北京理工大學
BEIJING INSTITUTE OF TECHNOLOGY

大数据处理架构 Hadoop

大数据处理技术

计算机学院

课程提纲

- Hadoop简介
- Hadoop生态系统
- Hadoop安装与使用方法
- Hadoop集群部署

Hadoop简介

- Hadoop是Apache软件基金会旗下的一个开源分布式计算平台，为用户提供了系统底层细节透明的分布式基础架构

- Hadoop是基于Java语言开发的，具有很好的跨平台特性，并且可以部署在廉价的计算机集群中

- Hadoop可以支持多种编程语言
  - C/C++, Java, Python etc.

Hadoop简介

- Hadoop的核心是分布式文件系统HDFS（Hadoop Distributed File System）和 MapReduce

- Hadoop被公认为行业大数据标准开源软件，在分布式环境下提供了海量数据的处理能力

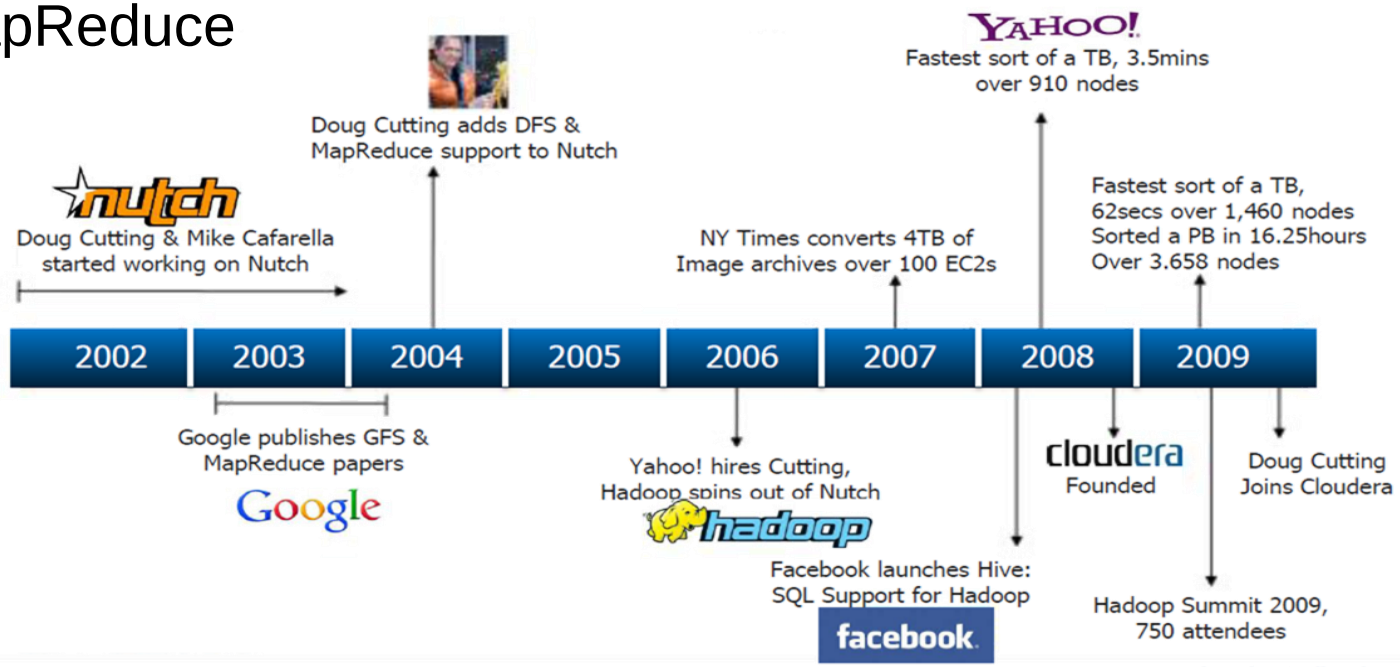
- 几乎所有主流厂商都围绕Hadoop提供开发工具、开源软件、商业化工具和技术服务，如谷歌、微软、Facebook、阿里、百度等

Hadoop简介

Hadoop 之父 Doug Cutting

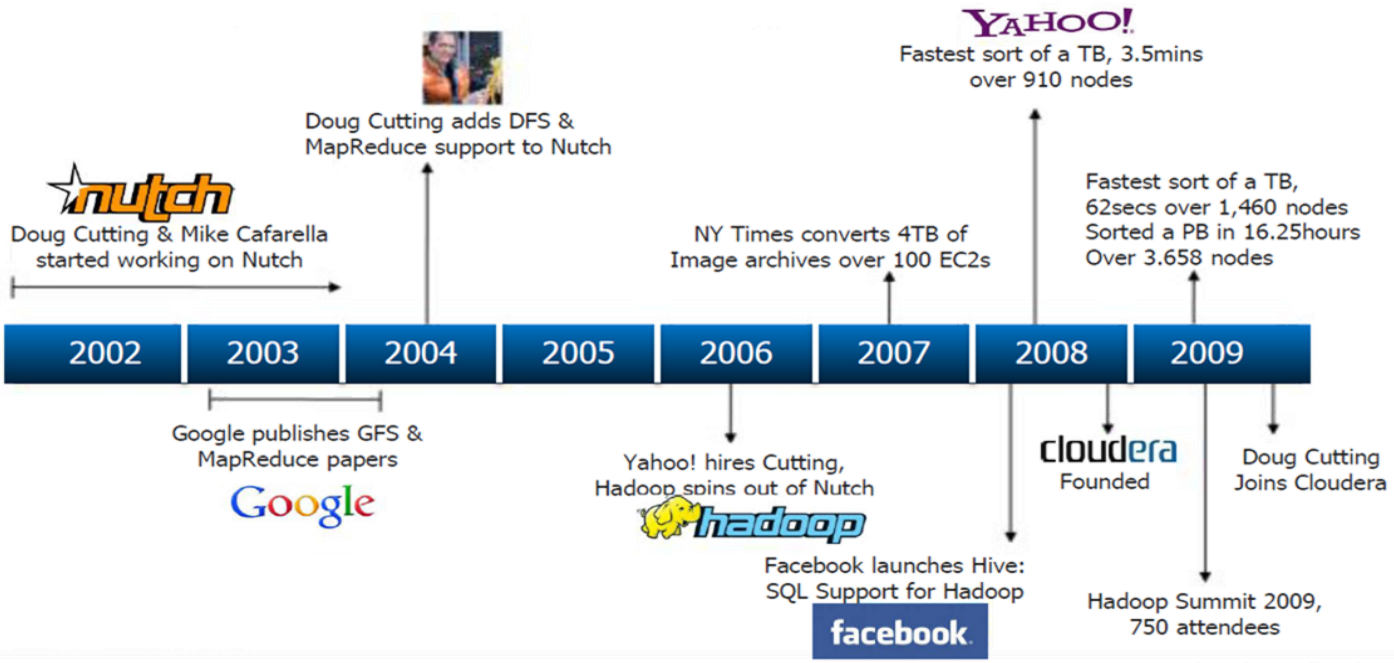
Hadoop简介

- Hadoop源自始于2002年的Apache Nutch项目——一个开源的网络搜索引擎并且也是Lucene项目的一部分
- 在2004年，Nutch项目也模仿GFS开发了自己的分布式文件系统NDFS（Nutch Distributed File System），即HDFS的前身
- 2004年，谷歌公司又发表了另一篇具有深远影响的论文，阐述了MapReduce分布式编程思想，Nutch开源实现了谷歌的MapReduce



