Department of Computer Science and Engineering
The Chinese University of Hong Kong

Cloud computing technologies and applications

Supervised by Prof. Michael R. Lyu

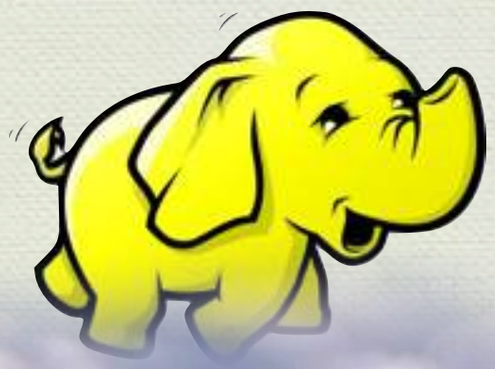
Group: LYU1005

Chu Yan Shing (1008618841)

Wu Bing Chuan, Bill(1008612783)

Road map

- Background information
- introduce cloud and what we did on it
- introduce hadoop and what we did on it
- Conclusion



Background

Background

- File sizes increase rapidly
- File formats are much more complicated

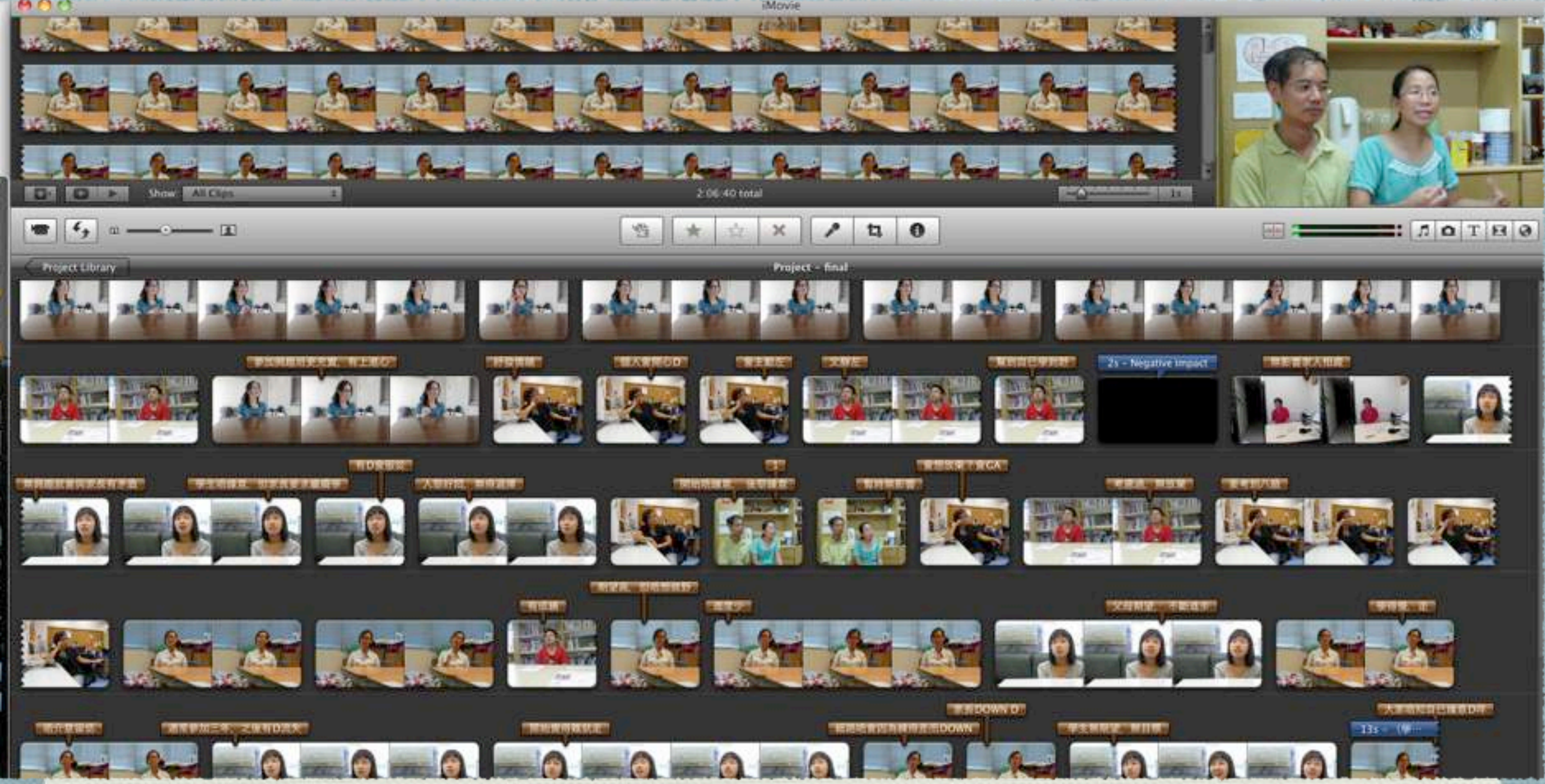
Text :a few KBs

Blu ray video:50 GB

Background

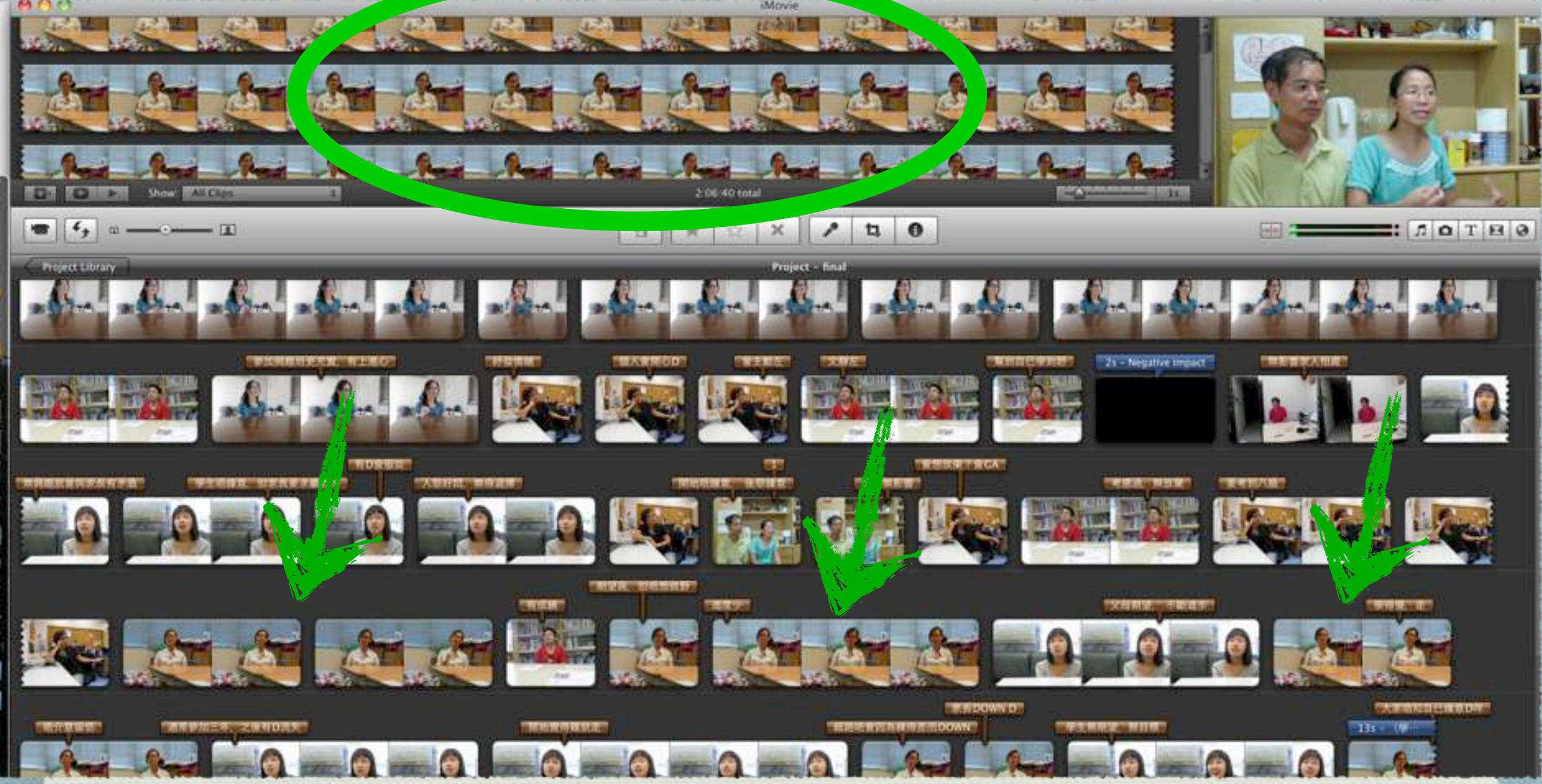
- Machine has many limitation
  - CPU clock rate, I/O time, Bandwidth,storage
- Run something parallel is the best solution
  - Single Google Query uses 1000 machines in 0.2 seconds

http://www.youtube.com/watch?v=6x0cAzQ7PVs&feature=player_embedded#!



One of our ideas:

Giving a video frame, find it from the video database



One of our ideas:

Giving a video frame, find it from the video database

Background

- but.....
  - buying expansive facilities is expansive
  - Develop a parallel application is also difficult
  - ...

Background

I make development simple

- but.....
  - buying expansive facilities is expansive
  - Develop a parallel application is also difficult

I reduce the cost

Background

I make development simple

- but.....
  - buying expansive facilities is expansive
  - Develop a parallel application is also difficult

I reduce the cost

In this term, we focus on understanding these technologies.

