



# 大数据处理技术





- 课程名称：**大数据开发技术Hadoop（Hadoop分布式计算系统）**
- 课时数：64课时（拟定）
- 上课时间：每周五下午、周六全天
- 授课起止时间：2021年11月12日~2021年12月31日
- 讲师：盛泳潘
- 班主任：王子暄（校内）、舒湛（校外）

# 目录

---

- **大数据处理技术**
- MapReduce并行计算技术
- Hadoop系统简介



# 大数据处理技术-大数据的技术特点

- 大数据的技术特点：

- **Volume (大体量)**：即可从数百TB到数十数百PB、甚至EB的规模。
- **Variety (多样性)**：大数据包含多种形式和形态的数据
- **Velocity (时效性)**：很多大数据需要在一定的时间限度下得到及时处理。
- **Veracity (准确性)**：处理的结果要保证一定的准确性。
- **Value (大价值)**：  
大数据包含很多深度的价值，  
为大数据分析挖掘和利用带来  
巨大的商业价值。

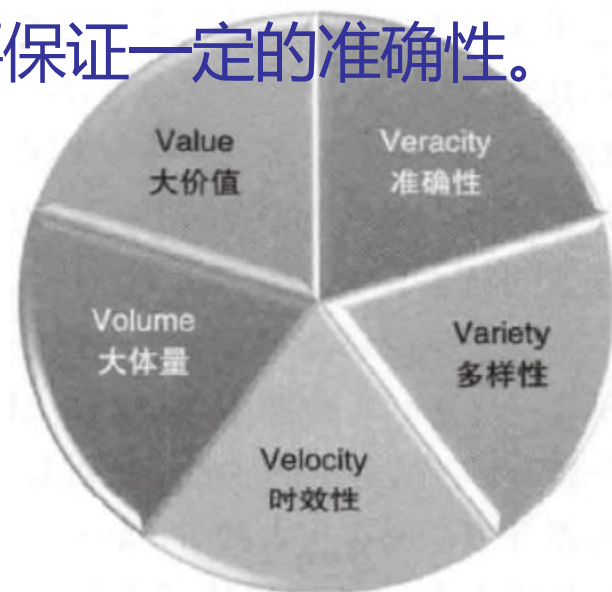


图1 大数据的5V特征



# 大数据处理技术-大数据的类型和计算特征

- 从数据结构特征角度看：
  - 结构化数据
  - 非结构化/半结构化数据
- 从数据获取处理方式看：
  - 批处理计算
  - 流式计算
- 从数据处理类型类型看：
  - 查询分析计算（传统）
  - 复杂数据挖掘计算（目前主流）
- 从大数据处理相应性能看：
  - 实时/准实时计算（例如，流式计算，查询分析类计算）
  - 非实时计算（例如，批处理计算，复杂数据挖掘计算）



# 大数据处理技术-大数据的类型和计算特征

- 从数据关系角度看：
  - 简单关系数据（例如，Web日志）
  - 复杂关系数据（例如，社交网络、分子结构、交通数据）
- 从迭代计算角度看：
  - 例如，机器学习等复杂计算任务
- 从并行计算体系结构特征角度看：
  - 并行计算（基于集群的分布式存储，并行计算体系结构和硬件平台）
  - 内存计算



# 大数据处理技术-主要目标和基本原则

- 大数据研究的**主要目标**是，以有效的信息手段和计算方法，获取、处理和分析各种应用行业的大数据，发现和提取数据的深度价值，为行业提供高附加值的应用和服务。
  - 核心目标：**价值发现**
  - 技术手段：**信息技术和计算方法**
  - 效益目标：**为行业提供高附加值的应用和服务**
- 大数据研究的**基本原则**是：
  - **应用需求为导向**
  - **领域交叉为桥梁**
  - **技术综合为支撑**



# 大数据处理技术-大数据研究的基本途径

- 从**数据关系**角度看：

- **寻找新算法降低计算复杂度**

在数据集很小时，很多在 $O(n)$ 、 $O(n\log n)$ 、 $O(n^2)$ 、 $O(n^3)$ 等线性或多项式复杂度的机器学习和数据挖掘算法都可以有效的工作，然而当数据规模增长到PB级尺度时，这些现有的串行算法将花费难以接受的时间开销，使得算法失效。

