Big Data and IBM Solution

Zeming Zhao

zhaozm@cn.ibm.com

November 1, 2014

Agenda

- Big Data

- Hadoop Brief Introduction

- HDFS Structure and Characteristic

- MapReduce Model

- Q&A

What is Big Data

○ Big data is being generated by everything **around us** at all times. Every digital process and social media exchange produces it. Systems, sensors and mobile devices transmit it.

○ Big data is an all-encompassing term for any collection of data sets so **large** and **complex** that it becomes difficult to process using traditional data processing applications.

3Vs Characteristics of Big Data

○ Volume: How much data

○ Velocity: How fast data is processed

○ Variety: The various types of data

Changes made by Big Data

○ ## Competitive advantage

Data is emerging as the world's newest resource for competitive advantage.

○ ## Decision making

Decision making is moving from the elite few to the empowered many.

○ ## Value of data

platform,analyzing, collection and visualization

Big Data Technology

○ ## Systems

Your infrastructure must capitalize on real-time information flowing through your organization. It must be optimized for analytics to respond dynamically to the increasing demands of big data.

○ ## Privacy

Your platform must comprise stringent policies and practices around privacy and insights on which your business relies.
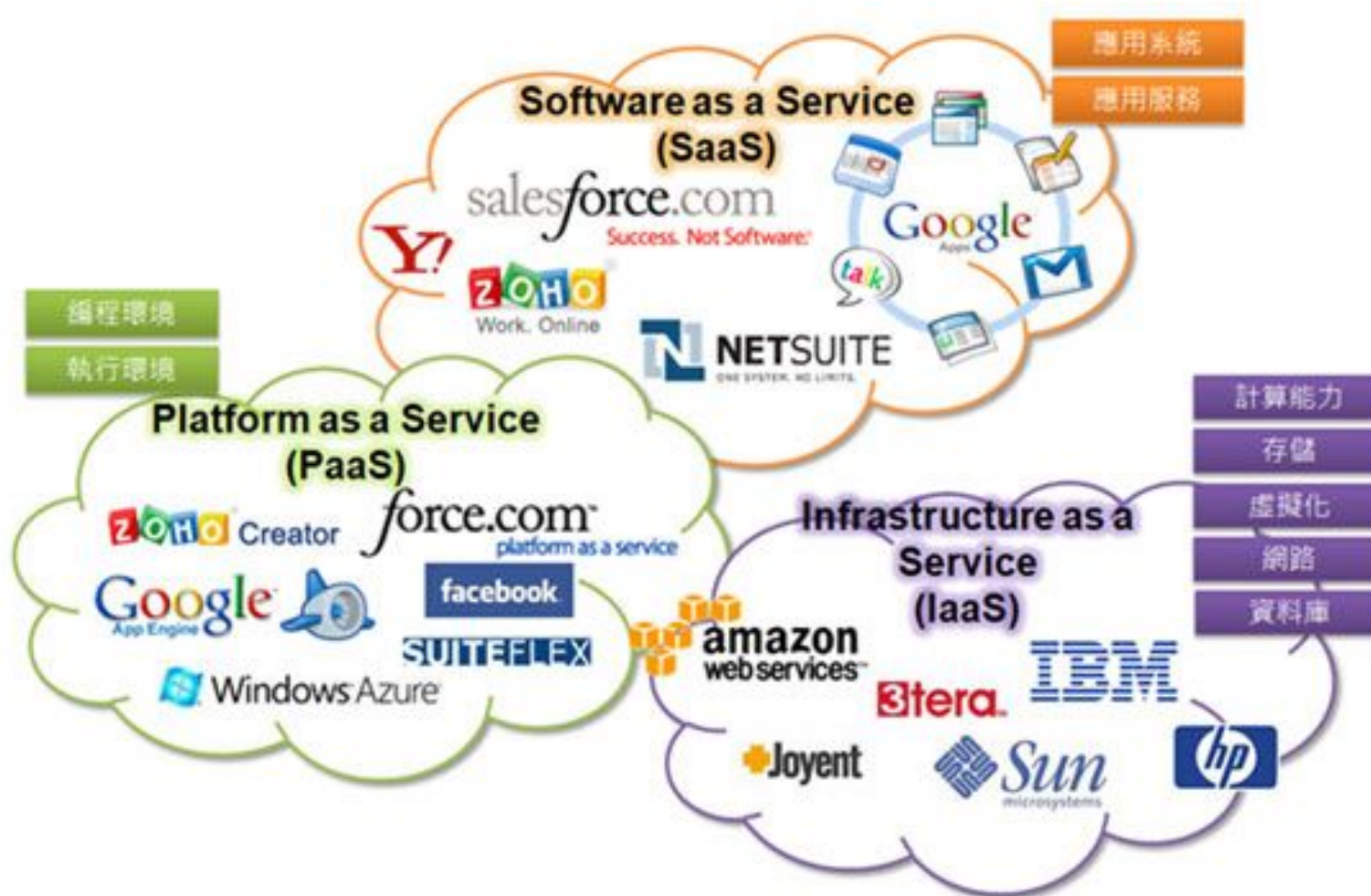
○ ## Storage