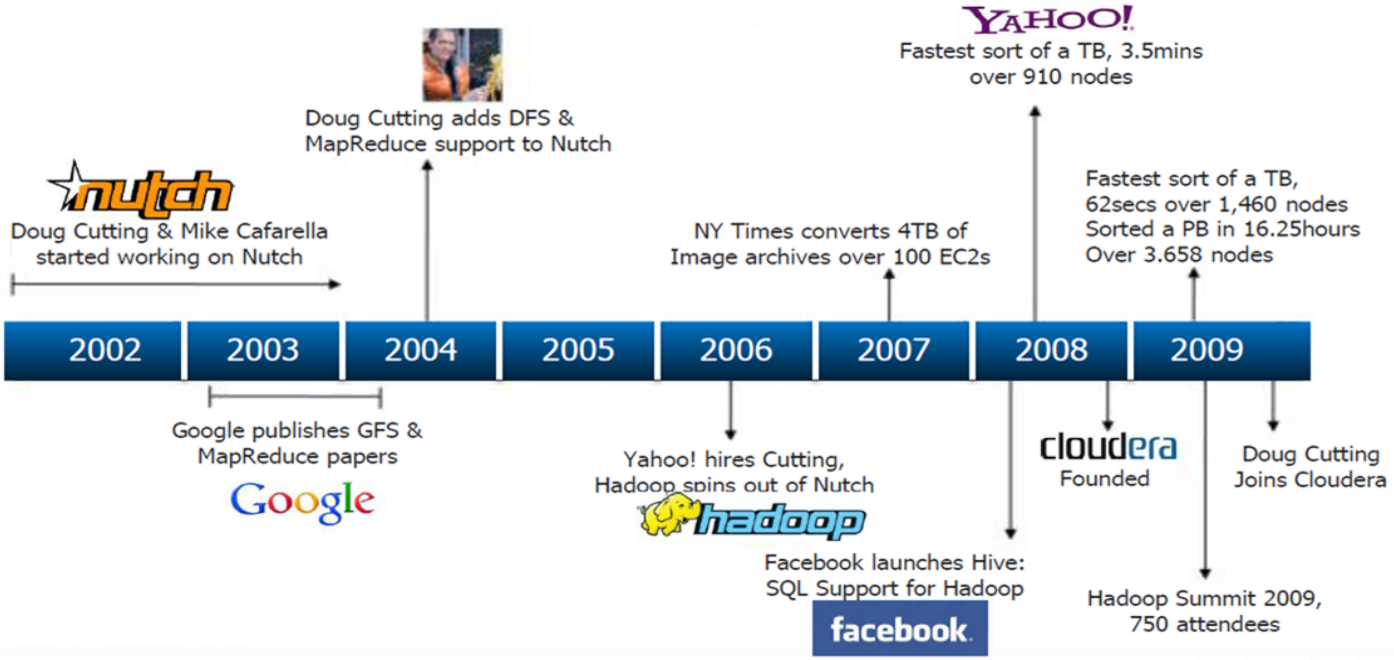
Hadoop简介

- 到了2006年2月，Nutch中的NDFS和MapReduce开始独立出来，成为Lucene项目的一个子项目，称为Hadoop，同时，Doug Cutting加盟雅虎
- 2008年1月，Hadoop正式成为Apache顶级项目，Hadoop也逐渐开始被雅虎之外的其他公司使用



Hadoop简介

- 2008年4月，Hadoop打破世界纪录，成为最快排序1TB数据的系统，它采用一个由910个节点构成的集群进行运算，排序时间只用了209秒
- 在2009年5月，Hadoop更是把1TB数据排序时间缩短到62秒。Hadoop从此名声大震，迅速发展成为大数据时代最具影响力的开源分布式开发平台，并成为大数据处理标准



