

特此声明

本视频由whoami博主（博客：itweet.cn）
Hadoop十周年庆开始公益录制，内容涉及
《Enterprise_Hadoop_Solutions》，声明此
教程使用图片大部分来自网络，版权属于广大
网友；任何个人或企业未经授权不得作商业用
途。想要持续关注更新或参与讨论请加入QQ
群：457710636！

Apache Hadoop前世今生



新浪微博 @Mr-Robot1992

作者: whoami



走向分布式

一个系统走向分布式，一定有其不得不为的理由。可扩展性是最常见的理由之一。我先简单的将“可伸缩”的需求分成两种：

- 1、 **Data Scalability:** 单台机器的容量不足以 (经济的) 承载所有资料，所以需要分散。如： NoSQL
- 2、 **Computing Scalability:** 单台机器的运算能力不足以 (经济的) 及时完成运算所以需要分散。如： 科学运算。不管是哪一种需求，在决定采用分布式架构时，就几乎注定要接受一些牺牲：
 - (1). 牺牲效率：网路延迟与节点间的协调，都会降低执行效率。
 - (2). 牺牲 AP 弹性：有些在单机上能执行的运算，无法轻易在分布式环境中完成。
 - (3). 牺牲维护维运能力：分散式架构的问题常常很难重现，也很难追踪.另外，跟单机系统一样，也有一些系统设计上的 tradeoffs(权衡)
 - (4). CPU 使用效率优化或是 IO 效率优化
 - (5). 读取优化或是写入优化
 - (6). 吞吐率优化或是 网络延迟优化
 - (7). 资料一致性或是资料可得性,选择了不同的 tradeoff，就会有不同的系统架构。

课纲

1、Hadoop生态概述

2、分布式存储HDFS

3、分布式计算MR

4、Why Hadoop2.0 ?

5、Hadoop2.0 介绍

6、Hadoop几大发行商

Hadoop生态圈 | 特点

源代码开源（免费）

社区活跃、参与者众多

涉及分布式存储和计算的方方面面

已得到企业界验证

Hadoop是什么？

- 分布式存储

- 分布式计算



- 弹性伸缩

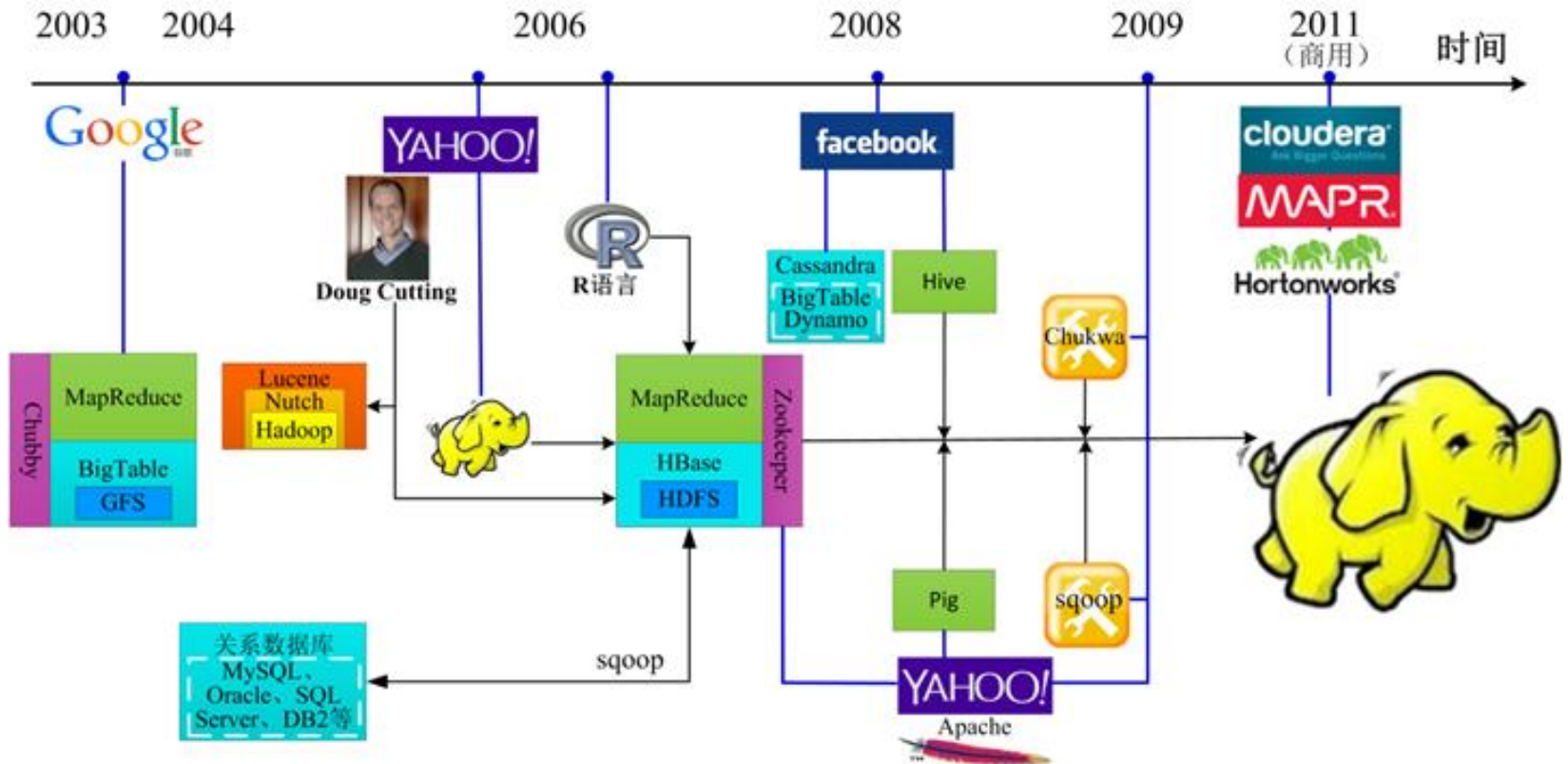
- 海量数据分析



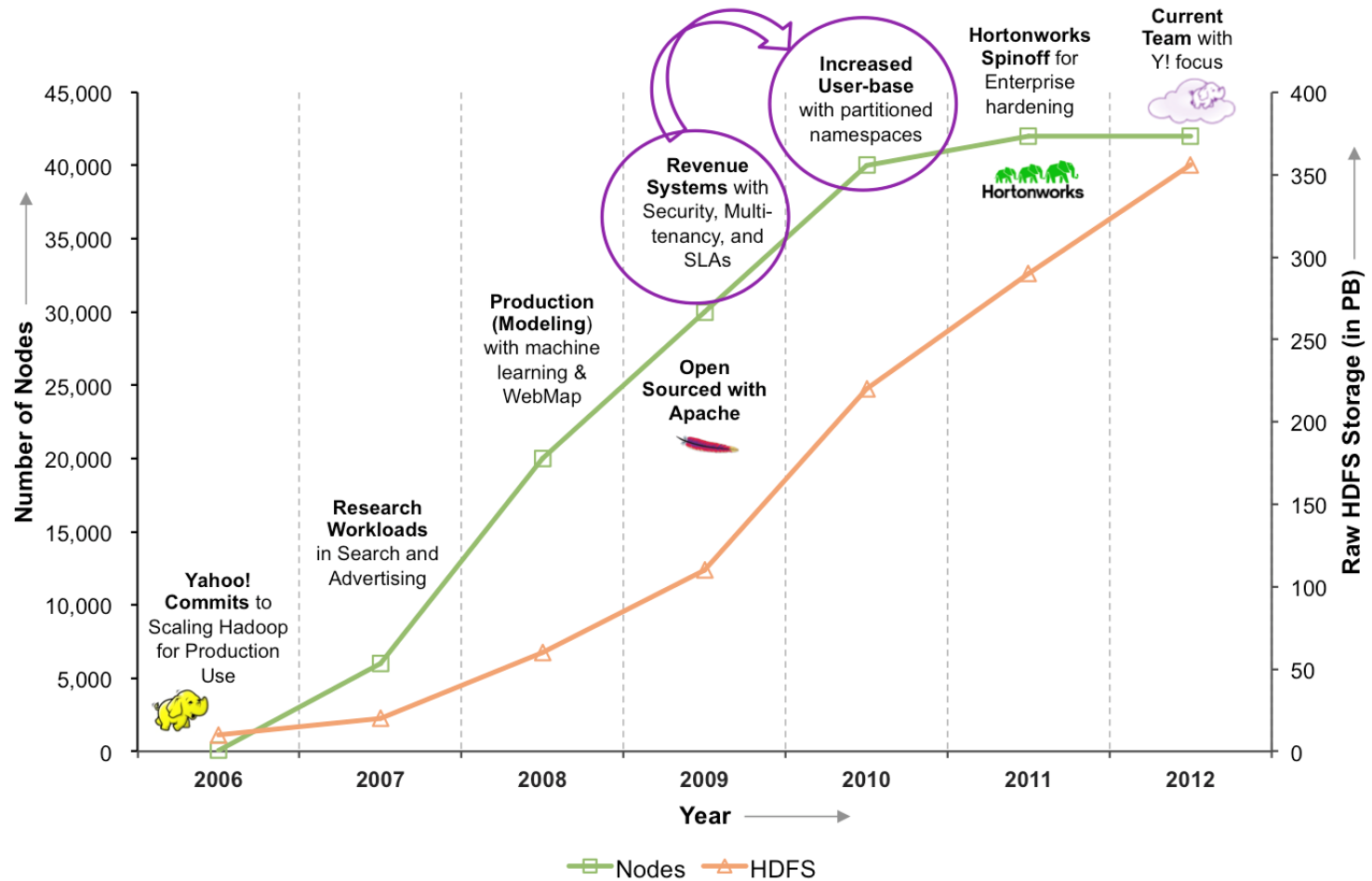
Hadoop version

- 2004年-- 最初的版本(现在称为HDFS和MapReduce)由Doug Cutting和Mike Cafarella开始实施。
- 2005年12月-- Nutch移植到新的框架，Hadoop在20个节点上稳定运行。 2006年1月-- Doug Cutting加入雅虎。
- 2006年2月-- Apache Hadoop项目正式启动以支持MapReduce和HDFS的独立发展。
- 2006年2月-- 雅虎的网格计算团队采用Hadoop。
- 2006年4月-- 标准排序(10 GB每个节点)在188个节点上运行47.9个小时。
- 2006年5月-- 雅虎建立了一个300个节点的Hadoop研究集群。
- 2006年5月-- 标准排序在500个节点上运行42个小时(硬件配置比4月的更好)。
- 2006年11月-- 研究集群增加到600个节点。
- 2006年12月-- 标准排序在20个节点上运行1.8个小时，100个节点3.3小时，500个节点5.2小时，900个节点7.8个小时。
- 2007年1月-- 研究集群到达900个节点。 07年4月-- 研究集群达到两个1000个节点的集群。
- 2008年4月-- 赢得世界最快1 TB数据排序在900个节点上用时209秒。
- 2008年10月-- 研究集群每天装载10 TB的数据。
- 2009年3月-- 17个集群总共24 000台机器。
- 2009年4月-- 赢得每分钟排序，59秒内排序500 GB(在1400个节点上)和173分钟内排序100 TB数据(在3400个节点上)。
- 2011年12月27日--1.0.0版本释出。标志着Hadoop已经初具生产规模。
- 2012年12月23日 --Apache Hadoop版本分为两代，我们将第一代Hadoop称为Hadoop 1.0，第二代Hadoop称为Hadoop 2.0。
- 2013年10月13日-- hadoop-2.2。High Availability for HDFS，HDFS Federation
- 2014-02-20 2.3 --HDFS in-memory caching，异构存储媒介的支持。
- 2014-04-07--ResourceMnager HA
-

Hadoop version



Multi-tenant Apache Hadoop at Yahoo!