Your infrastructure must embody a defensible disposal strategy that reduces the run rate of storage and risk.

○ ## Security

Your infrastructure must have strong security measures built in to guard your organization against internal and external threats.
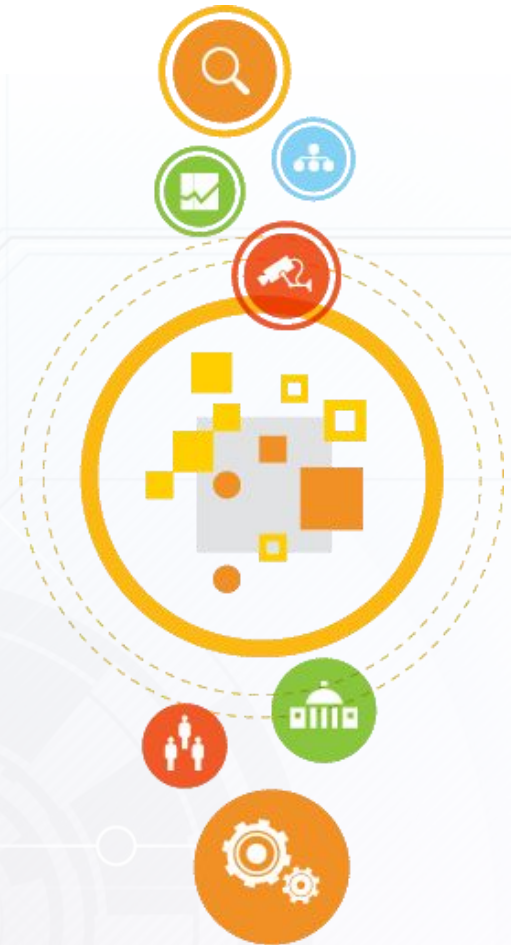
Categories of Data Technology

Big Data Technology

Type	Consumer	Service Provided by Cloud	Service Level Coverage	Customization
SaaS	End user	• Finished application	• Application uptime • Application Performance	• Minimal to no customization • Capabilities dictated by market or provider
PaaS	Application owner	• Runtime environment for application code • Cloud storage • Other Cloud services such as integration	• Environment availability • Environment performance • No application coverage	• High degree of application level customization available within constraints of the service offered • Many applications will need to be rewritten
IaaS	Application owner or IT provides OS, middleware, and application support	• Virtual server • Cloud storage	• Virtual server availability • Time to provision • No platform or application coverage	• Minimal constraints on applications installed on standardized virtual OS builds

8

Value/Opportunity of Big Data

○ **Discover the new role of data scientist**

4.4 million data andanalytics jobs is needed globally. only one-third of those jobs will be filled.