Hadoop特性

Hadoop是一个能够对大量数据进行分布式处理的软件框架，并且是以一种可靠、高效、可伸缩的方式进行处理的，它具有以下几个方面的特性：

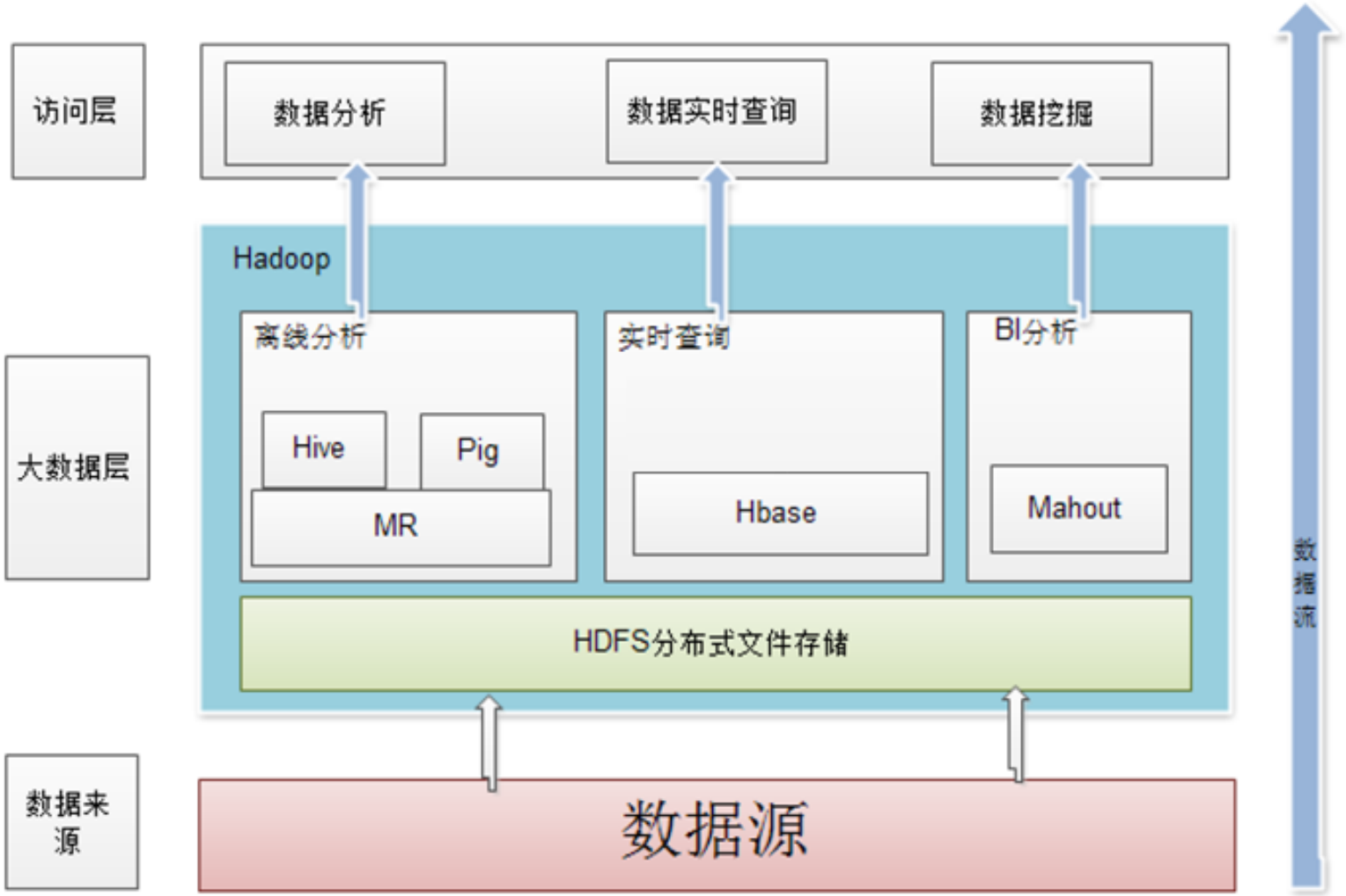
- 高可靠性
- 高效性
- 高可扩展性
- 高容错性
- 成本低

Hadoop的应用现状

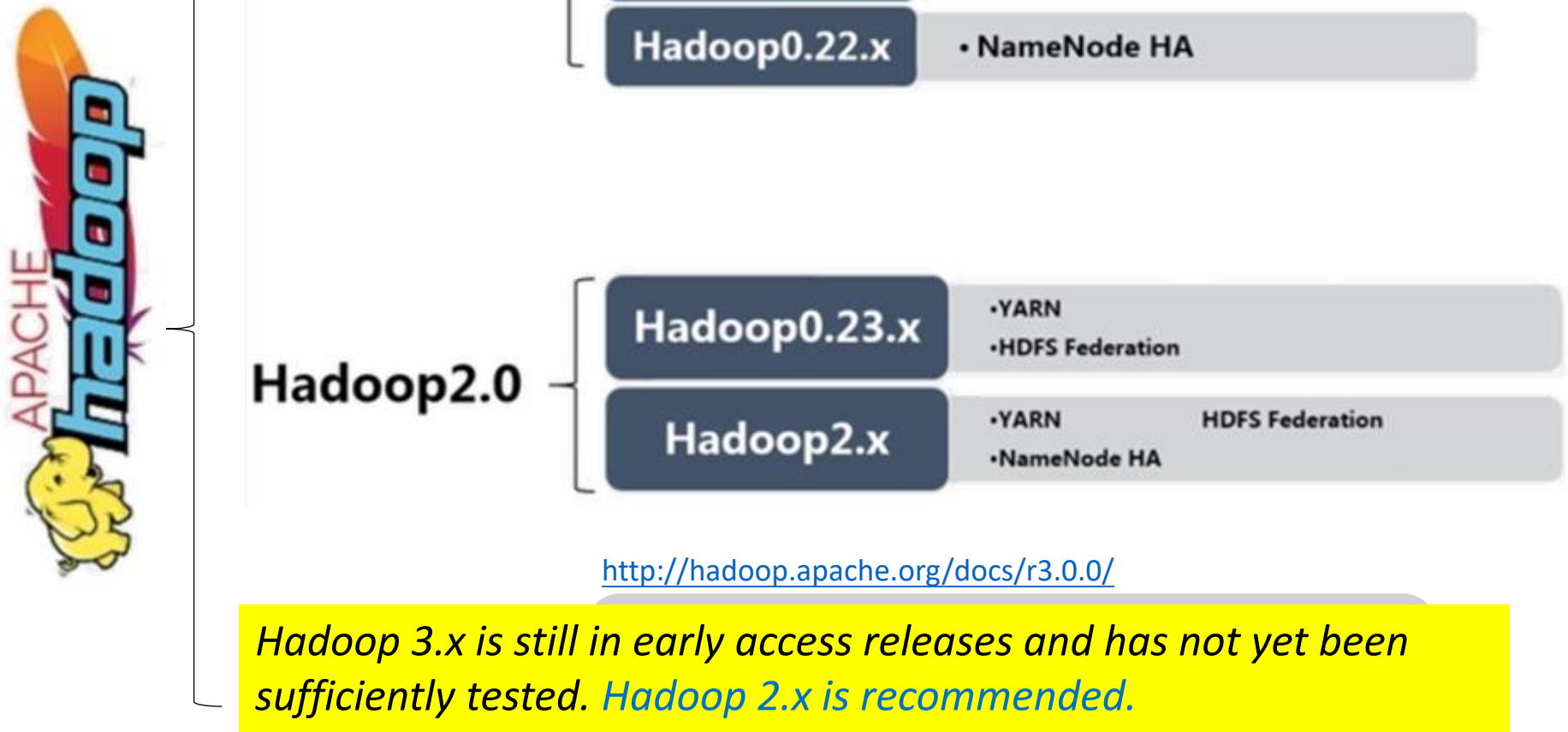
Hadoop凭借其突出的优势，已经在各个领域得到了广泛的应用，而互联网领域是其应用的主阵地

- Facebook主要将Hadoop平台用于日志处理、推荐系统和数据仓库等方面
- 国内采用Hadoop的公司主要有百度、阿里、网易、华为、中国移动等，其中，阿里的Hadoop集群比较大

Hadoop在企业中的应用架构



Apache Hadoop版本演变



http://hadoop.apache.org/docs/r3.0.0/

Hadoop 3.x is still in early access releases and has not yet been sufficiently tested. Hadoop 2.x is recommended.