Footprint: 42,000 nodes, 365PB HDFS, 10M daily slot hours | **Usage (Apr'13):** 411 projects, 805 users, 13.2M jobs

课纲

1、Hadoop生态概述

2、分布式存储HDFS

3、分布式计算MR

4、Why Hadoop2.0 ?

5、Hadoop2.0 介绍

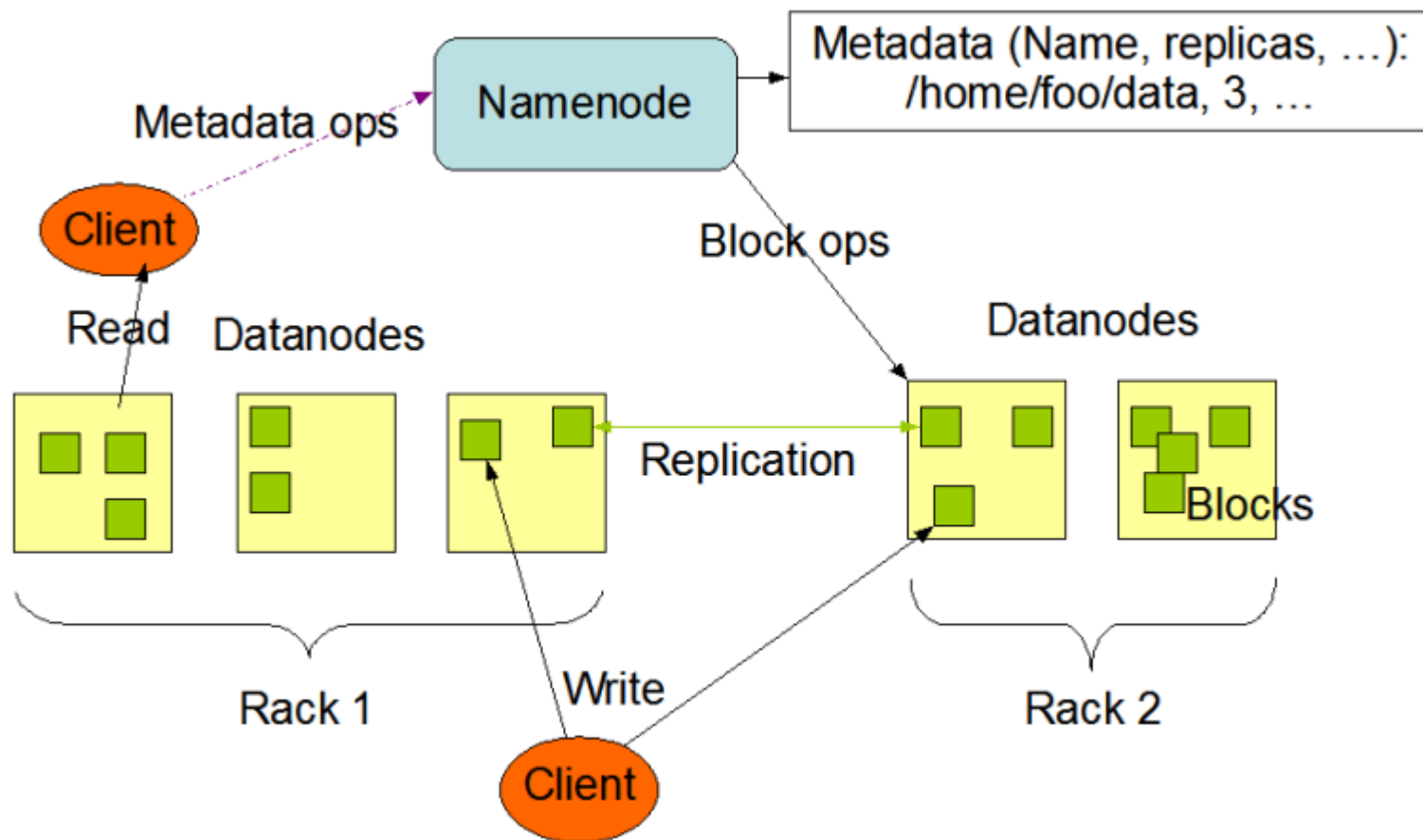
6、Hadoop几大发行商

分布式存储

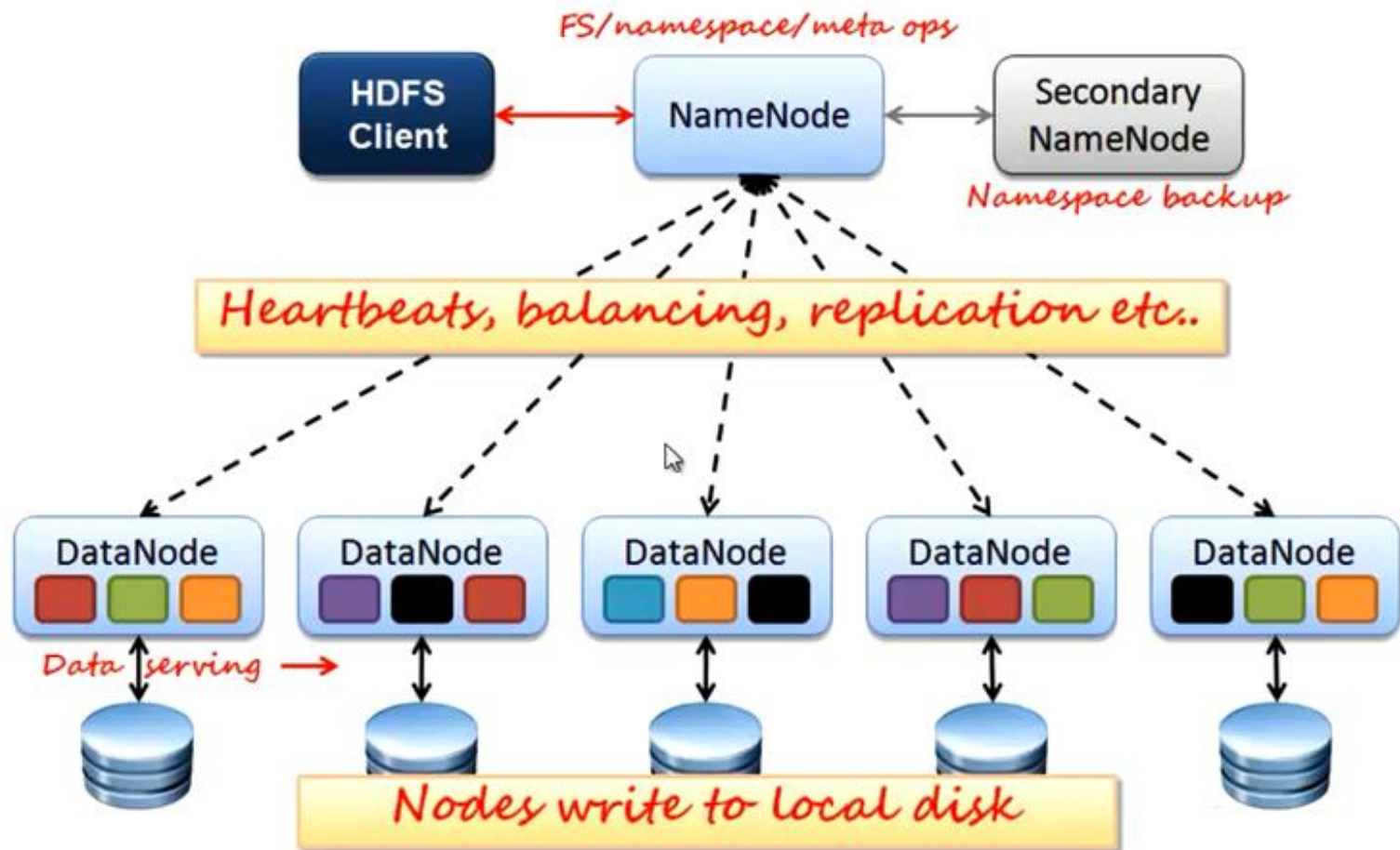
- HDFS (Hadoop Distributed File System)
- 高容错性
- 高吞吐量
- 部署在低廉的 (low-cost) 硬件上
- 集群 (master-slave)

分布式存储

v1.0+



分布式存储

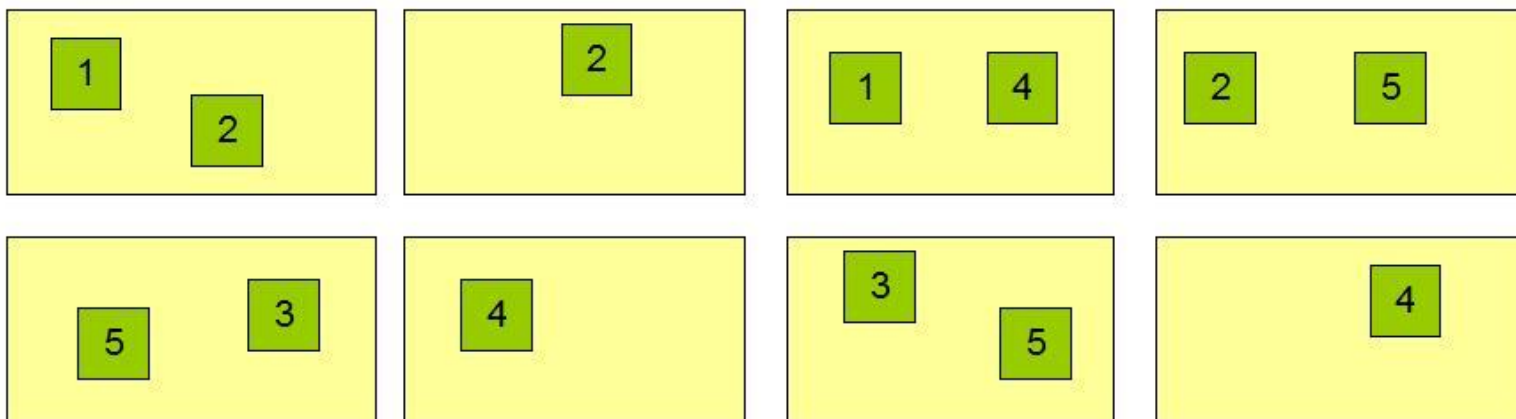


分布式存储 | 策略

Block Replication

Namenode (Filename, numReplicas, block-ids, ...)
/users/sameerp/data/part-0, r:2, {1,3}, ...
/users/sameerp/data/part-1, r:3, {2,4,5}, ...