- **寻找和采用降低数据尺度的算法**

在保证结果精度的前提下，用数据抽样或数据尺度无关的近似算法来完成大数据的处理。

- **分而治之的并行化处理**

目前为止，大数据处理中最为有效和重要的方法之一还是采用并行化方法，在大规模分布式存储和并行计算平台上完成大数据并行化处理。





# 大数据处理技术-大数据计算模式和系统

- 根据大数据多样性的需求，目前出现了多种典型的大数据计算模型，以及与它们相适应的计算系统和工具。

表1 大数据计算模式及其对应的典型系统和工具

大数据计算模式	典型系统和工具
大数据查询分析计算	HBase, Hive, Cassandra, Premel, Impala, Shark, Hana, Redis, 等
批处理计算	MapReduce, Spark, 等
流式计算	Scribe, Flume, Storm, S4, Spark Steaming, 等
迭代计算	HaLoop, iMapReduce, Twister, Spark, 等
图计算	Pregel, Giraph, Trinity, PowerGraph, GraphX, 等
内存计算	Dremel, Hana, Redis, 等



# 大数据处理技术-大数据主要技术层面

- 从信息系统的角度来看，大数据处理是一个涉及整个软硬件系统各个层面的综合性信息处理技术。

图1 从信息处理角度看大数据的主要技术层面和技术内容

应用层	大数据行业应用/服务层	电信/公安/商业/金融/遥感遥测/勘探/生物医药…… 领域应用/服务需求和计算模型
	应用开发层	分析工具/开发环境和工具/行业应用系统开发
并行化算法层	应用算法层	社会网络，排名与推荐，商业智能，自然语言处理，生物信息媒体分析检索，Web挖掘与检索，大数据分析可视化计算…
	基础算法层	并行化机器学习与数据挖掘算法
系统软件层	并行编程模型与计算框架层	并行计算模型与系统 批处理计算，流式计算，图计算，迭代计算 内存计算，混合式计算，定制式计算……
	大数据存储管理层	大数据查询（SQL，NoSQL NewSQL） 大数据存储（DFS，HBase RDFDB，MemD，RDB） 大数据采集与预处理
基础设施层	并行构架和资源平台层	集群，众核，GPU，混合式构架（如集群+众核，集群+GPU）云计算资源与支撑平台



# 目录

---

- 大数据处理技术
- **MapReduce并行计算技术**
- Hadoop系统简介



# MapReduce并行计算技术

## ■ 什么是MapReduce?

MapReduce是面向大数据并行处理的**计算模型、框架和平台**，它隐含了三层含义：

- MapReduce是一个基于集群的高性能**并行计算平台**（Cluster Infrastructure）。它允许用市场上普通的商务服务器构成一个包含数十、数百至数千个节点的分布和并行计算集群。
- MapReduce是一个**并行计算与运行软件框架**（Software Framework）。它提供了一个庞大但设计精良的并行计算软件框架，**能够自动完成计算任务的并行化处理，自动划分计算数据和计算任务，在集群节点上自动分配和执行任务，以及收集计算结果**，将数据分布存储、数据通信、容错处理等并行计算涉及到的很多**系统底层的复杂细节**交由系统负责处理。



# MapReduce并行计算技术

---

## ■ 什么是MapReduce?

MapReduce是面向大数据并行处理的**计算模型、框架和平台**，它隐含了三层含义：

- MapReduce是一个**并程序序设计模型与方法**（Programming Model & Methodology）。它遵循函数式程序设计语言Lisp的设计思想，用Map和Reduce两个函数编程（即并行编程接口）实现基本的并行计算任务。



# MapReduce并行计算技术

## ■ MapReduce的由来

- MapReduce最早由**Google公司**提出，其研发初衷主要是为了解决其搜索引擎中大规模网页数据的并行化处理。
- 2003年和2004年，Google在国际会议上发表了**GFS和MapReduce的论文**，公布了它们的基本原理和主要设计思想。
- 2004年，Lucene和Nutch的创始人**Doug Cutting**模仿Google MapReduce，开发了基于Hadoop的MapReduce。
- 随后，**Hadoop成为Apache下的顶级项目**。
- 马里兰大学的Jimmy Lin在《Data-Intensive Text Processing with MapReduce》书中写道：“MapReduce代表了第一个有别于冯诺依曼结构的计算模型，是在集群规模而非单个机器上组织大规模计算的新的抽象模型上的**第一个重大突破，是到目前为止所见到的最为成功的基于大规模计算资源的计算模型**”。