4.4MILLION data scientists needed by 2015

ONLY 1/3 FILLED

DATA SCIENTIST

○ **Privacy, security and governance**

While big data can provide significant value, it also presents significant risk.

○ **Create new business models with big data**

Data-driven marketing and ad targeting and something new

Agenda

○ Big Data

○ Hadoop Brief Introduction

○ HDFS Structure and Characteristic

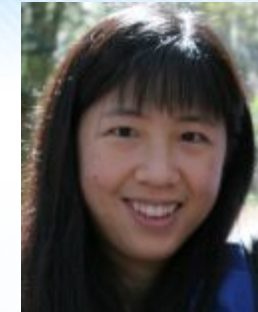○ MapReduce Model

○ Q&A

Three Papers from Google

December, 2004
Jeffrey Dean
Sanjay Ghemawat

MapReduce
Simplified Data Processing on Large Clusters

BigTable
A Distributed Storage System for Structured Data

November, 2006
Fay W. Chang
Jeffrey Dean
Sanjay Ghemawat
Wilson C. Hsie
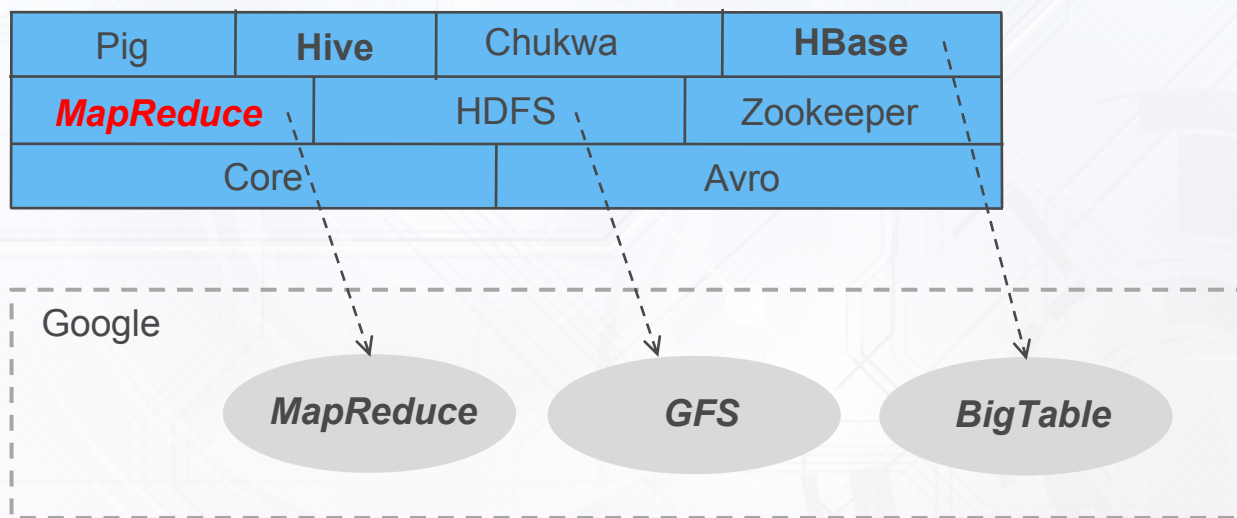etc.

The Google File System

October, 2003
Sanjay Ghemawat
Howard Gobioff
Shun-Tak Leung

Hadoop Brief Introduction

○ The Apache Hadoop project develops *open-source* software for reliable, *scalable*, *distributed computing*. Implemented in Java.

○ The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to *thousands of machines*, each offering local computation and storage. *Rather than rely on hardware to deliver high-availability*, the library itself is designed to *detect and handle failures at the application layer*, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

Pig	**Hive**	Chukwa	**HBase**
MapReduce	HDFS		Zookeeper
Core		Avro	

Google

MapReduce   *GFS*   *BigTable*

Doug Cutting

The Companies around Hadoop

The Companies around Hadoop

○ **Yahoo:**

More than 100,000 CPUs in more than 25,000 computers running it.

