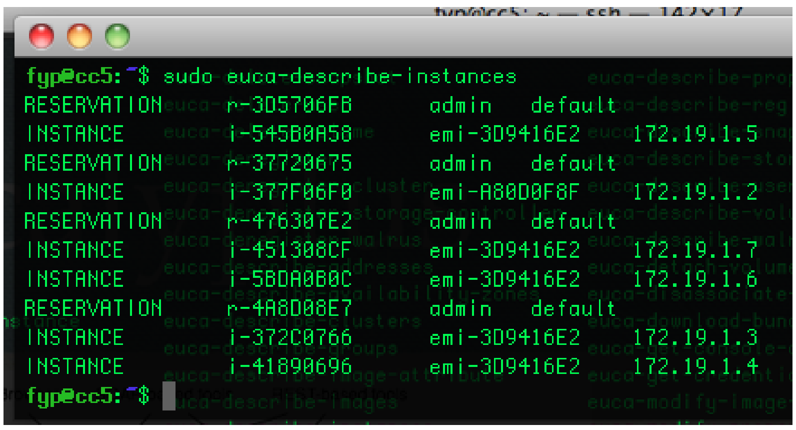
AMI ID	Manifest	State	Owner	Visibility
ami-a93adfc0	webficient/ec2_images/fc8-rails2-nginx-mysql/im...	available	02118698...	public
ami-6bcd2802	rails1-51193934437/rails1-51193934437.img.ma...	available	32841446...	public
ami-540aef3d	ec2onrails/ec2onrails-v0_9_7-x86_64.manifest.xml	available	51403004...	public
ami-5c0aef35	ec2onrails/ec2onrails-v0_9_7-i386.manifest.xml	available	51403004...	public
ami-f920c590	ec2onrails/ec2onrails-v0_9_6-x86_64.manifest.xml	available	51403004...	public
ami-e620c58f	ec2onrails/ec2onrails-v0_9_6-i386.manifest.xml	available	51403004...	public
ami-99c025f0	ec2onrails/ec2onrails-v0_9_5-x86_64.manifest.xml	available	51403004...	public
ami-86c025ef	ec2onrails/ec2onrails-v0_9_5-i386.manifest.xml	available	51403004...	public
ami-efc12486	ec2onrails/ec2onrails-v0_9_4-x86_64.manifest.xml	available	51403004...	public
ami-ecc12485	ec2onrails/ec2onrails-v0_9_4-i386.manifest.xml	available	51403004...	public
ami-a3f91cca	ec2onrails/ec2onrails-v0_9_3.manifest.xml	available	51403004...	public

Figure 8 Elasticfox a tool to control AWS liked cloud



↑	Username	Email	Name	Status
0	admin	n/a	admin	active & admin

Figure 8 Ubuntu cloud's web interface



```
fyp@cc5:~$ sudo euca-describe-instances
RESERVATION     r-3D5706FB      admin   default
INSTANCE        i-545B0A58      emi-3D9416E2    172.19.1.5
RESERVATION     r-37720675      admin   default
INSTANCE        i-377F06F0      emi-A80D0F8F    172.19.1.2
RESERVATION     r-476307E2      admin   default
INSTANCE        i-451308CF      emi-3D9416E2    172.19.1.7
INSTANCE        i-5BDA0B0C      emi-3D9416E2    172.19.1.6
RESERVATION     r-4A8D08E7      admin   default
INSTANCE        i-372C0766      emi-3D9416E2    172.19.1.3
INSTANCE        i-41890696      emi-3D9416E2    172.19.1.4
fyp@cc5:~$
```

Figure 9 Terminal is the most common user control

2.2.4 Why cloud is so powerful

First we define physical machines as cloud servers, and instance or virtual machine (VM) as the virtual server provided to the users.

Resource pool is large

Public cloud gives users surprise with its elastic property and giant resource pool with thousands of machines. With a few line of commands, user can use these machines to do anything they wanted to do.

Easy to manage

Private cloud gives users surprise with its easiness of managing resource and better use of resource. Private cloud users can send request similar to the public cloud users. Apart from that, they can almost get what they want immediately. All these actions are handled by cloud system and administrators need not to set up any new machines.

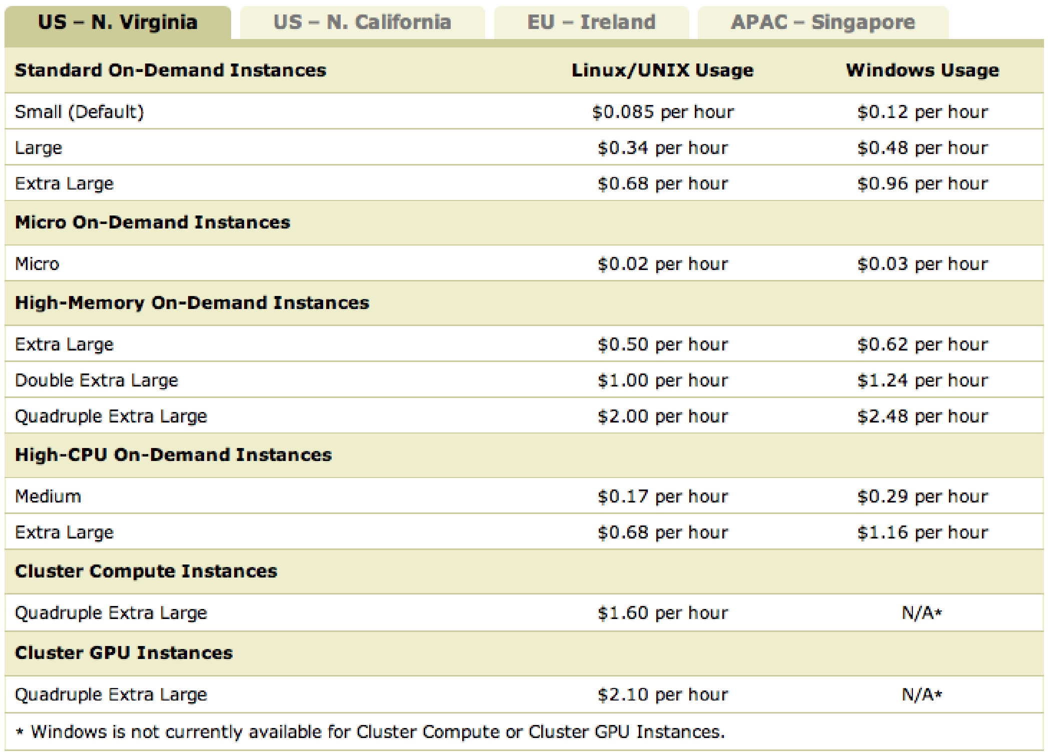
Virtualization allows physical machine divide resource and plays many roles

General cloud (Eucalyptus and AWS EC2) heavily use the virtualization technology and run many virtual machine on a single physical machine. Each virtual machine obtain environment with different amount of resource, like Size of RAM, Storage and number of virtual CPU cores. The resource allocated can be very small and extremely large.

2.2.5 Usage of public cloud

Public cloud usually charge user per user hour with a reasonable price. Within $20 dollar

Apart from the resource allocated can be adjust, platforms the instance running can be difference also like Windows, Linux. Since the instance is a virtual machine only, some application can be pre-installed inside the image and provide more different type of images.

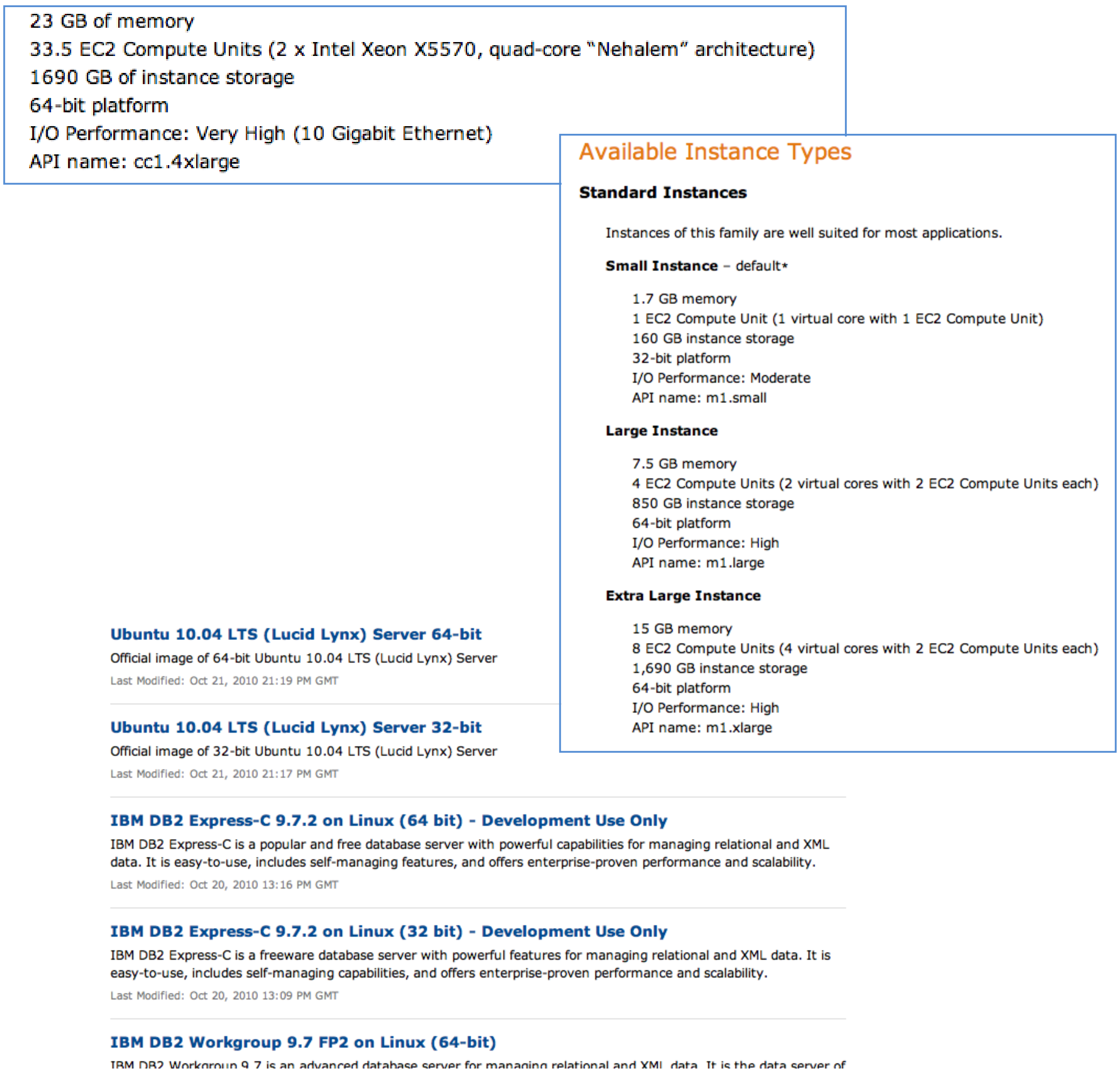


US – N. Virginia | US – N. California | EU – Ireland | APAC – Singapore

Standard On-Demand Instances	Linux/UNIX Usage	Windows Usage
Small (Default)	$0.085 per hour	$0.12 per hour
Large	$0.34 per hour	$0.48 per hour
Extra Large	$0.68 per hour	$0.96 per hour
Micro On-Demand Instances		
Micro	$0.02 per hour	$0.03 per hour
High-Memory On-Demand Instances		
Extra Large	$0.50 per hour	$0.62 per hour
Double Extra Large	$1.00 per hour	$1.24 per hour
Quadruple Extra Large	$2.00 per hour	$2.48 per hour
High-CPU On-Demand Instances		
Medium	$0.17 per hour	$0.29 per hour
Extra Large	$0.68 per hour	$1.16 per hour
Cluster Compute Instances		
Quadruple Extra Large	$1.60 per hour	N/A*
Cluster GPU Instances		
Quadruple Extra Large	$2.10 per hour	N/A*

* Windows is not currently available for Cluster Compute or Cluster GPU Instances.

Figure 10 the price list of AWS


23 GB of memory
33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core "Nehalem" architecture)
1690 GB of instance storage
64-bit platform
I/O Performance: Very High (10 Gigabit Ethernet)
API name: cc1.4xlarge

Available Instance Types

Standard Instances

Instances of this family are well suited for most applications.

Small Instance – default*

1.7 GB memory
1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit)
160 GB instance storage
32-bit platform
I/O Performance: Moderate
API name: m1.small

Large Instance

7.5 GB memory
4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)
850 GB instance storage
64-bit platform
I/O Performance: High
API name: m1.large

Extra Large Instance

15 GB memory
8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)
1,690 GB instance storage
64-bit platform
I/O Performance: High
API name: m1.xlarge

Ubuntu 10.04 LTS (Lucid Lynx) Server 64-bit
Official image of 64-bit Ubuntu 10.04 LTS (Lucid Lynx) Server
Last Modified: Oct 21, 2010 21:19 PM GMT

Ubuntu 10.04 LTS (Lucid Lynx) Server 32-bit
Official image of 32-bit Ubuntu 10.04 LTS (Lucid Lynx) Server
Last Modified: Oct 21, 2010 21:17 PM GMT

IBM DB2 Express-C 9.7.2 on Linux (64 bit) - Development Use Only
IBM DB2 Express-C is a popular and free database server with powerful capabilities for managing relational and XML data. It is easy-to-use, includes self-managing features, and offers enterprise-proven performance and scalability.
Last Modified: Oct 20, 2010 13:16 PM GMT

IBM DB2 Express-C 9.7.2 on Linux (32 bit) - Development Use Only
IBM DB2 Express-C is a freeware database server with powerful features for managing relational and XML data. It is easy-to-use, includes self-managing capabilities, and offers enterprise-proven performance and scalability.
Last Modified: Oct 20, 2010 13:09 PM GMT

IBM DB2 Workgroup 9.7 FP2 on Linux (64-bit)
IBM DB2 Workgroup 9.7 is an advanced database server for managing relational and XML data. It is the data server of

Figure 12 Pre-installed images of AWS

Actually, cloud is an IT resource hiring service; there are many usage of the cloud. The aim target of using cloud is reducing the cost. General public cloud allows user pay as you go and gets the only service they wanted.

Using public cloud as the only IT support

For example purely using public cloud can reduce or eliminate the cost of manage an IT system. Since company must put many resource on maintenances other than buying the facilities. Cloud providers usually provide many manage tool and some of them are free. These tools can manage the resource using according from usage and adjust number of instance automatically. Also some detailed report will be sent to user and alert the any issue their instances encountered.

Using expensive facilities

Also, cloud can also provide some expensive facilities with a reasonable price. For example, if you need a few high performance for some experiments for a few day. It does not make sense to buy this machine only for this single purpose. Let see *Picture 7*, AWS provide you with the most powerful machine with GPU supported and changes.

Using cloud to as part of the IT resource

In some situation, Company's bandwidth and storage may not enough to serve the user over the world. They may use cloud as storage as cloud pool usually has very large bandwidth.

Public cloud is a service

As a conclusion, cloud is a service and it is a business term rather than technical term. It separates the concern between the user and provider. The cloud is very powerful as it uses virtualization and has a giant resource pool. Using cloud can reduce the cost on IT management.

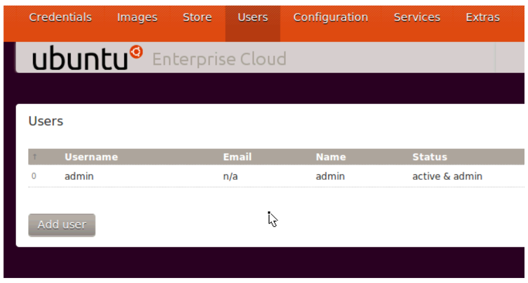
Many people misunderstood cloud

Cloud is a term make many people confuse. It seems to be the way to solve large scale problem and change the instance's computation power more than his host machine. Actually, a virtual machine running on a physical machine cannot run faster than the host in most case (Exception found in testing procedure, and discuss in the part of KVM). To make instance more powerful than it, the only way it using distributed system or similar framework like Hadoop. We will introducing the computation power of virtual machine later and also introduce Hadoop which is important part of our project.

2.2.6 Private cloud

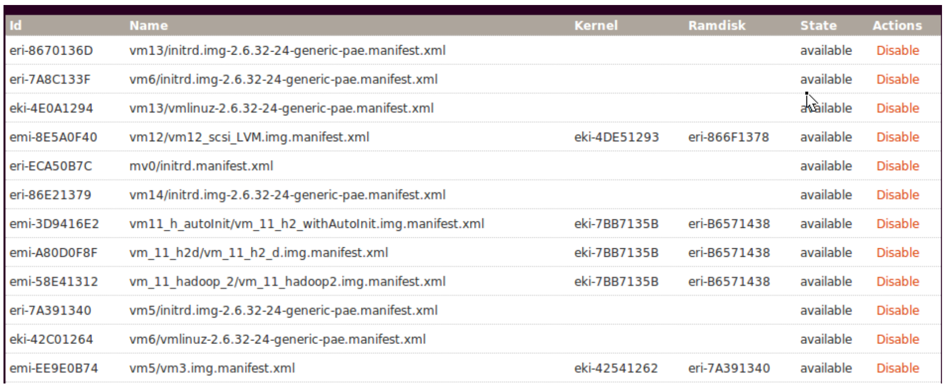
Owning a cloud is not the only right of big company. But of course, private cloud is weaker than public because of smaller resource pool.