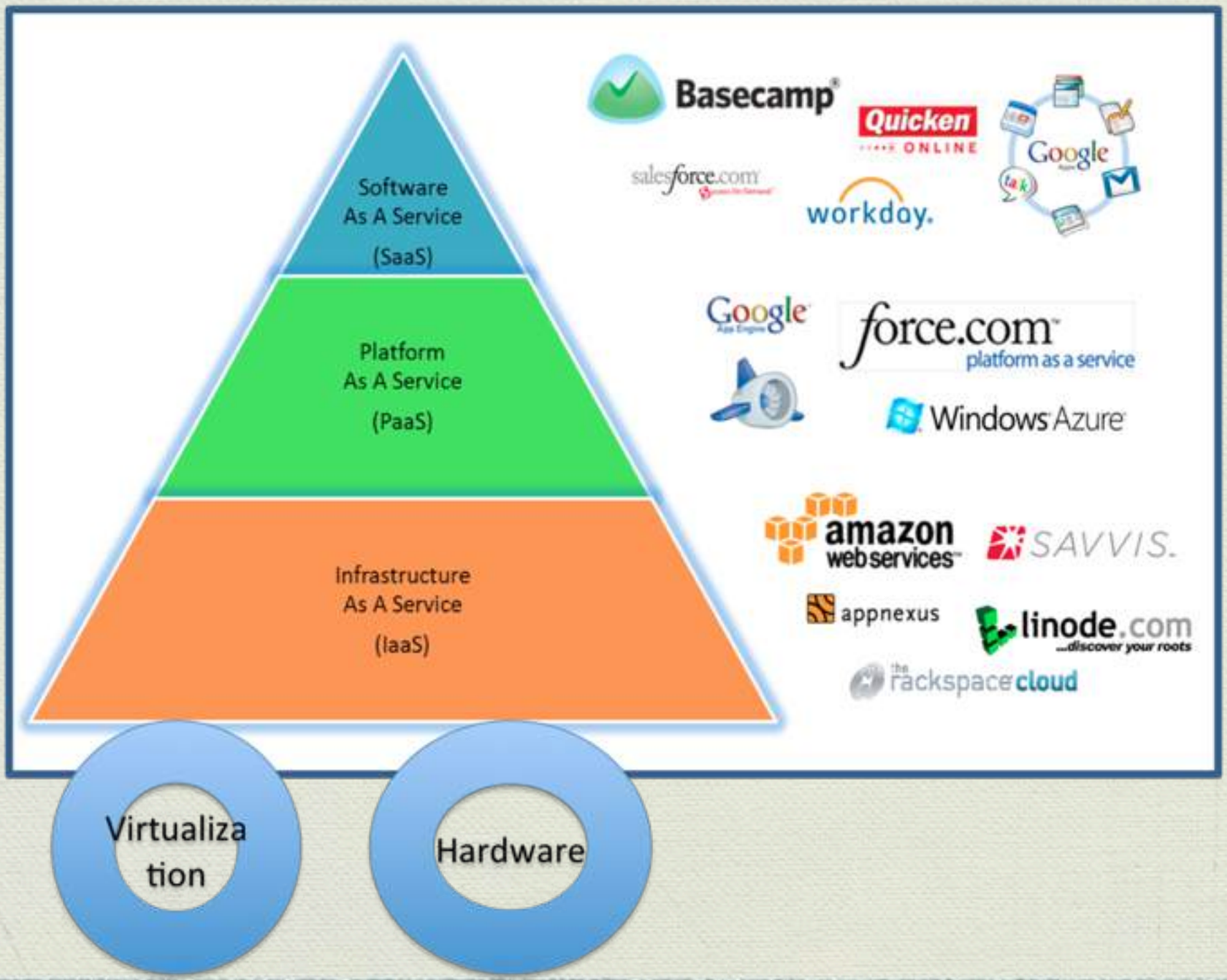
Cloud

Cloud is

a IT resource service through network

Elastic

Reduce cost



Reduce cost

- Pay as you go
- Fast response
- No maintenance cost

andard 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

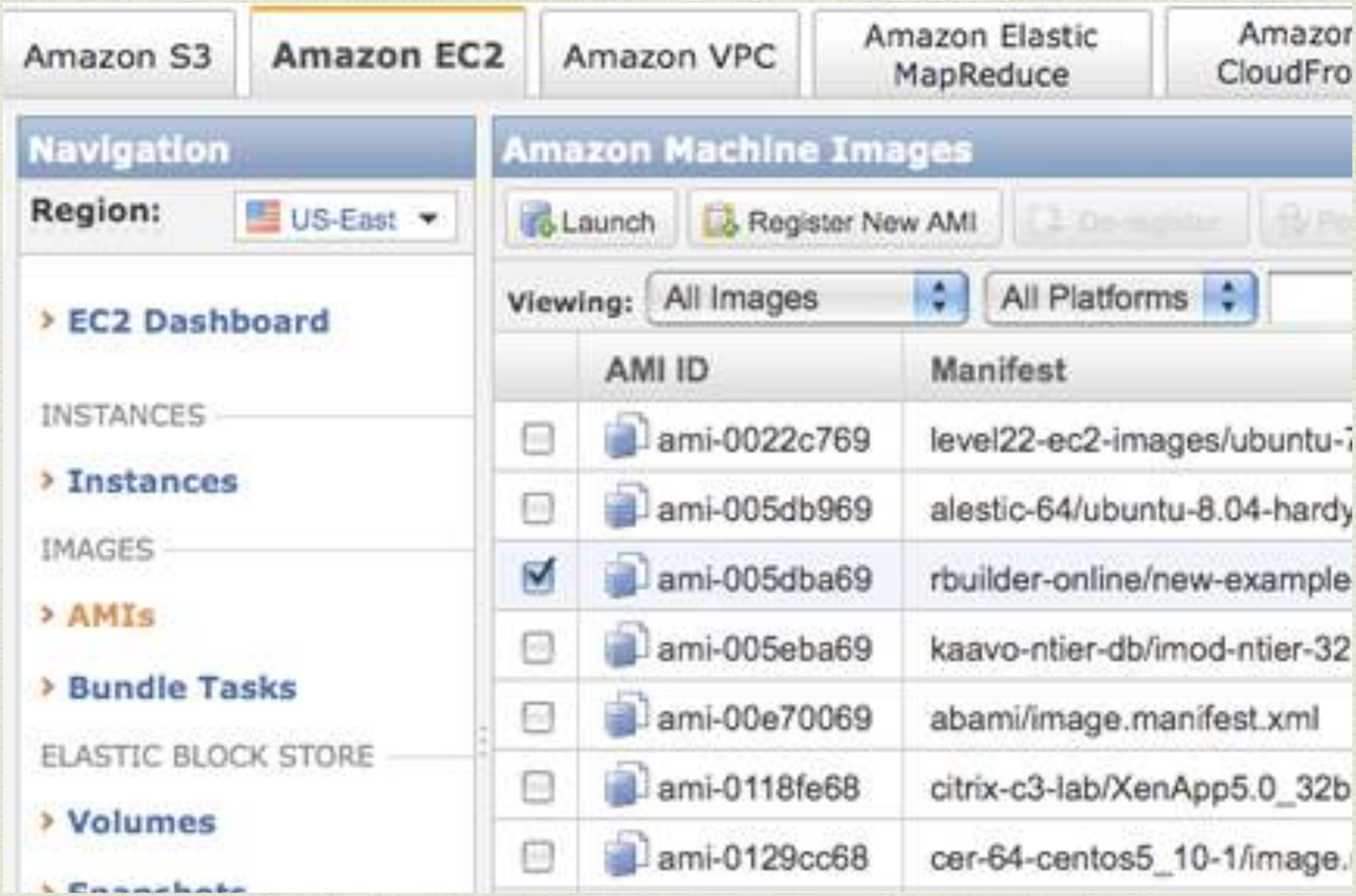
US – N. Virginia | US – N. California | EU – Ireland | APAC – Singapore

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

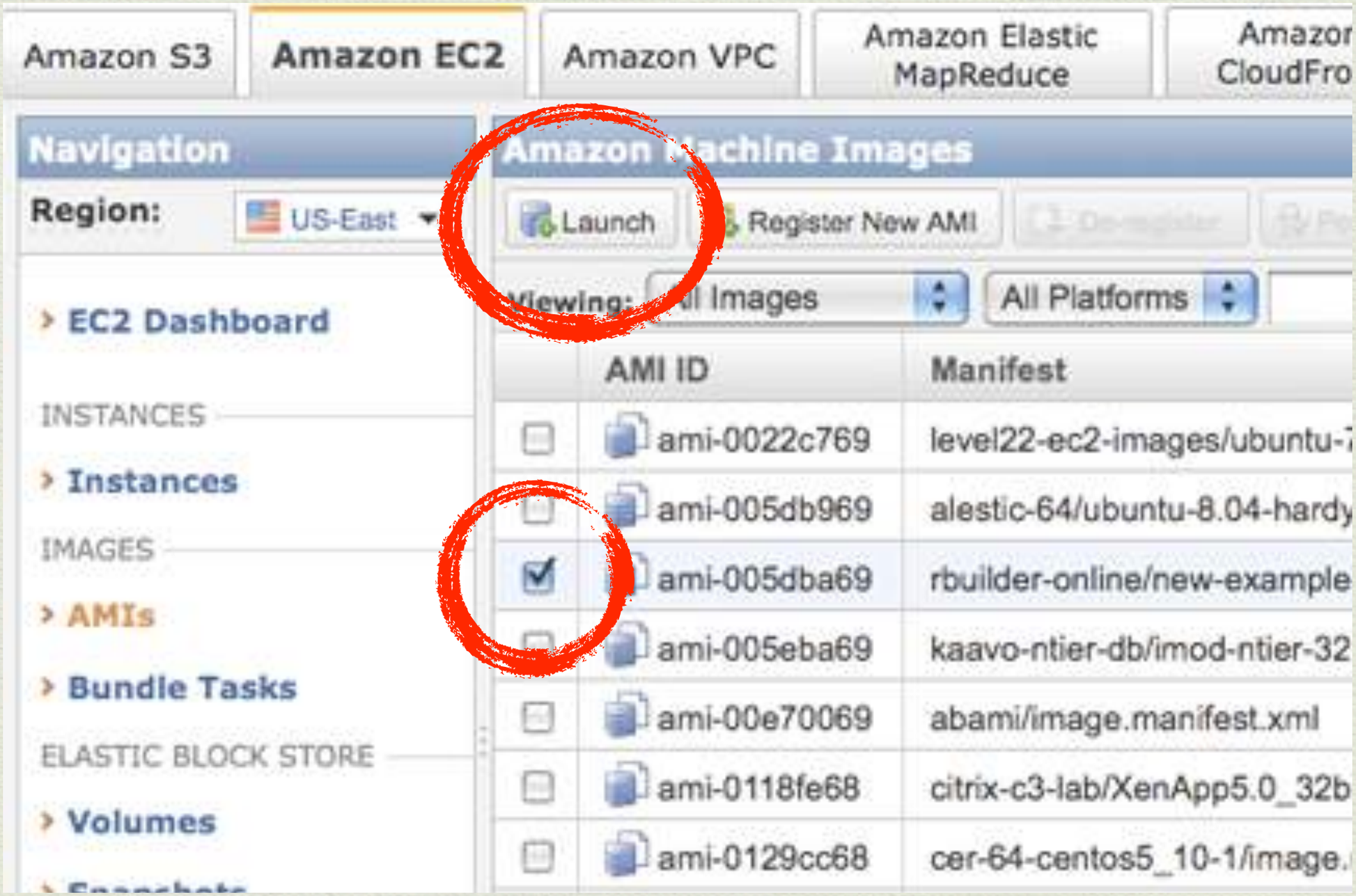
amazon web services

Prices and specification of AWS

Easy to use

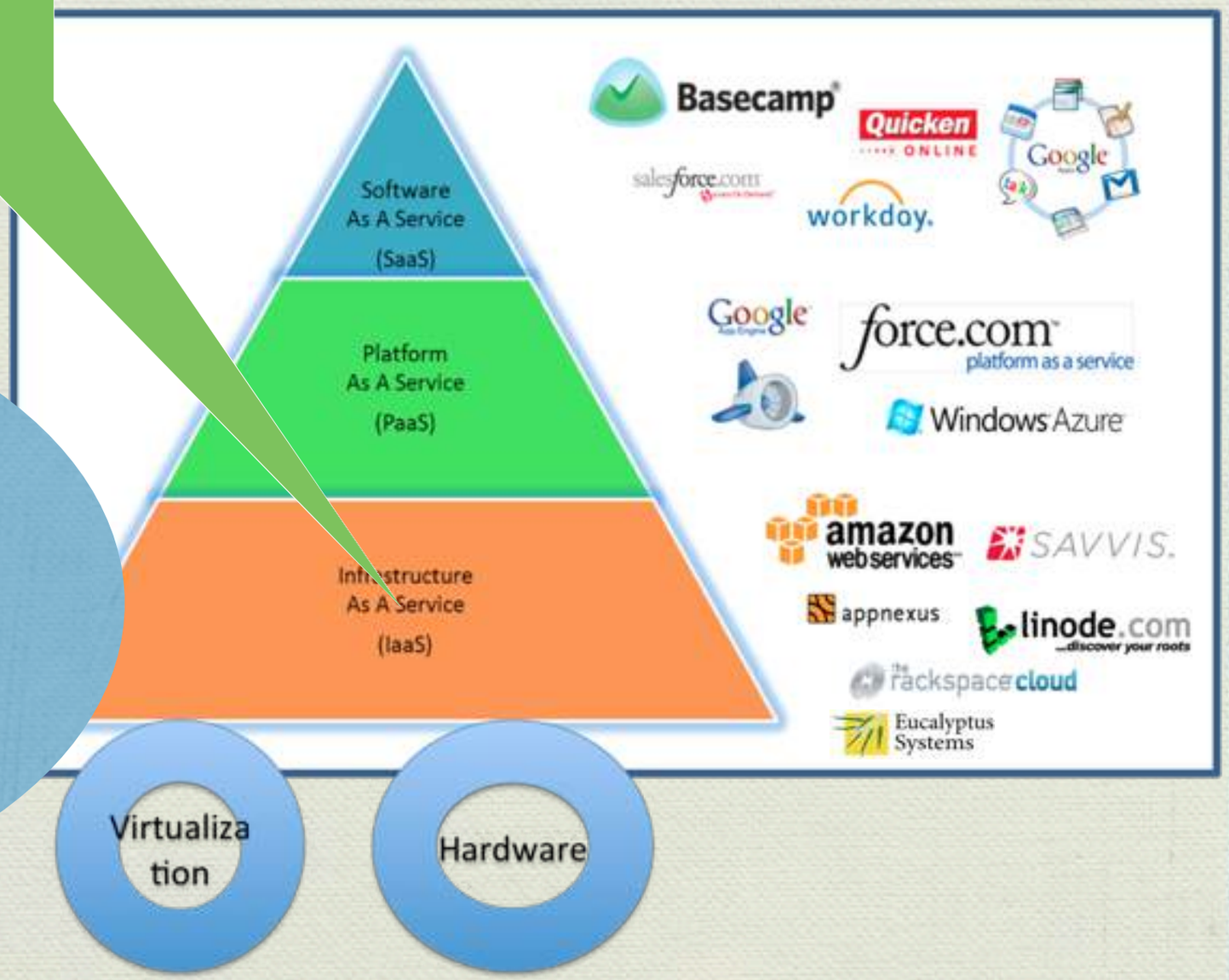

Amazon S3
Amazon EC2
Amazon VPC
Amazon Elastic MapReduce
Amazor CloudFro
Navigation
Region:
US-East
EC2 Dashboard
INSTANCES
Instances
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Amazon Machine Images
Launch
Register New AMI
Viewing:
All Images
All Platforms
AMI ID
Manifest
ami-0022c769
level22-ec2-images/ubuntu-
ami-005db969
alestic-64/ubuntu-8.04-hardy
ami-005dba69
rbuilder-online/new-example
ami-005eba69
kaavo-ntier-db/imod-ntier-32
ami-00e70069
abami/image.manifest.xml
ami-0118fe68
citrix-c3-lab/XenApp5.0_32b
ami-0129cc68
cer-64-centos5_10-1/image.