5. MapReduce task-level native optimization
6. …

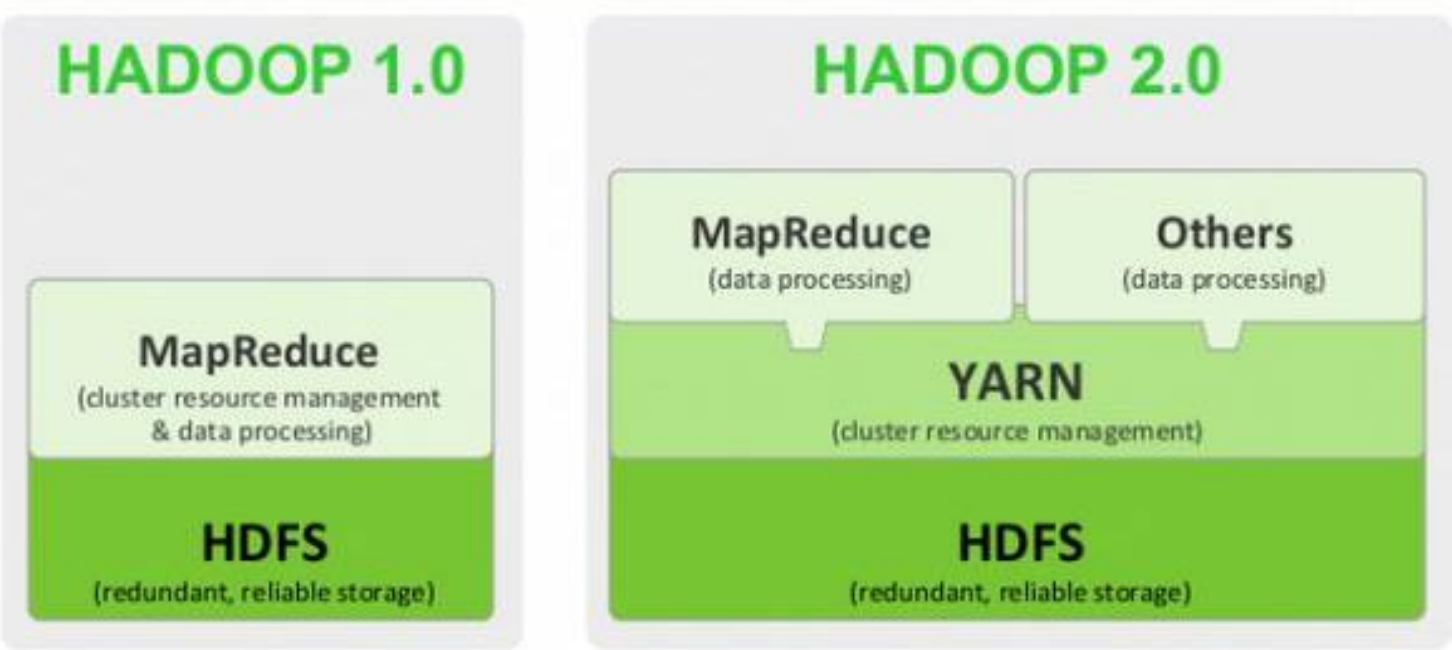
Apache Hadoop版本演变

- ✅ = Tested to be fully-functional
- ❌ = Known to not be fully-functional
- ⚠ = Not tested, may/may-not function

Hadoop version support matrix

	HBase-1.2.x, HBase-1.3.x	HBase-1.4.x	HBase-2.0.x	HBase-2.1.x
Hadoop-2.4.x	✅	❌	❌	❌
Hadoop-2.5.x	✅	❌	❌	❌
Hadoop-2.6.0	❌	❌	❌	❌
Hadoop-2.6.1+	✅	❌	✅	❌
Hadoop-2.7.0	❌	❌	❌	❌
Hadoop-2.7.1+	✅	✅	✅	✅
Hadoop-2.8.[0-1]	❌	❌	❌	❌
Hadoop-2.8.2	⚠	⚠	⚠	⚠
Hadoop-2.8.3+	⚠	⚠	✅	✅
Hadoop-2.9.0	❌	❌	❌	❌
Hadoop-2.9.1+	⚠	⚠	⚠	⚠
Hadoop-3.0.[0-2]	❌	❌	❌	❌
Hadoop-3.0.3+	❌	❌	✅	✅
Hadoop-3.1.0	❌	❌	❌	❌
Hadoop-3.1.1+	❌	❌	✅	✅

Apache Hadoop版本演变



组件	Hadoop1.0的问题	Hadoop2.0的改进
HDFS	单一名称节点，存在单点失效问题	设计了HDFS HA，提供名称节点热备机制
HDFS	单一命名空间，无法实现资源隔离	设计了HDFS Federation，管理多个命名空间
MapReduce	资源管理效率低	设计了新的资源管理框架YARN

Apache Hadoop版本演变

Hadoop1.0的核心组件（仅指MapReduce和HDFS，不包括Hadoop生态系统内的Pig、Hive、HBase等其他组件），主要存在以下不足：

- 抽象层次低，需人工编码
- 表达能力有限
- 开发者自己管理作业（Job）之间的依赖关系
- 难以看到程序整体逻辑
- 执行迭代操作效率低
- 资源浪费（Map和Reduce分两阶段执行）
- 实时性差（适合批处理，不支持实时交互式）

Apache Hadoop版本演变

Hadoop的优化与发展主要体现在两个方面：

- 一方面是Hadoop自身两大核心组件MapReduce和HDFS的架构设计改进
- 另一方面是Hadoop生态系统其它组件的不断丰富，加入了Pig、Tez、Spark等新组件