Datanodes



分布式存储 | command

文件副本,default: 3

```
$ hadoop fs -ls /tmp/logs/hsu/logs/application_1435053267354_0233
Found 9 items
-rw-r----- 3 hsu hadoop 12517634718 2015-06-27 03:13 /tmp/logs/hsu/logs/application_1435053267354_0233/server01_8041
-rw-r----- 3 hsu hadoop 12596068334 2015-06-27 03:13 /tmp/logs/hsu/logs/application_1435053267354_0233/server02_8041
```

文件block所在位置

```
$ hadoop fsck /tmp/logs/hsu/logs/application_1435053267354_0233/server01_8041 -files -blocks -locations
93. BP-1361292518-135.33.5.53-1404706202963:blk_1077544742_3813190 len=35386014 repl=3 [192.168.1.3:50010, 192.168.1.2:50010, 192.168.1.1:50010]
Status: HEALTHY
Total size: 12517634718 B
Total dirs: 0
Total files: 1
Total symlinks: 0
Total blocks (validated): 94 (avg. block size 133166326 B)
Minimally replicated blocks: 94 (100.0 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 0 (0.0 %)
Mis-replicated blocks: 0 (0.0 %)
Default replication factor: 3
Average block replication: 3.0
Corrupt blocks: 0
Missing replicas: 0 (0.0 %)
Number of data-nodes: 9
Number of racks: 1
FSCK ended at Sat Jun 27 11:34:43 CST 2015 in 3 milliseconds
```

分布式存储 | HDFS command

```
$ hadoop fs
Usage: hadoop fs [generic options]
    [-appendToFile <localsrc> ... <dst>]
    [-cat [-ignoreCrc] <src> ...]
    [-checksum <src> ...]
    [-chgrp [-R] <GROUP> PATH...]
    [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
    [-chown [-R] [OWNER][[:<GROUP>]] PATH...]
    [-copyFromLocal [-f] [-p] <localsrc> ... <dst>]
    [-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
    [-count [-q] <path> ...]
    [-cp [-f] [-p | -p[topax]] <src> ... <dst>]
    [-createSnapshot <snapshotDir> [<snapshotName>]]
    [-deleteSnapshot <snapshotDir> <snapshotName>]
    [-df [-h] [<path> ...]]
    [-du [-s] [-h] <path> ...]
    [-expunge]
    [-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
    [-getfacl [-R] <path>]
    [-getfattr [-R] {-n name | -d} [-e en] <path>]
    [-getmerge [-nl] <src> <localdst>]
    [-help [cmd ...]]
    [-ls [-d] [-h] [-R] [<path> ...]]
```

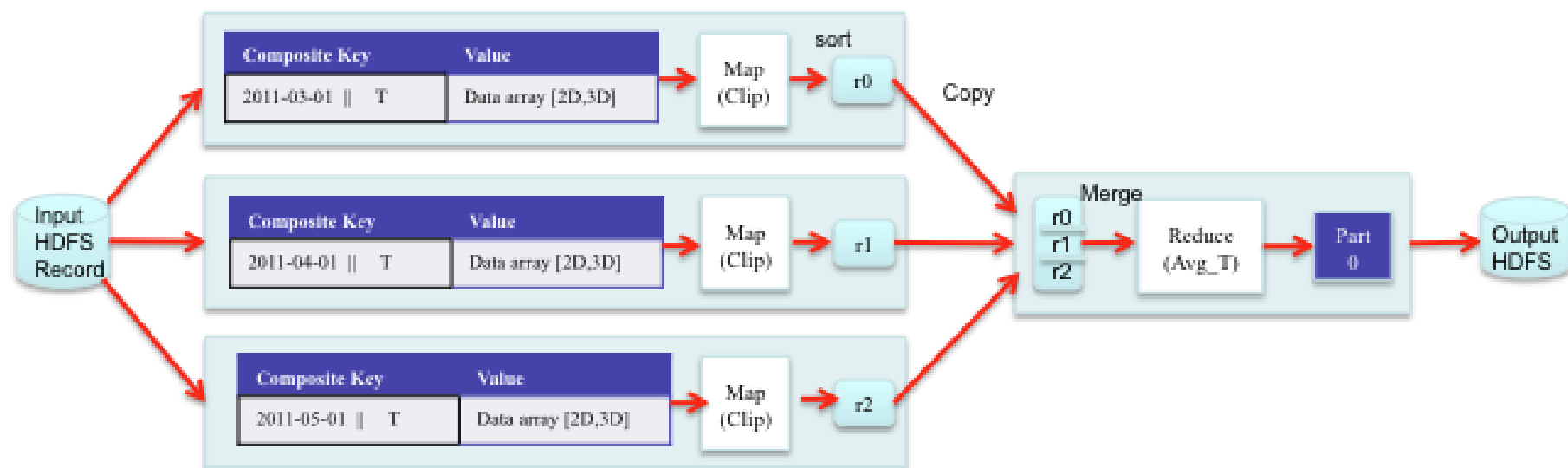
```
$ hadoop fs -ls /
Found 7 items
drwxr-xr-x - hsu      supergroup      0 2015-05-26 14:17 /data
drwxr-xr-x - hbase    hbase          0 2015-01-08 10:43 /hbase
drwxr-xr-x - hadoop   supergroup      0 2015-02-11 16:26 /home
drwxr-xr-x - hdfs     supergroup      0 2015-02-13 15:18 /system
drwxr-xr-x - hsu      supergroup      0 2015-05-28 18:02 /tachyon
drwxrwxrwt - hdfs     supergroup      0 2015-06-27 11:22 /tmp
drwxr-xr-x - hdfs     supergroup      0 2015-05-20 17:50 /user
```

<input type="checkbox"/>		名称
<input type="checkbox"/>		.
<input type="checkbox"/>		data
<input type="checkbox"/>		hbase
<input type="checkbox"/>		home
<input type="checkbox"/>		system
<input type="checkbox"/>		tachyon
<input type="checkbox"/>		tmp
<input type="checkbox"/>		user

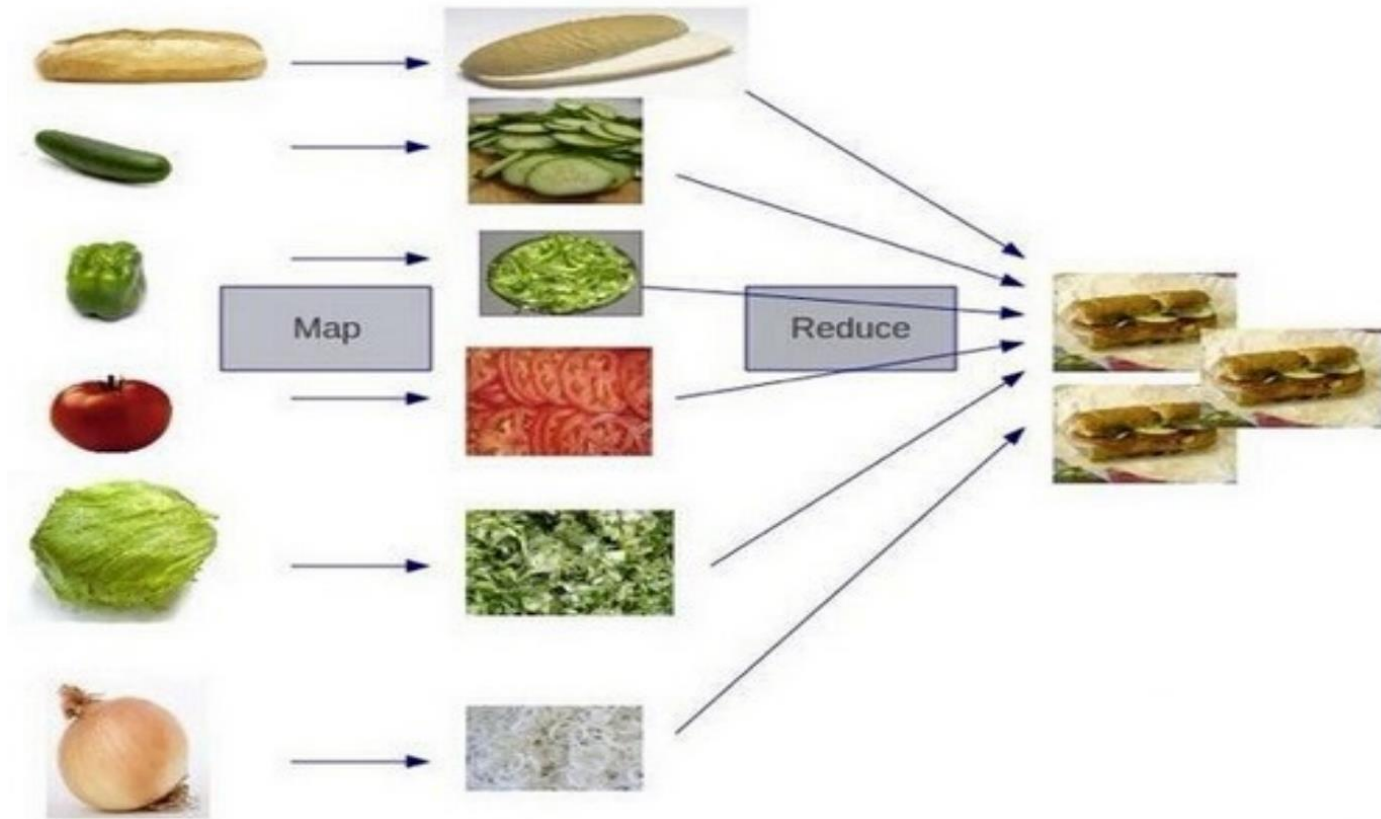
课纲

- 1、Hadoop生态概述
- 2、分布式存储HDFS
- 3、分布式计算MR
- 4、Why Hadoop2.0 ?
- 5、Hadoop2.0 介绍
- 6、Hadoop几大发行商