The biggest cluster: 4000 nodes,2 x 4CPU boxes, with 4 x 1 TB disk, and 16 GB RAM.

○ **Amazon :**

Process millions of sessions daily for analytics.

Using both Java and streaming APIs.

○ **Facebook:**

Use it to store copies of internal log and dimension data sources.

a source for reporting and analytics, with machine learning algorithms.

○ **BaiDu**

~10K cluster size

○ **Sina**

>1K cluster size

● ● ● ● ● ●

Agenda

○ Big Data

○ Hadoop Brief Introduction

○ HDFS Structure and Characteristic

○ MapReduce Model

○ Q&A

HDFS Brief Introduction

○ *Hadoop Distributed File System* (HDFS) is the primary 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.

○ For huge data (PB)

○ Large scale distributed file system

○ Cheap PC servers

○ Based on file system of OS

○ For rarely modified data

○ Reliable

HDFS Brief Introduction

○ **Master-Slave architecture**

Master: Name node (1, single-point failure)

Slave: Data node (N)   SecondaryNameNode(1)

○ **Distributed**

Every long file is split into blocks of 64MB, each block is allocated
to different storage node.

○ **Checkpoint:**
fsimage + edit log

○ **Reliability**

Multiple replicas for each block
(default: 3)

○ **Scalable**

Add disks/nodes

HDFS Architecture

Metadata ops

Namenode

Metadata (Name, replicas, …):
/home/foo/data, 3, …

Client

Read   Datanodes

Block ops

Datanodes

Replication

Blocks

Rack 1   Write   Rack 2

Client
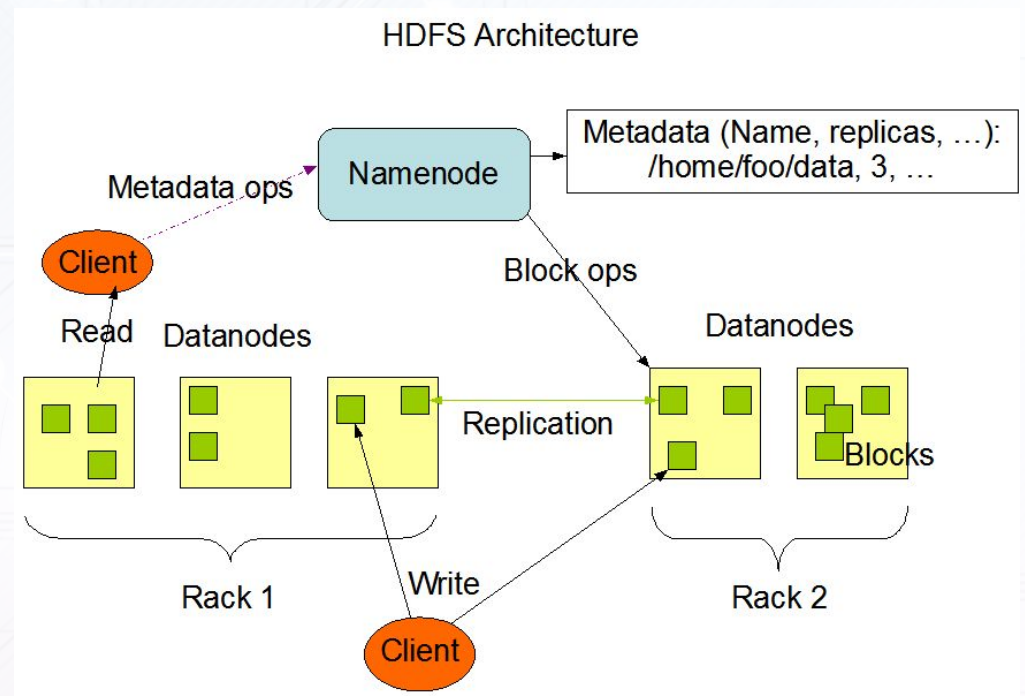
Upload file (140MB, /usr/file1)

Splitting:

A (64MB)	B (64MB)	C (12MB)

HDFS writes the replica into different nodes in different racks.
Enhance the reliability, but reduce the write speed.