These are a few open source cloud project and Eucalyptus is the most famous one in Linux environment. In our project, we chose to use Ubuntu Enterprise Cloud (UEC) as it has more support and documentation. Before we introduce Eucalyptus, we will introduce private cloud.



Users

↑	Username	Email	Name	Status
0	admin	n/a	admin	active & admin



Id	Name	Kernel	Ramdisk	State	Actions
eri-8670136D	vm13/initrd.img-2.6.32-24-generic-pae.manifest.xml			available	Disable
eri-7A8C133F	vm6/initrd.img-2.6.32-24-generic-pae.manifest.xml			available	Disable
eki-4E0A1294	vm13/vmlinuz-2.6.32-24-generic-pae.manifest.xml			available	Disable
emi-8E5A0F40	vm12/vm12_scsi_LVM.img.manifest.xml	eki-4DE51293	eri-866F1378	available	Disable
eri-ECA50B7C	mv0/initrd.manifest.xml			available	Disable
eri-86E21379	vm14/initrd.img-2.6.32-24-generic-pae.manifest.xml			available	Disable
emi-3D9416E2	vm11_h_autoInit/vm_11_h2_withAutoInit.img.manifest.xml	eki-7BB7135B	eri-B6571438	available	Disable
emi-A80D0F8F	vm_11_h2d/vm_11_h2_d.img.manifest.xml	eki-7BB7135B	eri-B6571438	available	Disable
emi-58E41312	vm_11_hadoop_2/vm_11_hadoop2.img.manifest.xml	eki-7BB7135B	eri-B6571438	available	Disable
eri-7A391340	vm5/initrd.img-2.6.32-24-generic-pae.manifest.xml			available	Disable
eki-42C01264	vm6/vmlinuz-2.6.32-24-generic-pae.manifest.xml			available	Disable
emi-EE9E0B74	vm5/vm3.img.manifest.xml	eki-42541262	eri-7A391340	available	Disable

Figure 13 image manage systemof our cloud

Private clouds usually provide only virtual machine and storage as the service. Private cloud has features very similar to the public cloud, private cloud user can get the service similar to the public user, but with less amount of resource.

Other than users can obtain certain benefits, administrator will feel cloud is powerful in point of view of management. It reduces the management cost and better use the IT resource.

Convenience

Private cloud usually provides different types of the user interfaces to the cloud user similar to the public cloud. For the administrators, there is another interface to manage the whole system. For example, user management and instance type is interface we use most in this project.

A single machine play different role

Cloud server generally provides the instance in form of virtual machine. Since we are not always require a very powerful machine. Cloud server can run many small instances on a machine and greatly reduce the cost on buying server. Apart from reduce the number of machines, the IT resource of instance can be dynamically adjust (reboot required like virtual box) or move to another machines even public cloud.

Fast virtual machines set up

Just imagine you are setting up 100 machines to run MapReduce, you may set up the machine one by one and repeat the jobs. It is surely a slow task. In the wrest case, if you need to install new software in each machine like FFmpeg (Encoder we used in this project), you must use a long time to modify such minor change in each machine. Using cloud is totally another case. We implemented a MapReduce program of movies conversion which required FFmpeg. We only need to modify a single image and upload to each server. It requires a few of hours only.

In our project, we only use a few hours to set up a Hadoop cluster with cloud, but almost a whole day in physical machine. Virtual machine environment also help us to prevent the technical driver.

Easy to replace the hardware

Generally, operation system binds to the physical machine. Replacing hardware usually requires reinstall all operation system, driver, applications and configuration. It surely cost a long time.

In cloud server, the image and the physical machine sis separate by the virtualization, so that even though the physical is replaced, the image can still obtain similar running environment. So whenever a hardware facility is replaced, it replaces only the computation power. After installing the virtualization software like KVM and XEN (two technologies widely applied in cloud computing), images can still run in the new server. Since administrators need not to worry about the configuration of image, they can upgrade and replace the system faster and easier.

Better use of the resource

Since the cloud is the resource pool shared through network and can be elastically adjusted. Administrator can allocate suitable amount and types of resource to different users according to their need. For example, it an application handling text data, can use less memory and CPU cores; an application simulating 3D world can allocation many more powerful instance

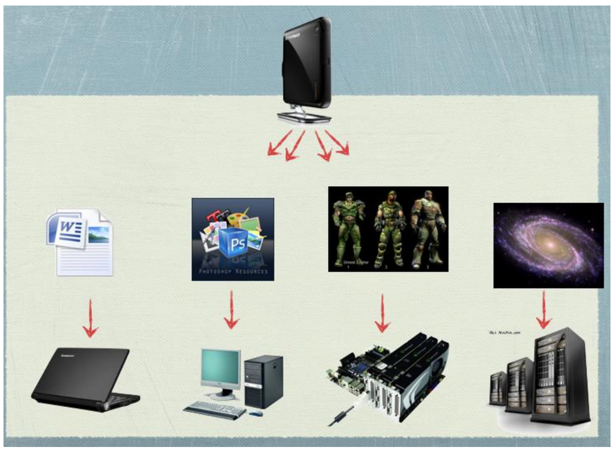


Figure 14 A office solution with cloud

even with GPU support.

A cloud company provides a solution that: every employee use only a very tiny PC and remote connect to the virtual machine. Each staff can have different type of department can have different type of configuration of virtual machine and can get a more powerful machine if necessary. Whenever the system need upgrade, if only upgrade the cloud server and images. The rest of the system keeps unchanged.

Security

Security is surely an important issue. Virus can easily propagate to other program within a server. It can cost a very serious damage to the server requires times to remove the infected file. Apart from virus, some application may crash and used up all the CPU resource. This prevents other program work properly.

Using cloud can separate the application with different virtual machine within one physical host. Since the virtual machine is independent, the infection change can significantly decrease.

A virtual machine can share limited number of virtual core. Even though an application hanged and used up its resource, other machine won't be affected and work properly.

2.3 From point of views of application structure

2.3.1 Using Cloud Layer

Usually, a cloud provide user with virtual machines. A virtual machine can be independent to others, just same as the servers inside the server room. But how can a cloud be elastic and fit to users' business?

There are two main approaches, but both of them do it by adjusting the number of virtual machines running but with two totally different mechanisms.
The first approach is giving the different virtual machine with different roles. The typical example is Microsoft Azure. Let take a large scale web application similar to YouTube as an example. The duty of the application are allowing user to upload the video and play the video.

Azure is actually a cloud layer running on a number of windows sever. It contains many technologies like crash handling, and workload distributed system. If a machine crash, the application can still run on other server and the database won't lose.

Azure requires programmers set up one or more roles (type of virtual machines) under some framework in the cloud application. Each role processes different types of job. For example:

a) Web Role process the website query, e.g. Display videos
b) Worker role process the background job, e.g. convert the videos to different format.
c) VM role are used by administrators to manage the application.

Such configuration even allows the role can run some large scale process in more than a virtual machine parallel.

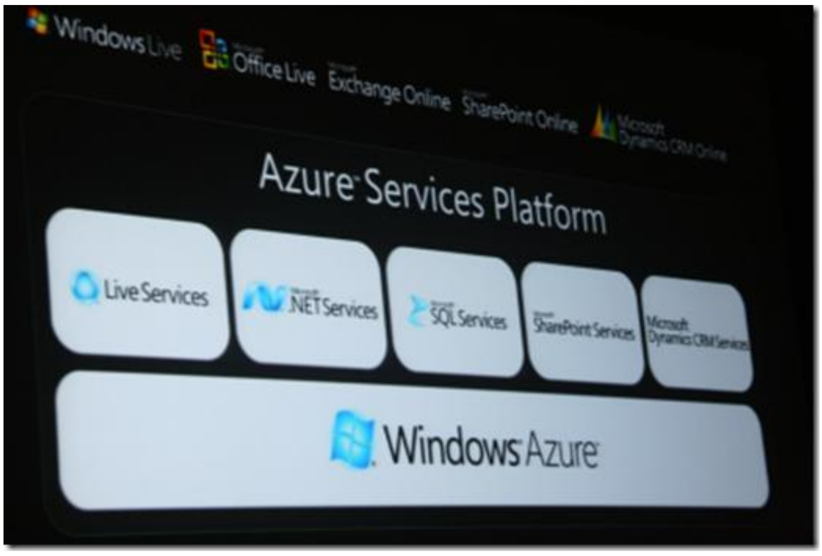


Figure 15 Windows Azure is a type of cloud provide a platform with special framework

The workload of each virtual machine is balanced by a Load Balancer. No of the virtual machines of each node can be adjust depending on the situation. For example, only a few users upload the video at mid-night, so, fewer virtual machines of worker role are used. This greatly reduces the cost of hiring the cloud service.

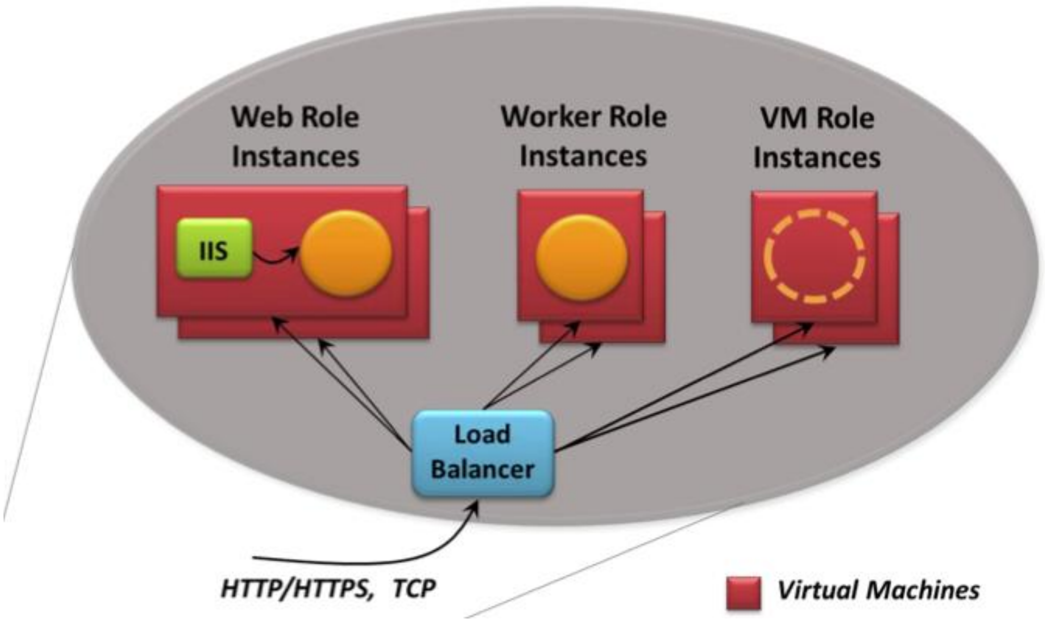


Figure 16 Architecture of an azure application

This approach cloud user to adjust the resource fit to the actual need. However, Azure require programmer to modify some features of the applications. In the worst case, it is necessary to redevelop the whole application. And the applications usually can't run on platform other than Azure.

2.3.2 Using Cloud instance with other framework

Another approach is migrating application directly into the cloud with form of virtual machine. Administrator is managing the system by managing the virtual machines. The typical example is AWS EC2. Cloud can run different number of machine on different image similar to the previous approach. However, AWS is providing only the instance but not a cloud layer for each user. If a server crash, no handler will take up the jobs. This approach requires the application has elastic property and handle the crash of the system. In order to has the property of elastic and handling crash of the system automatically. Some other technology like distributed system or MapReduce.

Figure 17 Azure is similar to instance with Hadoop

technology, Hadoop is the MapReduce application framework) as a layer to connect all machine.

2.4 Summary of concept of Cloud

Private cloud is a way to manage the system, private cloud is a business term rather than technical term. Cloud has feature of elastic and virtualization. The most important benefit of cloud is reducing cost, but not it powerfulness. The power of the cloud is upper limited by the size of the resource and heavily relies on the network condition.

2.5 Eucalyptus (UEC)

2.5.1 Principal of eucalyptus cloud

There is different cloud architecture and in general, a cloud will not provide physical machine but virtual machine or an application service to the users. We are introducing the Eucalyptus which is an open source project for private cloud and public cloud.

Eucalyptus has architecture similar to Amazon Elastic Compute Cloud (EC2). Actually, the image of eucalyptus can directly run on EC2 and the commands of these two systems is almost the same?

Eucalyptus highly relies on the Virtualization technology. The cloud usually provides a virtual machine to the user and a physical machine can run more than one virtual machine (instance). Users can run self-created images (operation system with self provided application) or provided default images. The architecture has many advantages:
A physical machine can play the rules of different operation systems, and less physical machines are needed in the network. This greatly reduces the budget of computer system set up.

All application can be independent and be more secure. Each virtual machine has its resource bound. Even though an application used up its resource in a virtual machine or crashed, it will not affect other virtual machine. Since all application is separated, it can help preventing attack from others and virus spreading.

Each virtual machine's resource can adjusted, some virtual machines can obtain more CPU cores, memory. If necessary, the virtual machine can be moved to other physical server and use up almost all the computation power of it.

Whenever a user suddenly wants a more virtual machine, he can directly add it first before buying more machines.

Greatly reduce the system upgrade effort. Since the virtual machine is actually an image file, so it the physical machine is updated, the system can be immigrate by copying the images files. Administrator need not to reinstall each virtual machine once.

Here is the basic architecture of the Eucalyptus, there are five components:

1.	Cloud Controller	Any request from user are receipted and handled by this part. All instances are delivered form it
2.	Walrus	However, it delivers the images to the Cloud Controller instead of the Node Controller. All images are compressed and encrypted.
3.	Cluster Controller	Since the cloud is large, Cluster Controller help share the workload of monitor.
4.	Storage Controller	Storing all disk images, kernel images, ram disk images and User storage here. This is the additional part of Eucalyptus. When an instance is terminated, Eucalyptus will not save the data inside, it need an external component to save these data.
5.	Node Controller	It generates the instances with Virtualization technology like KVM, XEN. (In Ubuntu Enterprise Cloud, it support only KVM)

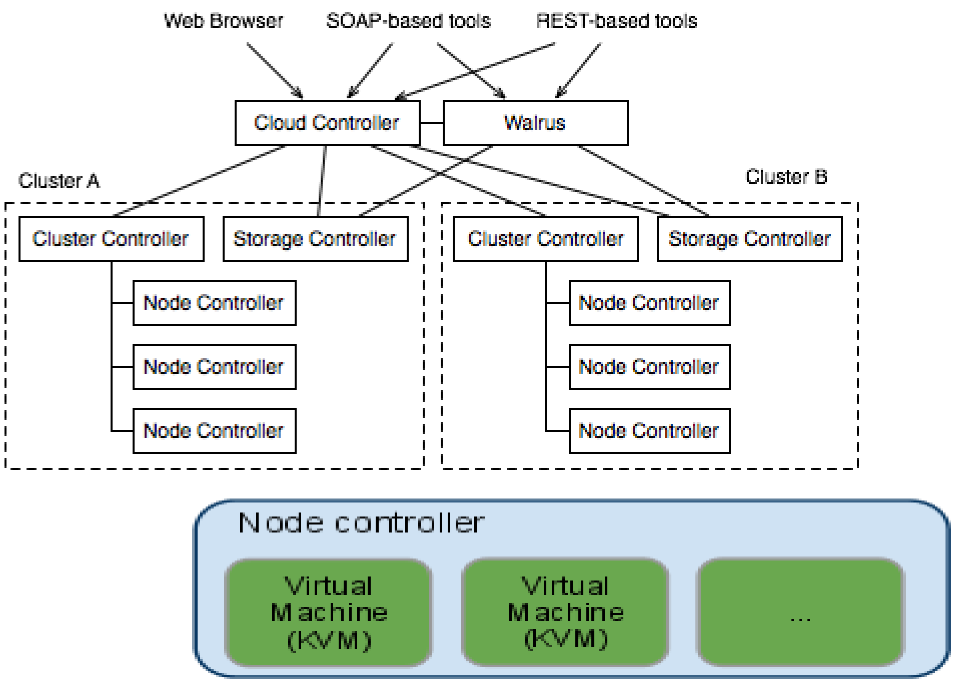


Figure 18 Design of eucalyptus cloud

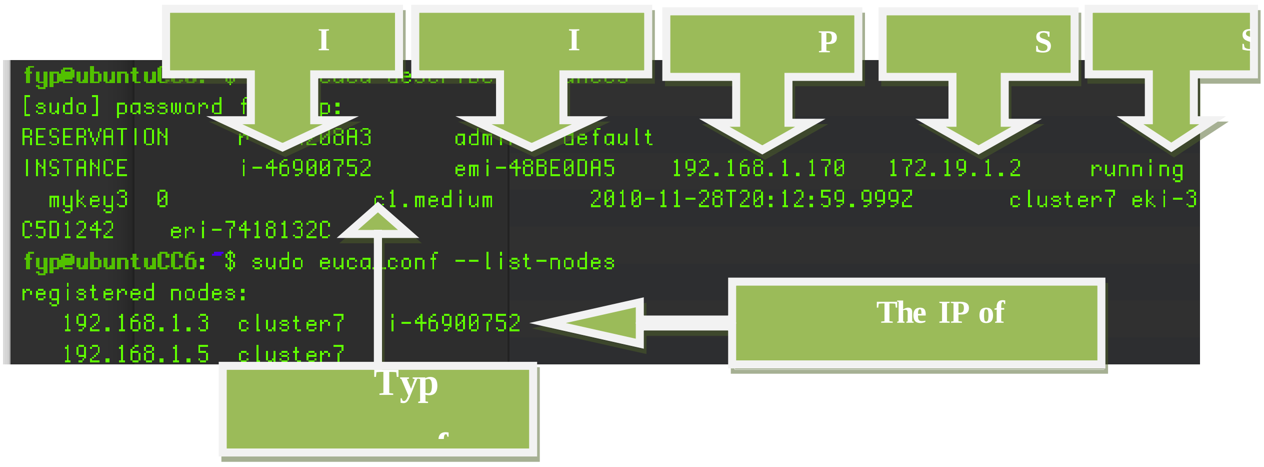
2.5.2 Our private cloud

We built our first private cloud with 4 old machines, the hardware equipment are shown in previous discussion. The most important requirement is having some CPU which supports VT-x.