Easy to use


Amazon S3
Amazon EC2
Amazon VPC
Amazon Elastic MapReduce
Amazor CloudFro
Navigation
Region:
US-East
EC2 Dashboard
INSTANCES
Instances
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Amazon Machine Images
Launch
Register New AMI
Viewing:
All Images
All Platforms
AMI ID
Manifest
ami-0022c769
level22-ec2-images/ubuntu-
ami-005db969
alestic-64/ubuntu-8.04-hardy
ami-005dba69
rbuilder-online/new-example
ami-005eba69
kaavo-ntier-db/imod-ntier-32
ami-00e70069
abami/image.manifest.xml
ami-0118fe68
citrix-c3-lab/XenApp5.0_32b
ami-0129cc68
cer-64-centos5_10-1/image.

Cloud application

Run our applications there

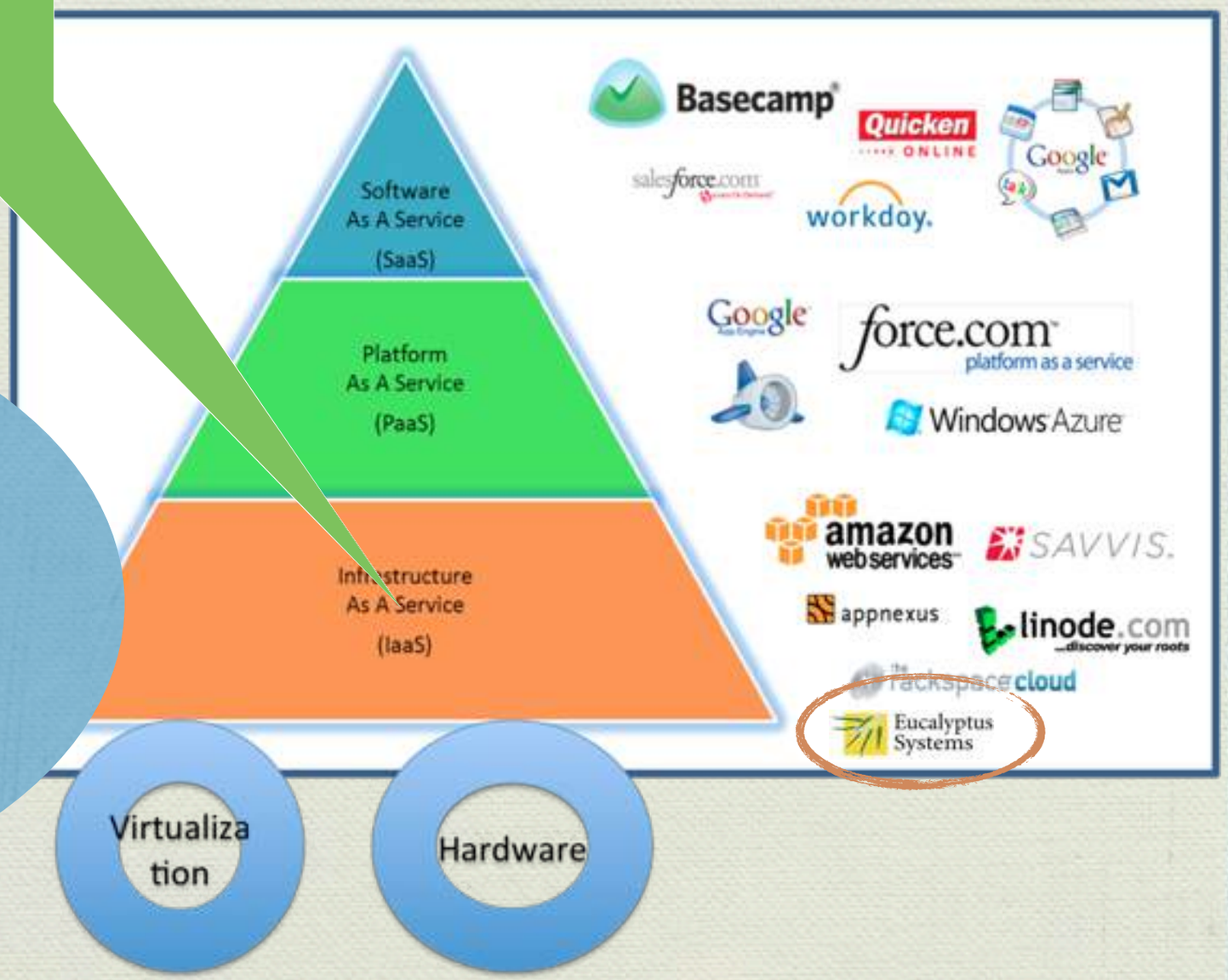


Why use cloud?
-Use a large resource pool
-Scalability
-Reduce cost

Cloud application

Run our applications there

Why use cloud?
-Use a large resource pool
-Scalability
-Reduce cost



Eucalyptus Systems

- Eucalyptus are compatible to Amazon Web Service
- Eucalyptus enable us to have a private cloud
- Private cloud is a group of technology:
  - Intel's VT-x , KVM, xen, Virtio.....

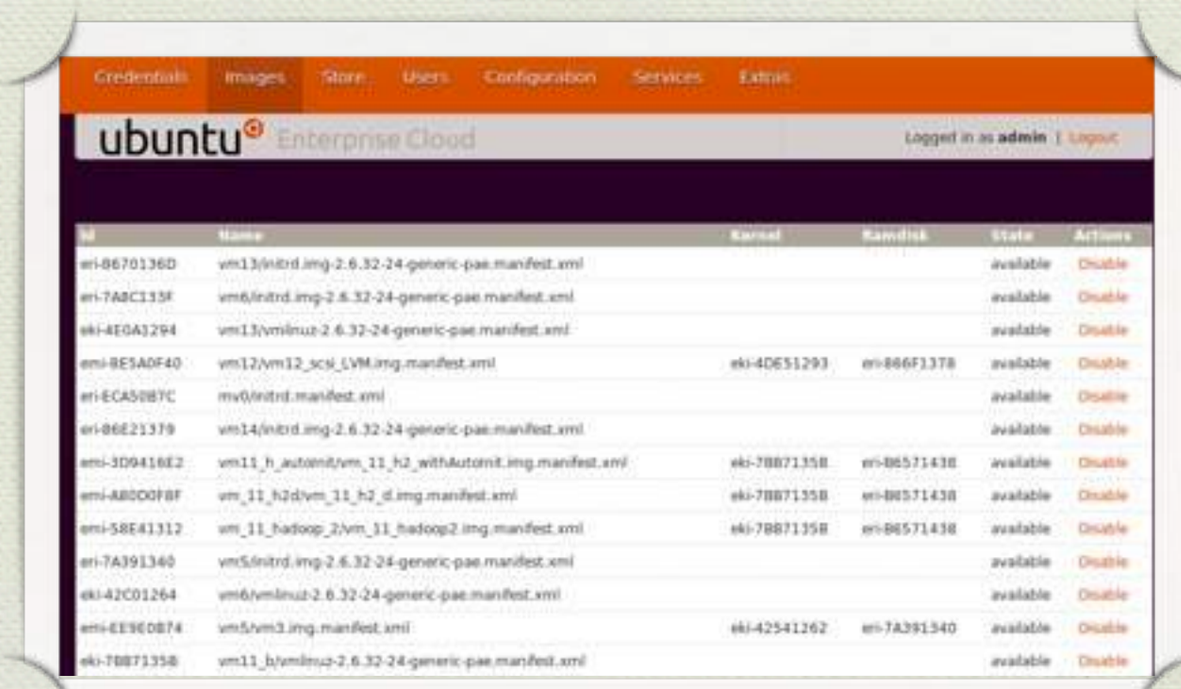
Structure of Eucalyptus

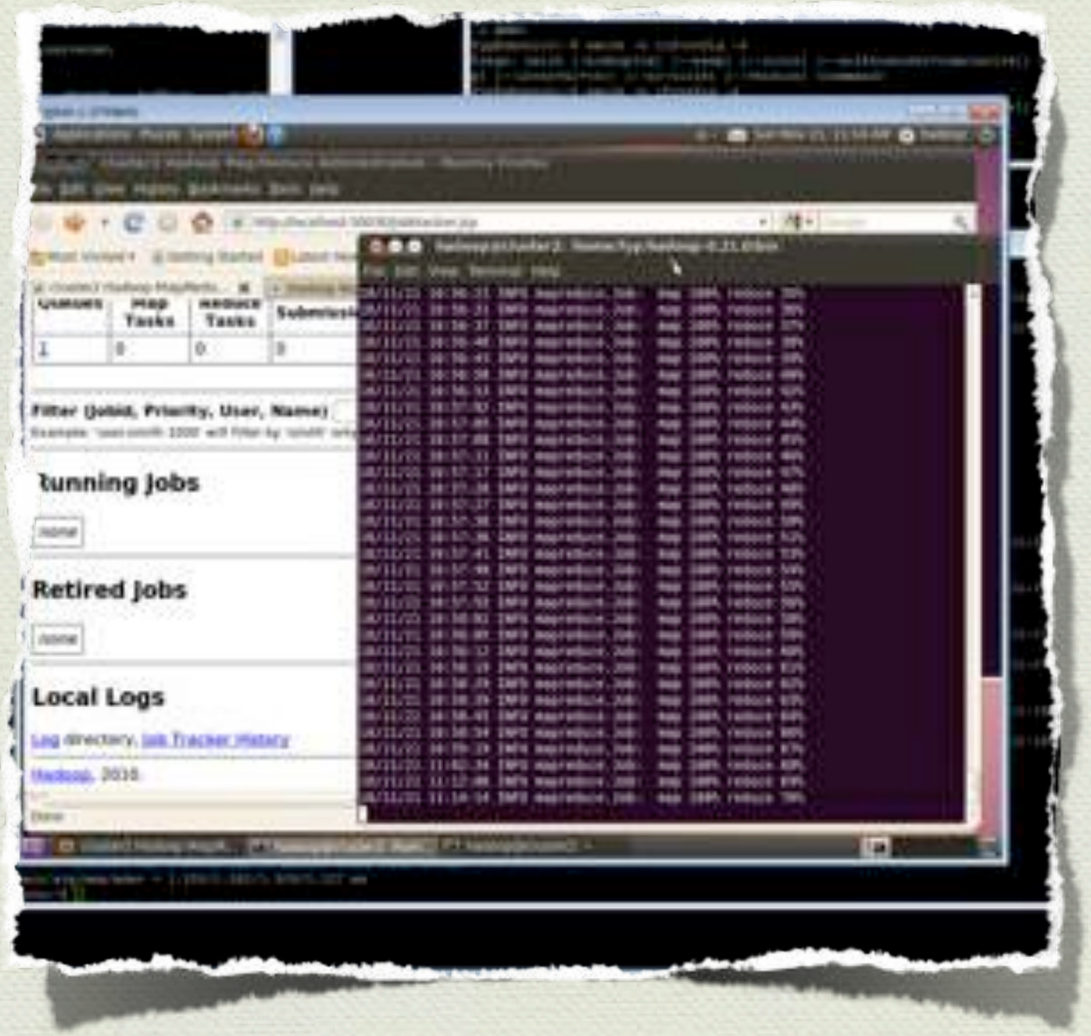
Controller

nodes

Our private cloud

- We built a tiny private cloud to understand the characteristics of cloud and Hadoop

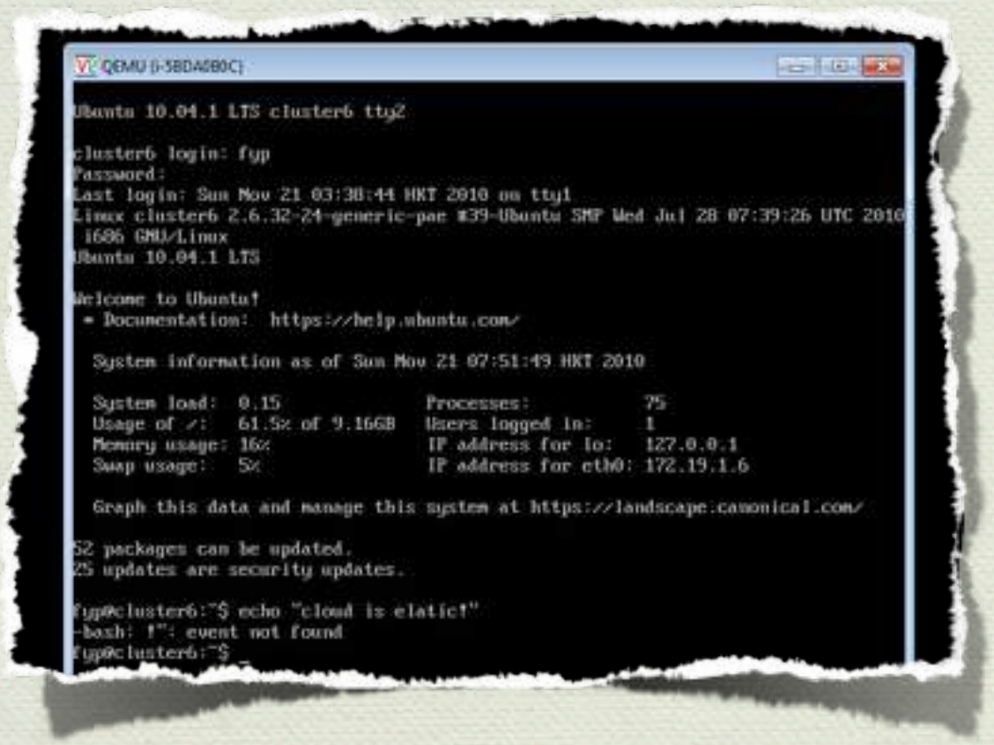




Ubuntu running Hadoop in Cloud

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

Our cloud's configuration



VNC play a role as display

Eucalyptus+Hadoop

- Allocate any number of virtual machines (elastic)
- Instance are independent to others.
- Put Hadoop on them

Eucalyptus+Hadoop

- Allocate any number of virtual machines (elastic)
- Instance are independent to others.
- Put Hadoop on them

Eucalyptus+Hadoop

- Allocate any number of virtual machines (elastic)
- Instance are independent to others.
  - Put Hadoop on them

An software layer to run a parallel application

Demo 1

- Start up Hadoop.