分布式计算



Map and Reduce



MapReduce | 栗子wordcount

```
public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable> {

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

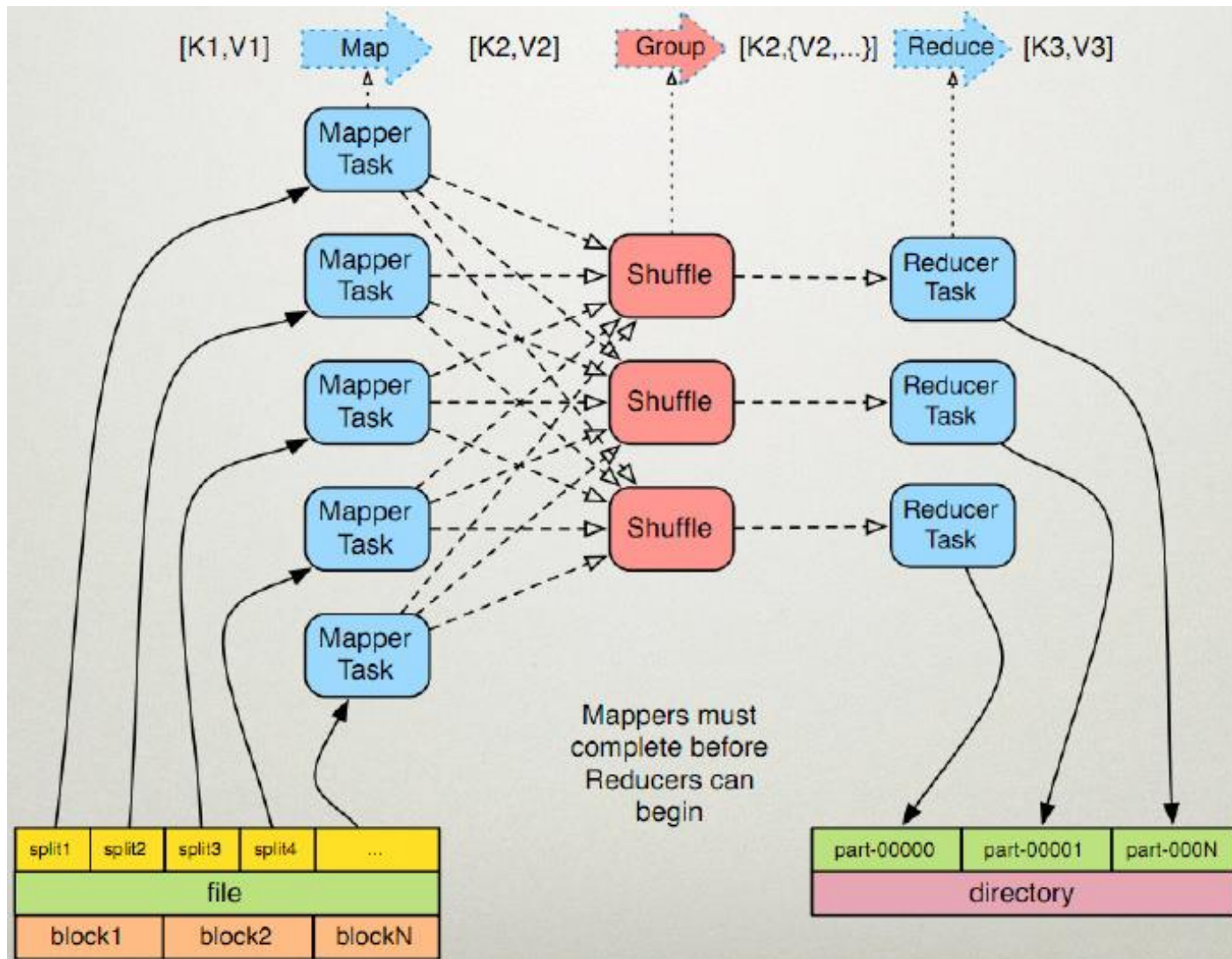
    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
        StringTokenizer itr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            context.write(word, one);
        }
    }
}

public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();

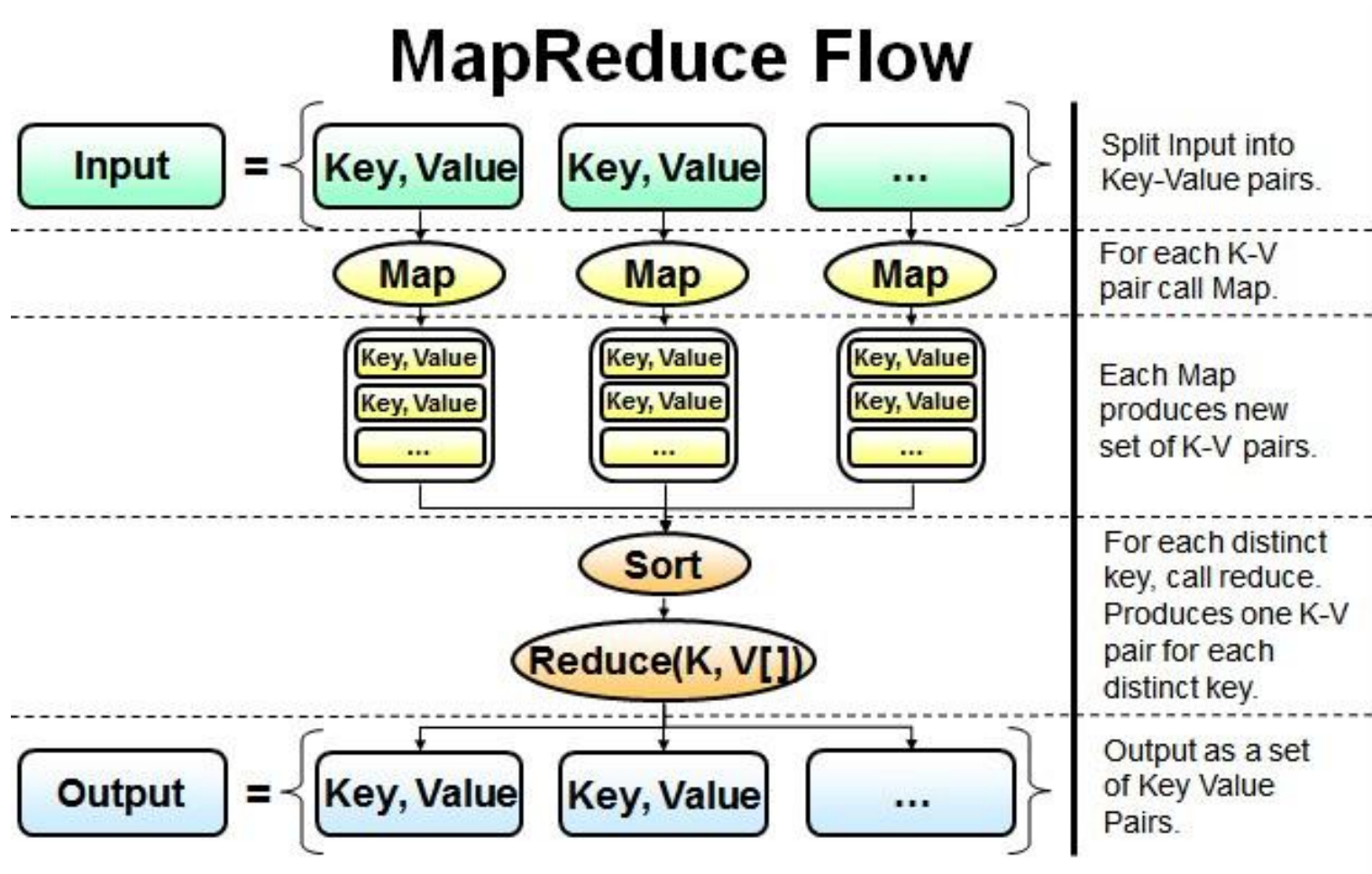
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}

public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = new GenericOptionsParser(conf, args)
        .getRemainingArgs();
    if (otherArgs.length != 2) {
        System.err.println("Usage: wordcount <in> <out>");
        System.exit(2);
    }
    Job job = new Job(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
```