After building a cloud, we built many images to test the actually performance of the cloud. Each image requires three file: kernel, ramdisk, and hard disk image. Each of these file has an image ID.

```
MAGE   eri-7A78133E    vm6
MAGE   eri-B834143E    vm1
MAGE   eki-4D591297    vm1
MAGE   emi-A8250FE6    vm_
MAGE   eki-7088131E    vm1
yp@cc4:~$ 
```

Figure 19 Our images list



```
fyp@ubuntu
[sudo] password f       p:
RESERVATION     r    208A3      adm     default
INSTANCE        i-46900752      emi-48BE0DA5    192.168.1.170   172.19.1.2      running
  mykey3  0             c1.medium       2010-11-28T20:12:59.999Z        cluster7 eki-3
C5D1242    eri-7418132C
fyp@ubuntuCC6:~$ sudo euca_conf --list-nodes
registered nodes:
   192.168.1.3  cluster7   i-46900752
   192.168.1.5  cluster7
```

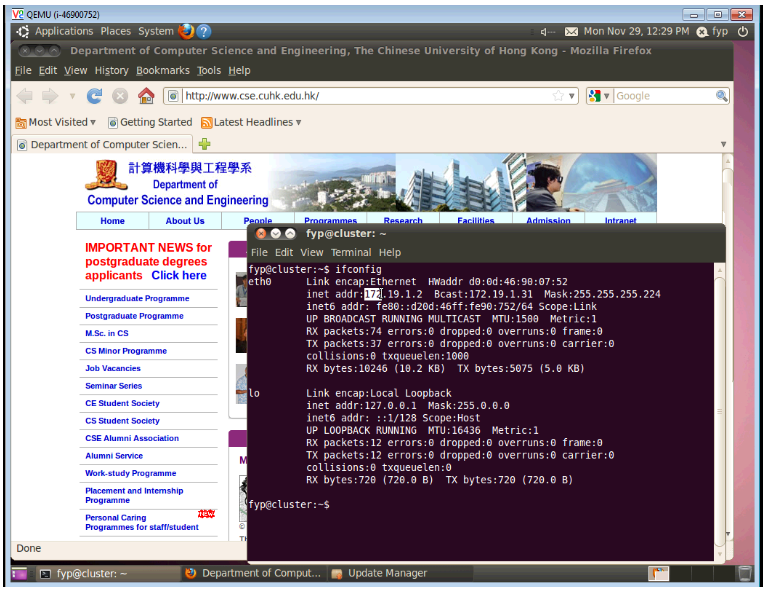
Figure 20 Intance control of our cloud

This is the status of an instance. There are seven important attribute:

Instance ID	This represent the ID of the virtual machine running, it is random generate by default.
Image ID	The original virtual uploaded to the cloud
Public IP	Use this IP to connect this virtual machine. This IP is bind to the cloud controller. The network traffic is first going to the cloud controller and than the host (node), so the bandwidth is bounded by the bandwidth of cloud controller.
Subnet IP	The virtual machine within the same cloud can communicate within this IP. This is a virtual IP bind to the host. So it is only bounded by the host.
Status	It show three status: Pending : preparing the instance Running: The boot signal is sent to kvm Terminated: the instance is killed
The IP of host	Using this IP to get the display of the instance
Type of instance	It refer to the configuration of the instance, e.g. no of virtual CPUs.

The instance is similar to the virtual machine in virtual and VMware. It can get the internet service like real machine. But instance only know his subnet IP but no idea of the public IP.

We can view the display of an instance by making a connection to the host with VNC viewer.

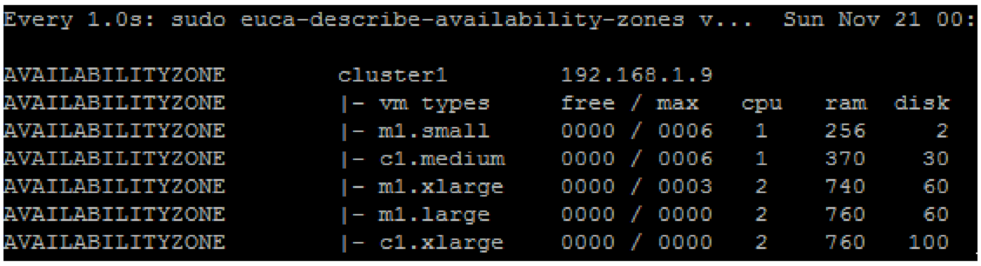


```
fyp@cluster:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr d0:0d:46:90:07:52
          inet addr:172.19.1.2  Bcast:172.19.1.31  Mask:255.255.255.224
          inet6 addr: fe80::d20d:46ff:fe90:752/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:74 errors:0 dropped:0 overruns:0 frame:0
          TX packets:37 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10246 (10.2 KB)  TX bytes:5075 (5.0 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:720 (720.0 B)  TX bytes:720 (720.0 B)

fyp@cluster:~$
```

Figure 21 A instance with internet connect



```
Every 1.0s: sudo euca-describe-availability-zones v...   Sun Nov 21 00:

AVAILABILITYZONE        cluster1        192.168.1.9
AVAILABILITYZONE        |- vm types     free / max   cpu   ram  disk
AVAILABILITYZONE        |- m1.small     0000 / 0006   1    256     2
AVAILABILITYZONE        |- c1.medium    0000 / 0006   1    370    30
AVAILABILITYZONE        |- m1.xlarge    0000 / 0003   2    740    60
AVAILABILITYZONE        |- m1.large     0000 / 0000   2    760    60
AVAILABILITYZONE        |- c1.xlarge    0000 / 0000   2    760   100
```

Figure 22 Specification of our instance

2.5.3 Technical trouble shoot

2.5.3.1 Image fails to run a image

After the cloud is set up, we found the cloud fail to run any images even the default images. It shows pending and then terminated with not error message displayed. The reason is that the CPUs do not support VT-x and KVM. There are two ways to solve this problem: use centOS and run xen instead of KVM; using the machine with VT-x support.

At the end, department give us three machines, so that we can install the cloud and run it. So we don't need to learn to set up cloud in a new environment.

2.5.3.2 Fail to add node server

After installed all required OS in Cluster Controller and nodes. The node can't discover and add any nodes. Only a few discussion on this problem, and we found this is a bug of Eucalyptus 1.6.

Eucalyptus doesn't release any guideline in handling this problem, but it can be fixed with a few simple steps by resting the cloud.

1) Clear all the clusters and nodes in Cluster Controller.

2) Reboot all machines

3) Clear all the .ssh/Known host record

4) Let Cluster Controller discover the nodes again.

2.5.3.3 Fail to de-registered

This is a bug similar to previous problem and can be fixed with the same way to handle previous bug.

2.5.3.4 Instance IPs problem

Whenever any machine's IP changed, the cluster won't work properly.

If Cluster Controller's IP changed, it can execute any "euca-*" command, all the setting of this cluster need to be rested at http://localhost:8443 and the update the credentials.

If Node's IP changed, it will be more complicated. Cluster Controllers may not find the node. The Cluster Controller will not add any other node which is using this IP, and the node with changed IP will be used and the certificate in the Cluster Controller doesn't accept the new IP. Cluster Controllers may give you some warning about some attacks. To fix this problem, you can let Cluster Controllers forget the old nodes by remove the "known hosts" file in Cluster Controller's

//var/lib/eucalyptus/.ssh (.ssh is a hidden folder)

The best way to handle this problem is bind the IPs with MAC address.

2.5.3.5 Connection fail to Instance

A key is required to access the Ubuntu instance and ssh client wills memory the IPs and public keys. If a key is used in different IPs, it will be treated as an attack from others.
This problem can be fixed by generate more keys for different IP or remove the record of the SSH client by remove the “known_hosts” file in ~/.ssh (.ssh is a hidden folder).

2.5.3.6 Slow start up speed **Problem**

This is mainly the problem of the network bandwidth and the size of the images. We can only build small images (4GB) and replace the cable to CAT-5E and reduce up the network transfer time.

2.5.3.7 Large image file problem

With the limitation of eucalyptus, how large the virtual machine, how large is the image file size. When we want to set up a virtual machine with 40GB storage, it starts up with a very long time.
We found KVM support smaller image file (Only 1/10 the original file size), but eucalyptus doesn’t support.
We did a lot of testing images and found eucalyptus has disabled this function.

2.5.4 Branch mark of our private cloud

To fully understand how cloud work and performance of the cloud. We did a series of testing on both physical machines and virtual machines.

2.5.4.1Startup time

Startup time of the virtual machines is heavily depended on the network status. Before running a virtual machine, the cloud runs a few steps:

1. Check availability of the cloud to check if there is enough resource.
2. Decrypt and decompress the image
3. At the same time, send the image to one of the node.
4. The node will save the image to the cache
5. And copy a new copy and run the virtual machine. The virtualization used

Size of the image	First start up time[1]
3GB	~5mins
5GB	~8mins
10GB	~17mins

Figure 23 Start up time of different instance

When an image is sent to the node server, it will copy will be save it node has sufficient space. Next time if the same image runs on it, it can directly copy the image and save the bandwidth. Second start up requires 1/3 of the first start up time only.

Startup of AWS images require two minutes only.

2.5.4.2 Overall performance

The overall performance of our private cloud (after optimization) is about 73% of the physical machine. It is similar to the value that eucalyptus discussed. The most important factor slowing down the performance is the IO performance of the virtual machine.

[1] It heavily rely on network status and it is an average time.

2.5.4.3IO performance

The IO time of the virtual machines is 1.47 time of the host (physical machine). The disk write speed is only 50.9MB/[2]s; it is just half of the physical machine’s 103MB/s. If two process is writing a hard disk at the same time, virtual machine can write data with speed 26MB/s, (65MB/s in physical machine). We can see write time of the virtual machine can reach to about 50% of the physical machines in our test case. The read disk speed is about 80% of the Physical machine.

The virtual machine can usually share ~50 %(20MB/s) bandwidth of the physical host (~40MB/s). It is fast enough for out project.

Some related report of hard disk branch mark show that virtual machine can reach ~90% performance of the physical machines. It seem to be the problem of configuration or the facilities'

```
#!/bin/bash
# init
function pause(){
 read -p “$*”
}

pause "1).Download Test"
time wget
http://www.cse.cuhk.edu.hk/~csci31
50/mp3/20100913_csci3150.mp3

pause "2).CompressFileTest"
time  tar -zcvf file.tar.gz
20100913_csci3150.mp3
rm -f 20100913_csci3150.mp3

pause "3).DecompressFileTest"
time  tar -zxvf   file.tar.gz
rm -f  file.tar.gz
```

```
pause "4).copyfile"
time cp 20100913_csci3150.mp3
20100913_csci3150_2.mp3
rm -f 20100913_csci3150.mp3
20100913_csci3150_2.mp3

pause "5).Write500MBFile"
time  dd if=/dev/zero bs=1M
count=500  of=~/500MB.file

pause "6).CompressFileTest2"
time  tar -zcvf file.tar.gz
500MB.file

pause "7).DecompressFileTest2"
rm -f 500MB.file
time  tar -zxvf   file.tar.gz
rm -f 500MB.file  file.tar.gz
pause "Finished."
```

Figure 24 Our testing script

[2] 1MB/s = 1MByte/s

2.5.4.4 Computation performance

We use CPU system time and user time to measure actually performance of the instance.

Over all system time of virtual machine is 1.512 x of the host (66% performance of host). The user time of instance is 1.11 x of the host (89.6% performance of host). To fully understand the actually performance of the CPU, we implemented a simple multi-threads programs which use 100% user time only. We find self-define function call can obtain 97% performance of the host machine. This really gives us surprise and guides us to implement the application with more user time than system time.

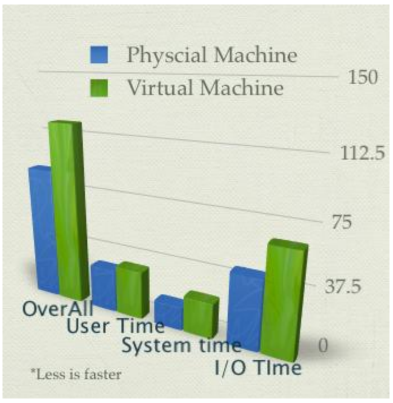
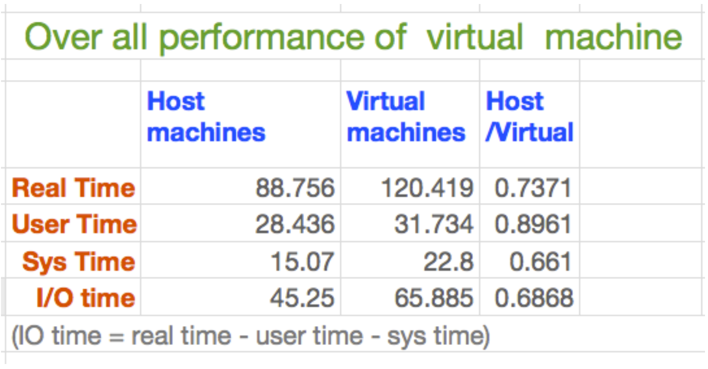


Figure 25 Running time of instance and host

Another testing is running two identical programs in the same host but different instance. The CPU time is almost the same.



Over all performance of virtual machine

	Host machines	Virtual machines	Host /Virtual
Real Time	88.756	120.419	0.7371
User Time	28.436	31.734	0.8961
Sys Time	15.07	22.8	0.661
I/O time	45.25	65.885	0.6868

(IO time = real time - user time - sys time)

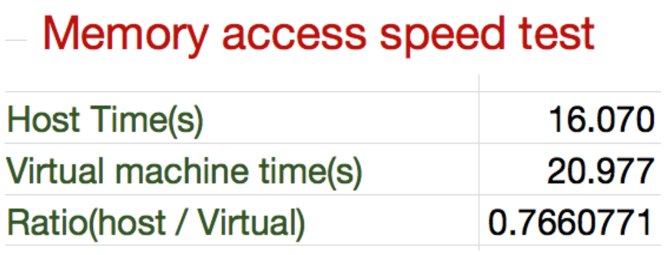
Figure 26 Result of our testing on computation

2.5.4.5 Memory Access time

Another testing of accessing large amount of memory shows that the memory access time of the Virtual machine is relatively slower than the host. It has only 76% of the Host machine. We can see that even VT-x and Virtio(will be discuss later) has improved the performance of virtual machine, virtual machine still can not directly access the physical memory.

```
int nu =__noOfLoop;
int mSize=100*MB;
int i,j;
char * list=malloc(sizeof(char)*mSize);
for(i=0;i<nu;i++)
        for(j=0;j<mSize;j++)
                list[j]++;
free(list);
return NULL;
```

Figure 27 our memory testing program



Memory access speed test

Host Time(s)	16.070
Virtual machine time(s)	20.977
Ratio(host / Virtual)	0.7660771

Figure 28 Result of our testing on memory access

2.5.4.6 Multi-platform

An important feature of cloud is supporting Linux and Windows running on it. After the cloud is set up, we tried to build our own image instead of the default image provided by eucalyptus.