# MapReduce并行计算-MapReduce的基本设计思想

## ■ 对付大数据并行处理：分而治之

- 一个大数据集若可以分为具有同样计算过程的数据块，并且这些数据块之间不存在数据依赖关系，则提高处理速度的最好方法就是采用“分而治之”的策略。
- MapReduce就是采用以上设计思想，**对相互间不具有或有较少数据依赖关系的大数据，用一定的数据划分方法对数据进行分片，然后将每个数据分片交由一个节点去处理，最后汇总处理结果。**

## ■ 上升到抽象模型：Map与Reduce

- MapReduce借鉴了Lisp函数式语言中的思想，通过Map和Reduce函数提供高层并行编程抽象模型和接口，开发者只需要实现这两个基本接口即可完成并行化程序设计。



# MapReduce并行计算-MapReduce的基本设计思想

## ■ 上升到抽象模型：Map与Reduce

- Map操作主要负责对一组数据记录进行某种重复处理；
- Reduce操作主要负责对Map的中间结果进行某种进一步的结果整理和输出。

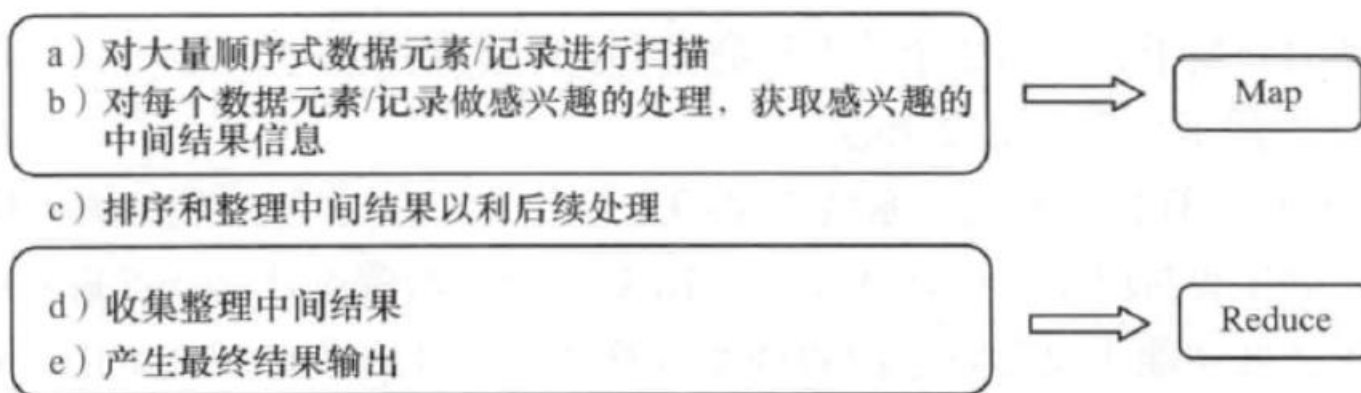


图2 典型的顺序式大数据处理过程和特征





# MapReduce并行计算-MapReduce的基本设计思想

## ■ 上升到架构：以统一架构为程序员隐藏系统层细节

MapReduce所提供的统一计算框架的主要目标是，实现自动并行化计算，为开发人员隐藏系统层细节。该框架可负责自动完成以下系统底层相关的处理：

- 计算任务的自动划分和调度
- 数据的自动化分布存储和划分
- 处理数据与计算任务的同步
- 结果数据的收集整理（sorting, combining, partitioning 等）
- 系统通信、负载平衡、计算性能优化处理
- 处理系统节点出错检验和失效恢复



# MapReduce并行计算-MapReduce的主要技术特征

- **MapReduce设计上具有以下主要技术特征**
  - 向“外”横向扩展，而非向“上”纵向扩展
  - 失效被认为是常态
  - 把处理向数据迁移
  - 顺序处理数据、避免随机访问数据
  - 为应用开发者隐藏系统层细节
  - 平滑无缝的可扩展性