Inside the cloud

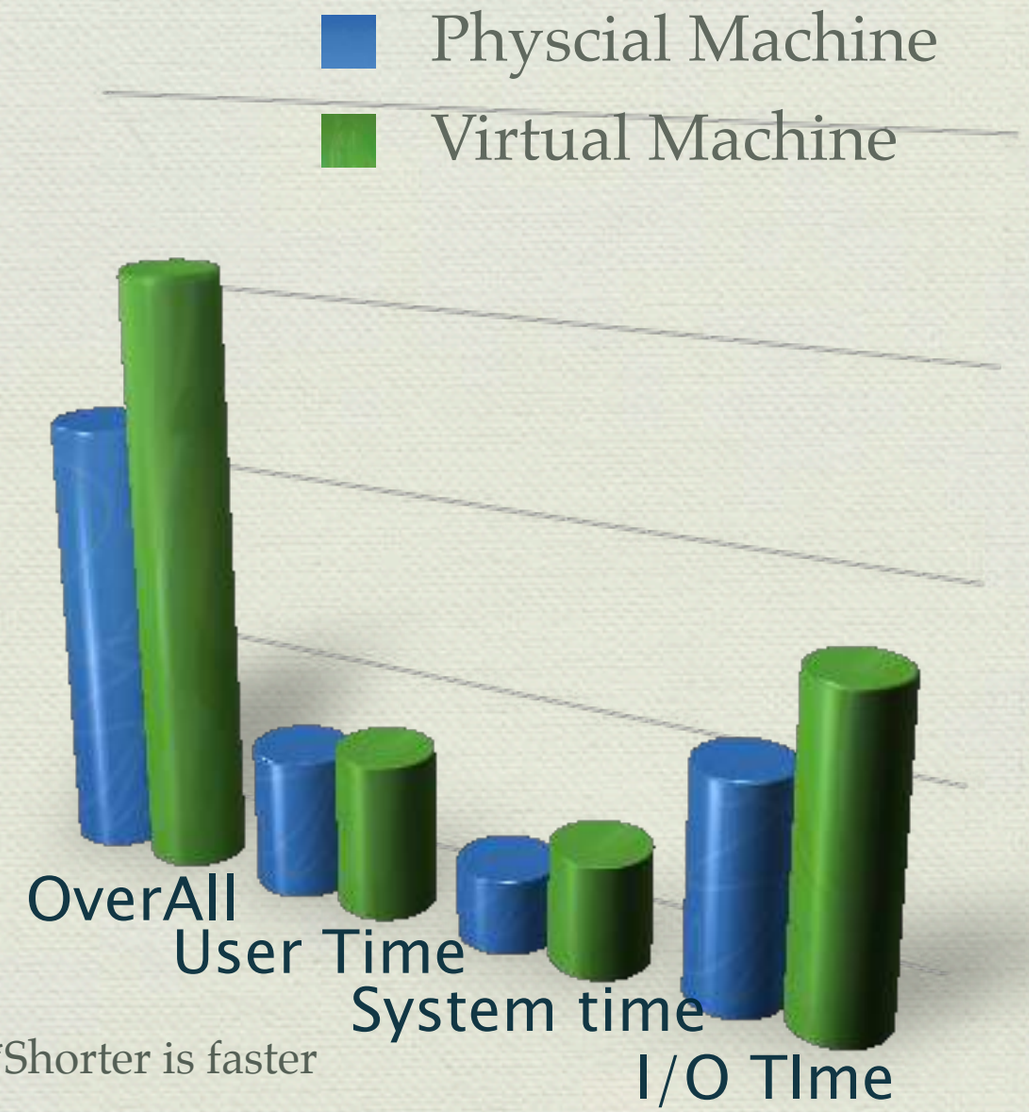
Initialize all node (e.g. change node names)

Eucalyptus is useful

- Reduce the time to start up a large application
- Environment is similar to host
- Run many applications at the same time.

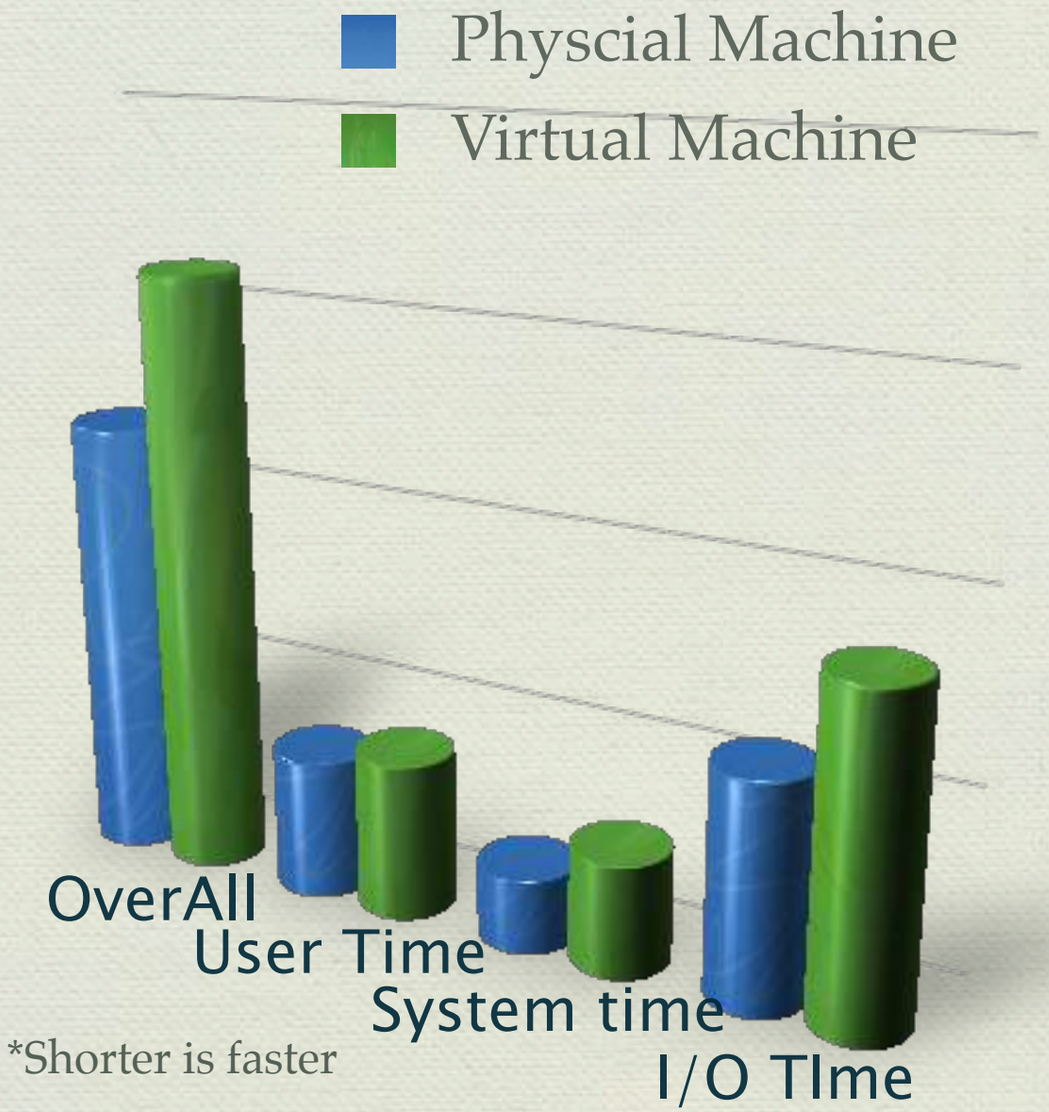
Performance Test(Stand alone)

- 1.3X overall Time(76%)
- 1.12X user time(89%)
- 1.6X system time(62.5%)
- 1.47X IO time(68%)



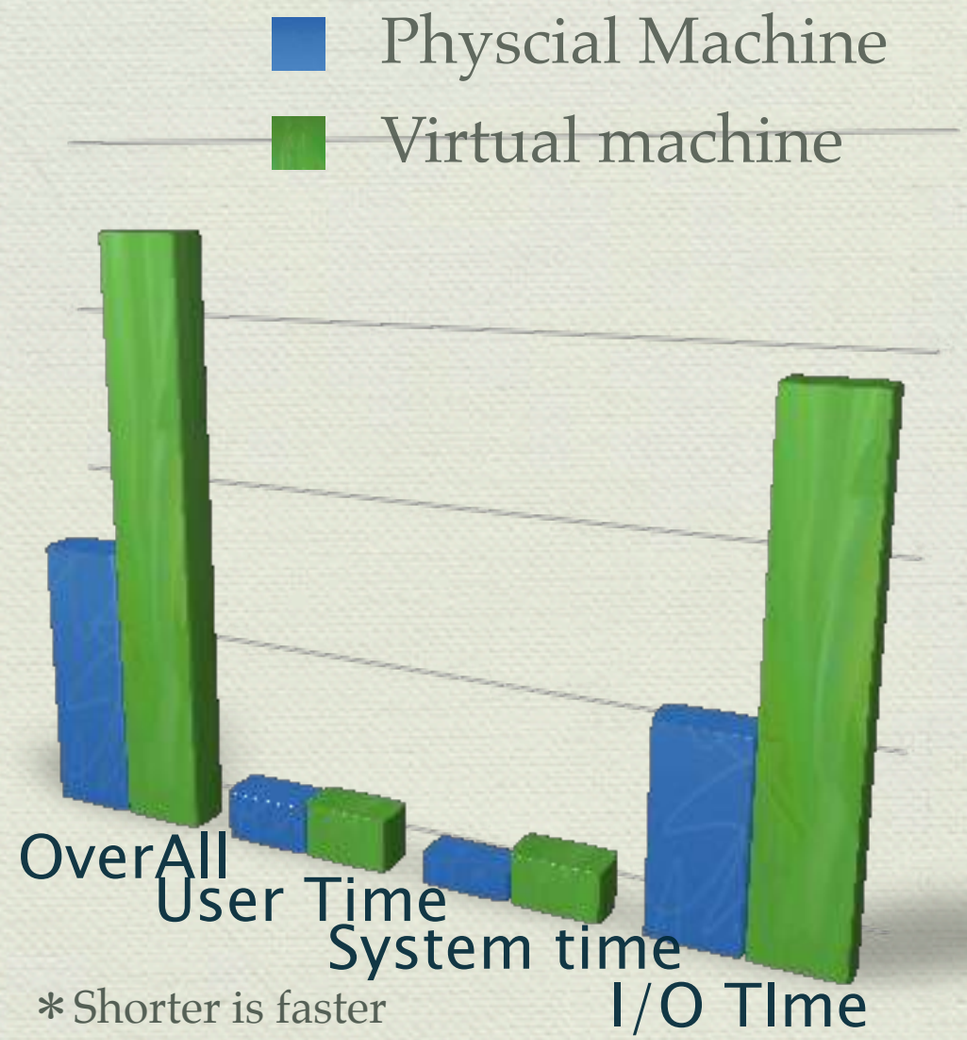
Performance Test(Stand alone)