Surely, setting up a Linux Image is easy and same as installing it on a physical machine. The performance of the self- built images is faster than the default image and support GUI Desktop version of cloud. There is a virtual monitor for each virtual machine.

We can install and setup the windows into an image, however, it show kernel panic when the image is upload up the cloud.

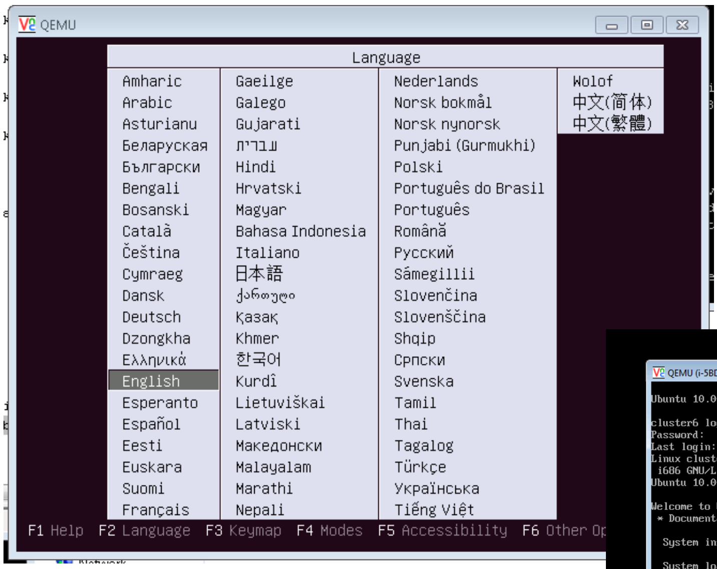
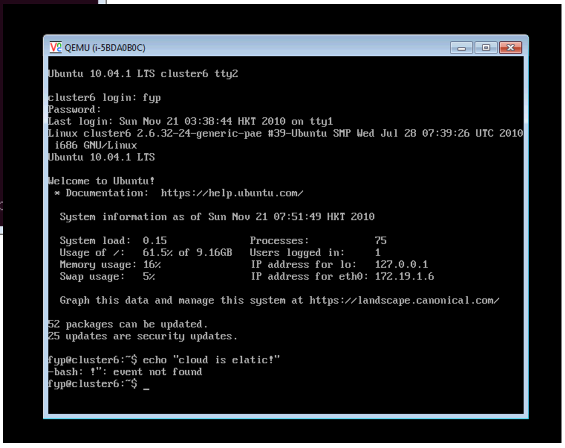


Figure 29 Building a Linux image


```
Ubuntu 10.04.1 LTS cluster6 tty2

cluster6 login: fyp
Password:
Last login: Sun Nov 21 03:38:44 HKT 2010 on tty1
Linux cluster6 2.6.32-24-generic-pae #39-Ubuntu SMP Wed Jul 28 07:39:26 UTC 2010
 i686 GNU/Linux
Ubuntu 10.04.1 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

  System information as of Sun Nov 21 07:51:49 HKT 2010

  System load:  0.15              Processes:           75
  Usage of /:   61.5% of 9.16GB   Users logged in:     1
  Memory usage: 16%               IP address for lo:   127.0.0.1
  Swap usage:   5%                IP address for eth0: 172.19.1.6

  Graph this data and manage this system at https://landscape.canonical.com/

52 packages can be updated.
25 updates are security updates.

fyp@cluster6:~$ echo "cloud is elatic!"
-bash: !": event not found
fyp@cluster6:~$ _
```

Figure 30 A Linux instance running on cloud

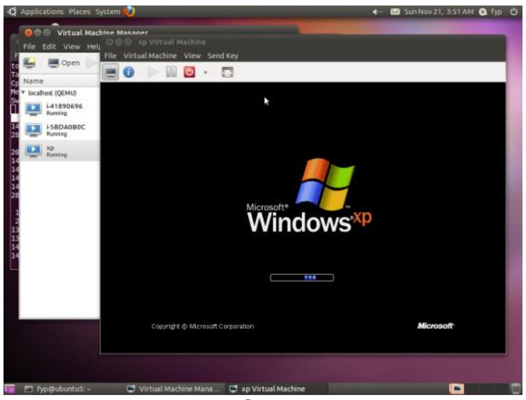


Figure 31 Windows XP running on KVM

2.5.4.7 KVM and XEN

Kvm and xen are the most common open source virtualization technology used in Eucalyptus and AWS EC2. Both of them are using the same image format and compatible to each other. However, their hardware require are different. Kvm requires host

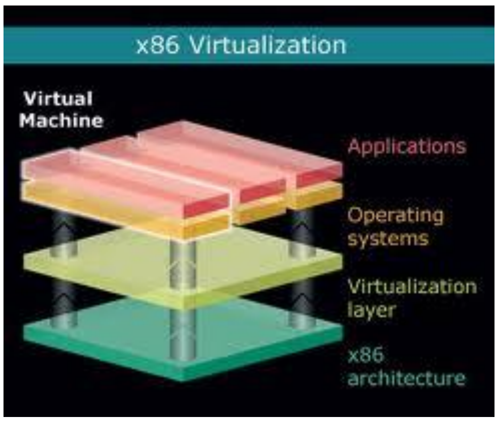


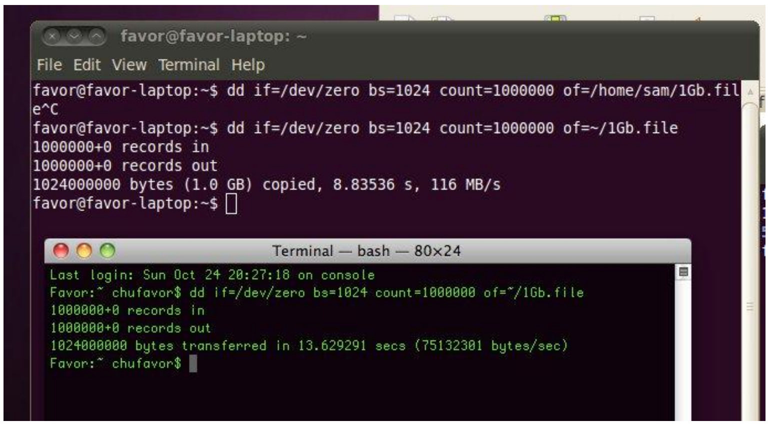
Figure 32 Architecture of AMD-v

CPU support Intel's VT-x or AMD-v. These two technologies are similar and allow the code of a virtual machine can directly run on host's CPU without many change. Xen and kvm both run very fast and kvm has better performance.

Ubuntu Enterprise Cloud doesn't support xen after Ubuntu 9.04 even eucalyptus still support it.

2.5.4.8 Exception in running speed

Virtual machine running slower than host seems to be trivial, but not always true, in some test case, if the host is very powerful and guest (virtual machine) is running some tedious jobs and heavily rely on the I/O, virtual machine can run faster than host. The reason is that virtual environment may cache up the hard disk read/write and greatly reduce the I/O time of the virtual machine.



```
favor@favor-laptop:~$ dd if=/dev/zero bs=1024 count=1000000 of=/home/sam/1Gb.fil
e^C
favor@favor-laptop:~$ dd if=/dev/zero bs=1024 count=1000000 of=~/1Gb.file
1000000+0 records in
1000000+0 records out
1024000000 bytes (1.0 GB) copied, 8.83536 s, 116 MB/s
favor@favor-laptop:~$
```

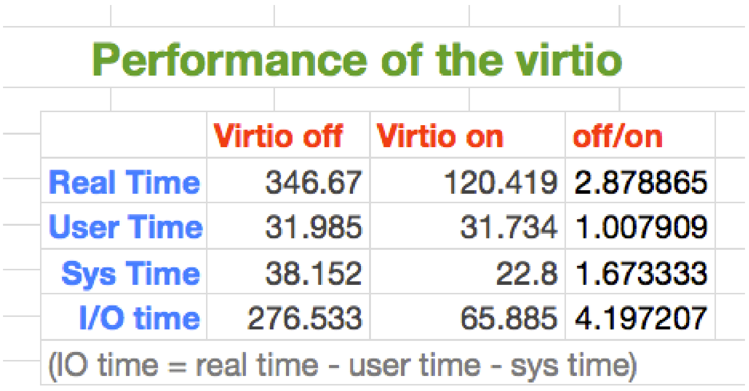
```
Last login: Sun Oct 24 20:27:18 on console
Favor:~ chufavor$ dd if=/dev/zero bs=1024 count=1000000 of=~/1Gb.file
1000000+0 records in
1000000+0 records out
1024000000 bytes transferred in 13.629291 secs (75132301 bytes/sec)
Favor:~ chufavor$
```

Figure 33 an instance runs faster than its host

2.5.4.9 Virtio

Virtualization generally slows down the overall performance of the virtual machine. There are two main technologies to optimize the running speed. The first one is Intel's VT-x or AMD-v. These CPU technologies allow virtual machine run the code directly in the Physical CPU. In previous discussion, we introduce the virtual machine can has almost the same user time of host. Another technology is Virtio[3], it allow the virtual machine directly read/write on the Physical machine.

[3] Eucalyptus default turn Virtio off



Performance of the virtio

	Virtio off	Virtio on	off/on
Real Time	346.67	120.419	2.878865
User Time	31.985	31.734	1.007909
Sys Time	38.152	22.8	1.673333
I/O time	276.533	65.885	4.197207

(IO time = real time - user time - sys time)

Figure 34 Result of testing on IO performance

After turning the Virtio on, Overall IO time reduce to 1/4 of the original. It greatly increases the write disk speed from 13MB/s to 50MB/s and bandwidth from 10MB/s to 20MB/s. The read disk speed is similar to the physical machine.

2.5.4.10 Limitation and weakness of Eucalyptus

Although eucalyptus is a pretty good open source cloud project with many powerful feature. IT has many limitations on it.

The instance won't be save after termination

Eucalyptus allows a user to copy an image to many instances. You can reboot the instance and the data in the hard disk still exist. However, if you terminated the instance, all the data in virtual machines will be freed. So before terminating an instance, user must upload the data to cloud storage like Walrus. Another approach is implementing the application only use the central storage. But this approach requires a very large bandwidth of the local network and may be charged a lot on public cloud. Getting worse, the poor IO performance many reduce the overall performance of the whole system.

Poor instance start up mechanism

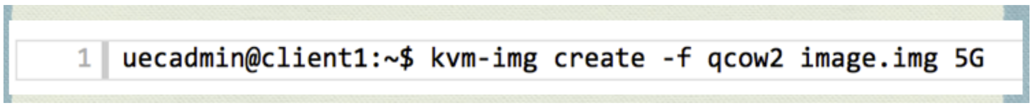
The design of Eucalyptus's instance start up mechanism is quit silly. The images uploaded up the cloud will be encrypted compressed and cut into many blocks. The decrypting and decompressing steps are process in the cloud controller before sending the images out. Whenever many instances (e.g. 10) are pending at the same time, the workload of the control is huge and limited to the bandwidth of the controller machine.

This design can actually be improved greatly by forming a peer to peer network within the nodes. The blocks of the images can be cache in the node if it was running on it. When a machine need to start up that image, just get the blocks of this images form all others having it and decompress on its own. This can greatly reduce the time to start up images and better use the bandwidth.

Large image file problem

The size of an image file is identical to the capacity of the virtual machine. That mean if I a virtual machine has a 10GB virtual hard disk, even 90% of the space is not been used.

Getting worse, this 10GB image must be sent to the node through local network and used up most of the bandwidth.



```
uecadmin@client1:~$ kvm-img create -f qcow2 image.img 5G
```

Figure 35 A command to create a small size image

The menu page of Kvm show that the kvm can create a special image format similar to the format using is Virtual Box and VMware. This format can free the space out space of the holes (large space not being used) in the images. We did many testing on this format and built some image and found that image support 10GB storage requires only 1GB disk space. But eucalyptus doesn't support this format and return error message of hard disk no found.

In our project, virtual machines play the roles of storage and computation and have size of tens of GBs. This surely gives our tiny private a great challenge.

Sensitive to the network

Since the cloud always transfer the images to nodes. It gives the local network a great challenge. Apart form that, if the cloud controller crash, all public IPs of the instance will become invalid and require reboot of all instance.

Non-static IP

Since the number of IPs allocated to the instance is limited. The cloud will reuse the IPs.

2.5.5 Difficulties in setting up private cloud

Setting up a cloud actually require only a few

2.5.6 Cloud Summary

Cloud is actually a very powerful technology. The most significant feature of cloud is elastic and reduces cost. It is not a tool to implement faster application since virtual machines seldom run faster than his host. But virtualization makes manage and share resource to different possible and allocate resource fit to any situation.

However, we have shows some limitation of typical. There are many trades off and silly design in cloud technology, and the performance of the cloud is limited to the power of the hosts and the network stability.

Chapter 3: Apache Hadoop

3.1 Introduction to Hadoop

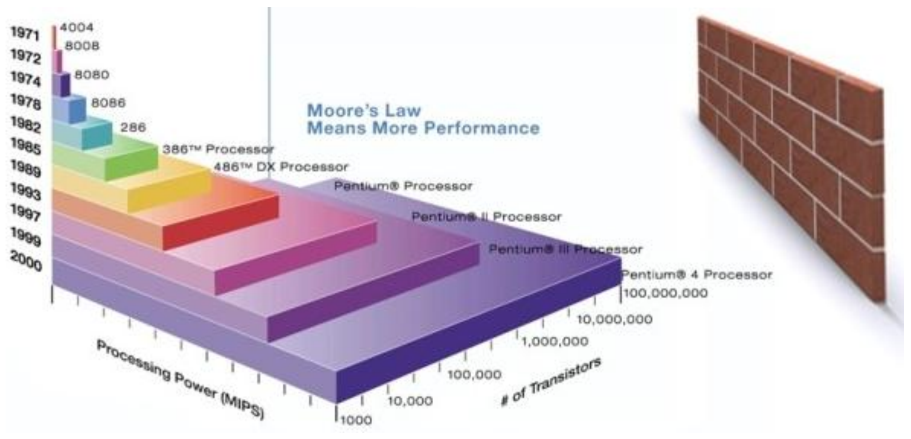


Figure 36 The computation power of machines has a bound

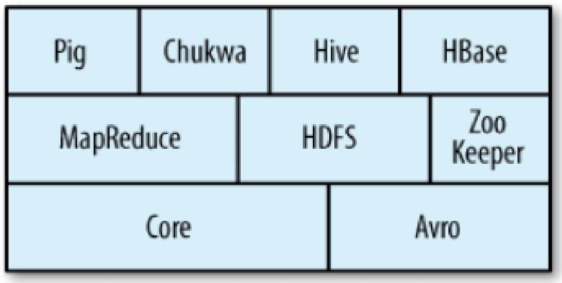


Figure 37 Hadoop family

In 1965, Gordon E. Moore claim that transistor counts had doubled every year. Recent years, the technology has reached the limit of what is possible with one CPU, that's why parallel computing comes, due to the convenience of management, could computing architecture and the most widely-used, open source framework Hadoop comes out.

Apache Hadoop is a software framework that supports data-intensive distributed applications; it enables applications to work with thousands of nodes and petabytes of data.

Hadoop contains

1. The projects Hadoop Common: the common utilities that support the other Hadoop subprojects.
2. Hadoop distribute file system(HDFS),
3. Hadoop MapReduce: a software framework for distributed processing of large data sets on compute clusters.
4. Avro: A data serialization system.
5. Chukwa: A data collection system for managing large distributed systems.
6. HBase: A scalable, distributed database that supports structured data storage for large tables.
7. Hive: A data warehouse infrastructure that provides data summarization and ad hoc querying.
8. Mahout: A Scalable machine learning and data mining library.
9. Pig: A high-level data-flow language and execution framework for parallel computation.

In our project, we mainly focus on HDFS, MapReduce and Hive.

Who is using Hadoop?

facebook
YAHOO!
The New York Times
Adobe
amazon.com
hulu
Baidu 百度
AOL
Microsoft

Hadoop benchmark

In year 2009, Hadoop sorts 100 TB in 173 minutes (0.578 TB/min) on 3452 nodes x (2 Quad core Xeons, 8 GB memory, 4 SATA)

HDFS

Hadoop Distributed File System (HDFS) is an open source storage system used by Hadoop applications. HDFS creates multiple replicas of data blocks and distributes them on compute nodes throughout a cluster to enable reliable, extremely rapid computations; it is designed in fault tolerance.

MapReduce

MapReduce is a programming model and software framework introduced by Google to support distributed computing on large data sets on clusters of computers. Hadoop MapReduce is an open source project from Apache, it allow us develop parallel applications without any parallel programming techniques, and the applications could be easy deployed and executed, it works with the HDFS and processes huge datasets (e.g. more than 1TB) on distributed clusters (thousands of CPUs) in parallel. It allow programmer writing the code easily and fast.