MapReduce | wordcount 全局图

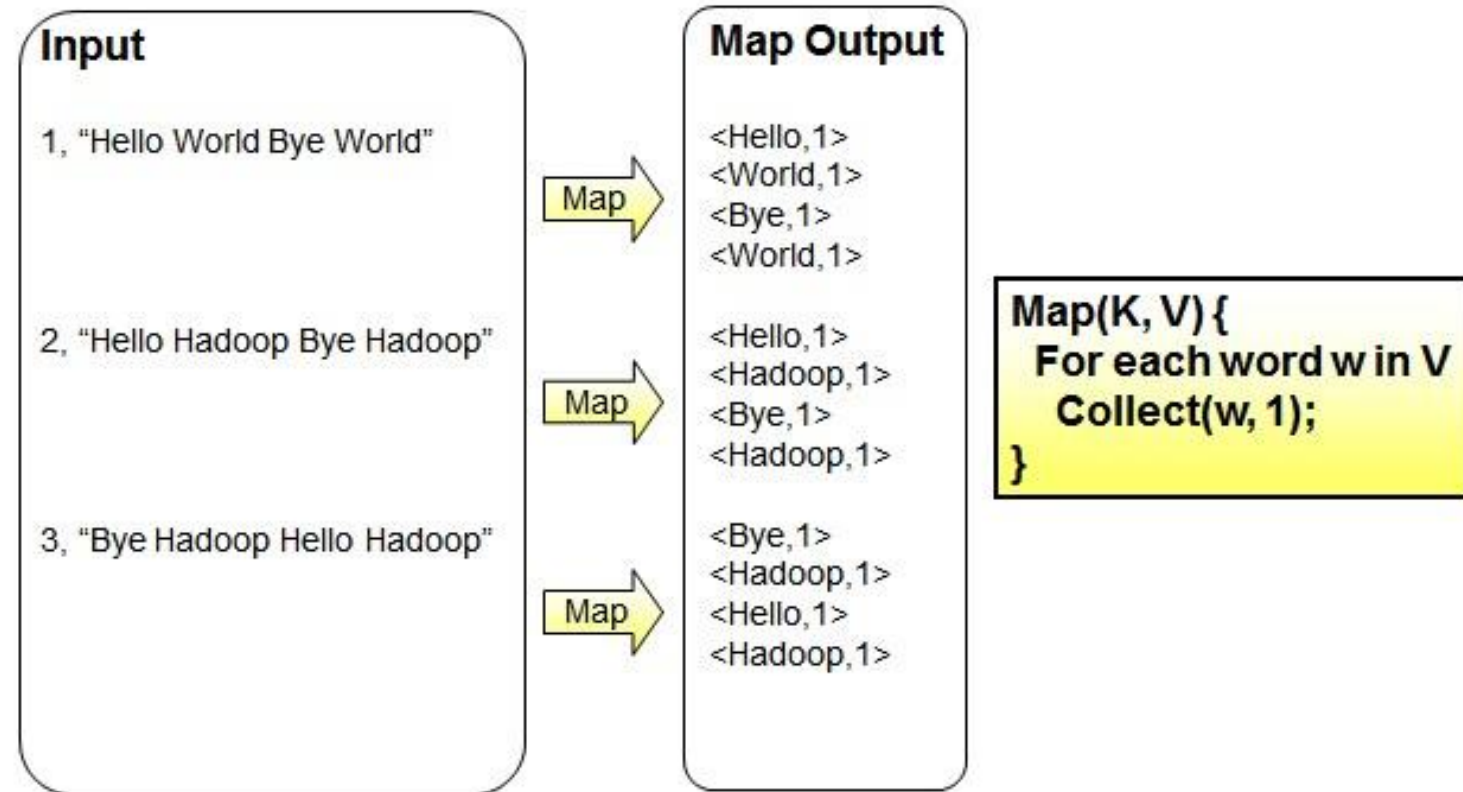


MapReduce | wordcount



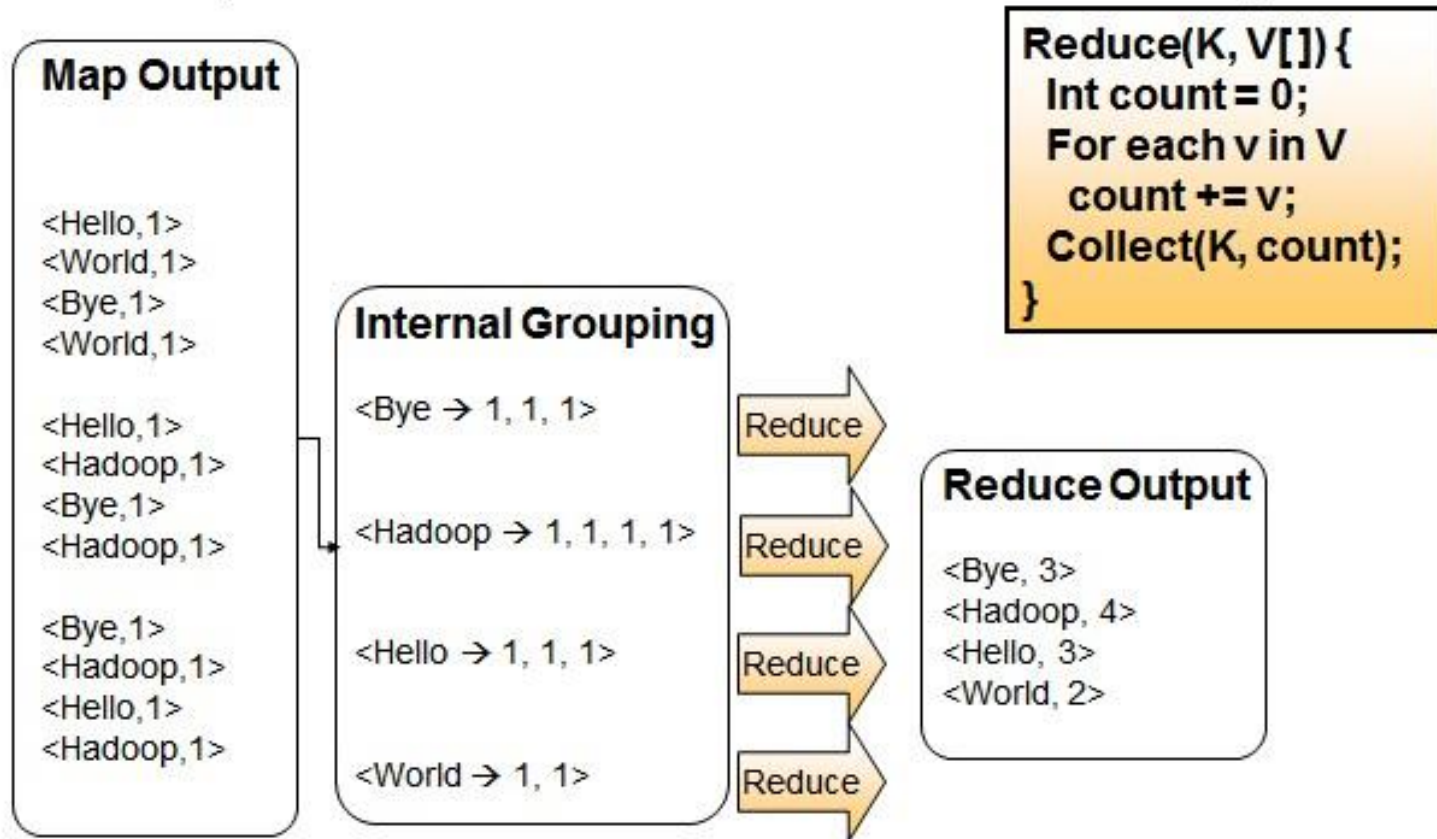
MapReduce | Map

MapReduce WordCount Example

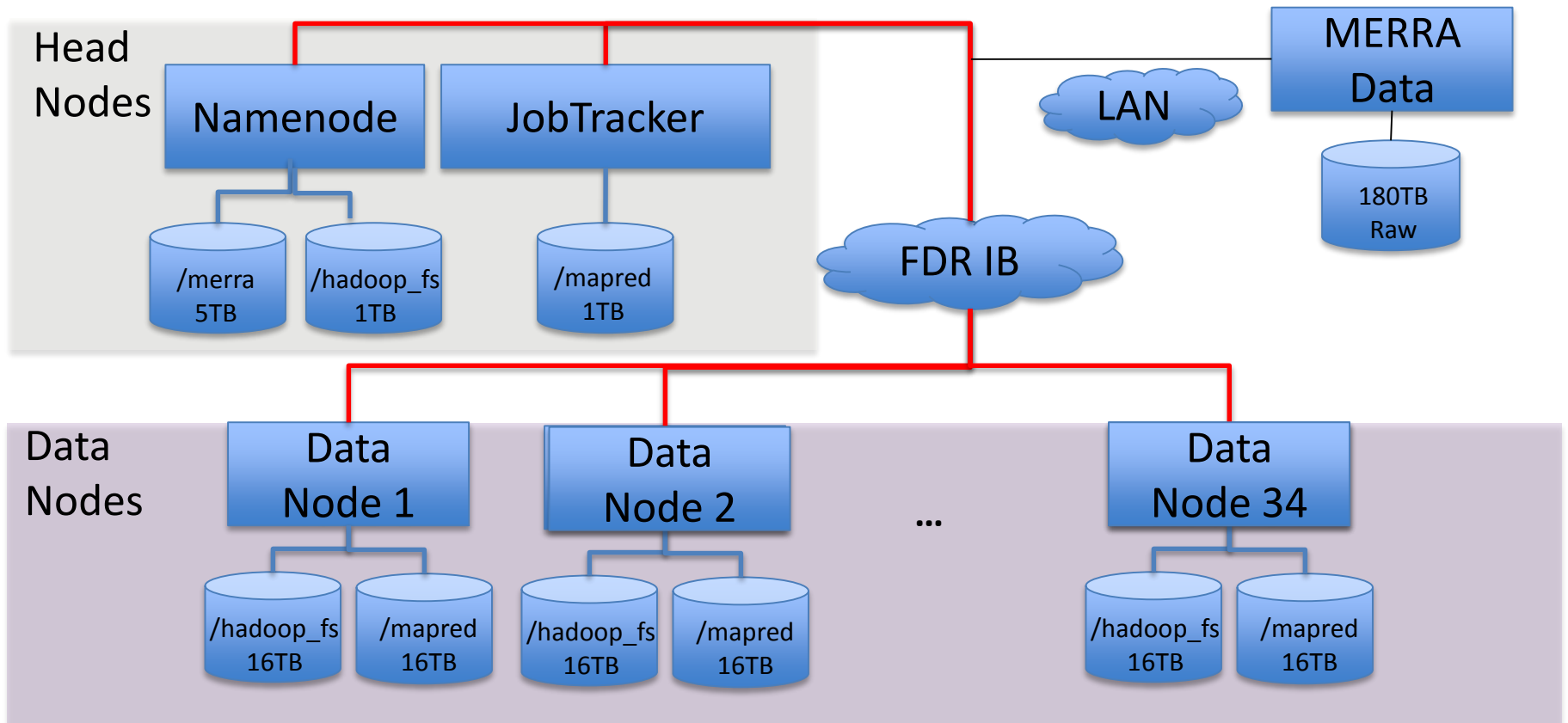


MapReduce | Reduce

MapReduce WordCount Example



Hadoop Cluster 1.0



HDFS: Namenode, datanode, SecondaryNamenode

MapReducer: JobTracker, taskTracker

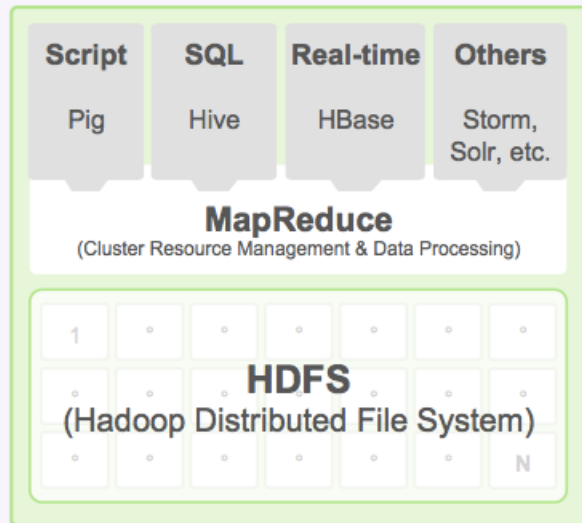
课纲

- 1、Hadoop生态概述
- 2、分布式存储HDFS
- 3、分布式计算MR
- 4、Why Hadoop2.0 ?
- 5、Hadoop2.0 介绍
- 6、Hadoop几大发行商

Why?