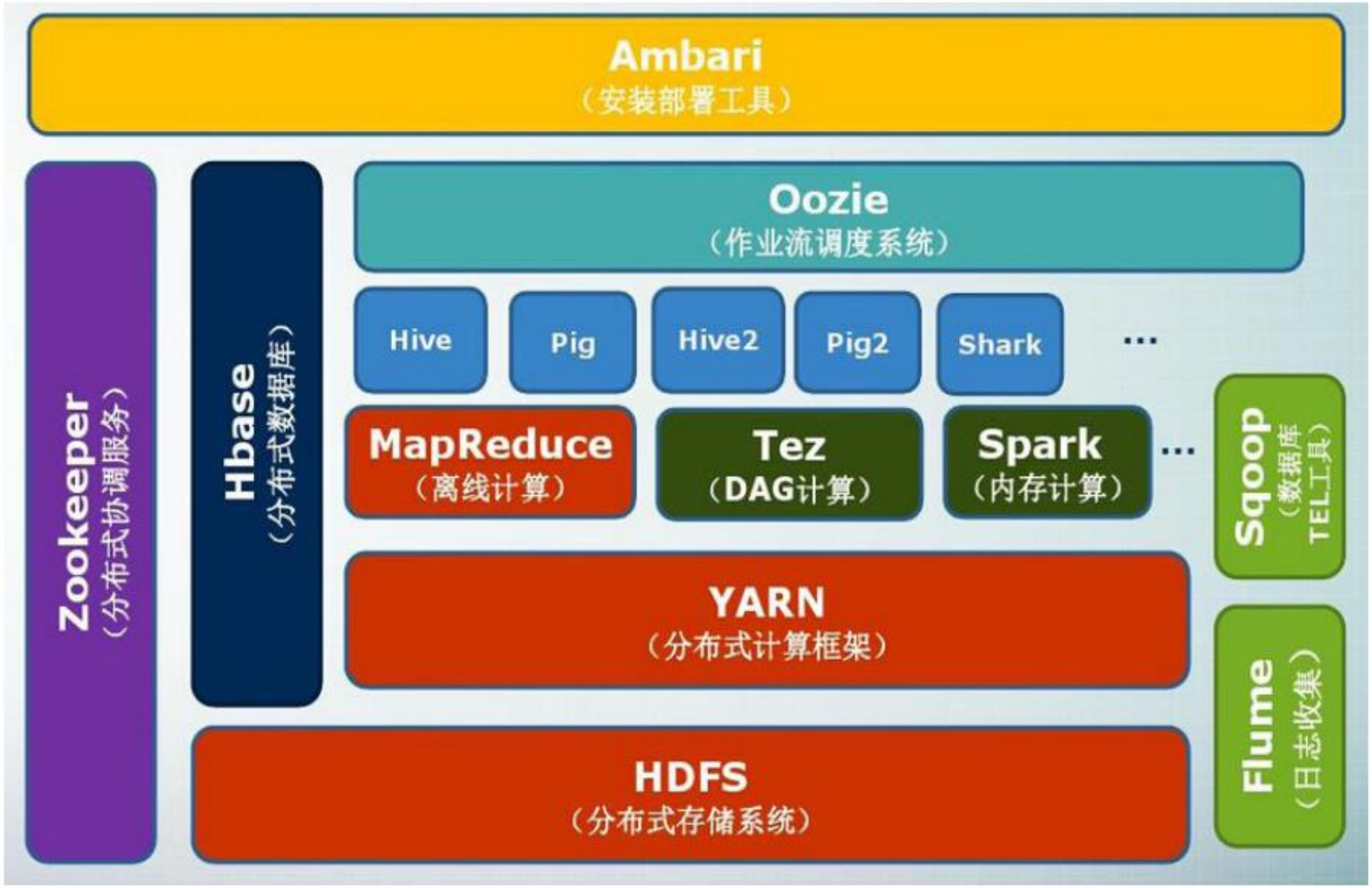
Apache Hadoop版本演变

不断完善的Hadoop生态系统

组件	功能	解决Hadoop中存在的问题
Pig	处理大规模数据的脚本语言，用户只需要编写几条简单的语句，系统会自动转换为MapReduce作业	抽象层次低，需要手工编写大量代码
Spark	基于内存的分布式并行编程框架，具有较高的实时性，并且较好支持迭代计算	延迟高，而且不适合执行迭代计算
Oozie	工作流和协作服务引擎，协调Hadoop上运行的不同任务	没有提供作业（Job）之间依赖关系管理机制，需要用户自己处理作业之间依赖关系
Tez	支持DAG作业的计算框架，对作业的操作进行重新分解和组合，形成一个大的DAG作业，减少不必要操作	不同的MapReduce任务之间存在重复操作，降低了效率
Kafka	分布式发布订阅消息系统，一般作为企业大数据分析平台的数据交换枢纽，不同类型的分布式系统可以统一接入到Kafka，实现和Hadoop各个组件之间的不同类型数据的实时高效交换	Hadoop生态系统中各个组件和其他产品之间缺乏统一的、高效的数据交换中介

Hadoop生态系统

Hadoop的项目结构不断丰富发展，已经形成一个丰富的Hadoop生态系统



Hadoop生态系统: Hive

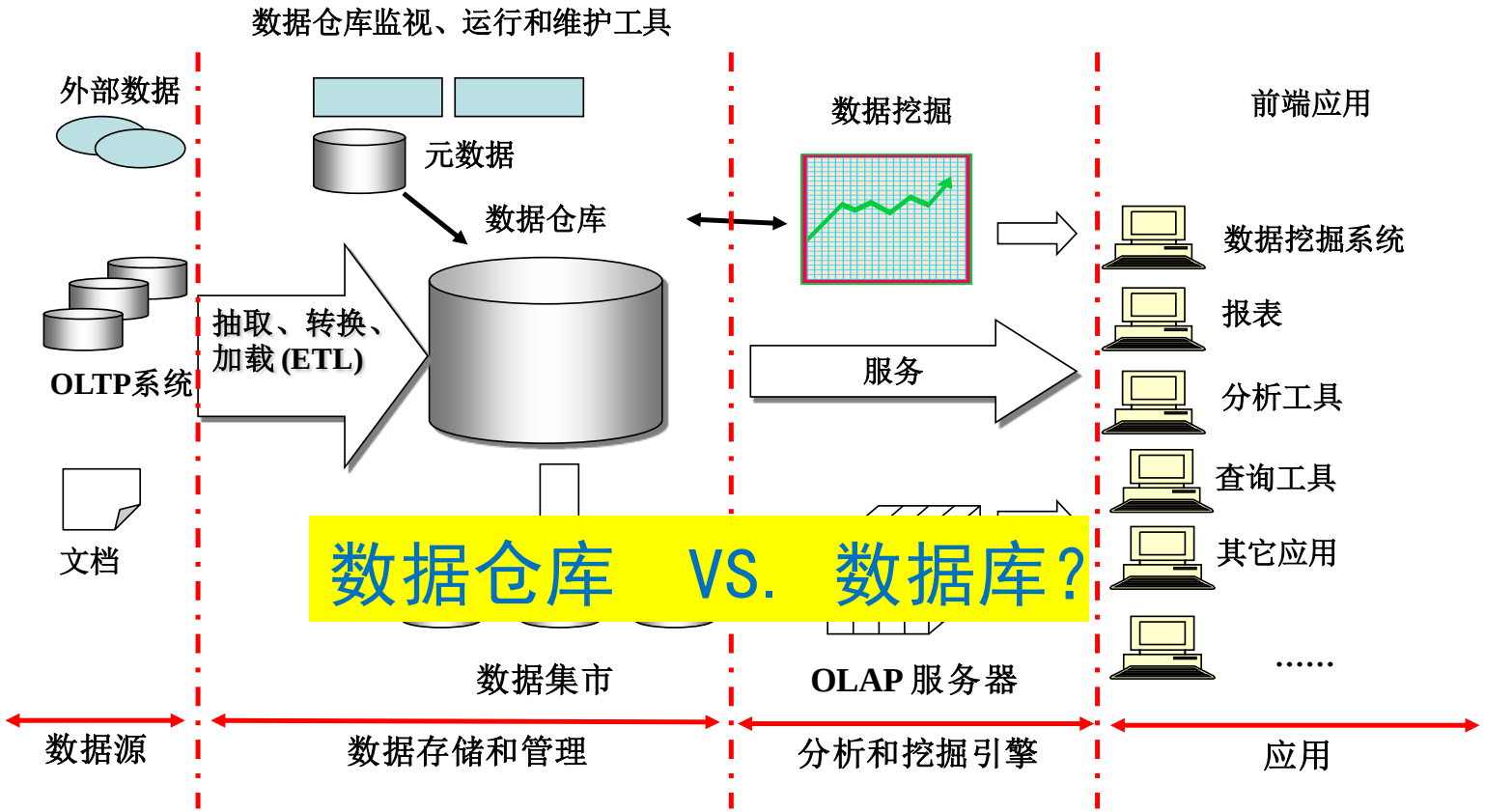
- Hive: 基于Hadoop的数据仓库工具

http://hive.apache.org/

- Hive架构在MapReduce之上，可以用于对存储在Hadoop文件中的数据集进行数据整理、特殊查询和分析处理
- Hive的学习门槛比较低，它提供了类似于关系数据库SQL语言的查询语言Hive QL
- Hive自身可以将HiveQL语句快速转换成MapReduce任务进行运行，而不必开发专门的MapReduce应用程序

Hadoop生态系统: Hive

数据仓库（Data Warehouse）是一个面向主题的（Subject Oriented）、集成的（Integrated）、相对稳定的（Non-Volatile）、反映历史变化（Time Variant）的数据集合，用于支持管理决策。



Hadoop生态系统: Hive

传统数据仓库面临的挑战：

（1）无法满足快速增长的海量数据存储需求

（2）无法有效处理不同类型的数据

（3）计算和处理能力不足

Hadoop生态系统: Hive

- Hive是一个构建于Hadoop顶层的数据仓库工具
- 支持大规模数据存储、分析，具有良好的可扩展性
- 某种程度上可以看作是用户编程接口，本身不存储和处理数据
- 依赖分布式文件系统HDFS存储数据
- 依赖分布式并行计算模型MapReduce处理数据
- 定义了简单的类似SQL 的查询语言——HiveQL
- 用户可以通过编写的HiveQL语句运行MapReduce任务
- 可以很容易把原来构建在关系数据库上的数据仓库应用程序移植到Hadoop平台上