Working Environment:

Currently the newest version of Hadoop is 0.21 which is not well tested, it may get problems when starting the HDFS service, so, we choose the version 0.20, it was supported by Linux environment and tested with 2000 nodes, theoretically speaking, it also works in windows, but rare people tried it. As it is Java based system, so, get Java installed is a must.

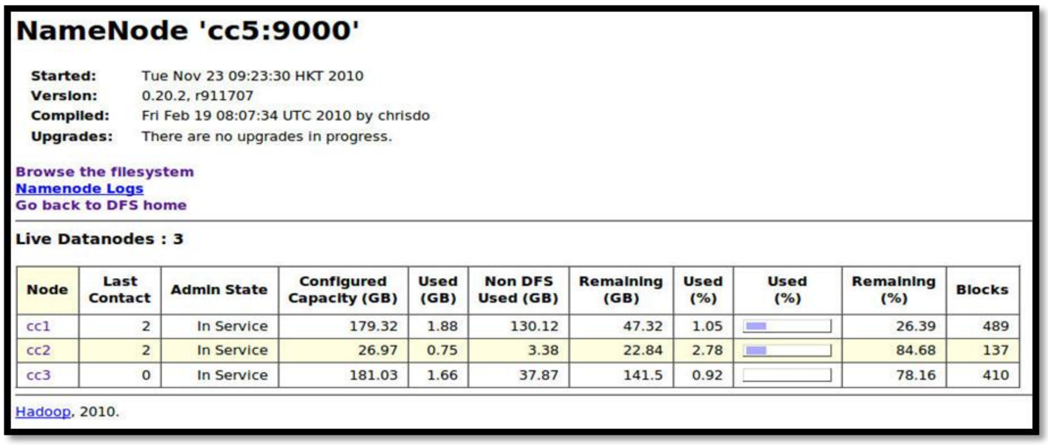
3.2 HDFS

The average of bandwidth of the world is increasing, lots of service provider needs more and more storage, such as Facebook, in 2009, 4TB compressed new data was generated to their storage each day, 135TB compressed data scanned each day, parallel computing is the choose, this also works for many years, but the needs is still growing, they needs more and more machines, hard disks, more than 1000 machines working together is not special now. So, the other big problem appears: hardware failure, a report from Google claims that the hard drive gets 7% failure rate per year, so, in average, they have hard drive failure every day. One of the possible solutions is choosing higher level hard disks which get smaller failure rate, but the cost is extremely expensive. HDFS is designed to handle this large scale data read and write and hard drive failure, data stored in different machines maybe 3 or more copies, if hard drive failure detected, the rebalance will do the work to keep the file completed.

3.2.1 HDFS Structure

3.2.1.1 Namenode

The Namenode server in HDFS is the most important part, it store all the metadata of the stored files and directories, such as the list of files, list of blocks for each file, list of DataNodes for each block and also File attributes. The other role is to serve the client queries, it allows clients to add/copy/move/delete a file, it will records the actions into a transaction log. For the performance, it save the whole file structure tree in RAM and hard drive. A HDFS only allow one running namnode, that's why it is a single point of failure, if the namenode failed or goes down, the whole file system will goes offline too. So, for the namenode machine, we need to take special cares on it, such as adding more RAM to it, this will increase the file system capacity, and do not make it as DataNode, JobTracker and other optional roles.



NameNode 'cc5:9000'

Started:	Tue Nov 23 09:23:30 HKT 2010
Version:	0.20.2, r911707
Compiled:	Fri Feb 19 08:07:34 UTC 2010 by chrisdo
Upgrades:	There are no upgrades in progress.

Browse the filesystem
Namenode Logs
Go back to DFS home

Live Datanodes : 3

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
cc1	2	In Service	179.32	1.88	130.12	47.32	1.05	▭	26.39	489
cc2	2	In Service	26.97	0.75	3.38	22.84	2.78	▭	84.68	137
cc3	0	In Service	181.03	1.66	37.87	141.5	0.92	▭	78.16	410

Hadoop, 2010.

Figure 38 Namenode monitoring

3.2.1.2 SecondaryNamede

From the name, it may like the active-standby node of the NameNode, in fact, it is not. What it works is to backup the metadata and store it to the hard disk, this may helping to reduce the restarting time of NameNode. In HDFS, the recent actions on HDFS will be stored in to a file called EditLog on the NameNode, after restarting HDFS; the NameNode will replay according to the Editlog. SecondaryNameNode will periodically combines the content of EditLog into a checkpoint and clear the Editlog File, after that, the NameNode will replay start from the latest checkpoint, the restarting time of NameNode will be reduced.

SecondaryNamede was running on the same machine as NameNode in default which is not a good idea, it is because if the NameNode machine crashed, it will hard to restore. Placing the SecondaryNamede on the other machine, it may help the new NameNode to restore the file structure.

3.2.1.3 Datanode

Datanode is the scaled up part of HDFS, it may have thousands, it mainly use to store the file data. On startup, DataNode will connect to the NameNode and get ready to respond to the operations from NameNode. After the NameNode telling the position of a file to the client, the client will directly talk to the DataNode to access the files. DataNodes could also talk to each other when they replicating data. The DataNode will also periodically send a report of all existing blocks to the NameNode and validates the data block checksums (CRC32).

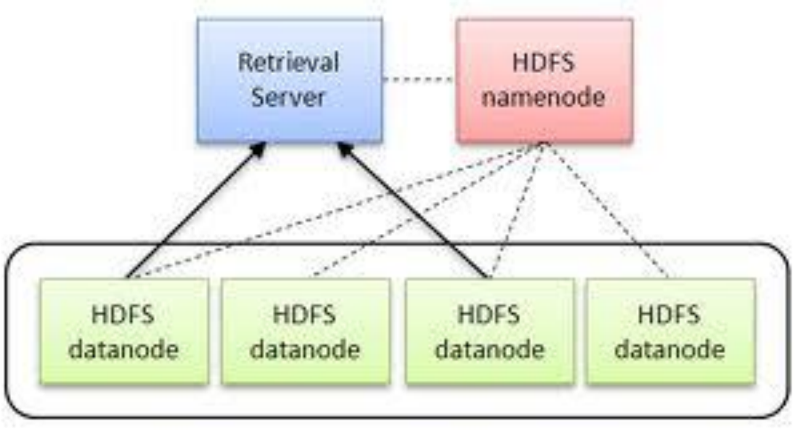


Figure 39 Simple structure of HDFS

3.2.1.4 Other Features

HeartBeats

DataNodes send heartbeat to the NameNode, and NameNode used heartbeats to detect DataNode failure. If failure detected, the NameNode will choose

New DataNodes for new replicas to balances disk usage and communication traffic to DataNodes.

Rebalancer

The rebalance preferred similar percentage of disk usage of each DataNode. It works when DataNodes are added or removed. It is throttled to avoid network congestion.

Block Placement

Any single file stored in Hadoop will be stored in different machines, for the metadata, it will be stored on NameNode, the file content will be split to data blocks and store each block to different DataNodes, the block side is 64MB in default, and it will has 3 replicas, the first one stored on local node, the 2nd and 3rd replicas are stored on remote nodes, for the other replicates, it will be placed randomly. When the client read a file, it will be directed to the nearest replicate.

Data Correctness

In HDFS, if use CRC32 to do the checksum validation, if validation failed, the client will try the other replicas.

Data Pipelining

When the client want to write a file to the file system, he will retrieve a list of DataNodes on which to place replicas of a block and writes block to the first DataNode, the first DataNode forwards the data to the next DataNode in the Pipeline

When all replicas are written, the client moves on to the next block of the file.

3.2.1.5 Compare with NFS

Network File System(NFS) is one of the famous DFS, the design is straightforward, it provides remote access to single logical volume stored on a single machine, but there has limitations such as the files in an NFS volume all reside on a single machine, lots of information as can be stored in one machine, and does not provide any reliability guarantees if that machine goes down, another limitation is that all the data is stored on a single machine, all the clients must go to this machine to retrieve their data. This can overload the server if a large number of clients must be handled. HDFS solve the problems.

- HDFS is designed to store a very large amount of information (terabytes or petabytes). This requires spreading the data across a large number of machines. It also supports much larger file sizes than NFS.
- HDFS should store data reliably. If individual machines in the cluster malfunction, data should still be available.
- HDFS should provide fast, scalable access to this information. It should be possible to serve a larger number of clients by simply adding more machines to the cluster.
- HDFS should integrate well with Hadoop MapReduce, allowing data to be read and computed upon locally when possible.
- Applications that use HDFS are assumed to perform long sequential streaming reads from files. HDFS is optimized to provide streaming read performance; this comes at the expense of random seek times to arbitrary positions in files.
- Data will be written to the HDFS once and then read several times; updates to files after they have already been closed are not supported. (An extension to Hadoop will provide support for appending new data to the ends of files; it is scheduled to be included in Hadoop 0.19 but is not available yet.)
- Due to the large size of files, and the sequential nature of reads, the system does not provide a mechanism for local caching of data. The overhead of caching is great enough that data should simply be re-read from HDFS source.
- Individual machines are assumed to fail on a frequent basis, both permanently and intermittently. The cluster must be able to withstand the complete failure of several machines, possibly many happening at the same time (e.g., if a rack fails all together). While performance may degrade proportional to the number of machines lost, the system as a whole should not become overly slow, nor should information be lost. Data replication strategies combat this problem.

3.2.1.6 Limitation of HDFS

The latest version of Hadoop is just 0.21, so there exists some limitations, for the NameNode, it is a single point of failure and it allows single Namespace for entire cluster, the development team claims that they are working on work on Backup Node to solve this problem. It's also does not rebalance based on access patterns or load, the average performance will be reduced. It's designed in Write-once-read-many access model, then the client can only append to existing files, no general file modify operation.

3.3 MapReduce

3.3.1 Machine Roles

There has 2 roles for MapReduce jobs, one is JobTracker and TaskTracker.

JobTracker

The JobTracker is used to farm out the MapReduce tasks to the clusters, it will allocate the task to the cluster which have the data that the task needed in highest priority to maximize the performance. When the client submit a job to the JobTracker, it will talks to the NameNode to find the data location, then it could determine which node is the nearest to the data and submits the task to the chosen TaskTracker nodes. When the TaskTracker finished the task, JobTracker will get the signal and assigns another task to it. It runs in the NameNode machine in default.

TaskTracker

TaskTracker just like the DataNode, it will be running in every clusters, it received the task from JobTracker and mainly do the Map, Reduce, Shuffle operations.

Every TaskTracker has a set of slots, each slot accept a task. When the JobTracker tries to find TaskTackers to run a MapReduce task, it first looks for an empty slot on the same server that hosts the

DataNode containing the data, and if not, it looks for an empty slot on a machine in the same rack. The TaskTracker runs the actual work in separate JVM process, if the task crashed, it will not take down the TaskTracker. When the task is running, TaskTracker may monitoring these processes and store the output and return value. It also send heartbeat signals to JobTracker and let the JobTracker knows that it was alive, the heartbeat signal also including the information such as number of empty slots. After finishing the task, it sends a signal to notify the JobTracker.

3.3.2 Programming Model

MapReduce is an easy programming model for computer or non-computer scientists. The whole structure of a MapReduce program is mainly work by 2 functions, the Mapper and the Reducer. One key-value pair as a input for the Mapper, and the

Map functions run in parallel, generates any pairs of intermediate key-value pairs from different input data sets, and these will be sent to the Reduce function as input, and the reducer will sort the input key-value pairs according to the key and combine the value of key-value pairs into a list which have the same key, at last, the reducers which are also run in parallel generates the output key-value pairs. All the tasks are working independently.

3.3.3 Execution flow

The user program splits the data source into small ones; normally it will be split into 16~64MB blocks, and copy the program to the selected JobTrackers to get ready for processing data.

The JobTracker distributes the job to JobTaskers.

The Mapper Read the records from the data source such as lines of text files, data row from database, list of text as file path and so on. After mapping process, it sends the intermediate key-value pairs to reducers. Prototype of Map function:

Map (in_key, in_value) →list (out_key,intermediate_value)

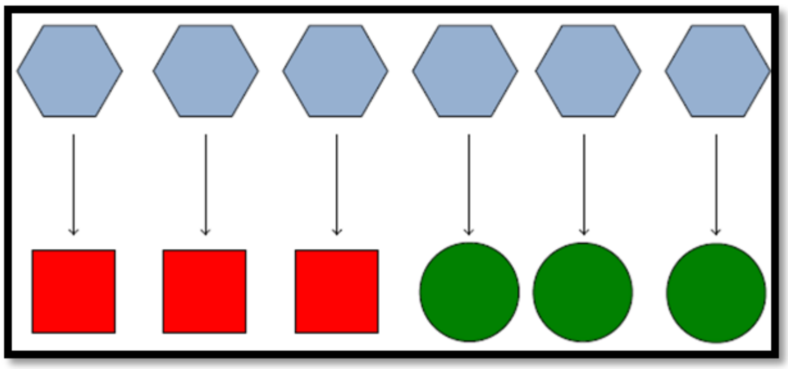
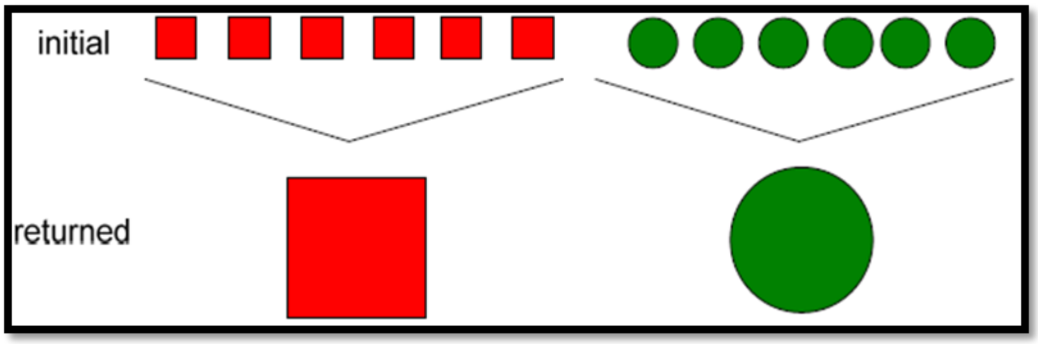


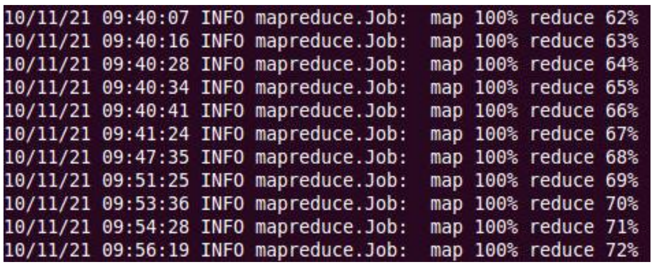
Figure 40 Output of a runing Hadoop program

The Reducer received the key-value pairs from mappers, it process on the key-value pairs and finally it generates the output key-value pairs

Prototype of Reduce function

reduce (out_key, list(intermediate_value)) →list(out_value)