- 1.3X overall Time(76%)
- 1.12X user time(89%)
- 1.6X system time(62.5%)
- 1.47X IO time(68%)



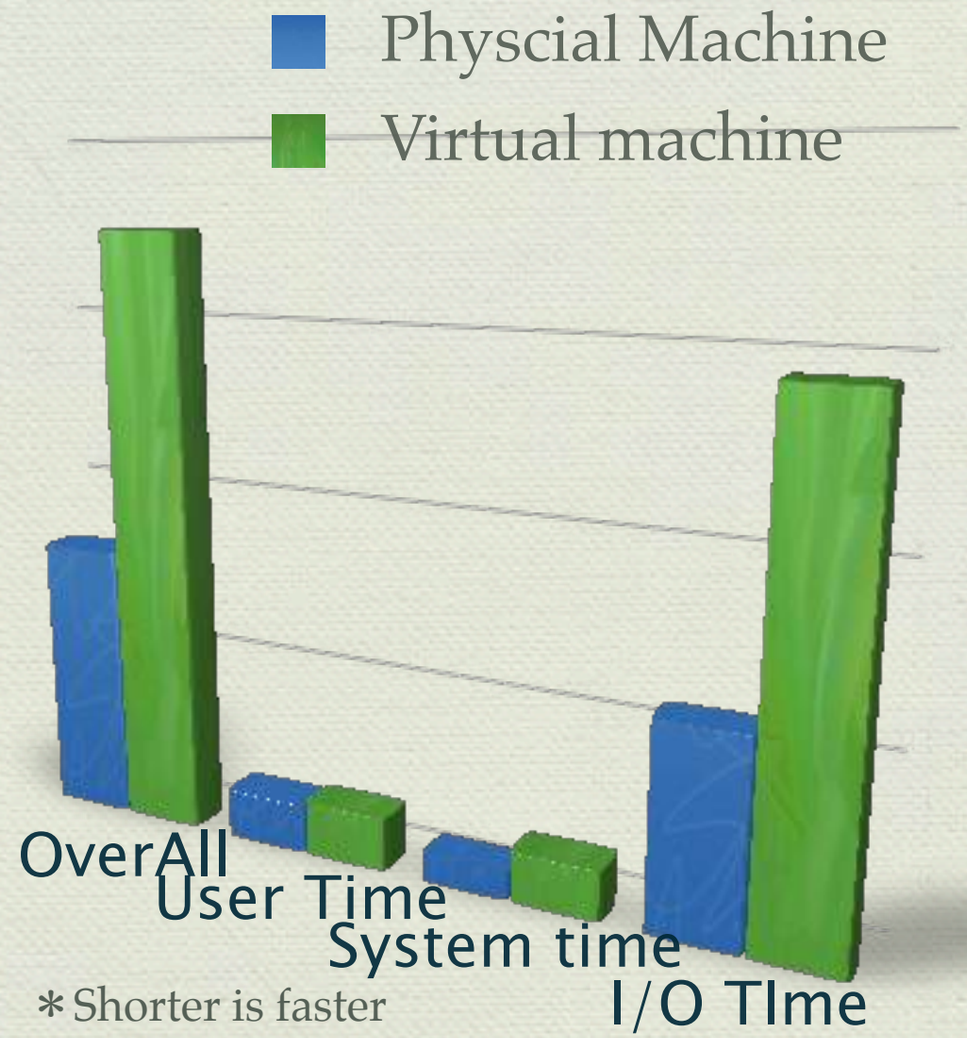
Performance Test(run two process at the same time)

- 2.3 X IO time(43%)



Performance Test(run two process at the same time)

- 2.3 X IO time(43%)



Performance Test

(Pure computation , memory access & bandwidth)

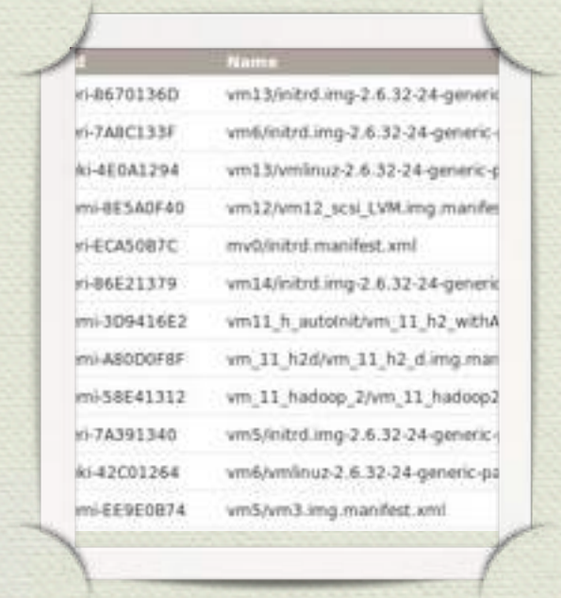
Computation	Memory access	Bandwidth
97.5%	76%	~50% (~20MBytes/s)

Start up time

- 5GB ~10mins
- 10GB~20MINS
- AWS ~ 2 mins
  - faster network

Advantage of using Eucalyptus

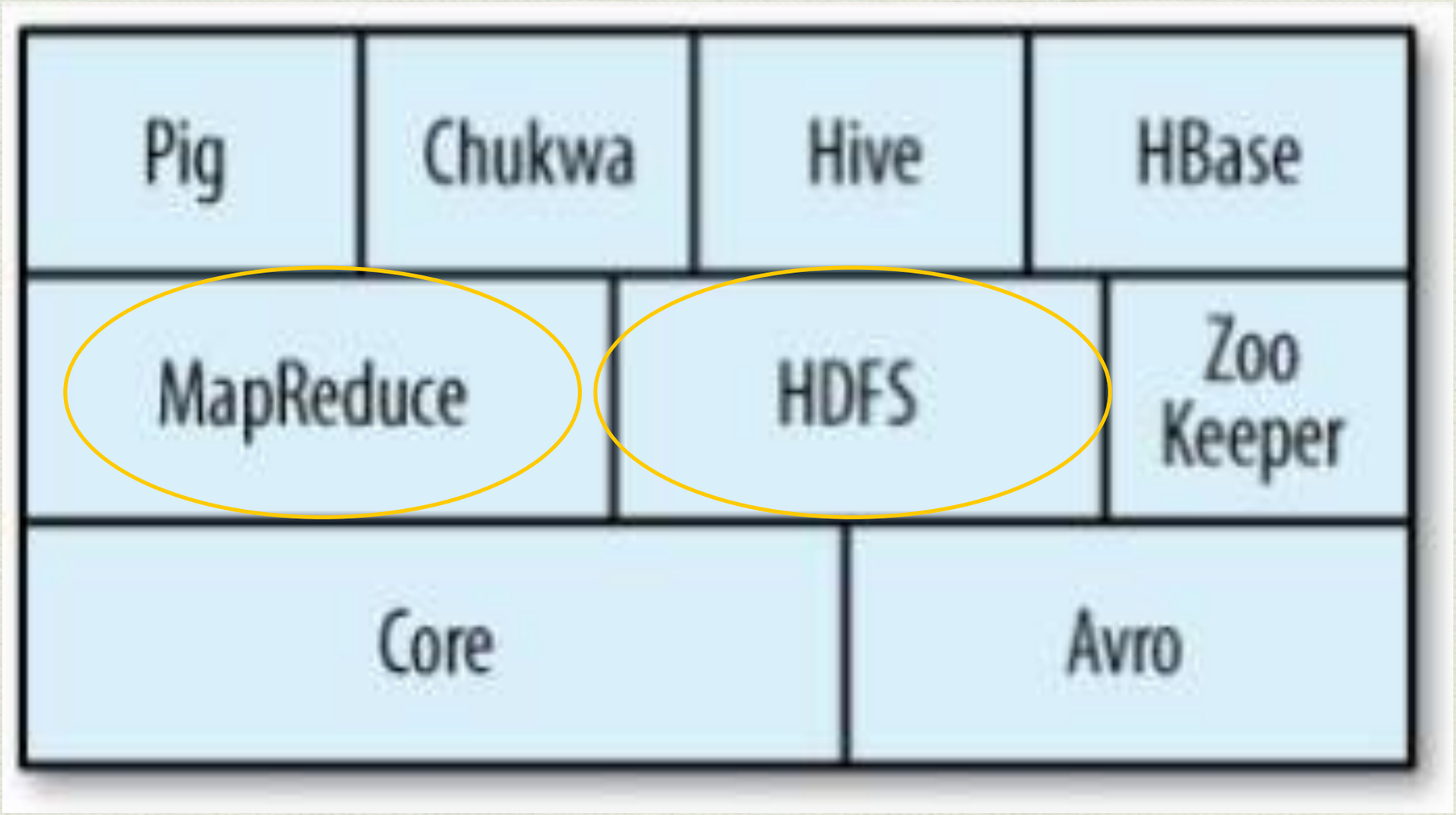
- Fast system set up
- make general resource shareable
- reduce cost
- Support many platform



Weakness Eucalyptus

- Performance trades off
- Instance can’t be saved
- Slow instance start up
- Need a large resource pool
  - Runing large scale application on public cloud is a good solution

hadoop


Pig
Chukwa
Hive
HBase
MapReduce
HDFS
Zoo Keeper
Core
Avro

WHO ARE USING HADOOP

facebook

YAHOO!

The New York Times

Adobe

amazon.com

hulu

Baidu 百度

AOL

Microsoft

Facebook:

4 TB of compressed new data added per day

135TB of compressed data scanned per day

Size:

4800 cores, 5.5 PetaBytes storage

12 TB per node

Terasort Benchmark:

100 TB in 173 minutes

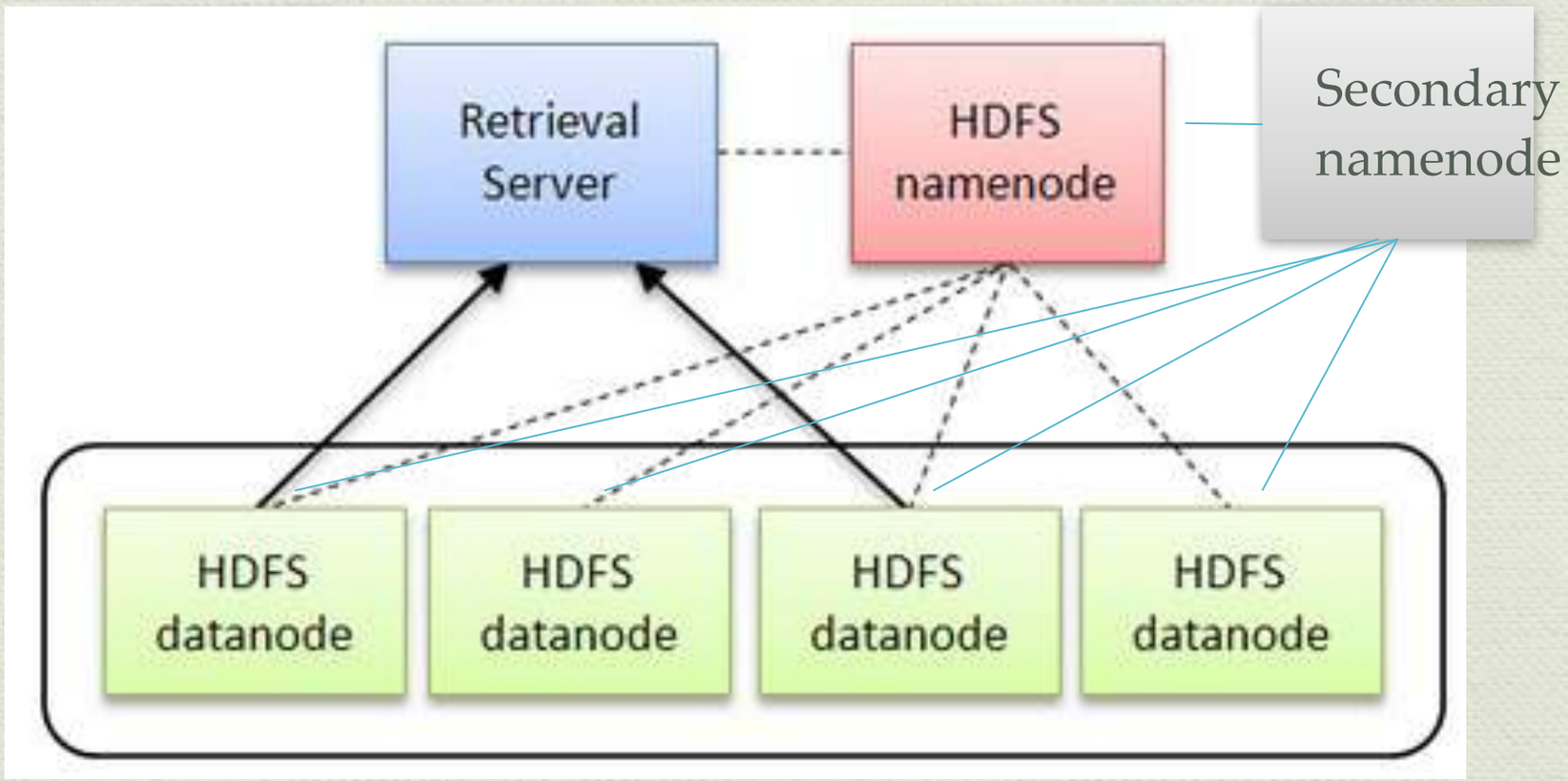
3452 nodes x (2 Quadcore Xeons, 8 GB memory, 4 SATA)