# 目录

---

- 大数据处理技术
- MapReduce并行计算技术
- **Hadoop系统简介**



# Hadoop系统简介

## ■ Hadoop的发展历史

- Hadoop系统最初的源头来自于Apache Lucene项目下的搜索引擎子项目Nutch，该项目的负责人是**Doug Cutting**。
- 2003年，Google为了解决其搜索引擎中大规模Web网页数据的处理，研究发明了一套称为MapReduce的大规模数据并行处理技术。
- 2004年，在OSDI会议上发表了题为：“**MapReduce: Simplified Data Processing on Large Clusters**”的论文，简要介绍了MapReduce的基本设计思想。
- Doug Cutting发现Google MapReduce所解决的大规模搜索引擎数据处理问题，正是他同样面临并急需解决的问题。因此，他尝试模仿Google MapReduce的设计思路，用Java设计实现了一套新的MapReduce并行处理系统，并将其与Nutch分布式文件系统NDFS结合，用以支持Nutch搜索引擎的数据处理。
- 2006年，将NDFS和MapReduce从Nutch项目中分离出来，成为一套独立的软件系统，并用Doug Cutting小儿子当时牙牙学语称呼自己玩具小象的名字“Hadoop”命名了这个系统。



# Hadoop系统简介

---

## ■ Hadoop的发展历史

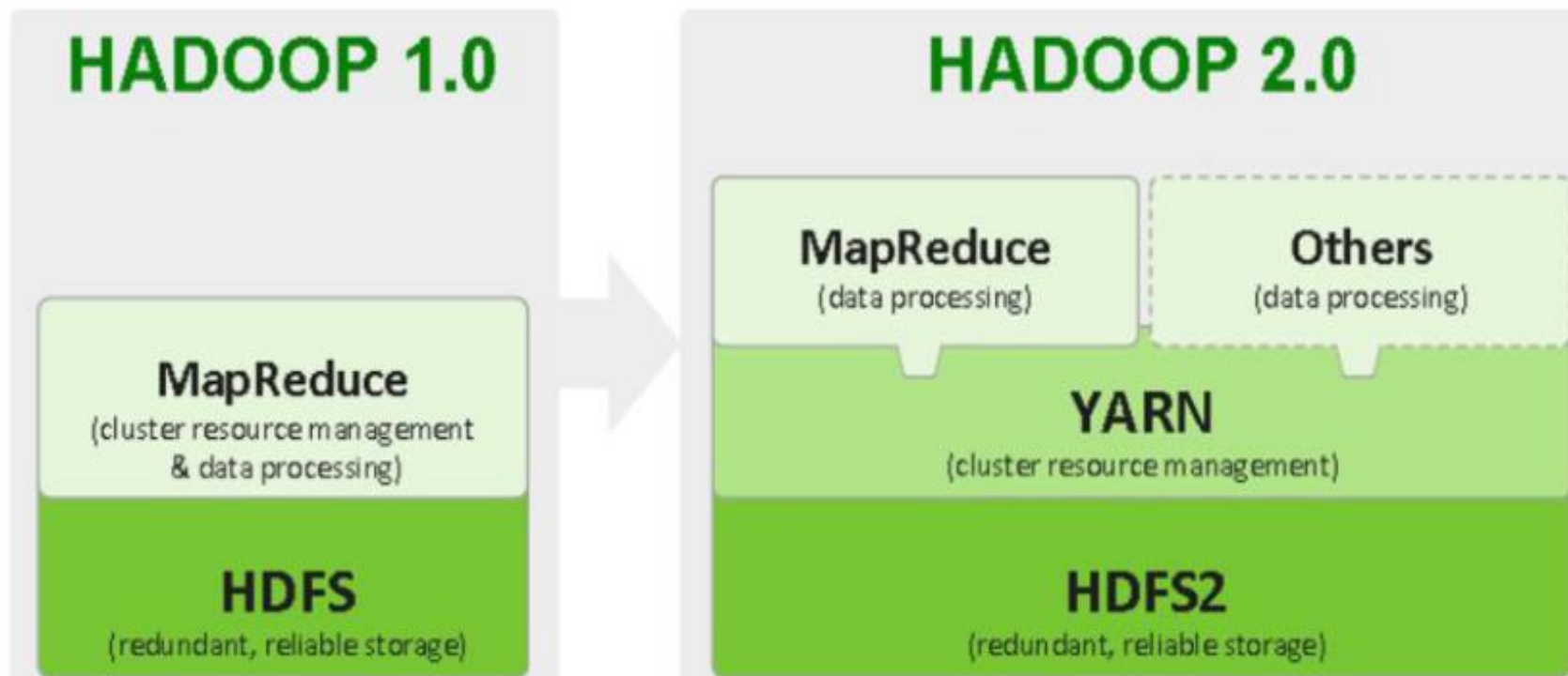
- 2008年，他们将Hadoop贡献出来，成为Apache的一个开源项目。
- 2011年12月，Hadoop在0.20.205版基础上发布了Hadoop 1.0.0版本，该版本到2012年3月发展为Hadoop 1.0.1稳定版。1.0版继续发展，到2013年8月发展为Hadoop 1.2.1稳定版。
- 2011年10月，Hadoop 0.23.0测试版发布，该版本最终演化为Hadoop 2.0版本（提出了YARN架构）。
- 2013年10月，基于YARN的Hadoop已发展出Hadoop 2.2.0稳定版。