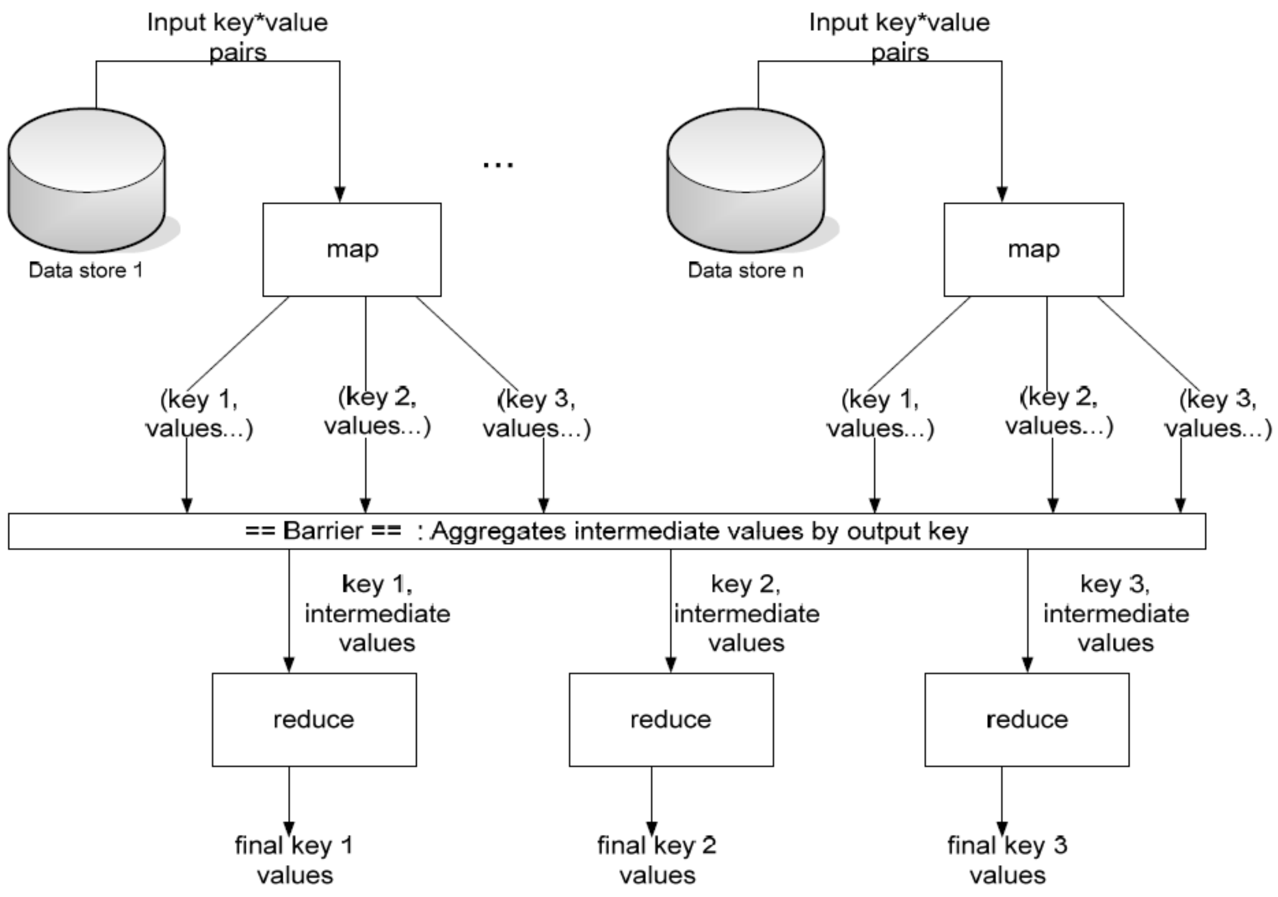




```
10/11/21 09:40:07 INFO mapreduce.Job:  map 100% reduce 62%
10/11/21 09:40:16 INFO mapreduce.Job:  map 100% reduce 63%
10/11/21 09:40:28 INFO mapreduce.Job:  map 100% reduce 64%
10/11/21 09:40:34 INFO mapreduce.Job:  map 100% reduce 65%
10/11/21 09:40:41 INFO mapreduce.Job:  map 100% reduce 66%
10/11/21 09:41:24 INFO mapreduce.Job:  map 100% reduce 67%
10/11/21 09:47:35 INFO mapreduce.Job:  map 100% reduce 68%
10/11/21 09:51:25 INFO mapreduce.Job:  map 100% reduce 69%
10/11/21 09:53:36 INFO mapreduce.Job:  map 100% reduce 70%
10/11/21 09:54:28 INFO mapreduce.Job:  map 100% reduce 71%
10/11/21 09:56:19 INFO mapreduce.Job:  map 100% reduce 72%
```

Figure 41 output of Hadoop

The full map of MapReduce flow



3.3.4 Anatomy of a Job

1. Client creates a job, configures it, and submits it to job tracker
2. JobClient computes input splits (on client end)
3. Job data (jar, configuration XML) are sent to JobTracker
4. JobTracker puts job data in shared location, enquires tasks
5. TaskTrackers poll for tasks

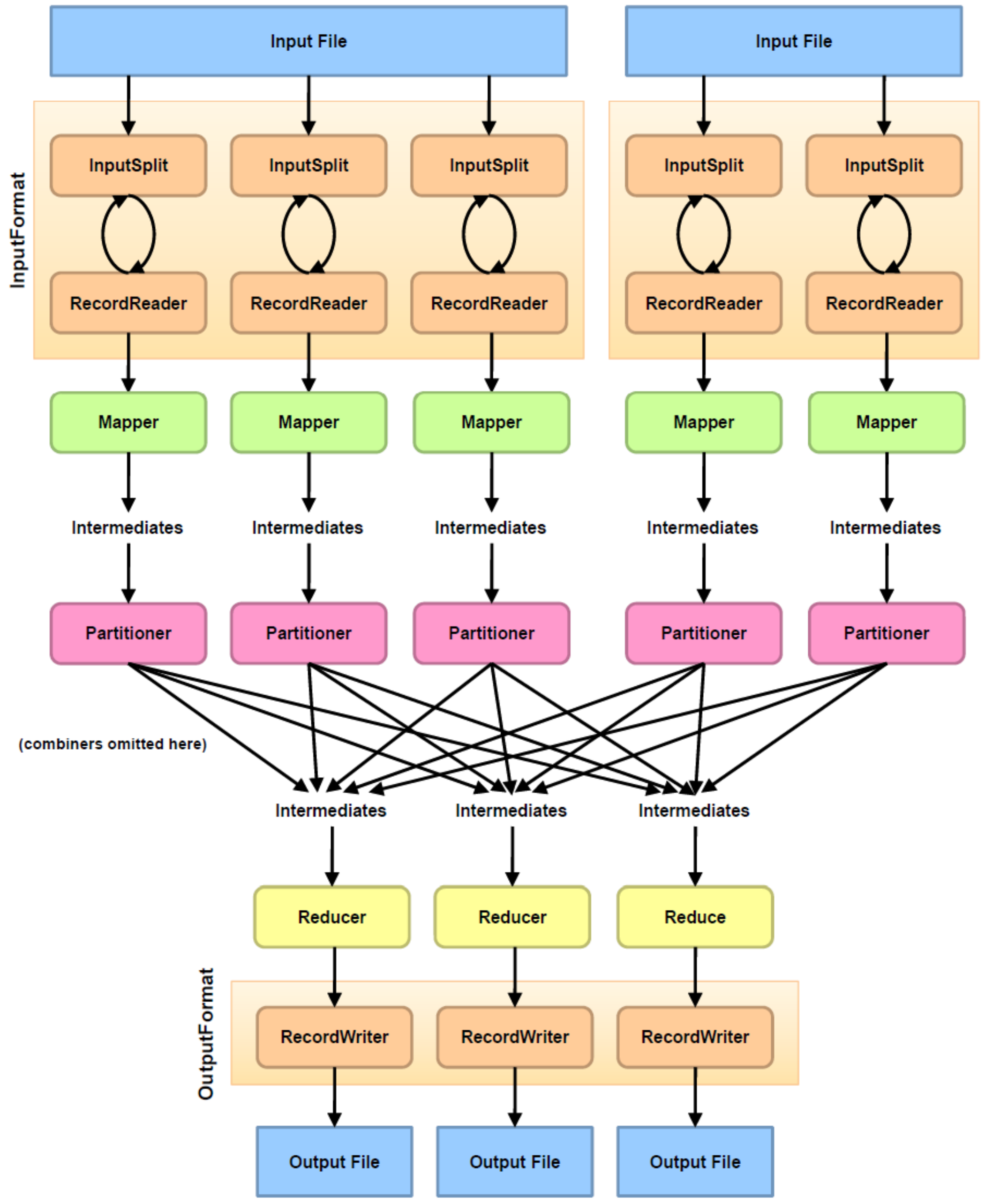


Figure 42 Coding flow

3.3.5 Features

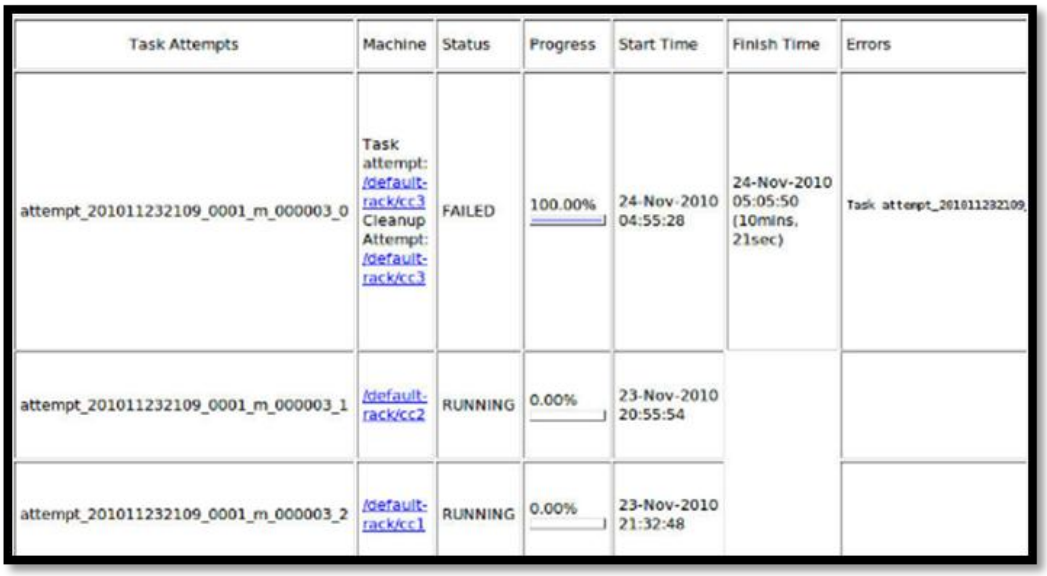
Automatic parallelization and distribution

When we run our application, the JobTracker will automatically handle all the messy things, distribute tasks, failure handling, report progress.

Fault-tolerance ›

The TaskTracker nodes are monitored. If they do not submit heartbeat signals in a period of time, they are deemed to have failed and the work is scheduled on a different TaskTracker. If a task failed for 4 times (in default), the whole job will failed.

A TaskTracker will notify the JobTracker when a task fails. The JobTracker decides what to do then: it may resubmit the job elsewhere, it may mark that specific record as something to avoid, and it may even blacklist the TaskTracker as unreliable.



Task Attempts	Machine	Status	Progress	Start Time	Finish Time	Errors
attempt_201011232109_0001_m_000003_0	Task attempt: /default-rack/cc3 Cleanup Attempt: /default-rack/cc3	FAILED	100.00%	24-Nov-2010 04:55:28	24-Nov-2010 05:05:50 (10mins, 21sec)	Task attempt_201011232109_
attempt_201011232109_0001_m_000003_1	/default-rack/cc2	RUNNING	0.00%	23-Nov-2010 20:55:54		
attempt_201011232109_0001_m_000003_2	/default-rack/cc1	RUNNING	0.00%	23-Nov-2010 21:32:48		

Figure 43 Fault tolerance

Locality optimization

He JobTracker will assign the splits which have the same keys to nearest clusters; this may reduce the job distribution time in Map procedure and collection the output key-value pairs in Reduce procedure.

Backup Tasks

In MapReduce, some operations may be come straggler and increase the total running time, one of the most important reason is that some of the Mapper/Reducer will fight for the local resources such as CPU, Memory, local disk, and network bandwidth, etc. these may increase the delay latency. A simple solution in Hadoop is to have more than 1 copies of the Map/Reduce job running in different machines if there has some empty resources/slots. If a copy finished the job, the others will be killed. The overhead of this procedure is small but the total time reduced significantly in large scale clusters operations.

3.3.6 Our testing

JobTracker is a single point of failure for MapReduce service, if it crashed, all the jobs will be failed.

Hadoop accounts are not isolated; they share the same file pool and resources.

Our Pilot programs

We want to have our own MapReduce application, but not just word count (the example), so we implement all the basic interface of the MapReduce framework: InputFormat, InputSplit, RecordReader, Writable, RecordWriter, OutputFormat, Mapper and Reducer. Base on these interfaces, we could write any MapReduce programs easily.

3.3.6.1 Image processing

A program the runs edge detection algorithm on images, we built it on MapReduce by Java, the input parameters are: [input file directory on HDFS] [output path on HDFS]

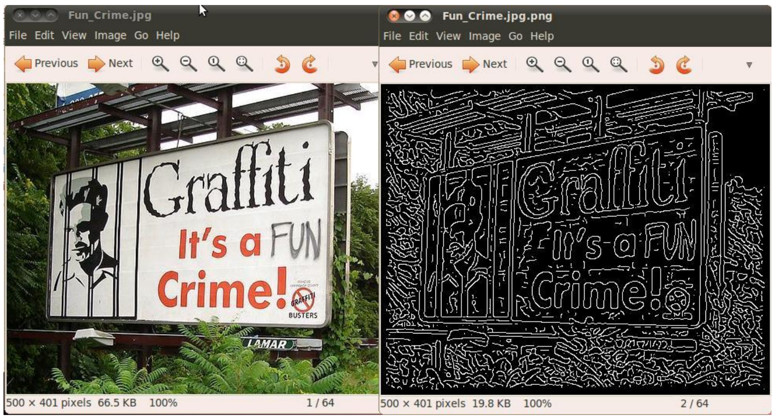

Fun_Crime.jpg
File Edit View Image Go Help
Previous Next
500 × 401 pixels 66.5 KB 100% 1 / 64
Fun_Crime.jpg.png
File Edit View Image Go Help
Previous Next
500 × 401 pixels 19.8 KB 100% 2 / 64

Figure 44Output of image processing testing

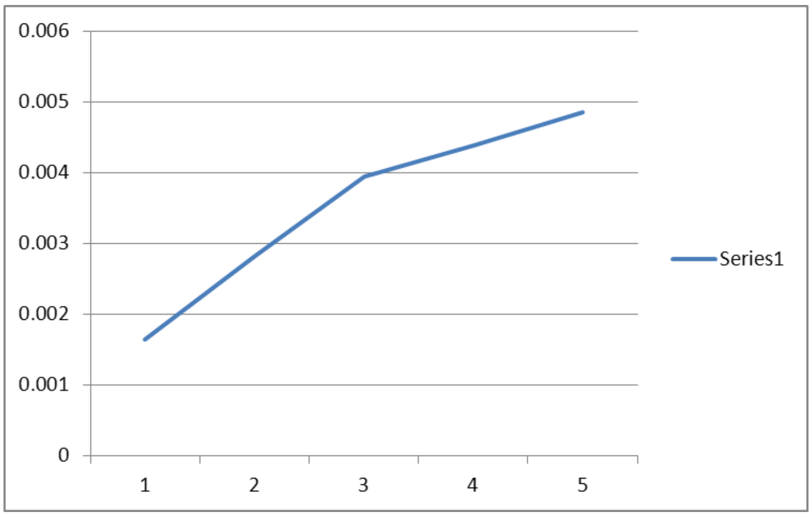
Working flow

The program will get the file list from input directory, and create a split array by treating each file path as a split.

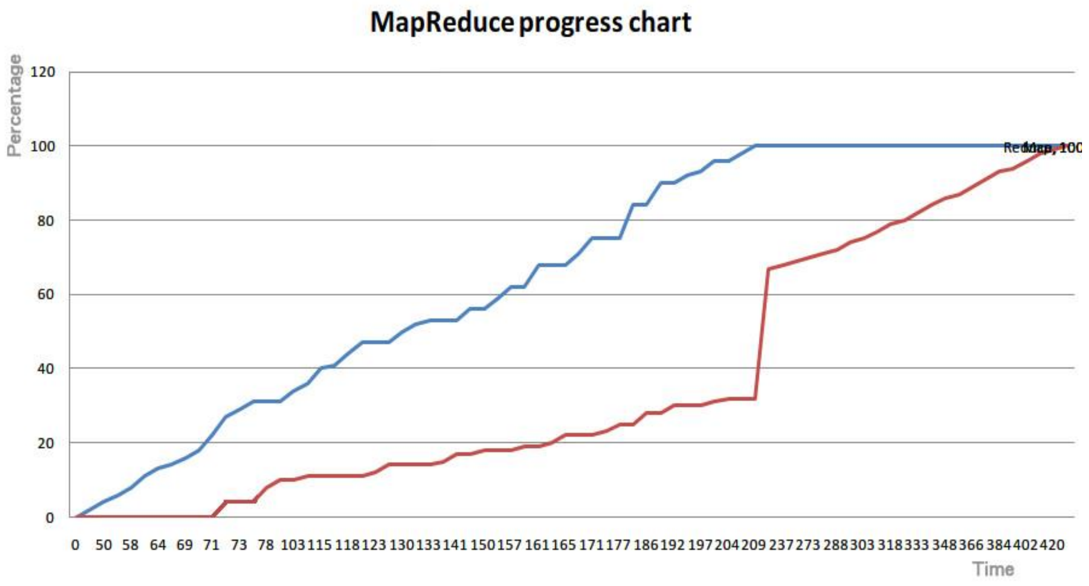
1. Treat each file path as a key, file content as value.
2. Map function received the key-value, runs the edge detection algorithm on it, file path as the intermediate key, and processed image content as intermediate value.
3. Reducer received the key-value pair and save the value to HDFS as output.

Test result

Image processing			
# of node	min	sec	total sec
1	10	8	608
2	5	54	354
3	4	14	254
4	3	48	228
5	3	26	206



As the mapper and reducer are both placed in a single machine, so, if the mapper finished its tasks, the reducer may speed up. This will be significant in the small scale of clusters (not enough nodes).