Hadoop生态系统: Hive

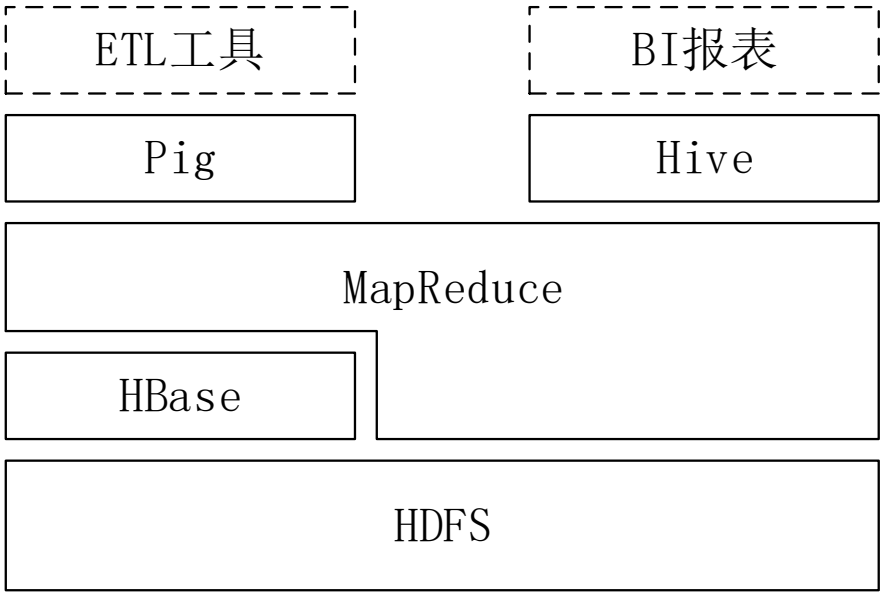
Hive具有的特点非常适用于数据仓库

- 采用批处理方式处理海量数据
  - Hive需要把HiveQL语句转换成MapReduce任务进行运行
  - 数据仓库存储的是静态数据，对静态数据的分析适合采用批处理方式，不需要快速响应给出结果，而且数据本身也不会频繁变化
- 提供适合数据仓库操作的工具
  - Hive本身提供了一系列对数据进行提取、转换、加载（ETL）的工具，可以存储、查询和分析存储在Hadoop中的大规模数据
  - 这些工具能够很好地满足数据仓库各种应用场景

Hadoop生态系统: Hive

Hive与Hadoop生态系统中其他组件的关系

- Hive依赖于HDFS 存储数据
- Hive依赖于MapReduce 处理数据
- 在某些场景下Pig可以作为Hive的替代工具
- HBase 提供数据的实时访问



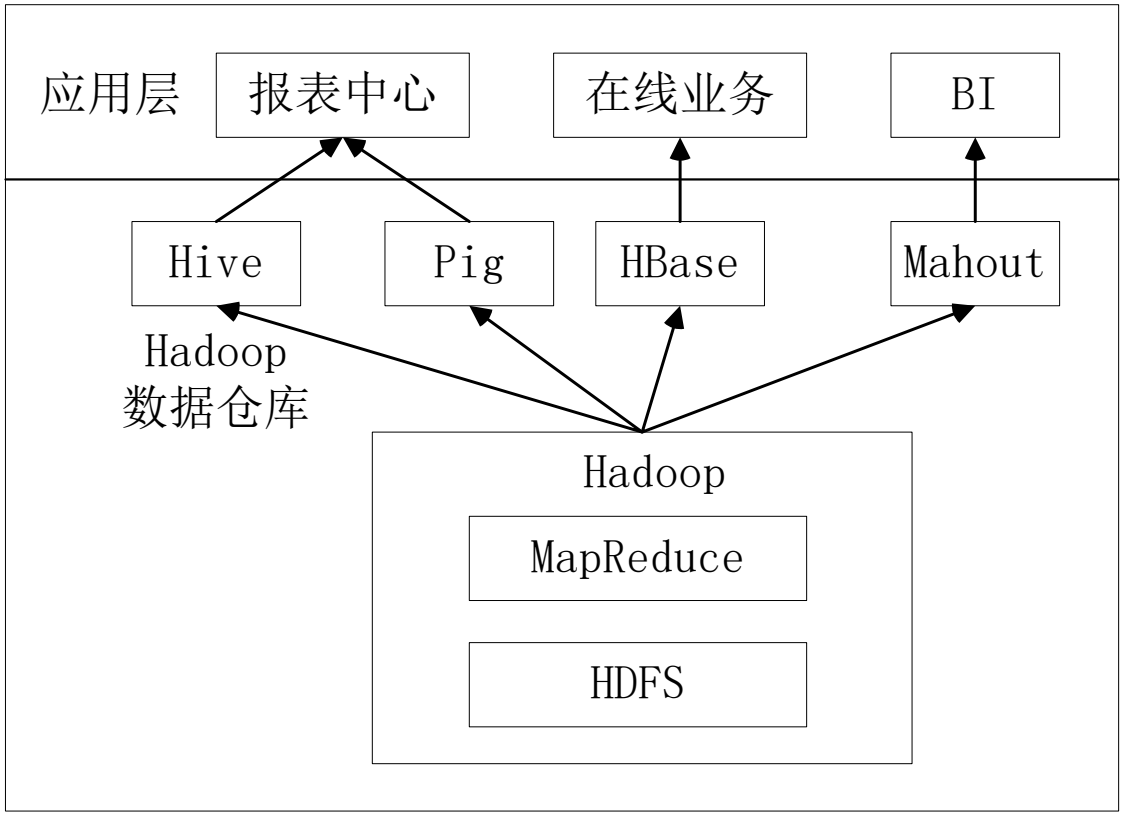
Hadoop生态系统: Hive

Hive在很多方面和传统的关系数据库类似，但是它的底层依赖的是HDFS和MapReduce，所以在很多方面又有别于传统数据库

对比项目	Hive	传统数据库
数据插入	支持批量导入	支持单条和批量导入
数据更新	不支持	支持
索引	支持	支持
执行延迟	高	低
扩展性	好	有限

Hadoop生态系统: Hive

Hive在企业大数据分析平台中的应用



Hive应用实例：WordCount

- 词频统计任务：创建input目录，在input文件夹中创建两个测试文件file1.txt和file2.txt，内容分别为“hello world”， "hello hadoop"

```
$ hive
 hive> create table docs(line string);
 hive> load data inpath 'input' overwrite into table docs;
 hive> create table word_count as
    select word, count(1) as count from
    (select explode(split(line,' '))as word from docs) w
    group by word
    order by word;
```

```
hive> select * from word_count;
OK
hadoop  1
hello   2
world   1
Time taken: 0.043 seconds, Fetched: 3 row(s)
```

Hive应用实例：WordCount

WordCount算法在MapReduce中的编程实现和Hive中编程实现的主要不同点：

- 采用Hive实现WordCount算法需要编写较少的代码量
- 在MapReduce的实现中，需要进行编译生成jar文件来执行算法，而在Hive中不需要
  - HiveQL语句的最终实现需要转换为MapReduce任务来执行，这都是由Hive框架自动完成的，用户不需要了解具体实现细节

Hadoop生态系统: Pig

- Pig: 同样可以简化MapReduce的编程
- 提供类似SQL的查询语言Pig Latin

http://pig.apache.org/

- 允许用户通过编写简单的脚本来实现复杂的数据分析，而不需要编写复杂的MapReduce应用程序
- Pig会自动把用户编写的脚本转换成MapReduce作业在Hadoop集群上运行，而且具备对生成的MapReduce程序进行自动优化的功能
- Hive一般处理的是结构化的数据，Pig可以处理非结构化数据
- 处理流程： LOAD→转换→STORE/DUMP

Pig应用实例：WordCount

- 词频统计任务：创建input目录，在input文件夹中创建两个测试文件file1.txt和file2.txt，内容分别为“hello world”， "hello hadoop"