Replication:

Rack 1

DataNode 1	DataNode 2	DataNode 3
A B	B' C	A' C'

Rack 2

DataNode 4	DataNode 5	DataNode 6
A'' B''	C''	
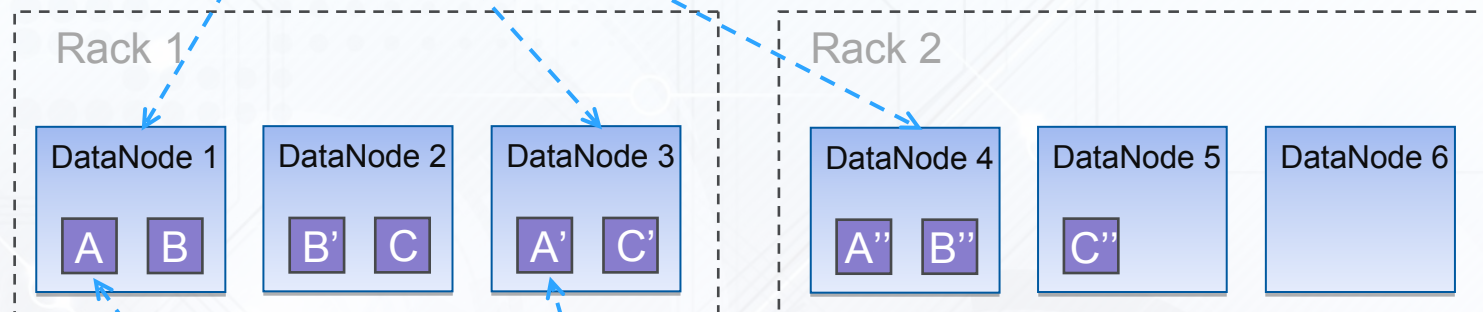
Consistency:

Read the nearest replica

NameNode

/usr/file1

A	B	C
Rack1-DataNode1-A	Rack1-DataNode1-B	Rack1-DataNode2-C
Rack1-DataNode3-A'	Rack1-DataNode2-B'	Rack1-DataNode3-C'
Rack2-DataNode4-A''	Rack2-DataNode4-B''	Rack2-DataNode5-C''
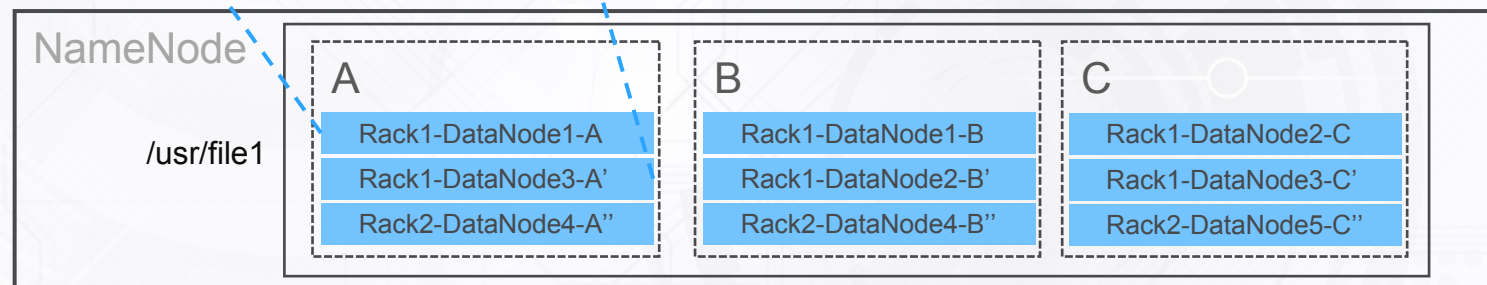
HDFS create/write

FSDataOutputStream is returned by DistributedFileSystem after contacting NameNode.