3.3.6.2Hadoop Word count example

Description

Find out all the words in the text files, and count how many times it appears.

Test on 10 * 64MB text files.

Running environment:

private cloud environment, from 1 to 5 instance

instance specification:

OS: Ubuntu 10.04

RAM: 370MB

CPU: Single virtual core of Pentium D

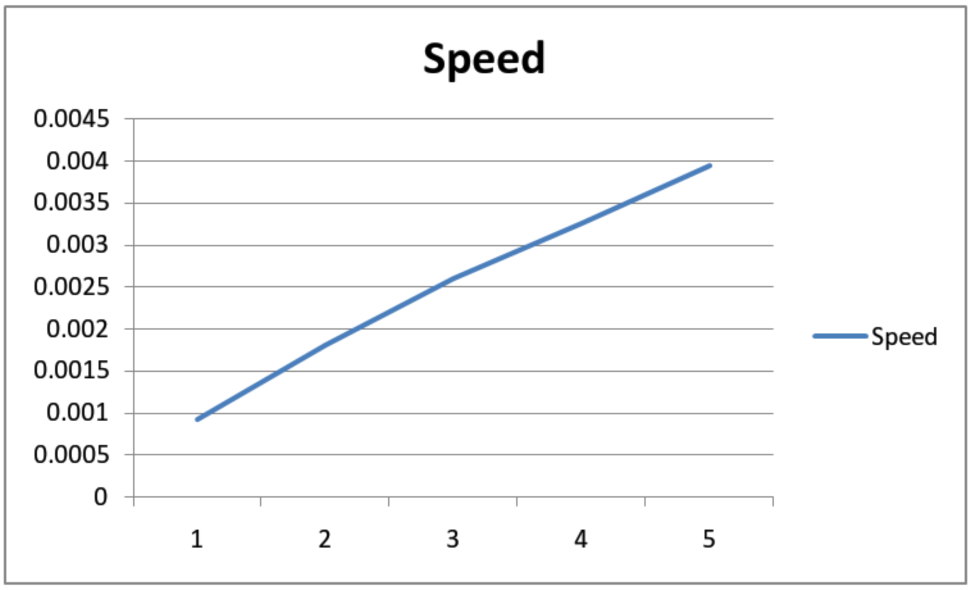
Program prototype

```
Map (String docid, String text):
for each word w in text:
        Emit(w, 1);

Reduce(String term, Iterator<Int> values):
int sum = 0;
for each v in values:
        sum += v;
Emit(term, value);
```

Test result

# of node	min	sec	total sec
1	17	56	1076
2	9	12	552
3	6	25	385
4	5	6	306
5	4	13	253

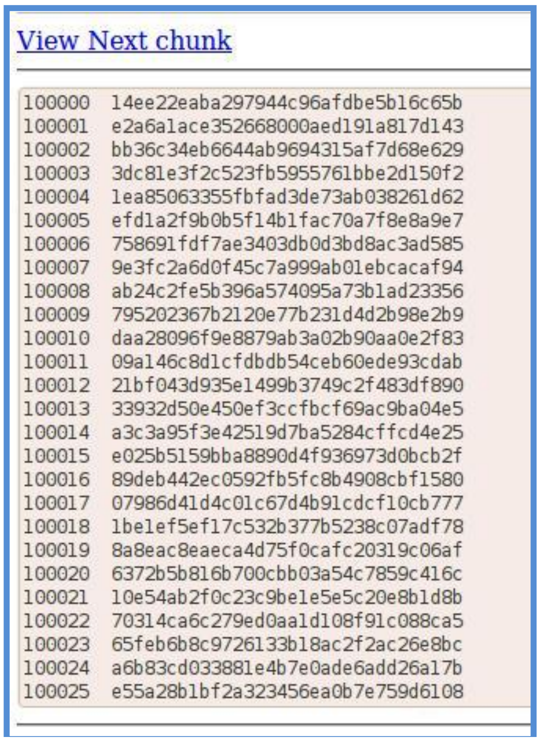


From the graph we see that speed is almost linear proportional to the number of nodes. This program tests on the I/O of HDFS, Job splitting, Shuffling and Sorting of Hadoop, the result shows that Hadoop is good on data processing.

3.3.6.3 MD5 Generator

Description

This program is used to calculate the MD5 checksum of 1 to n digits numbers, the result may be stored to HDFS, and after that, we could checking the MD5 mapping in few seconds by Hive. The bottleneck of this program is on CPU and uploading data to HDFS.
Calculate the md5 checksum of numbers from 1 to 10^7



```
100000  14ee22eaba297944c96afdbe5b16c65b
100001  e2a6a1ace352668000aed191a817d143
100002  bb36c34eb6644ab9694315af7d68e629
100003  3dc81e3f2c523fb5955761bbe2d150f2
100004  1ea85063355fbfad3de73ab038261d62
100005  efd1a2f9b0b5f14b1fac70a7f8e8a9e7
100006  758691fdf7ae3403db0d3bd8ac3ad585
100007  9e3fc2a6d0f45c7a999ab01ebcacaf94
100008  ab24c2fe5b396a574095a73b1ad23356
100009  795202367b2120e77b231d4d2b98e2b9
100010  daa28096f9e8879ab3a02b90aa0e2f83
100011  09a146c8d1cfdbdb54ceb60ede93cdab
100012  21bf043d935e1499b3749c2f483df890
100013  33932d50e450ef3ccfbcf69ac9ba04e5
100014  a3c3a95f3e42519d7ba5284cffcd4e25
100015  e025b5159bba8890d4f936973d0bcb2f
100016  89deb442ec0592fb5fc8b4908cbf1580
100017  07986d41d4c01c67d4b91cdcf10cb777
100018  1be1ef5ef17c532b377b5238c07adf78
100019  8a8eac8eaeca4d75f0cafc20319c06af
100020  6372b5b816b700cbb03a54c7859c416c
100021  10e54ab2f0c23c9be1e5e5c20e8b1d8b
100022  70314ca6c279ed0aa1d108f91c088ca5
100023  65feb6b8c9726133b18ac2f2ac26e8bc
100024  a6b83cd033881e4b7e0ade6add26a17b
100025  e55a28b1bf2a323456ea0b7e759d6108
```

Figure 45 Output of MD5 testing

Running environment:

From 1 to 5 machines
Machine specification:
OS: Ubuntu 10.04

RAM: 1GB

CPU: Pentium 4

Working flow

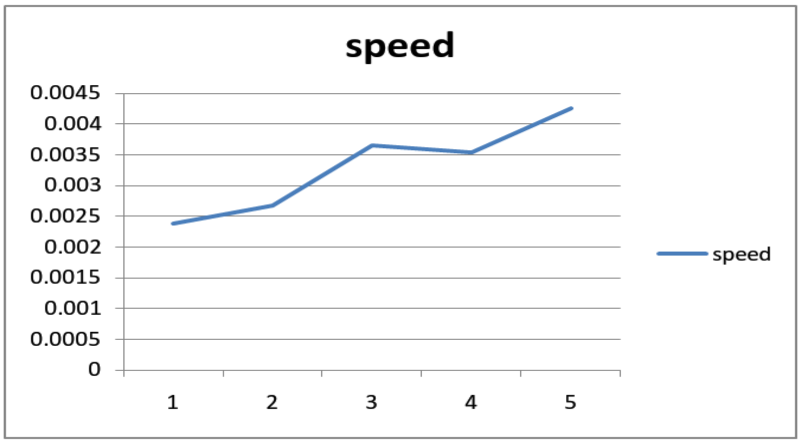
The program needs 3 parameters, start_from max_number and output directory.

For each split, it contains 2 values, the offset and length. Offset is the key. We set the length for all the splits to 10000. In map function, it will loop from offset to offset+length to calculate the MD5 checksum, for each iteration, it add an intermediate key-value pair: <number, md5 checksum>

In reduce function, it collects all the intermediate key-value pair and stored it to HDFS.

Test result

# of node	min	sec	total sec
1	6	59	419
2	6	14	374
3	4	34	274
4	4	42	282
5	3	55	235



This program mainly tests on file appending on HDFS and computation performance of TaskTrackers. Form this graph, we have a unexpected result, 3 nodes is faster than 4 nodes, we tried to run the test on 3 and 4 nodes more times, but the result is similar.

3.3.6.3 Video Conversion

Description

This program aims to convert the video files in parallel.

Convert 30 x 50MB mp4 files to flv files.

Running environment:

From 3 to 5 machines

Machine specification:

OS: Ubuntu 10.01

RAM: 1GB

CPU: Pentium 4

Working flow

It takes 2 parameters: input_dir as the input directory in HDFS, output_dir as the output path.

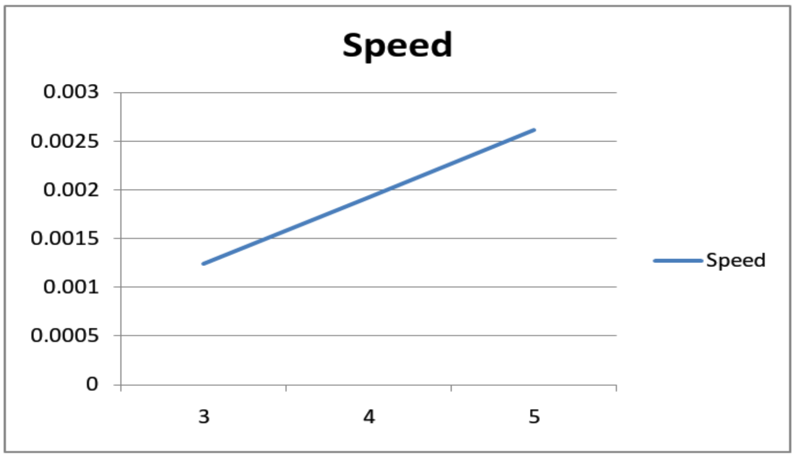
For each split, it contains 1 value, the file pat.

In map function, it will download the video file from HDFS according to the key value, then, it uses the shell to call the FFmpeg to convert the video file to any format, after finished the conversion, and it uploads the converted file to HDFS. It will not return any output key-value pairs.

In reduce function, it will do nothing.

Test result

# of node	min	sec	total sec
3	13	25	805
4	8	41	521
5	6	22	382



In this program, it only uses the Mappers and also I/O of HDFS. The result graph is nice, the speed is also linear increasing with the number of node, and the programming is easy and flexible.

3.4 Trouble shoot

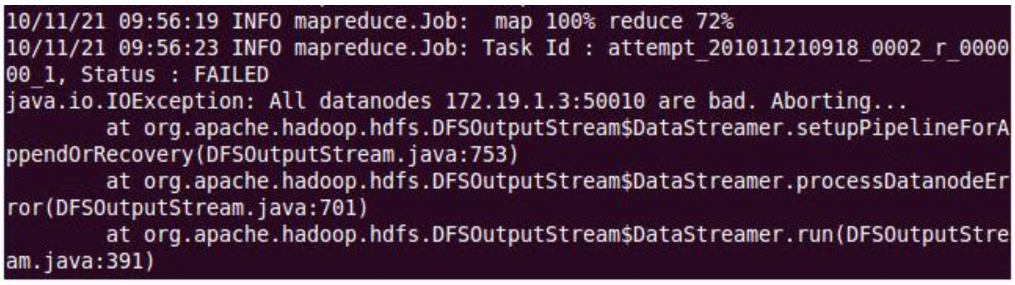
Failed to start the NameNode : Installing Hadoop is relative simple, but there may have some problems, after setting up the environment, we need to specify the Java installation path by setting the parameter JAVA_HOME in $Hadoop_path/conf/hadoop-env.sh, otherwise, the HDFS will get error when it try to boot up.

No TaskTracker available: the NameNode runs normally and also exist some DataNode, but no TaskTracker exist, that's because if HDFS is not ready, the JobTracker will no accept any TaskTracker to join, HDFS will be in safe mode when it start, it will not allow any modifications to the file system, if the number of DataNode is not enough, it will report error and state in safe mode. It may return “The ratio of reported blocks 0.9982 has not reached the threshold 0.9990. Safe mode will be turned off automatically." Turn on enough DataNode to reach the threshold will solve this problem, if some of DataNode were crashed and the available ratio is not enough, the only choice is to re-format the whole HDFS. So, less than 3 DataNodes is dangerous.

MapReduce job always 0 Map input and 0 reduce input: in the RecordReader, the "next" function should return true although there has no next value, we should use a local variable as a flag to check if any other value pair exists.

In map function, not HDFS instance: We could access the HDFS by using the shell to call the HDFS API directly.

Heap Space Error: Memory on TaskTracker is not enough, in default, the heap size of JVM is 128MB, so, we could set it in client programs: conf.set("mapred.child.java.opts","-Xmx1024m"); Lack of memory may cause a lots of problems, such as the task may always failed and the whole job failed, or too much content swapping, the performance will extremely low.



```
10/11/21 09:56:19 INFO mapreduce.Job:  map 100% reduce 72%
10/11/21 09:56:23 INFO mapreduce.Job: Task Id : attempt_201011210918_0002_r_0000
00_1, Status : FAILED
java.io.IOException: All datanodes 172.19.1.3:50010 are bad. Aborting...
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.setupPipelineForA
ppendOrRecovery(DFSOutputStream.java:753)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.processDatanodeEr
ror(DFSOutputStream.java:701)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStre
am.java:391)
```

Figure 46 error message of our application

3.5 Hadoop summary

For those test cases, the scalability of Hadoop is very significant, most of them show the linear increasing trend, we just test on 5 machines, if the number of clusters is large enough, and it will be very powerful for any jobs. On the performance aspect, it is not the same as we expected at beginning, for empty jobs, it may also takes 30 to 60 seconds to run. The overhead of Hadoop on small jobs is very significant, that’s why Hadoop is for large scale, data intensive jobs, otherwise, MPI will be the better choice.

Hadoop is also not good for frequently I/O access, the MD5generator shows that the performance is much lower than running on a machine locally, that's because frequently appending data to a file is very slow. For the word count example, we are very surprise that processing 40MB text and 400MB text data are using nearly the same time, it's because each task is too small for 40MB data, the overhead is larger than the effective working time, in average, on our environment, each task will use at least 2 seconds on the whole flow, if the effective working time is near to this value, the efficiency may become very low, if the effective working time is too long, a task may hold the machine and occupied lots of resources, if the task failed, the overhead will be much large as the task will start from zero again. The suggestion from Cloudera which is the enterprise Hadoop developer is 14 to 17 seconds for a task runs in map function, this may get better efficiency, but some of the Job could not be controlled to fit this optimal value, such as the video conversion, if we need to meet the time, we need to cut a video into small pieces, each piece may used about that period of time to convert, after the conversion, we need an additional step to combine the pieces together, the cutting and combining task may also use up lots of time, the overall efficient may be lower, but for the task such as MD5generator, we could turn the length of each split and let the map function run for any time we expect. For the jobs that have a lot of small pieces of input files, we may pack the files together, it may get performance improvement.

Theoretically speaking, any jobs could be ran in a single machine, it could be paralleled by MapReduce in Hadoop, the programming is easy, but it is not always a good choice, Jobs mainly focused on computation is not good in Hadoop, Jobs do not consume too much data may have better choices other than Hadoop. For the convenience of parallel programming, Hadoop maybe the easiest one to learn, the programmer don't need to have any parallel programming skills, it will easily make it parallel by MapReduce, it is good for the non- computer scientist.

Hadoop is made for scalability not performance, it is good in data processing!

Chapter 4: Overall conclusion

We can see cloud is powerful IT resource pool (public cloud), or management tool (private cloud). Its elastic property allows application resizing itself to any scale. The almost important benefit of using cloud is paying and using the IT resource as you go. How much has you used, how much you should pay. These features of cloud enable small company or organization like school running large scale application or experiment with a reasonable cost.

Even though the cloud provide users with elastic resource, not all application can directly run on the cloud, it re-development of application and put it into cloud layer (e.g. Azure), or distribute the workload on its own (e.g. AWS). In order to fully shows up the performance of cloud. Second approach requires some powerful framework on top of all virtual machines and fully used up all resource form each virtual machine.

Hadoop (MapReduce) is one of the very powerful frameworks that enable easy development on data-intensive application. It objective is help building a supplication with high scalability with thousands of machines. We can see Hadoop is very suitable to data-intensive background application and perfect fit to our project's requirements.

Apart from running application in parallel, Hadoop provides some job monitoring features similar to Azure. If any machine crash, the data could be recovered by other machines, and it will take up the jobs automatically.

When we put Hadoop into cloud, we also see the convenience in setting up Hadoop. With a few command lines, we can allocate any number of clusters to run Hadoop, this may save lot of time and effort.

We found the combination of cloud and Hadoop is surely a common way to setup large scale application with lower cost, but higher elastic property.

4.1 Difficulty of this project

There are three main difficulty of our project.