PigLatin Script

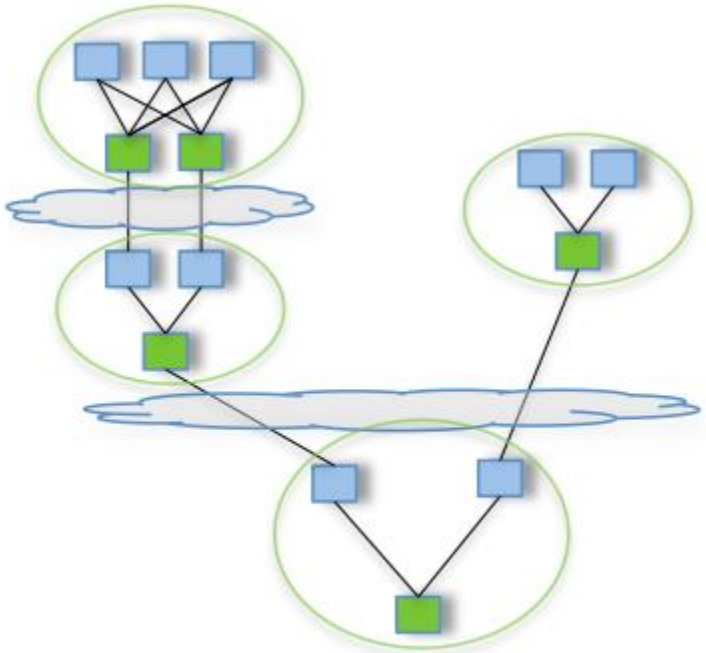
```
docs = LOAD '/usr/local/hadoop/input' AS (line:chararray);
words = FOREACH docs GENERATE FLATTEN(TOKENIZE(line)) AS word;
groupd = GROUP words BY word;
wordcount = FOREACH groupd GENERATE group, COUNT(words);
DUMP wordcount;
```

Hadoop生态系统

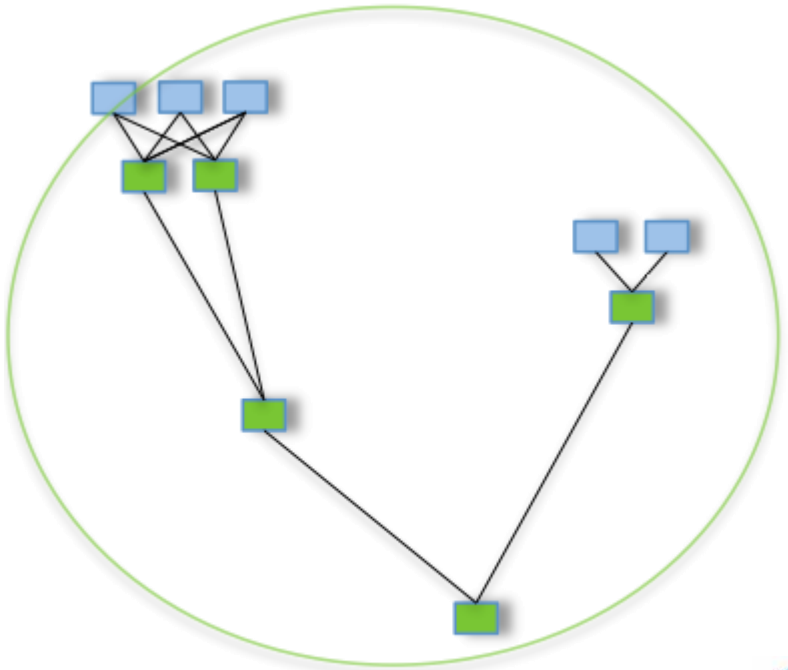
- Tez: 支持DAG作业的计算框架

https://tez.apache.org/

- 核心思想是将Map和Reduce两个操作进一步拆分,分解后的元操作可以任意灵活组合，产生新的操作,这些操作经过一些控制程序组装后，可形成一个大的DAG作业



Pig/Hive - MR



Pig/Hive - Tez

Hadoop生态系统

组件	功能
HDFS	分布式文件系统
MapReduce	分布式并行编程模型
YARN	资源管理和调度器
Tez	运行在YARN之上的下一代Hadoop查询处理框架
Hive	Hadoop上的数据仓库
HBase	Hadoop上的非关系型的分布式数据库
Pig	一个基于Hadoop的大规模数据分析平台，提供类似SQL的查询语言Pig Latin
Sqoop	用于在Hadoop与传统数据库之间进行数据传递
Oozie	Hadoop上的工作流管理系统
Zookeeper	提供分布式协调一致性服务
Storm	流计算框架
Flume	一个高可用的，高可靠的，分布式的海量日志采集、聚合和传输的系统
Ambari	Hadoop快速部署工具，支持Apache Hadoop集群的供应、管理和监控
Kafka	一种高吞吐量的分布式发布订阅消息系统，可以处理消费者规模的网站中的所有动作流数据
Spark	类似于Hadoop MapReduce的通用并行框架

Hadoop安装与使用方法

准备工作：

- 安装Linux虚拟机 或 安装双操作系统
- 熟悉Linux常用命令

Hadoop安装与使用方法

Hadoop安装方式

- 单机模式：Hadoop 默认模式为非分布式模式（本地模式），无需进行其他配置即可运行。
- 伪分布式模式：Hadoop 可以在单节点上以伪分布式的方式运行，Hadoop 进程以分离的 Java 进程来运行，节点既作为 NameNode 也作为 DataNode，同时，读取的是 HDFS 中的文件
- 分布式模式：使用多个节点构成集群环境来运行 Hadoop

Hadoop安装与使用方法

实验步骤：

- 创建Hadoop用户
- SSH登录权限设置
- 安装Java环境
- 单机安装配置
- 伪分布式安装配置

https://hadoop.apache.org

http://dblab.xmu.edu.cn/blog/install-hadoop/

Hadoop安装与使用方法

SSH登录权限设置

- Hadoop名称节点（NameNode）需要启动集群中所有机器的Hadoop守护进程，这个过程需要通过SSH登录来实现。Hadoop并没有提供SSH输入密码登录的形式，因此，为了能够顺利登录每台机器，需要将所有机器配置为名称节点可以无密码登录它们

Hadoop安装与使用方法

单机安装配置

- 下载Hadoop安装文件
- 将文件解压后即可使用
- 输入命令hadoop version来检查 Hadoop 是否可用，成功则会显示 Hadoop 版本信息

```
$ cd /usr/local/hadoop
$./bin/hadoop version
```

- 运行实例

Hadoop安装与使用方法

伪分布式安装实验步骤：

- 伪分布式方式需要修改2个配置文件
  - core-site.xml
  - hdfs-site.xml
- 初始化文件系统 hdfs namenode –format
- 启动Hadoop start-dfs.sh
- 访问web界面，查看Hadoop信息
- 运行实例

Hadoop安装与使用方法

修改配置文件 **core-site.xml**

```
<configuration>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>file:/usr/local/hadoop/tmp</value>
        <description>A base for other temporary directories.</description>
    </property>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

伪分布式虽然只需要配置 fs.defaultFS 和 dfs.replication 就可以运行（官方教程如此），不过若没有配置 hadoop.tmp.dir 参数，则使用默认的临时目录，而这个目录在重启时有可能被系统清理掉，导致必须重新执行 format 才行

Hadoop安装与使用方法

修改配置文件 **hdfs-site.xml**

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/hadoop/tmp/dfs/name</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
       <value>file:/usr/local/hadoop/tmp/dfs/data</value>
    </property></configuration>
```