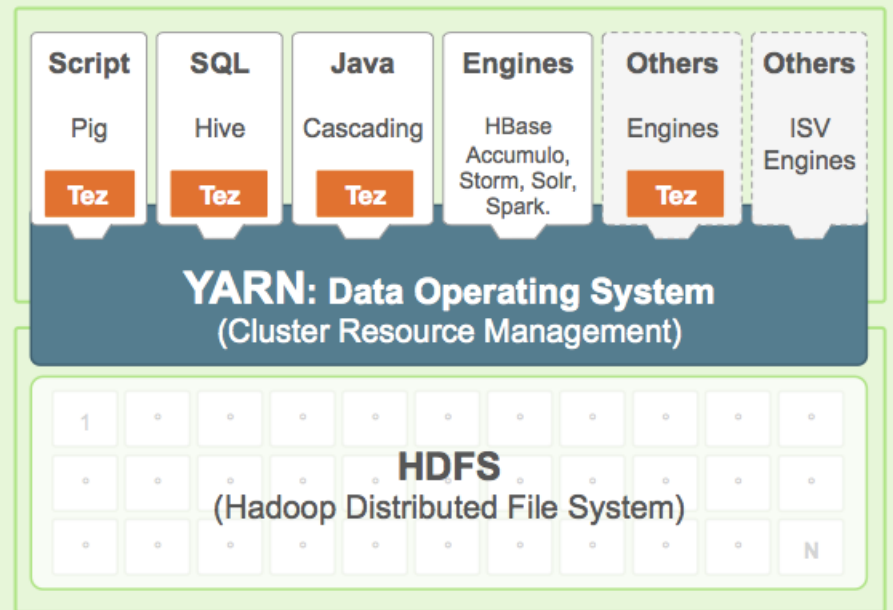
Hadoop 1

- Silos & Largely batch
- Single Processing engine



Hadoop 2 w/ Tez

- Multiple Engines, Single Data Set
- Batch, Interactive & Real-Time

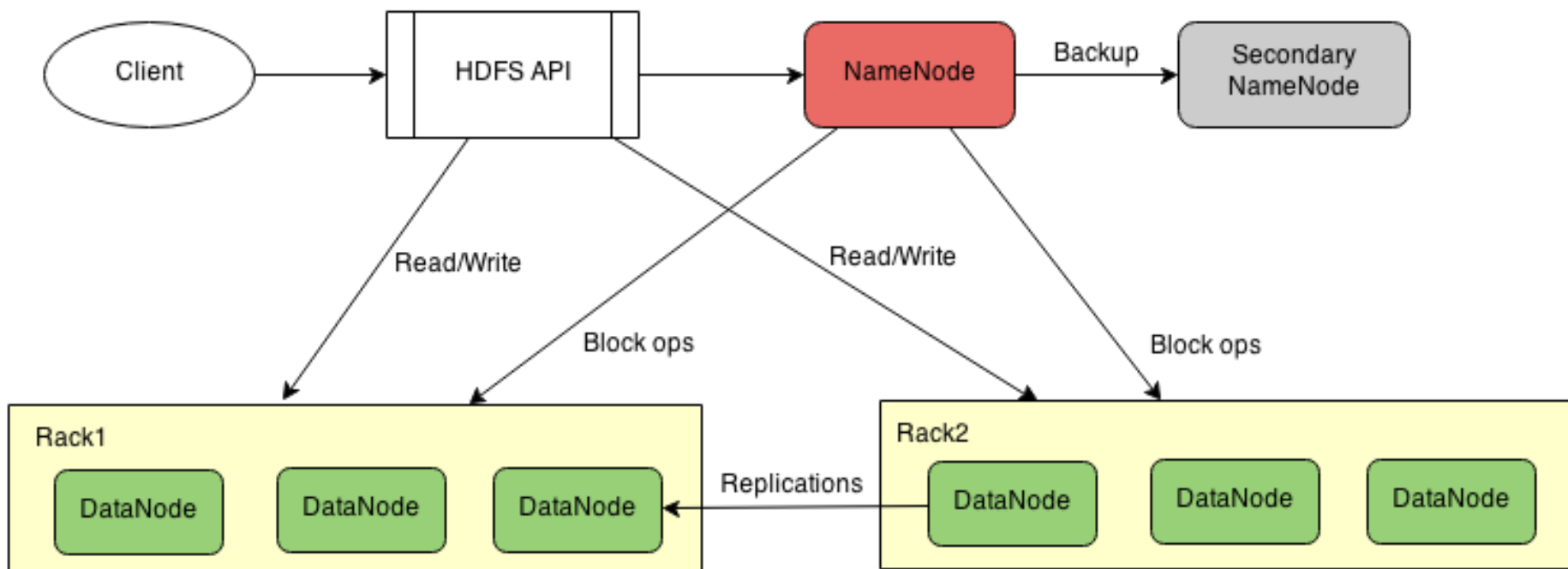


课纲

- 1、Hadoop生态概述
- 2、分布式存储HDFS
- 3、分布式计算MR
- 4、Why Hadoop2.0 ?
- 5、Hadoop2.0 介绍
- 6、Hadoop几大发行商