Upload file is split into multiple packets. Only after the *first* block has been stored into all the replica, the *second* block begins to upload.
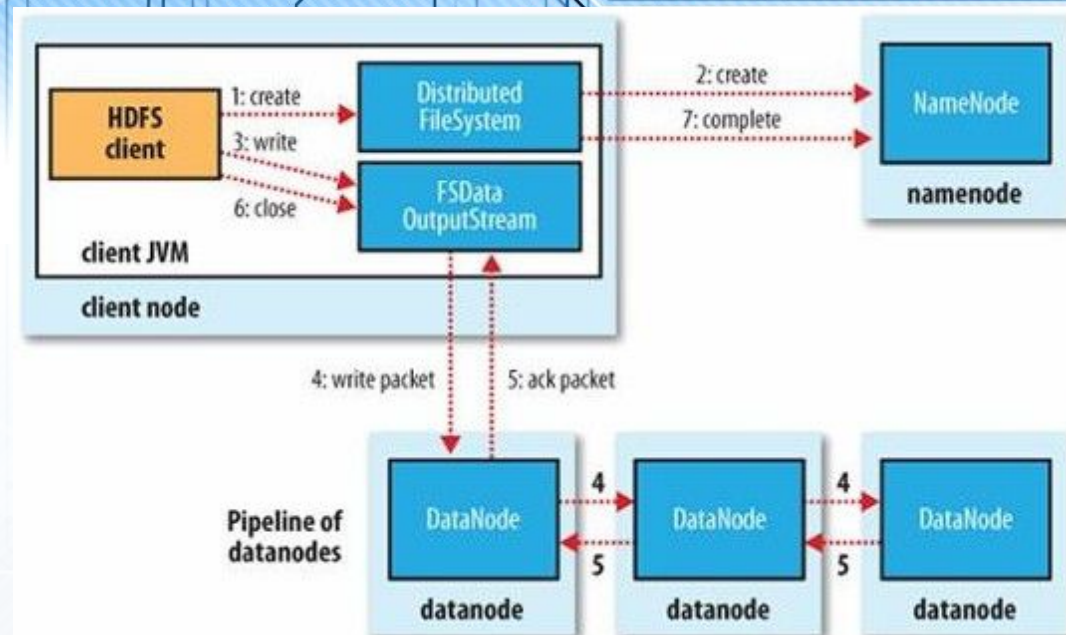
HDFS read

FSDataOutputStream is returned by DistributedFileSystem after contacting NameNode to find the block address.

FSDataOutputStream read the block from the *nearest* datanode, if fails it goes to the replica.
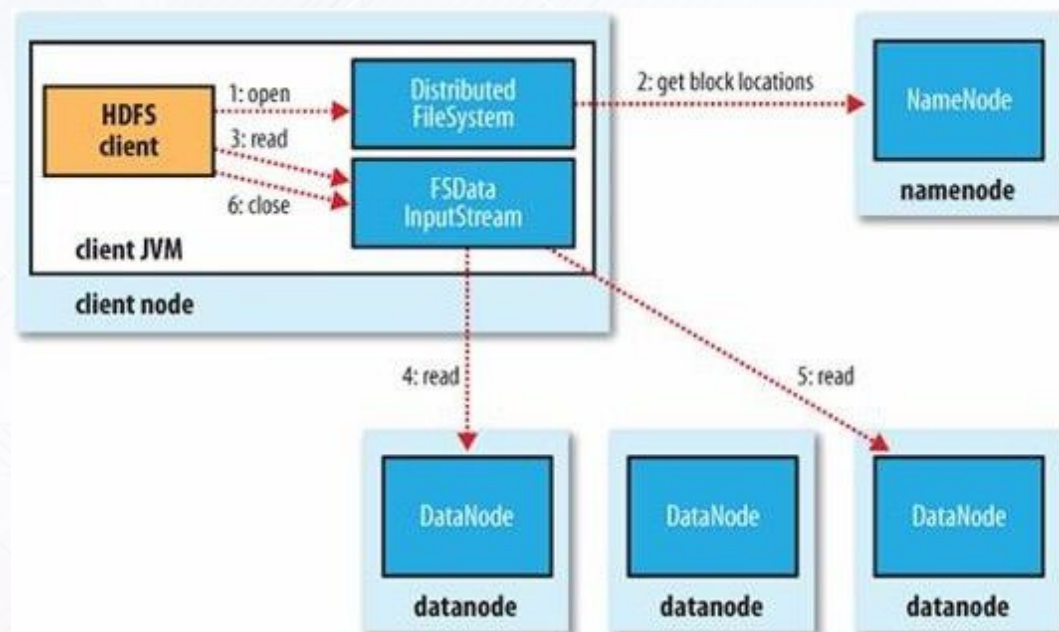
Agenda