Rapid update of soft wares

Both Hadoop and Eucalyptus are not mature application. There is not enough documentation and user manual, Apart from that, both of them keep changing and has different usage method. Most guidelines are written by user and wrong. We put lot of effort on set up Eucalyptus and Hadoop. We sent a email to Cloudera (Hadoop supporting website), they haven't rely us until now.

Testing time is very long

It is different form running a single program on a single machine. Our testing cycle can last for a few hours to build an image. Since cloud is and Hadoop are sensitive to network status, the network always crash and require us to repair and re-set up the environment with an hour.

Limited resource

Our cloud is very tiny, we can only allocate 370MB memory to each instance and they have only about 80MB free memory. Such environment can't support any large scale computation, like movie conversion in cloud. Even run all application in physical machines, we still need a few more machines to evaluate the performance of Hadoop.

Chapter 5: What we will do next

In this semester, we have not finished the video search yet. We have done lots of effort on building the cloud environment and configuring Hadoop, writing applications to learn the characteristic of Hadoop DFS and MapReduce, in the next semester, we will mainly focused on video searching algorithm in Hadoop MapReduce, such as the data locality, file splitting, comparing spitted data, and also using the Hive interface to store and search the comparison result.

Chapter 6: Acknowledgement

We would like to thank our final year project supervisor: Professor Michael Lyu. He gives us valuable advices for our project and how to present well. Also, he reminds us to take a good scheduling.

Besides, we would like to thank Mr.Edward Yan and Mr. Un Tze Lung in VIEW Lab. They provide us with facilities and technical support when we are setting up the cloud, and some algorithm related to video processing.

Lastly, we thank to Kenny (PhD Student RM 101). He gives us guideline on cloud computing and Mapreduce. so we can better prepare out platform for our project.

Chapter 7: Reference list

[1] Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, "Cloud computing : a practical approach". New York : McGraw-Hill, c2010.

[2] Johnson D, Kiran Murari, Murthy Raju, Suseendran RB, Yogesh Girikumar ,"Eucalyptus Beginner's Guide - UEC Edition (Ubuntu Server 10.04 - Lucid Lynx)", ICT Services Technology Asoption, May 10, 2010

[3] Zibin Zheng, M.R. Lyu, "Component Recommendation for Cloud Applications", ICSE 2010, Cape Town, South Africa, May 4, 2010, pp 323-345, Volume 15, Issue 4

[4] Kiranmurari ,"UEC: CC and NC on a single machine", http://kiranmurari.wordpress.com/2010/03/19/uec-cc-nc-single-machine/

[5] Kiranmurari, "UEC: Bundling Windows Image", http://kiranmurari.wordpress.com/2010/03/29/uec-bundling-windows-image/

[6] Ben Selinger , "Slow disk IO for all guests with all disc modes and types.", http://web.archiveorange.com/archive/v/4feNTkhyMpte1aDL4w3h

[7] "Virtualization Studies", http://vmstudy.blogspot.com/2010/04/network-speed-test-iperf-in-kvm-virtio.html

[8] "UECImages", https://help.ubuntu.com/community/UEC/Images

[9] "UECCreateYourImage", https://help.ubuntu.com/community/UEC/CreateYourImage

[10] "Eucalyptus user handbook", http://trac.nchc.org.tw/grid/wiki/Eucalyptus/User

[11] "How to create a new kvm image for eucalyptus UEC", http://bsd.dischaos.com/2009/11/15/how-to-create-a-new-kvm-image-for-eucalyptus-uec/

[12] "How To Customize UEC Images",

http://www.ossramblings.com/customizing-uec-images

[13] "Run your own Ubuntu Enterprise Cloud, part 1-3",
http://fnords.wordpress.com/2009/10/04/run-your-own-uec-part-1/
http://fnords.wordpress.com/2009/10/07/run-your-own-uec-part-2/
http://fnords.wordpress.com/2009/10/13/run-your-own-uec-part-3/

[14] "Private cloud setup using eucalyptus and xen",
http://knol.google.com/k/private-cloud-setup-using-eucalyptus-and-xen#

[15] ""
http://highscalability.com/blog/2010/7/8/cloud-aws-infrastructure-vs-physical-infrastructure.html

[16] David Chappell," INTRODUCING WINDOWS AZURE", MICROSOFT CORPORATION,2010

[17] "KVM-Virtio"
http://www.linux-kvm.org/page/Virtio

[18] "Virtio",
http://wiki.libvirt.org/page/Virtio

[19] "Paravirtualization is State of the Art Virtualization",
http://staging.xen.org/about/paravirtualization.html

[20] "Video Surveillance Through Cloud Computing"
http://www.ivedasolutions.net/index.php/cloud-computing/

[21] "Surveillance Video Mining",
http://www.zdnetasia.com/whitepaper/surveillance-video-mining_wp-1387385.htm

[22] "MapReduce and HDFS", University of Washington, 2009 Cloudera, Inc.

[23] Tyson Condie , Neil Conway, Peter Alvaro, Joseph M. Hellerstein , "MapReduce Online", UC Berkeley, Yahoo! Research

[24] Gong Zhang, Ling Liu, "GeoGrid Project and Experiences with Hadoop", Distributed Data Intensive Systems Lab, Center for Experimental Computer Systems Research

[25] konstantin V. shVachko, "HDFS scalability: the limits to growth", ;LOGIN: VOL. 35, NO. 2

[26] Introduction To Hadoop , "Introduction To Hadoop", Google Inc, January 14, 2008

[27] Dhruba Borthakur , “Hadoop Architecture and its Usage at Facebook”, Presented at Microsoft Research, Seattle October 16, 2009

[28] Jake Hofman,“Large-scale social media analysis with Hadoop”, Yahoo! Research, May 23, 2010

[29] Jimmy Lin, “Hadoop: Nuts and Bolts”, University of Maryland, Data-Intensive Information Processing Applications — Session #2, Data-Intensive Information Processing Applications — Session #2

[30] “Cloudera » Apache Hadoop for the Enterprise”, http://www.cloudera.com/

Appendix

Progress table

Period	Chu Yan Shing	Wu Bing Chuan
Summer	Studying the Cloud Computing programming model Studying Linux Trying to set up a Cloud but fail with limited resource	Studying the Cloud Computing programming model Studying the surveillance video analysis algorithms and techniques.
September	Building a new cloud Testing the cloud environment Building images	Studying MapReduce Studying Hadoop
October	Optimizing the cloud installing MapReduce to our cloud	Testing Hadoop with virtual machines Implemented application framework our environment
November		Implement a few Hadoop applications
	Testing cloud + Hadoop performance Testing Hadoop application with large amount of data	
	-	Further testing on Hadoop
	Writing report	

Cloud performance testing outputs

Performance testing of our private cloud

(Single process running)

		Physical Machine		Instance		Instance		
				without virtio		with virtio		Virtio / SCSI
		time (s)	mb/s	s	mb/s	s	mb/s	
Download	Real Time	15.406	2.02	14.757	2.04	12.457	2.39	1.1846
a 30MB mp3 file	User Time	0.072		0.044		0.152		0.2895
	Sys Time	0.486		0.816		0.948		0.8608
	Real Time	15.447		14.962		11.434		1.3086
	User Time	0.052		0.048		0.12		0.4
	Sys Time	0.42		0.984		1.192		0.8255
Compress	Real Time	3.144		3.48		3.469		1.0032
a 30MB mp3 file	User Time	2.94		3.12		3		1.04
	Sys Time	0.148		0.264		0.252		1.0476
	Real Time	3.05		3.595		3.754		0.9576
	User Time	2.892		3.268		2.984		1.0952
	Sys Time	0.16		0.316		0.292		1.0822
Decompress this file	Real Time	0.368		1.758		0.707		2.4866
	User Time	0.332		0.388		0.368		1.0543
	Sys Time	0.112		0.144		0.116		1.2414
	Real Time	0.385		0.643		0.77		0.8351
	User Time	0.34		0.4		0.388		1.0309
	Sys Time	0.152		0.108		0.112		0.9643
Copy a 30MB file	Real Time	0.074		0.115		0.528		0.2178
	User Time	0		0.004		0.004		1
	Sys Time	0.072		0.1		0.04		2.5
	Real Time	0.104		0.101		0.226		0.4469
	User Time	0.004		0		0		-

	Sys Time	0.076		0.096		0.072		1.3333
Write a 500MB File	Real Time	5	103	61.87	8.3	10.153	50.9	6.0938
	User Time	0.164		0.388		0.548		0.708
	Sys Time	2.724		5.2		3.584		1.4509
	Real Time	4.903	105	58.555	8.7	11.215	45.8	5.2211
	User Time	0.156		0.428		0.48		0.8917
	Sys Time	2.784		5.54		3.128		1.7711
Compress this file	Real Time	8.337		13.854		12.68		1.0926
490kb	User Time	7.476		7.888		8.449		0.9336
	Sys Time	1.432		4.968		3.092		1.6067
	Real Time	8.307		18.001		13.293		1.3542
	User Time	7.46		8.285		8.333		0.9942
	Sys Time	1.492		4.968		3.372		1.4733
Decompress this file	Real Time	5.781		68.137		11.675		5.8361
	User Time	2.2		2.456		2.34		1.0496
	Sys Time	1.856		5.528		2.248		2.4591
	Real Time	4.907		67.056		11.033		6.0778
	User Time	2.228		2.46		2.328		1.0567
	Sys Time	1.988		5.38		2.184		2.4634
MD5 checksum	Real Time	6.613		10.091		8.411		1.1997
a 500MB file	User Time	1.08		1.372		1.184		1.1588
	Sys Time	0.54		1.608		0.944		1.7034
	Real Time	6.93		9.695		8.614		1.1255
	User Time	1.04		1.436		1.056		1.3598
	Sys Time	0.628		2.132		1.224		1.7418
		time (s)		time (s)	ratio	time (s)	ratio	Virtio / SCSI
Overall	Real Time	88.756		346.67	0.25602	120.42	0.7371	2.8789
	User Time	28.436		31.985	0.88904	31.734	0.8961	1.0079
	Sys Time	15.07		38.152	0.395	22.8	0.661	1.6733
	I/O time	45.25		276.533	0.16363	65.885	0.6868	4.1972
	(IO time = real time - user time - sys time)							

Performance testing of our private cloud

(Multi processes running)

		Host				2 instance at the same node					2 instance at the same node with virtio				Virtio / SCSI
		Proc A		Proc B		Proc A		Proc B			Proc A		Proc B		
		s	mb/s	s	mb/s	s	mb/s	s		mb/s	s	mb/s	s	mb/s	
Download	Real Time	29.029	1.15	23.965	1.12	28.733	1.06mb/s	26.178		905kb/s	25.829	1.76	23.988	1.895	1.10225
a 30MB mp3 file	User Time	0.096		0.064		0.088		0.092			0.172		0.184		0.50562
	Sys Time	0.552		0.552		0.84		0.796			1.224		1.228		0.66721
	Real Time	26.766	1.85	26.7	970kb/s	29.23	1.16kb/s	27.786		706kb/s	24.537	1.13	23.936	1.43	1.17624
	User Time	0.068		0.124		0.044		0.092			0.128		0.2		0.41463
	Sys Time	0.452		0.516		1.124		0.98			1.16		1.188		0.89608
Compress	Real Time	3.268		3.28		3.663		3.899			2.69		3.839		1.15822
A 30MB mp3 file	User Time	2.928		3.06		3.176		3.508			2.268		3.016		1.26495
	Sys Time	0.116		0.088		0.38		0.252			0.228		0.292		1.21538
	Real Time	3.33		3.126		9.707		10.118			3.795		3.703		2.64404
	User Time	3.088		2.968		3.136		3.156			3.068		2.992		1.03828
	Sys Time	0.1		0.076		0.352		0.344			0.26		0.272		1.30827
Decompress this file	Real Time	0.451		0.473		1.954		14.671			0.816		1.117		8.60062
	User Time	0.328		0.344		0.348		0.36			0.26		0.388		1.09259
	Sys Time	0.128		0.12		0.24		0.192			0.216		2.48		0.16024
	Real Time	0.464		0.531		2.919		5.413			0.547		0.546		7.62306
	User Time	0.316		0.324		0.348		0.336			0.344		0.316		1.03636
	Sys Time	0.12		0.156		0.172		0.188			0.16		0.2		1
Copy a	Real Time	0.083		0.087		0.174		10.351			0.09		0.09		58.4722

30MB file															
	User Time	0		0		0.004		0.004			0		0.004		2
	Sys Time	0.08		0.084		0.084		0.092			0.068		0.076		1.22222
	Real Time	0.104		0.118		0.316		0.338			0.107		0.09		3.3198
	User Time	0.004		0		0.008		0.004			0.004		0		3
	Sys Time	0.08		0.08		0.252		0.264			0.073		0.076		3.46309
Write a 500MB File	Real Time	7.676	65	7.648	65	223.44	2.3	222.05		2.3	19.613	26.3	19.445	26.6	11.4058
	User Time	0.148		0.14		0.372		0.312			0.584		0.412		0.68675
	Sys Time	4.044		4.184		5.236		5.056			4.084		4.456		1.20515
	Real Time	14.1	36	8.962	57.2	204.93	2.5	209.79		2.4	17.628	28.562	19.774	25.461	11.088
	User Time	0.152		0.132		0.32		0.368			0.512		0.384		0.76786
	Sys Time	3.752		2.824		4.988		5.352			3.824		4.38		1.26036
Compress this file	Real Time	16.062		16.275		45.058		44.386			52.801		52.547		0.84903
490kb	User Time	7.428		7.448		8.405		8.333			8.365		8.165		1.01258
	Sys Time	1.396		1.364		5.06		4.864			3.092		3.228		1.57025
	Real Time	14.178		14.763		42.423		43.07			49.587		49.046		0.86678
	User Time	7.536		7.476		8.857		8.393			8.137		8.213		1.05505
	Sys Time	1.472		1.116		5.024		5.396			3.532		2.96		1.60505
Decompress this file	Real Time	12.562		12.152		229.11		219.14			19.957		20.961		10.9549
	User Time	2.228		2.236		2.368		2.46			2.58		2.572		0.93711
	Sys Time	1.68		1.692		5.336		5.276			3.912		4.076		1.32849
	Real Time	11.992		14.996		214.38		210.26			20.107		19.716		10.663
	User Time	2.156		2.196		2.312		2.564			2.5		2.404		0.99429
	Sys Time	1.956		2.26		5.464		5.284			3.972		3.876		1.36952
MD5 checksum	Real Time	15.293		14.869		34.154		35.86			55.292		54.673		0.63669
a 500MB file	User Time	1.064		1.116		1.384		1.332			1.24		1.22		1.10407

	Sys Time	0.596		0.52		1.932		2.744			1.788		1.552		1.4
	Real Time	15.395		15.071		38.133		39.995			57.21		57.911		0.67866
	User Time	1.032		0.984		1.368		1.348			1.296		1.228		1.07607
	Sys Time	0.552		0.592		1.836		2.8			1.732		1.753		1.33027
									host/ virtual					host/ virtual	Virtio / SCSI
Overall	Real Time	170.75		163.02		1108.3		1123.3	0.1496		350.61		351.38	0.47546	3.17899
	User Time	28.572		28.612		32.538		32.662	0.8771		31.458		31.698	0.90544	1.03236
	Sys Time	17.076		16.224		38.32		39.88	0.4258		29.325		32.093	0.54219	1.27324
	IO Time	125.11		118.18		1037.5		1050.8	0.1165		289.82		287.59	0.42134	3.6165
	(IO time = real time - user time - sys time)														

Performance testing of pure user time

		Host		Instance		Host				2 instance at the same node				
		32bit	64bit			Proc A		Proc B		Proc A		Proc B		
		time (s)	time (s)	s		s	mb/s	s	mb/s	s	mb/s	s	mb/s	
Pure user time Test	Real Time	46.165	45.447	90.946		92.253		91.453		92.59		90.939		
4threads run a few lines of	User Time	89.438	89.45	90.926		89.502		89.49		92.558		90.918		
code 1000million times	Sys Time	0.06	0.01	0.02		0.028		0.06		0.024		0.012		
	Real Time	46.02	45.443	90.953		92.085		91.744		91.851		91.868		

	User Time	89.49	89.44	90.91		89.514		89.514		91.734		91.822		
	Sys Time	0.024	0.03	0.016		0.02		0.012		0.028		0.04		
Overall	Real Time	92.185		181.9		184.34		183.2		184.44		182.81		
	User Time	178.93		181.84		179.02		179		184.29		182.74		
	Sys Time	0.084		0.036		0.048		0.072		0.052		0.052		
Performance ratio	Real Time	0.5068		overall	0.9843					1.0008		overall	0.9755	
	User Time	0.984								0.9754				
	Sys Time	2.3333								1.1538				

Memory access speed test

Host Time(s)	16.07
Virtual machine time(s)	20.977
Ratio(host / Virtual)	0.76608