- Store and process Petabyte of Data
- Handle frequent hard drive failure
- Have efficient I/O in large scale of data

HDFS

- Store and process Petabyte of Data
- Handle frequent hard drive failure
- Have efficient I/O in large scale of data

Stores all metadata In RAM and HDD



Heartbeats

Block Placement

Rebalancer

Data Correctness

Data Pipelining

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

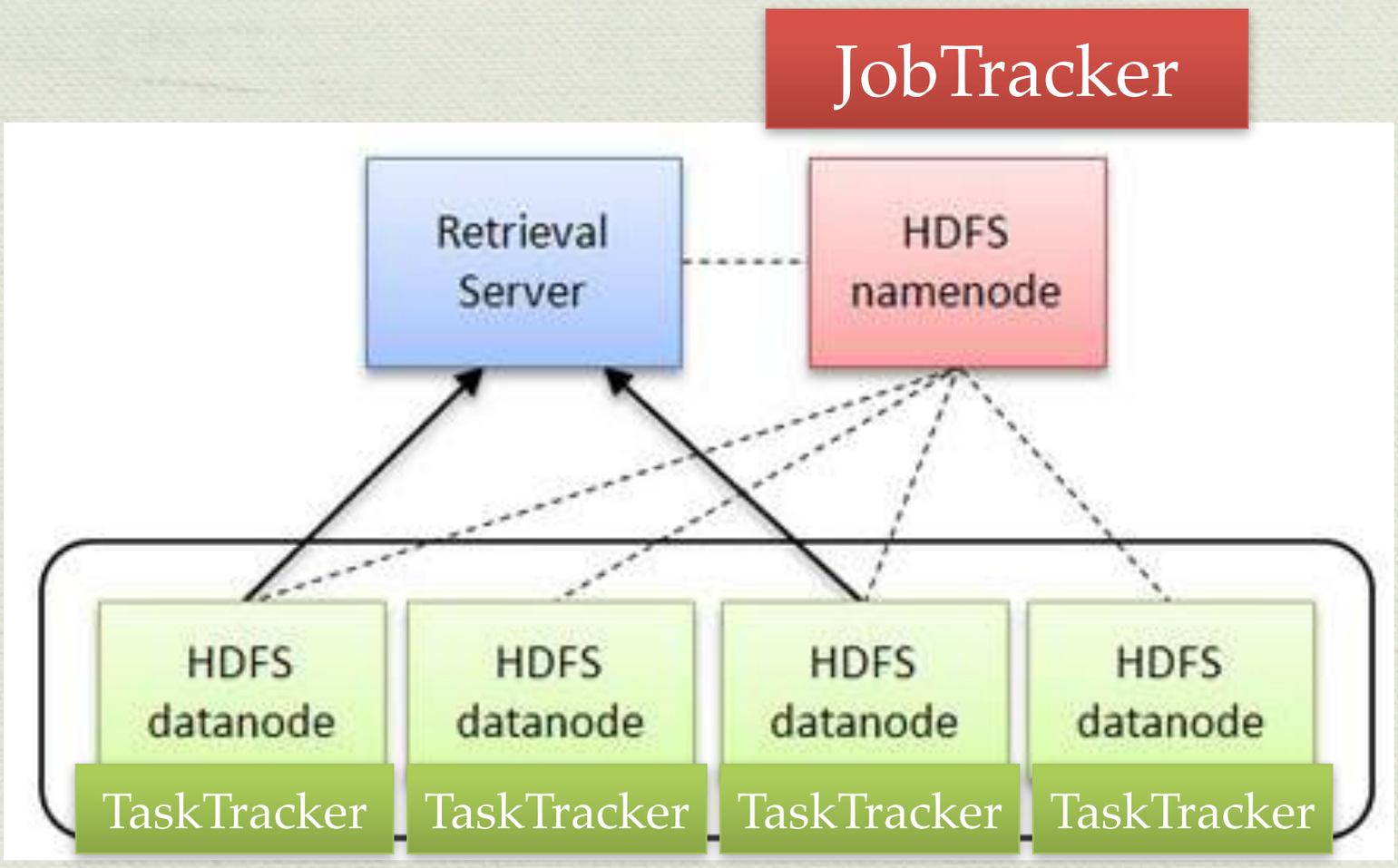
Live Datanodes : 3

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.

Map Reduce

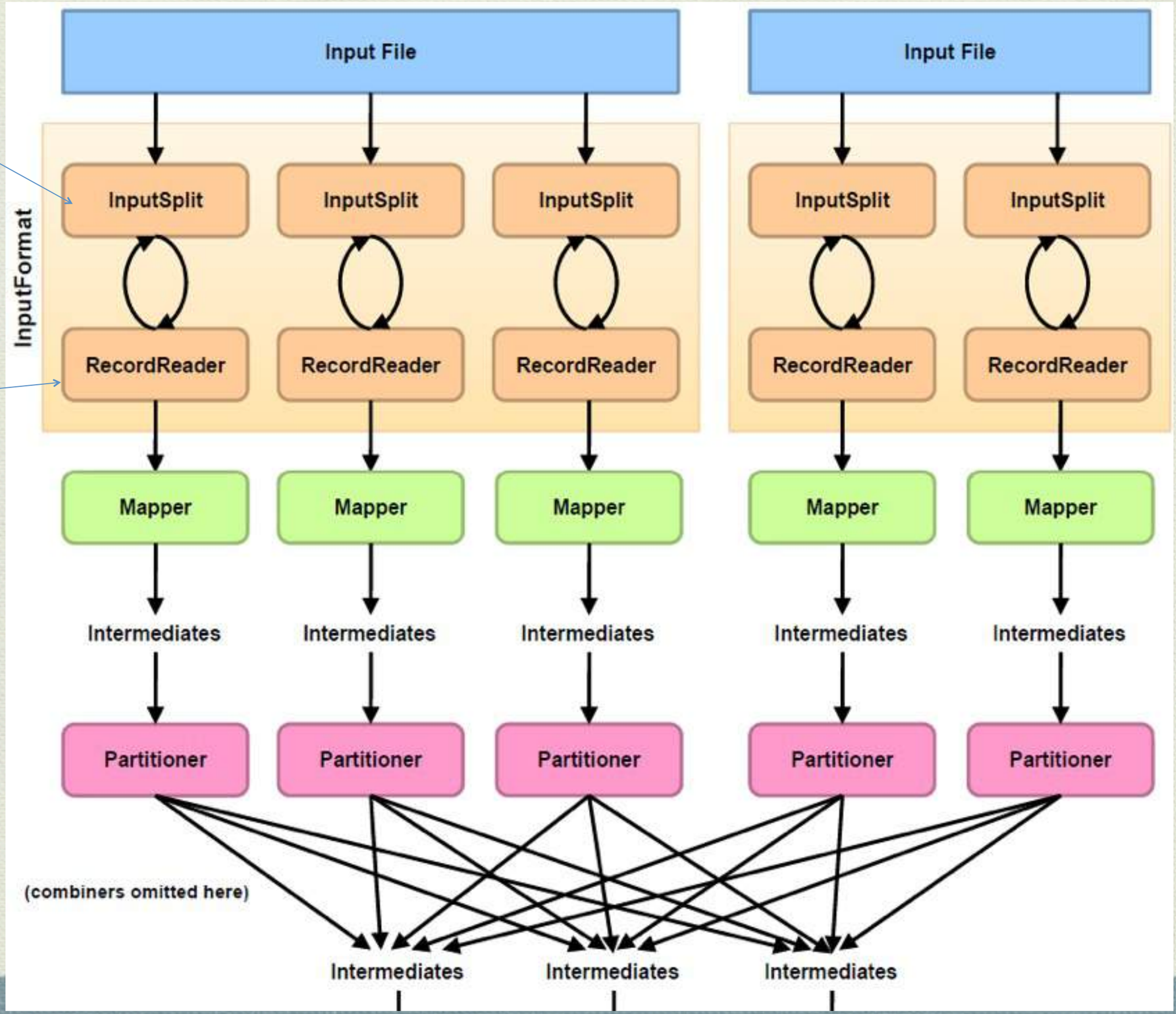
- One JobTracker
- Many TaskTrackers

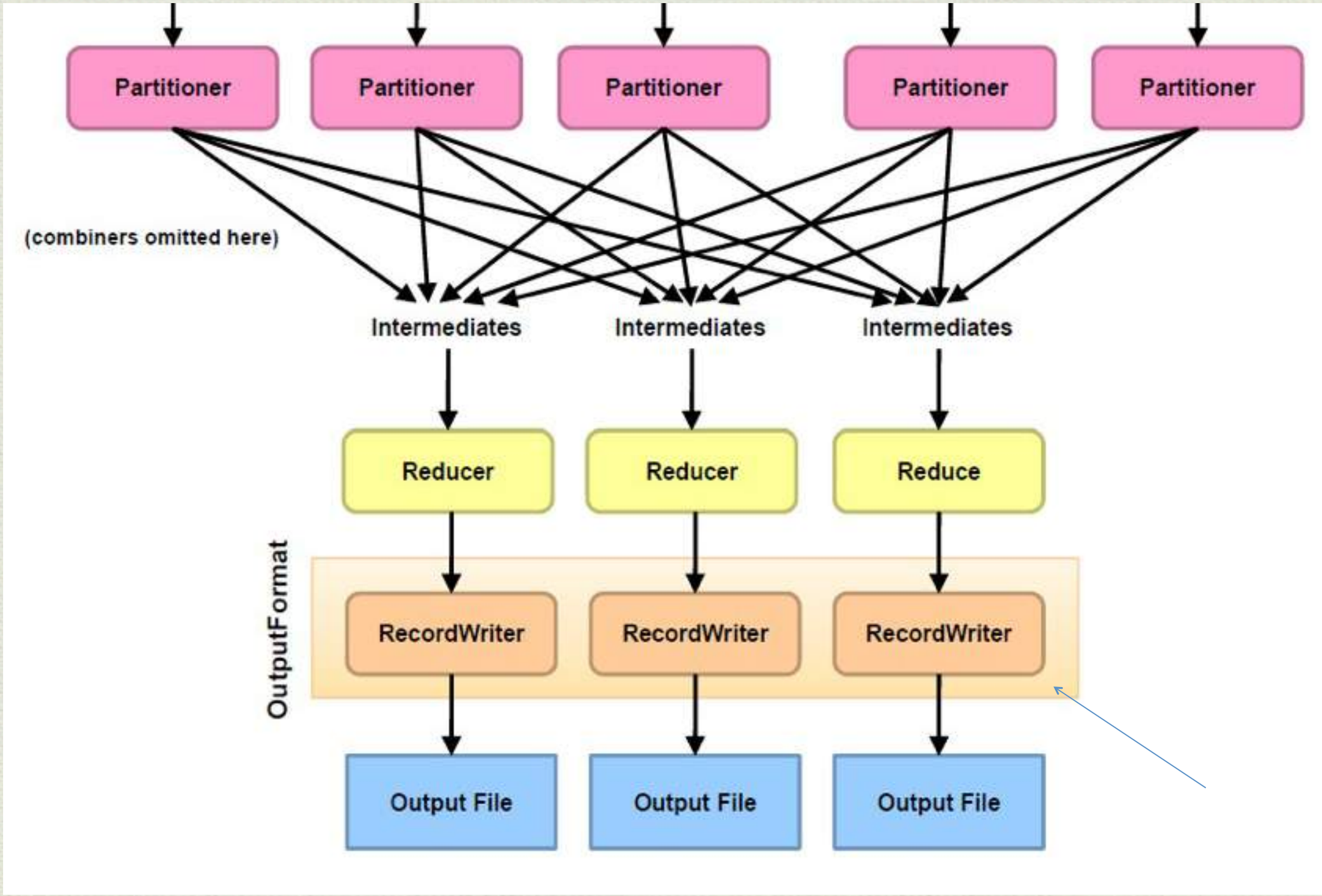


Map Reduce

- Automatic parallelization and distribution
- Fault-tolerance
- Backup Tasks

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		


Input File
Input File
InputFormat
InputSplit
InputSplit
InputSplit
InputSplit
InputSplit
RecordReader
RecordReader
RecordReader
RecordReader
RecordReader
Mapper
Mapper
Mapper
Mapper
Mapper
Intermediates
Intermediates
Intermediates
Intermediates
Intermediates
Partitioner
Partitioner
Partitioner
Partitioner
Partitioner
(combiners omitted here)
Intermediates
Intermediates
Intermediates