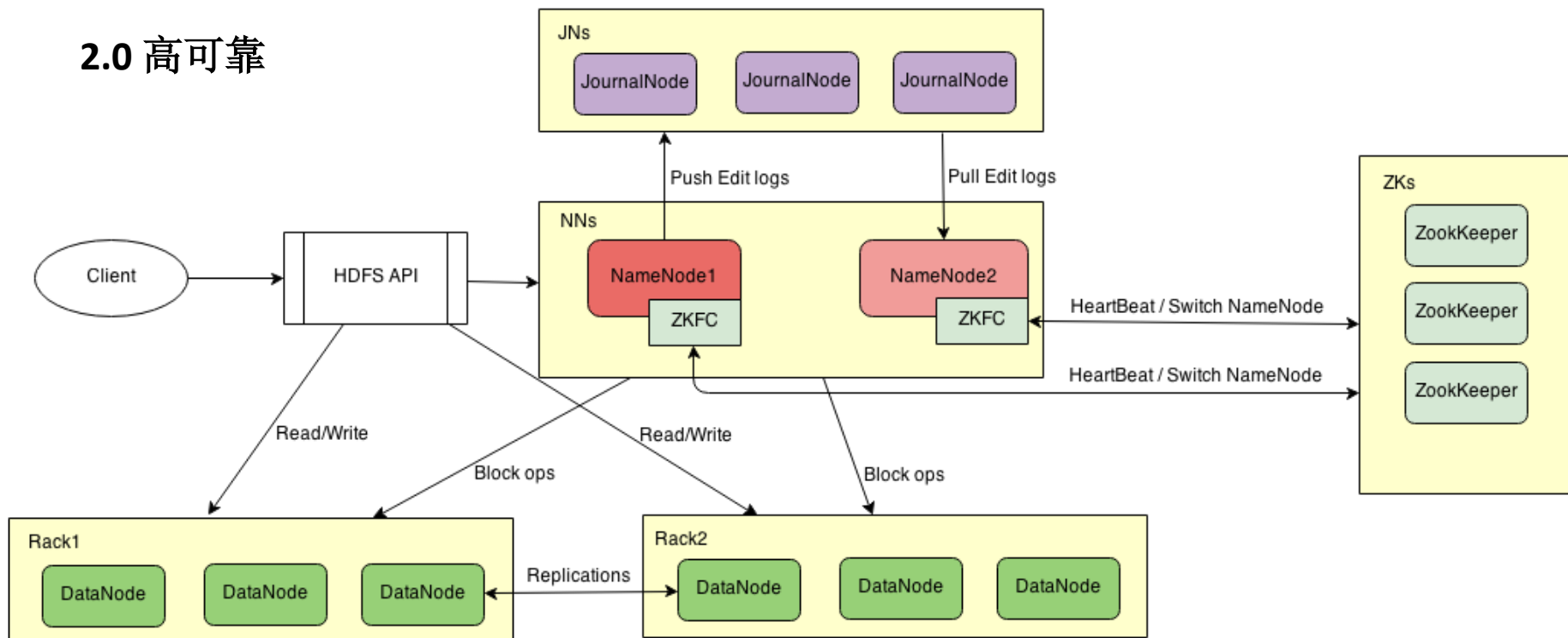
分布式存储 HA

1.0 冷备



分布式存储 HA

2.0 高可靠



Hadoop2.0

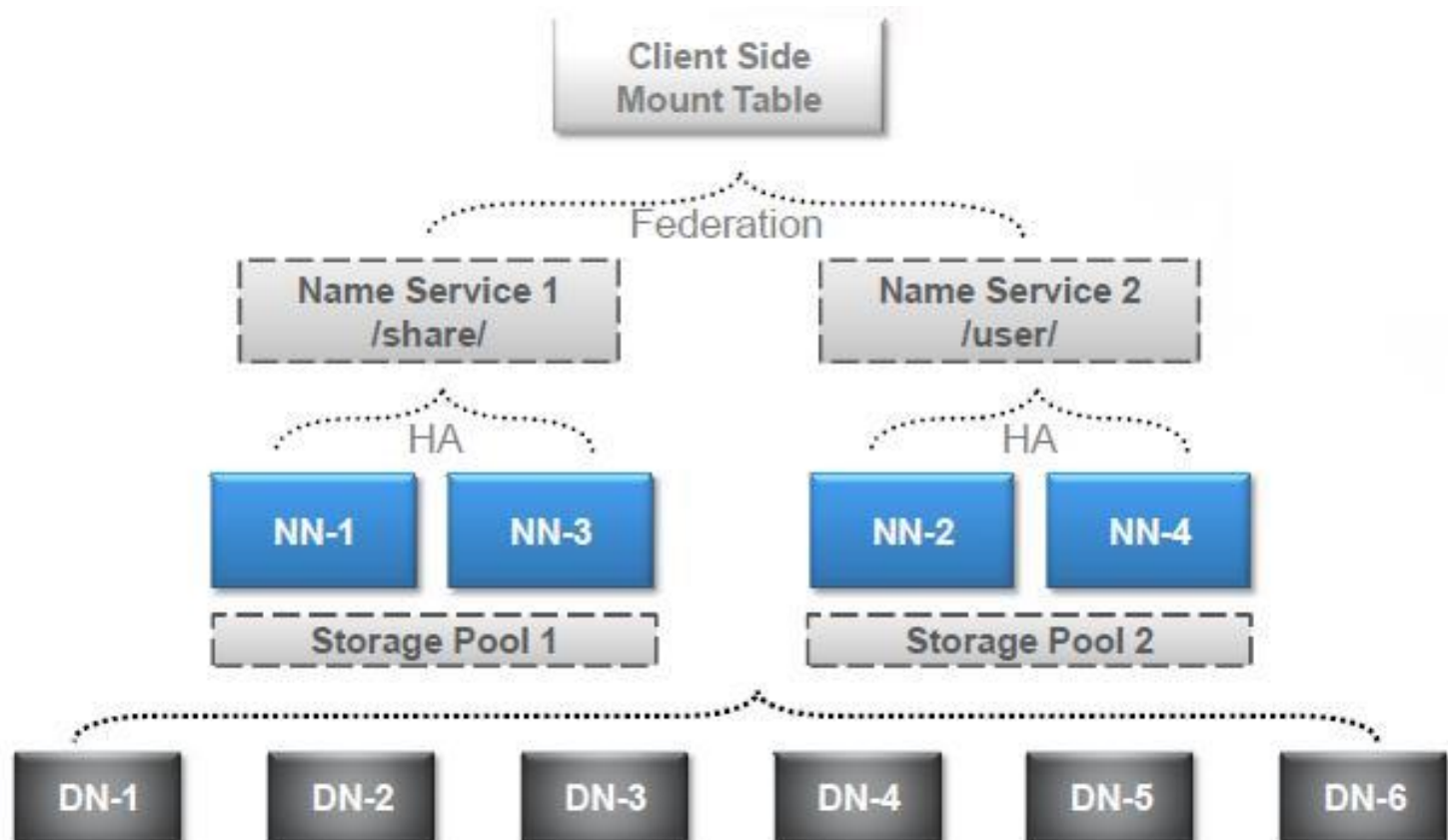
. HDFS

namenode(HA),
SecondaryNamenode,journalnode,datanode

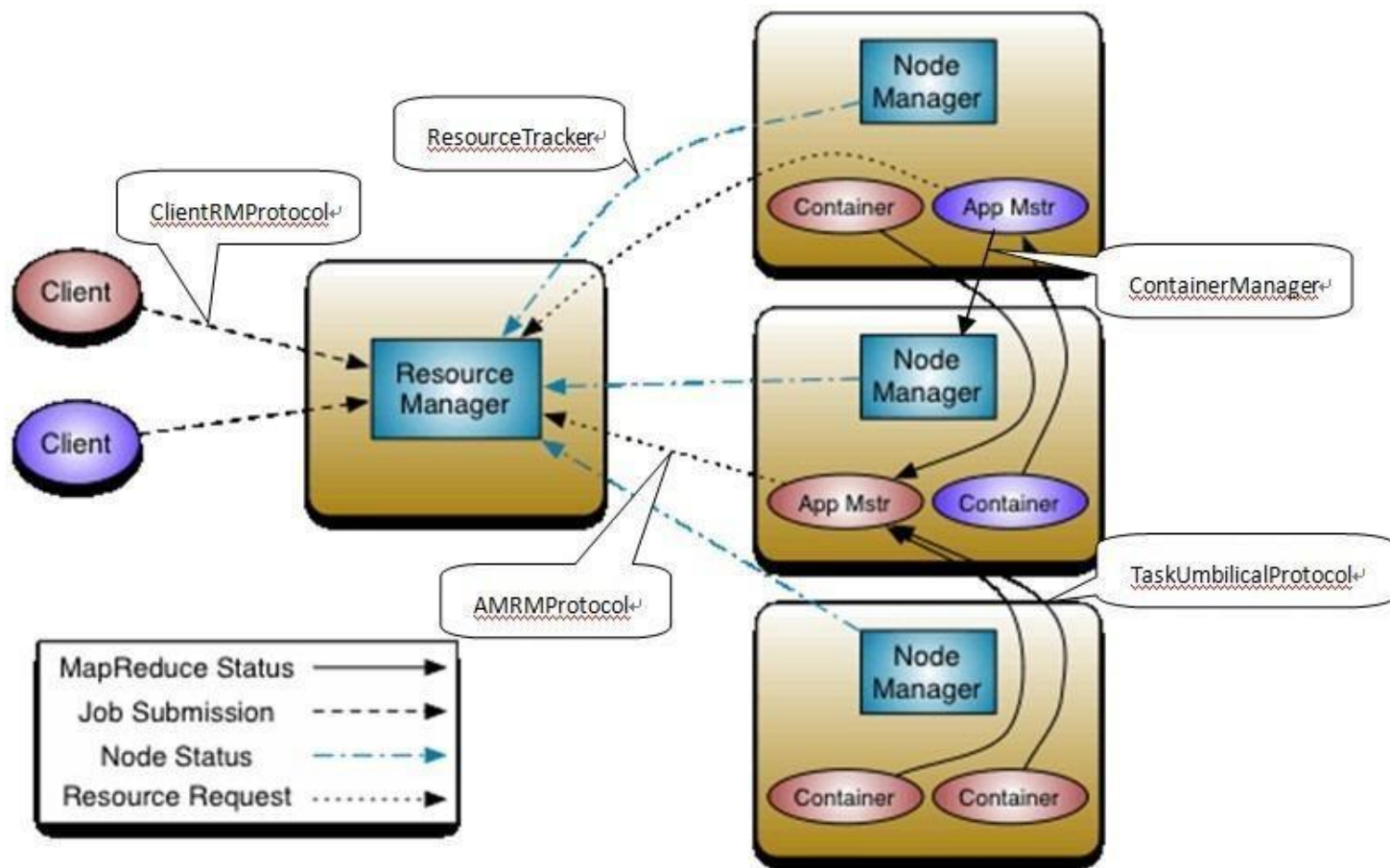
.Yarn

ResourceManager,NodeManager

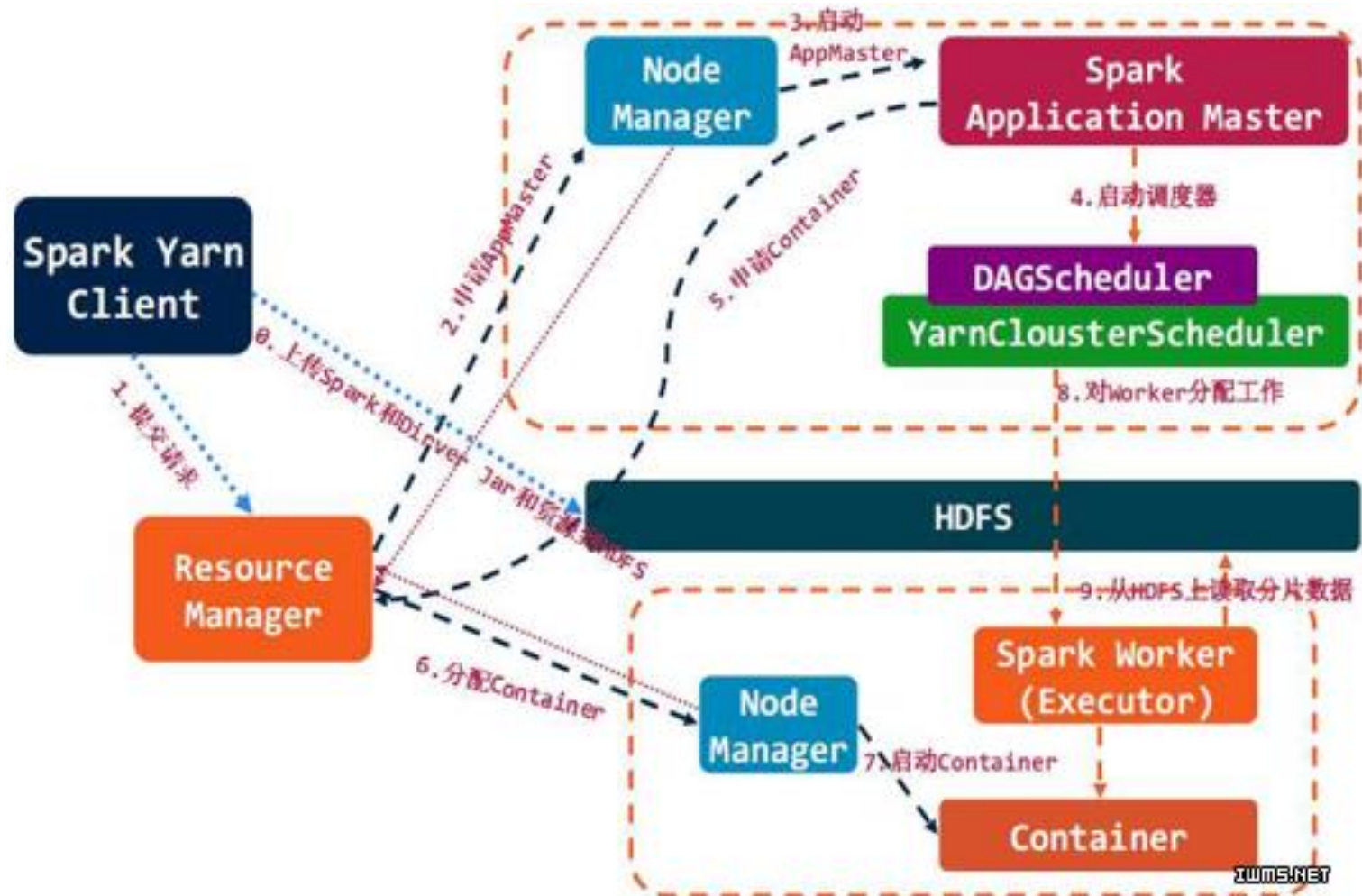
分布式存储 Federation



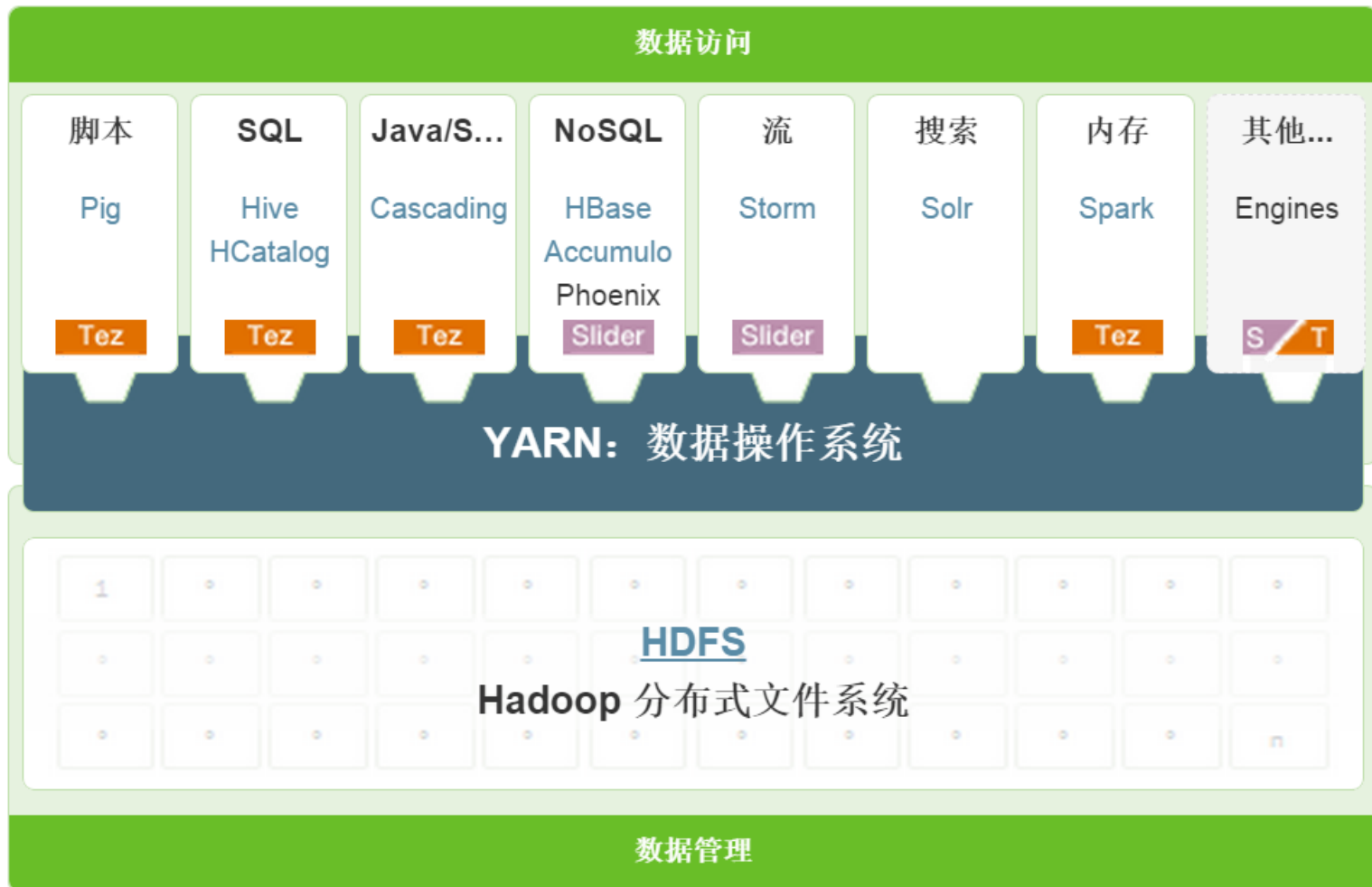
分布式计算(Yarn)