○ Big Data

○ Hadoop Brief Introduction

○ HDFS Structure and Characteristic

○ MapReduce Model

○ Q&A

MapReduce Model

/usr/file1

```
hello world
hi how are you
```

/usr/file2

```
hello suzhou
you are the best
```

......

/usr/file3

```
hello guys
suzhou is the
best
```

MapReduce

/usr/wc

are	2
best	2
guys	1
hello	3
hi	1
how	1
is	1
suzhou	2
the	2
world	1
you	2
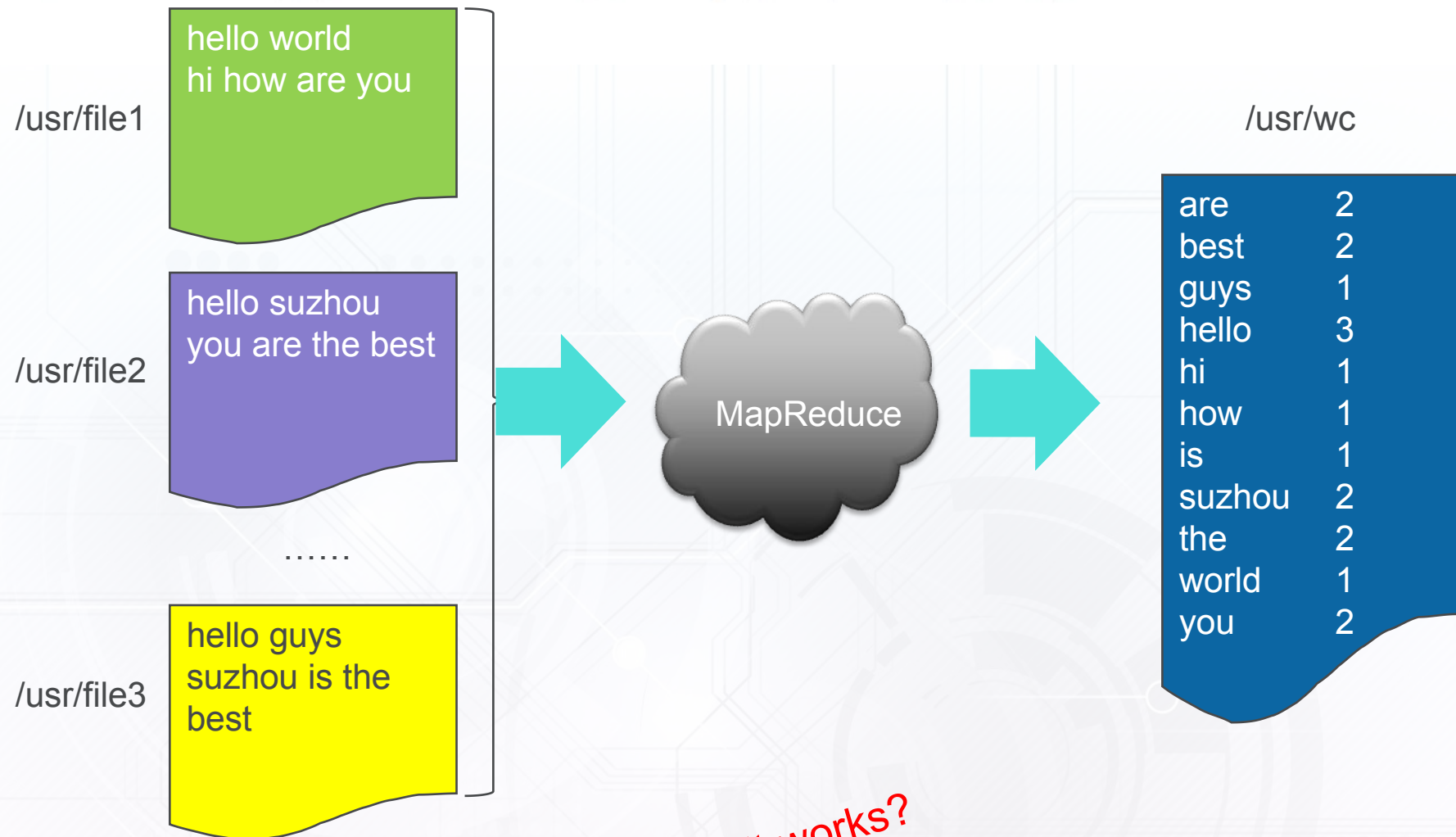
How it works?