Partitioner
Partitioner
Partitioner
Partitioner
Partitioner
(combiners omitted here)
Intermediates
Intermediates
Intermediates
Reducer
Reducer
Reduce
OutputFormat
RecordWriter
RecordWriter
RecordWriter
Output File
Output File
Output File

```java
// mapper
private static class MapClass extends MapReduceBase implements
        Mapper<String, ImageWritable, Text, ImageWritable> {

    @Override
    public void map(String key, ImageWritable value,
            OutputCollector<Text, ImageWritable> output, Reporter reporter)
            throws IOException {

        Date startD = new java.util.Date();
        System.out.println("###Map:"+key+" Date:"+startD);

        EdgeDetector detector = new EdgeDetector();
        detector.setLowThreshold(0.5f);
        detector.setHighThreshold(1f);
        detector.setSourceImage(value.bufimage);
        detector.process();
        output.collect(new Text(key), new ImageWritable(detector.getEdgesImage()));

        //output.collect(new Text(key), value);
        System.out.println("###Map.Time_delay:"+new java.util.Date()+ " (ms)"+(new java
    }
}
```

```java
// reducer
private static class ReduceClass extends MapReduceBase implements
        Reducer<Text, ImageWritable, Text, ImageWritable> {

    @Override
    public void reduce(Text key, Iterator<ImageWritable> values,
            OutputCollector<Text, ImageWritable> output, Reporter reporter)
            throws IOException {
            //now is single image per job only
        Date startD = new java.util.Date();
        System.out.println("###Reduce:"+key+" Date:"+startD);
        if(values.hasNext()){
            ImageWritable val = values.next();
            output.collect(key, val);
        }
        System.out.println("###Reduce.Time_delay:"+new java.util.Date()+ " (ms)"+(new
    }
}
```

Pilot applications

- WordCount
- Image Edge detection
- Md5 Calculation
- Video Conversion

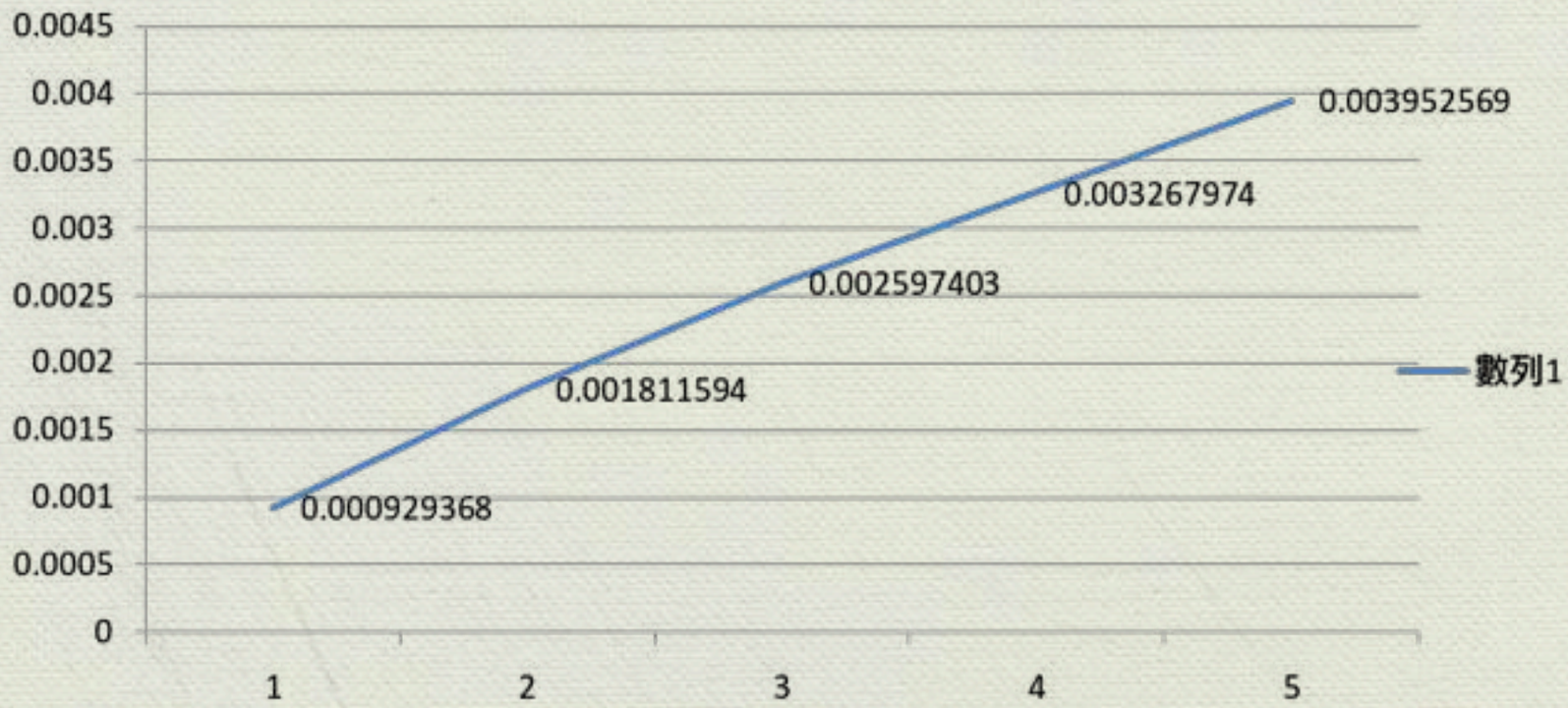
Test 2

edge detection

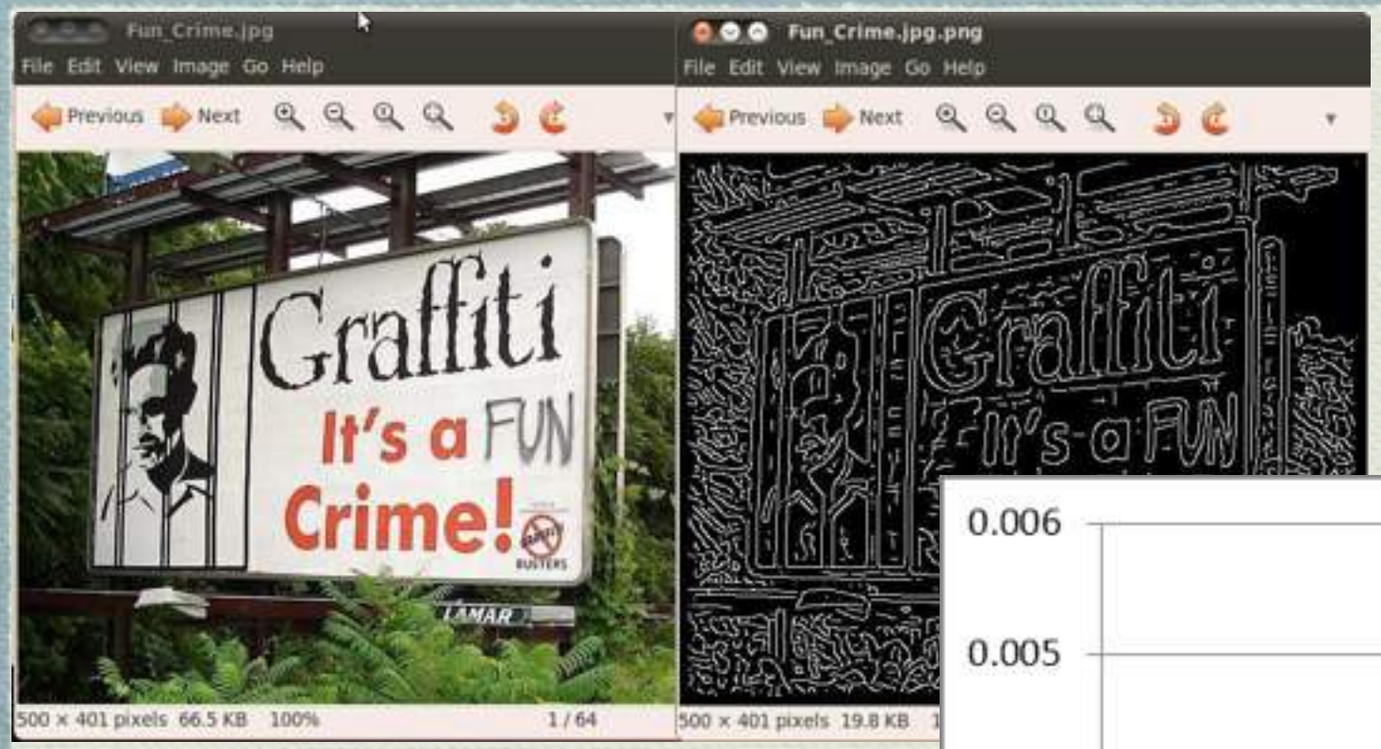
WordCount

Test on I/O of HDFS
Full function of MapReduce
Map,Reduce,Combine,Sort

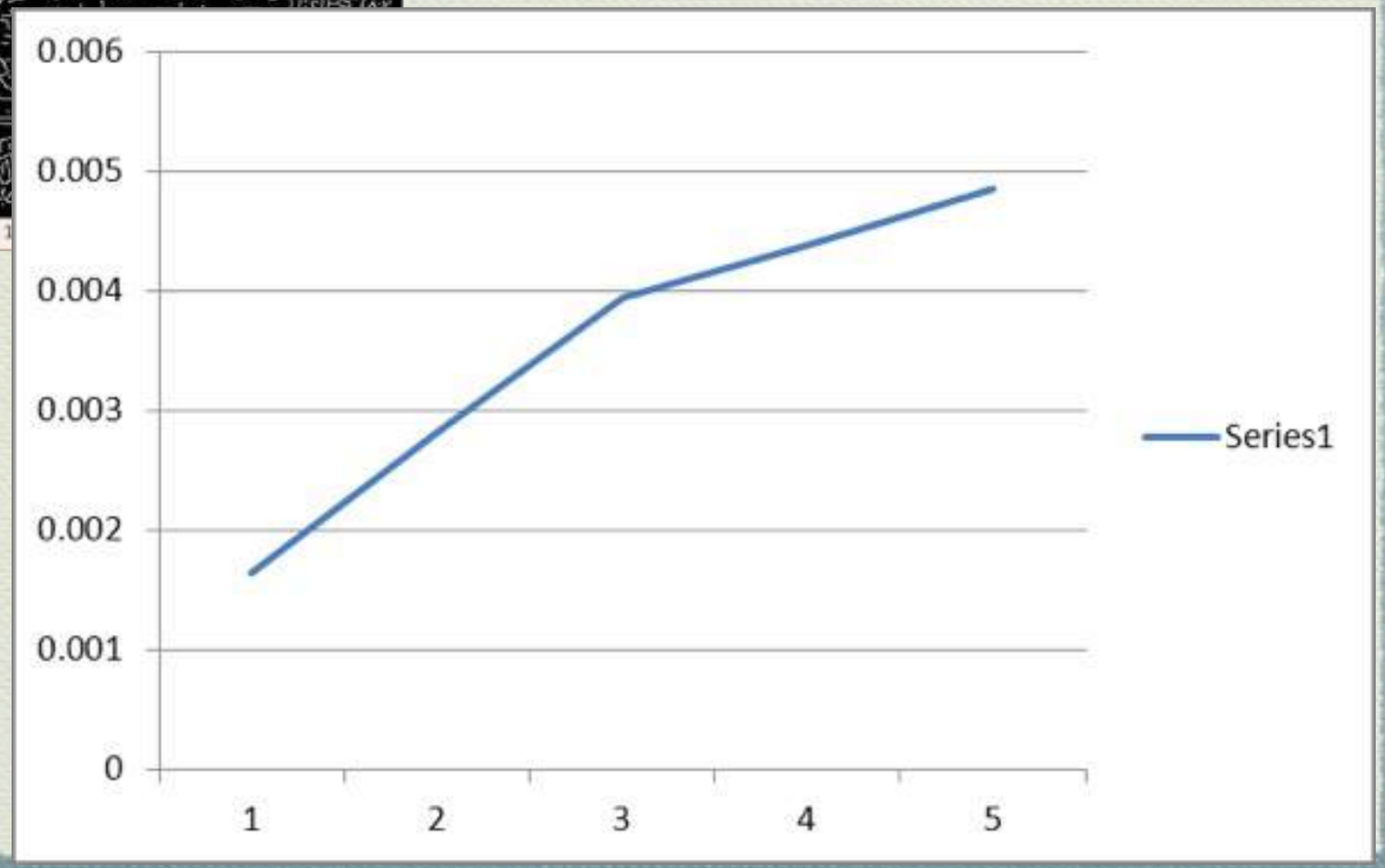
```
a     3001
the   2023
as    240
....
....
```