Yarn工作原理



Hadoop Cluster 2.0



课纲

- 1、Hadoop生态概述
- 2、分布式存储HDFS
- 3、分布式计算MR
- 4、Why Hadoop2.0 ?
- 5、Hadoop2.0 介绍
- 6、Hadoop几大发行商

Hadoop几大发行商？

目前大数据解决方案



其他方案

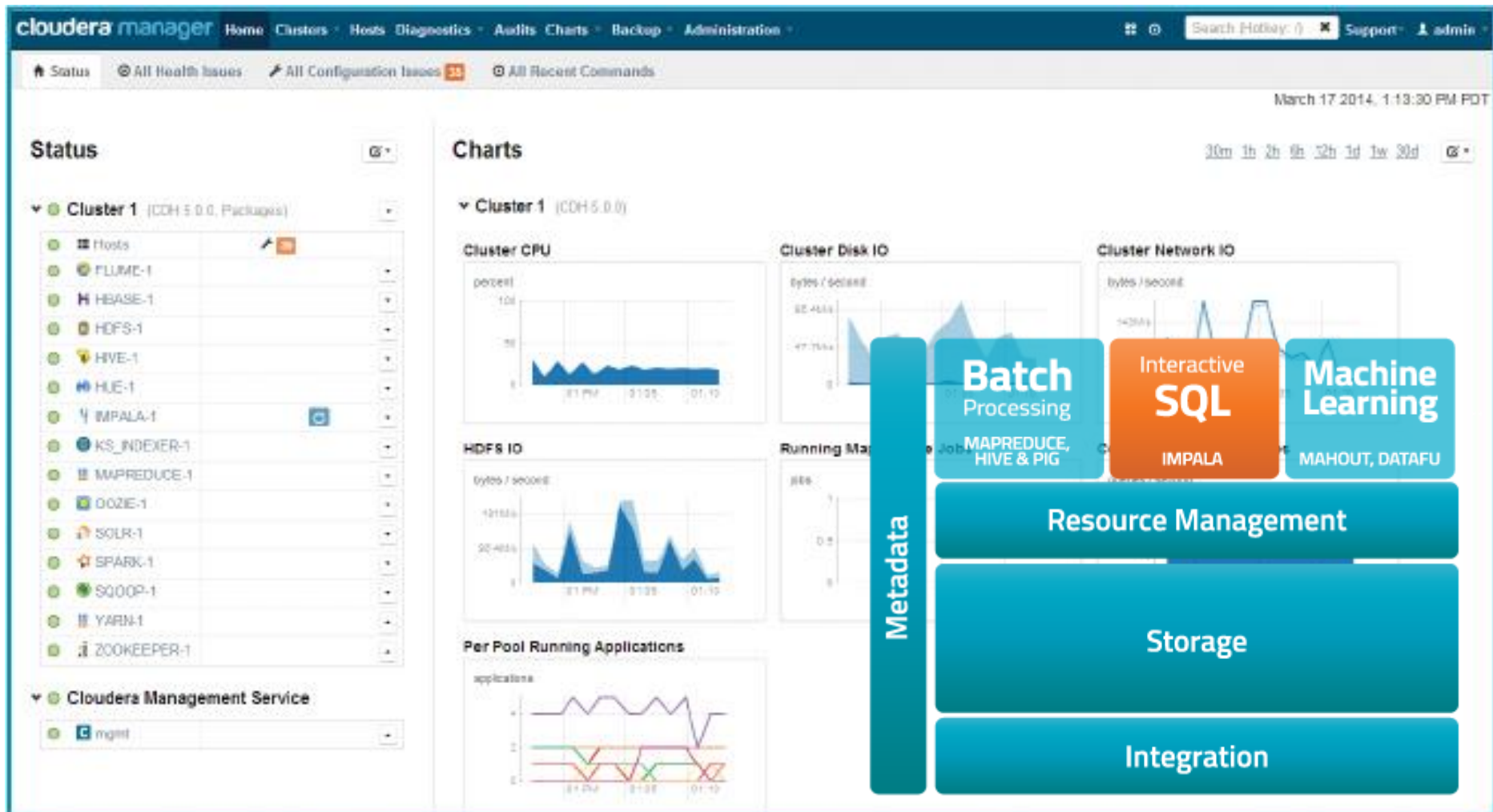


这是一个伟大的时代！

时间来证明？

一切，刚刚开始！

Cloudera Cluster

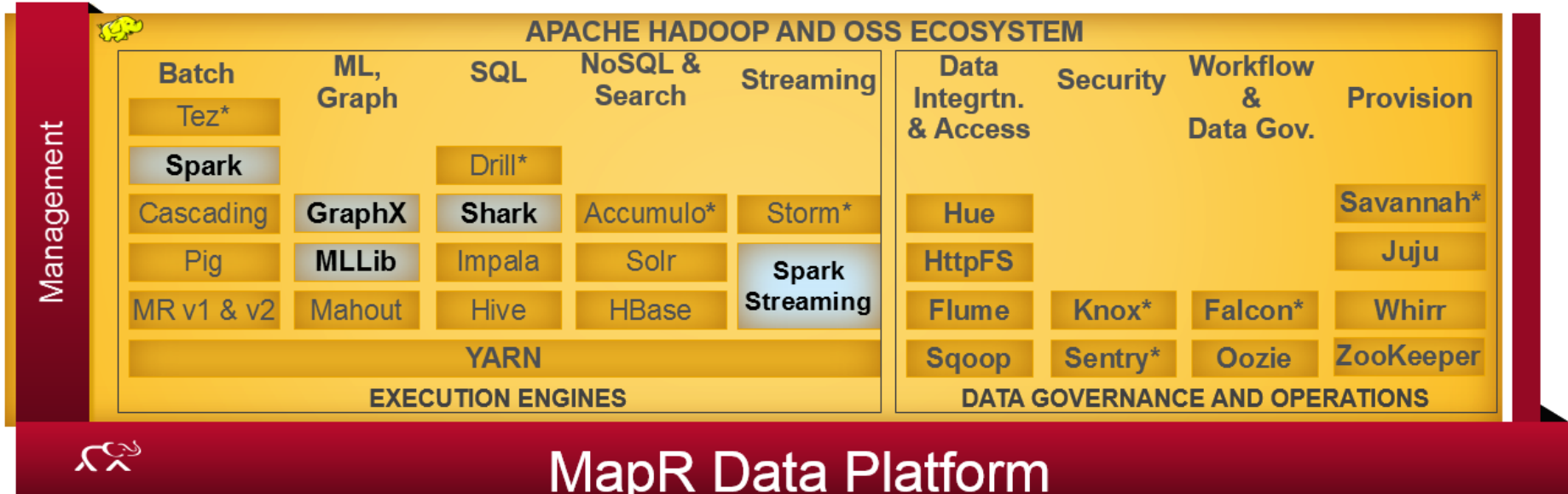


Hortonworks



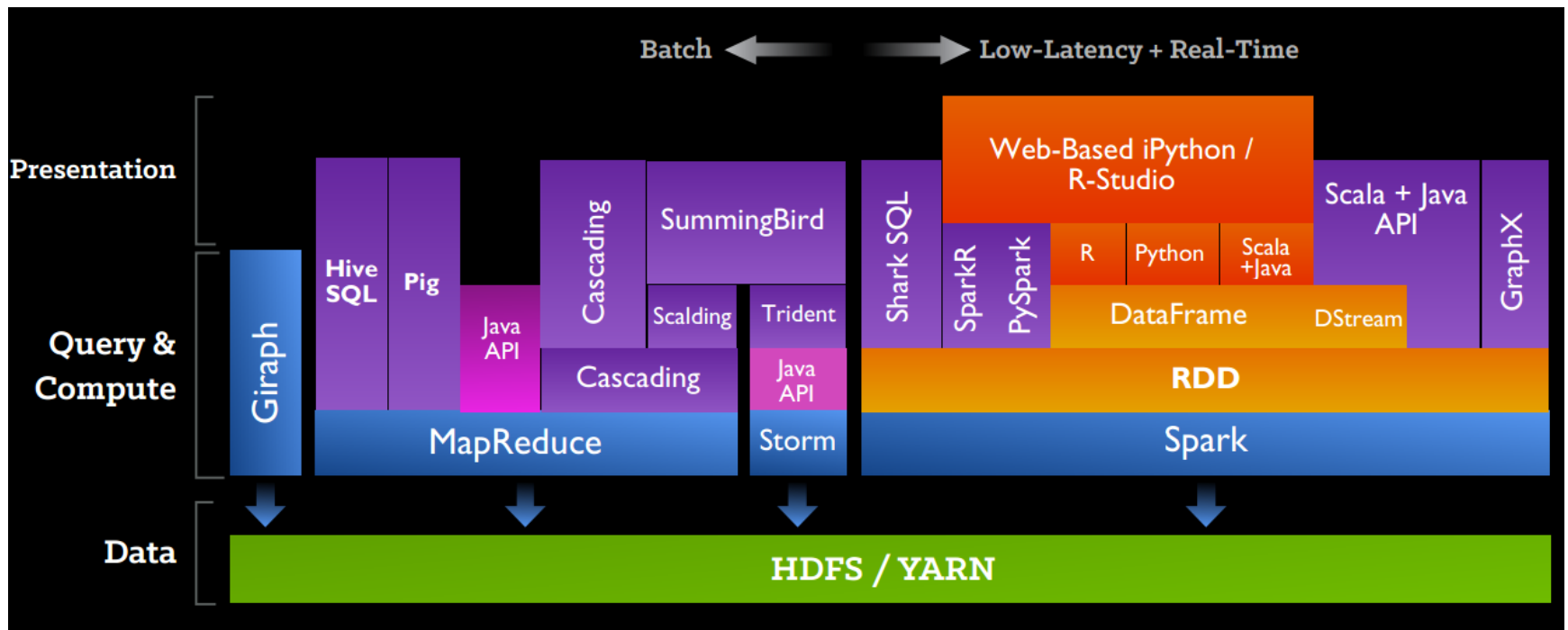
Hadoop Stack MapR?

The Complete Spark Stack on Hadoop



* 2014 TIMELINE

Hadoop on spark stack



演示



介绍性质

大数据



Thank you

提问时间?

Blog: <http://www.itweet.cn>

PPT: <https://github.com/itweet/course>

Video: <http://www.tudou.com/home/sparkjvm/>