MapReduce Model

○ Two major functions – map and reduce

○ Map function

Written by user

Takes input pairs and produce a set of intermediate key/value pairs

Intermediate key/value pairs are grouped and passed to reduce function

○ Reduce function
Accepts an intermediate key and a set of values for the key
Merges together these values

Major components of Hadoop MapReduce 1.x

- Job Tracker

  Cluster manager

  Job manager

  Job scheduler

- Task tracker

  Task JVM spawner

  Monitor

  Shuffle server

Client

(1) Create splits

(2) Submit job

Job Tracker

(3) Dispatch map tasks

(3) Dispatch reduce tasks

output

Input split 1 — (4) read → Map task 1

P1: k1, k3
P2: k2, k4

(7) shuffle

Reduce task 1 — (9) → k1,k3,k5,k7

(6) generate intermediate data

Input split 2 → Map task 2

P1: k1, k5
P2: k4, k6

Input split 3 → Map task 3

P1: k1, k3, k7
P2: k2, k6

Reduce task 2 → k2,k4,k6,k8

(8) call reduce() of user

Input split 4 → Map task 4

P1: k3, k5
P2: k2, k8

(5) call map() of user

The partition number
Is the number of
reduce tasks

/usr/file1

hello world
hi how are you

→ <0,hello world>
<12, Hi how are you>

→ <hello, 1>
<world, 1>
<hi,1>
<how,1>
<are,1>
<you,1>

sort

/usr/file2

hello suzhou
you are the best

→ <0,hello suzhou>
<13, you are the best>

MAP →

<hello, 1>
<suzhou, 1>
<you,1>
<are,1>
<the,1>
<best,1>

sort

/usr/file3

hello guys
suzhou is the best

→ <0,hello guys>
<11, suzhou is the best>

→ <hello, 1>
<guys, 1>
<suzhou,1>
<is,1>
<the,1>
<best,1>

sort

Shuffle

/usr/wc

are	2
best	2
guys	1
hello	3
hi	1
how	1
is	1
suzhou	2
the	2
world	1
you	2

REDUCE ←

<are, **list<1,1>** >
<best, **list<1,1>** >
<guys, **list<1>** >
<Hello, **list<1,1,1>** >
<hi, **list<1>** >
<how, **list<1>** >
<is, **list<1>** >
<suzhou, **list<1,1>**
<the, **list<1,1>** >
<world, **list<1>** >
<you, **list<1,1>**
>

Q&A

参考资料

http://hadoop.apache.org