Edge detection

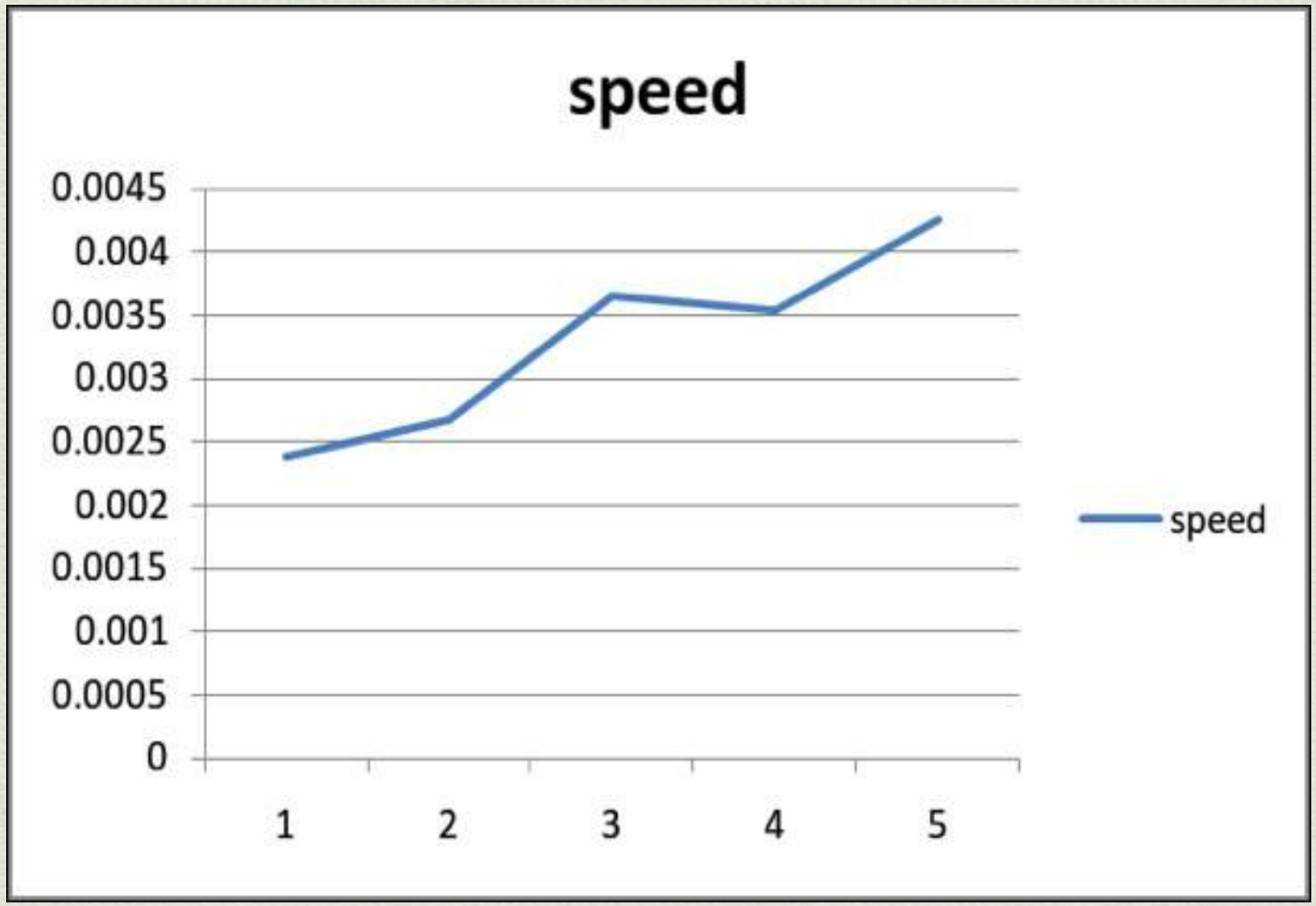


Normal network traffic
Normal I/O access on HDFS



Md5 Calculation

- Frequently appending text to file in HDFS
- Low network traffic



View Next chunk

```
100000  14ee22eaba297944c96afdbe5b16c65b
100001  e2a6alace352668000aed191a817d143
100002  bb36c34eb6644ab9694315af7d68e629
100003  3dc81e3f2c523fb5955761bbe2d150f2
100004  lea85063355fbfad3de73ab038261d62
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
```

Video conversion

using the FFmpeg library

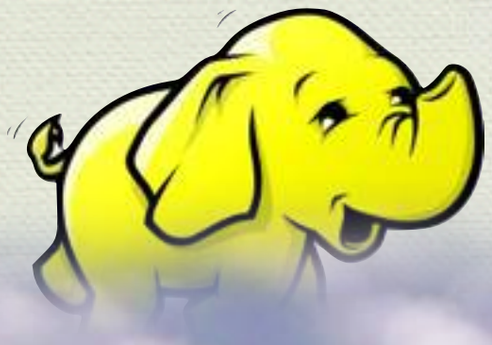
only uses the Mappers and I/O of HDFS.
High network traffic

Conclusion on Hadoop

- Write once read many access model
- Good in Scalability
- Not efficient for computation intensive jobs
- Fault-tolerance in both HDFS/MapReduce
- Easy parallelization(coding and deploying)
- Easy large-scale computations

Why Hadoop needs to run in Cloud

- Flexible in Hadoop cluster managements
- Account isolation
- Reduce cost



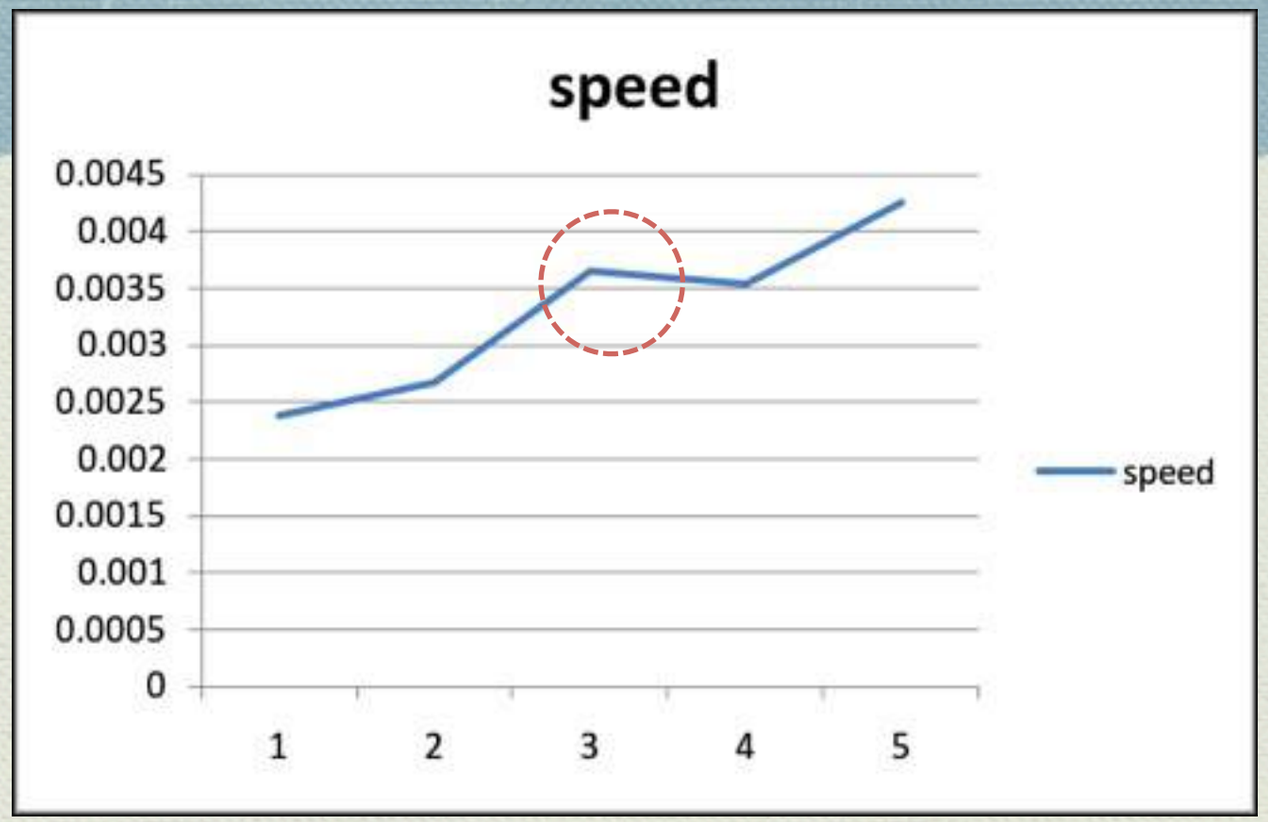
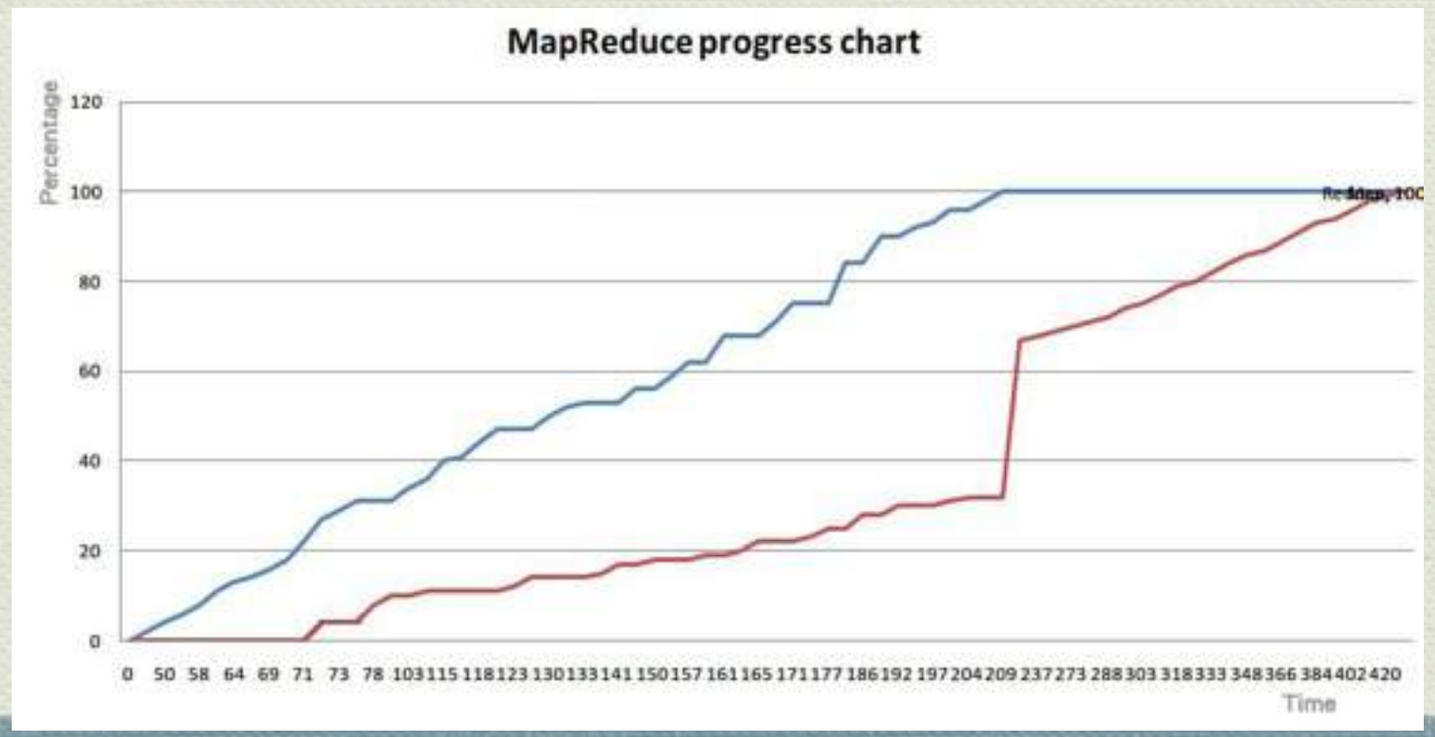
What will we do next…

- Hive and HBase

- Make use of Hadoop and Cloud

e.g. Video searching

Data mining

…

Unsolved problems