- dfs.replication表示副本的数量，伪分布式要设置为1
- dfs.namenode.name.dir表示本地磁盘目录，是存储fsimage文件的地方
- dfs.datanode.data.dir表示本地磁盘目录，HDFS数据存放block的地方

Hadoop集群的部署

- Hadoop框架中最核心的设计是为海量数据提供存储的HDFS和对数据进行计算的MapReduce
- MapReduce的作业主要包括：（1）从磁盘或从网络读取数据，即IO密集工作；（2）计算数据，即CPU密集工作
- Hadoop集群的整体性能取决于CPU、内存、网络以及存储之间的性能平衡。因此运营团队在选择机器配置时要针对不同的工作节点选择合适硬件类型
- 一个基本的Hadoop集群中的节点主要有
  - NameNode：负责协调集群中的数据存储
  - DataNode：存储被拆分的数据块
  - JobTracker：协调不同机器上数据计算任务
  - TaskTracker：负责执行由JobTracker指派的任务
  - SecondaryNameNode：帮助NameNode收集文件系统运行的状态信息

Hadoop集群的部署

- 在集群中，大部分的机器设备是作为Datanode和TaskTracker工作的，Datanode/TaskTracker的硬件规格可以参考以下方案：
  - 4个磁盘驱动器（单盘1-2T）
  - 2个4核CPU
  - 16-24GB内存
  - 千兆以太网

Hadoop集群的部署

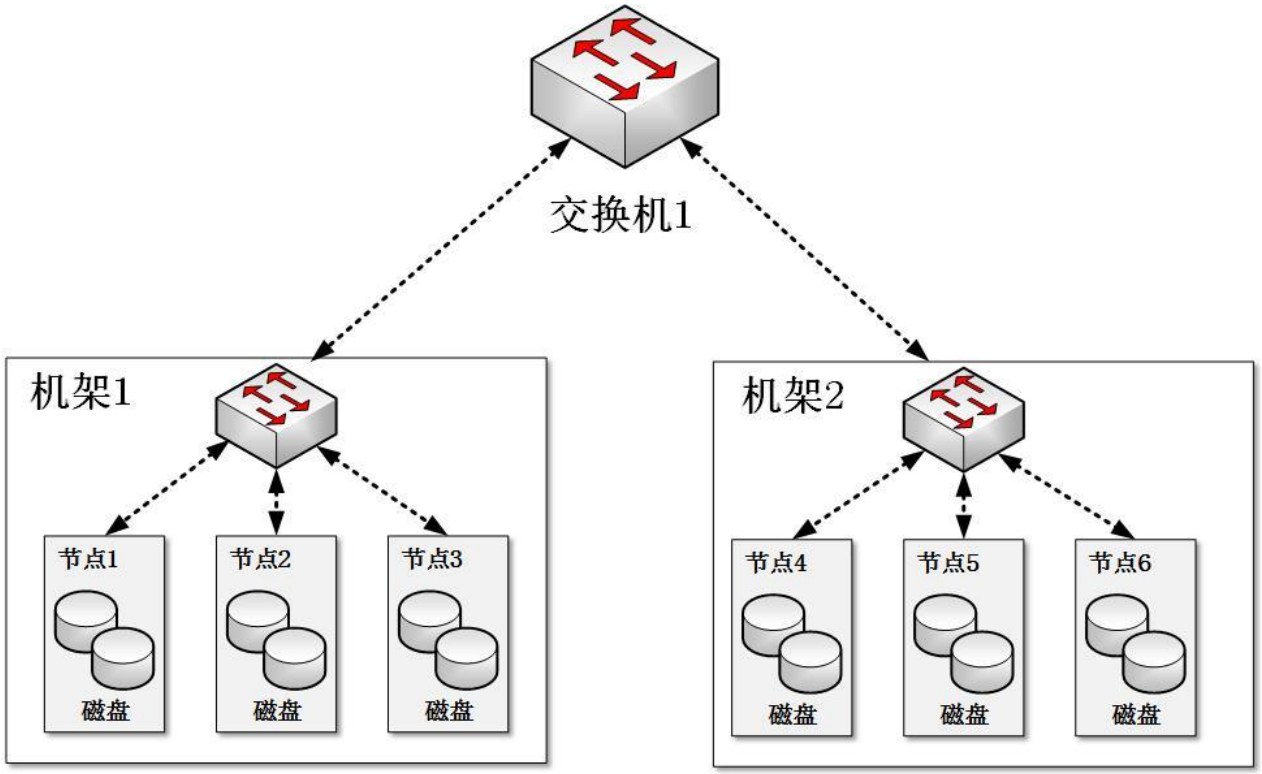
- NameNode提供整个HDFS文件系统的NameSpace(命名空间)管理、块管理等所有服务，很多元数据是直接保存在内存中的，因此需要更多的RAM，与集群中的数据块数量相对应，并且需要优化RAM的内存通道带宽。硬件规格可以采用参考方案：
  - 8-12个磁盘驱动器（单盘1-2T）
  - 2个4核/8核CPU
  - 16-72GB内存
  - 千兆/万兆以太网
- SecondaryNameNode在小型集群中可以和NameNode共用一台机器，较大的群集可以采用与NameNode相同的硬件

Hadoop集群的部署

- Hadoop集群规模可大可小，初始时，可以从一个较小规模的集群开始，比如包含10个节点，然后，规模随着存储器和计算需求的扩大而扩大
- 如果数据每周增大1TB，并且有三个HDFS副本，即每周需要3TB作为原始数据存储。要允许一些中间文件和日志（假定30%）的空间，由此，可以算出每周大约增加4T，需要增加一台新机器。存储两年数据的集群，大约需要100台机器
- 对于一个小的集群，名称节点（NameNode）和JobTracker运行在单个节点上，通常是可以接受的。但是，随着集群和存储在HDFS中的文件数量的增加，名称节点需要更多的主存，这时，名称节点和JobTracker就需要运行在不同的节点上
- 第二名称节点（SecondaryNameNode）会和名称节点可以运行在相同的机器上，但是，由于第二名称节点和名称节点几乎具有相同的主存需求，因此，二者最好运行在不同节点上

Hadoop集群的部署

- 普通的Hadoop集群结构由一个两阶网络构成
- 每个机架（Rack）有30-40个服务器，配置一个1GB的交换机，并向上传输到一个核心交换机或者路由器（1GB或以上）



小结

- Hadoop被视为事实上的大数据处理标准，本章介绍了Hadoop的发展历程，并阐述了Hadoop的高可靠性、高效性、高可扩展性、高容错性、成本低、运行在Linux平台上、支持多种编程语言等特性
- Hadoop目前已经在各个领域得到了广泛的应用，雅虎、Facebook、百度、淘宝、网易等公司都建立了自己的Hadoop集群
- 经过多年发展，Hadoop项目已经变得非常成熟和完善，包括HDFS、MapReduce、HBase、Hive、Pig等很多子项目，其中，HDFS和MapReduce是Hadoop的两大核心组件
- 本章最后介绍了如何在Linux系统下完成Hadoop的安装和配置，以及Hadoop集群的部署