# Hadoop系统简介

## ■ Hadoop的发展历史

### ■ 从Hadoop 1.0到Hadoop 2.0

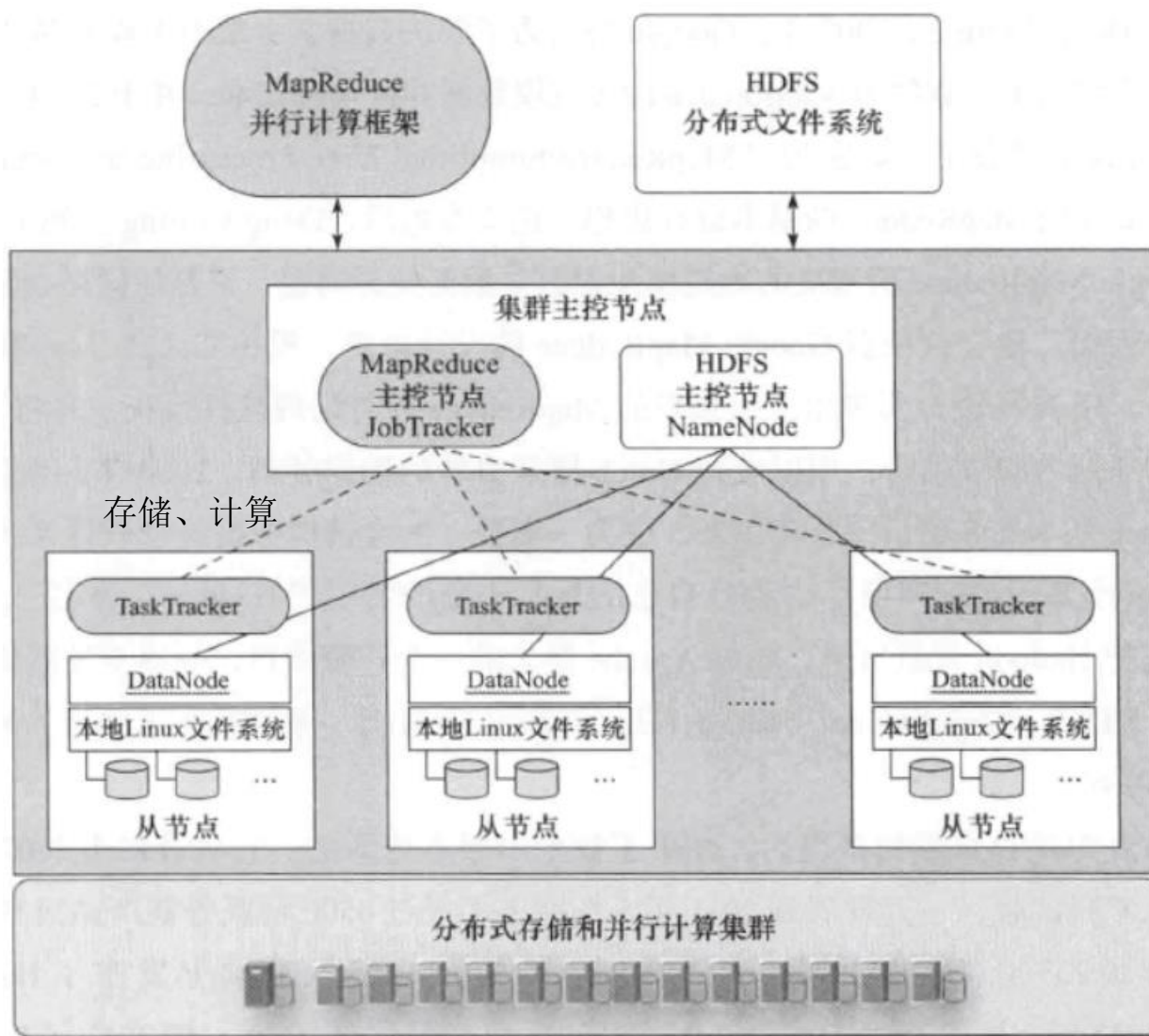


<http://www.89cool.com/801.html>



# Hadoop系统简介

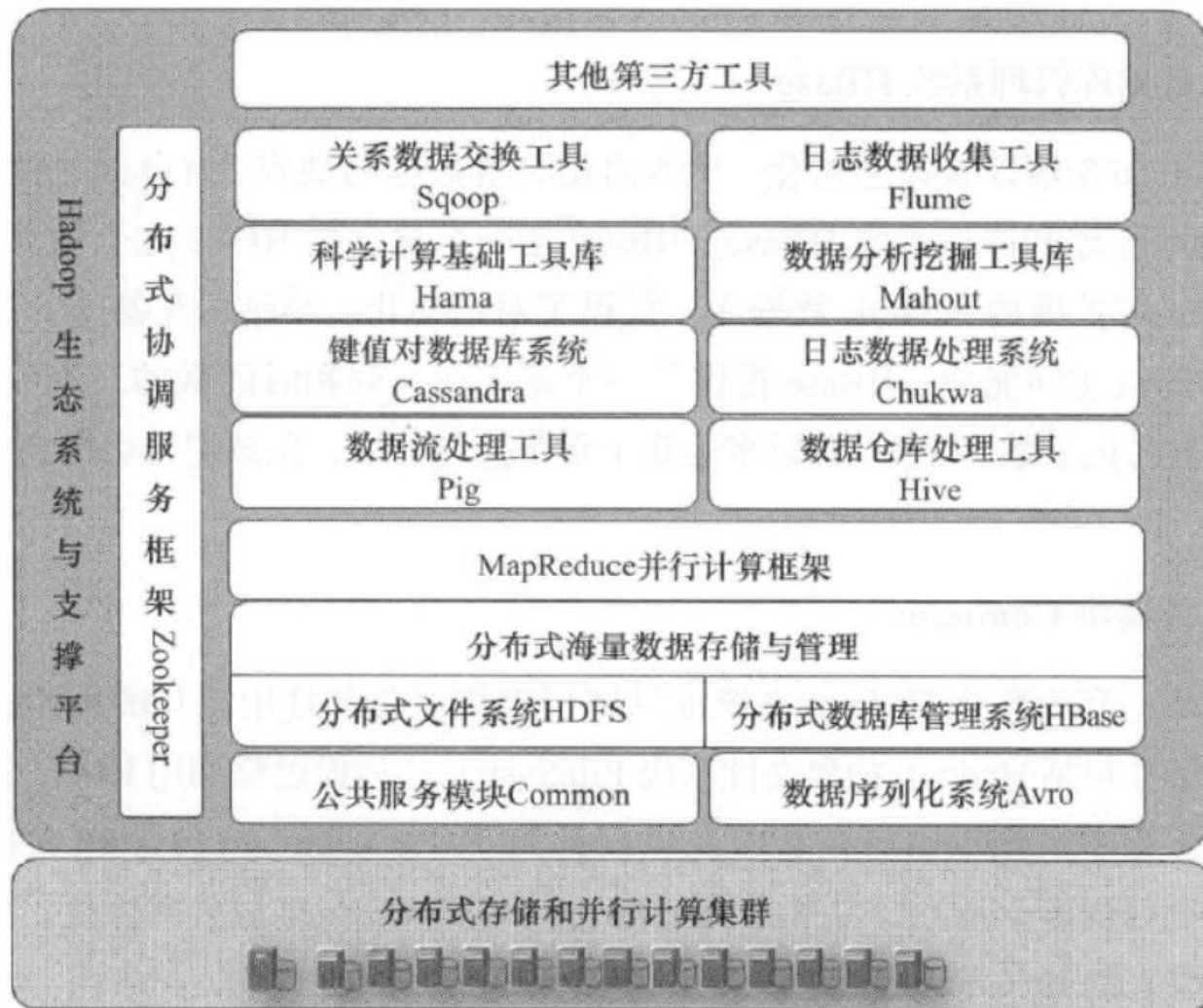
## ■ Hadoop系统分布式存储与并行计算架构





# Hadoop系统简介

## ■ Hadoop平台的基本组成与生态系统





# Hadoop平台的基本组成与生态系统

## 1. MapReduce并行计算框架

- MapReduce并行计算框架是一个并行化程序执行系统，它提供一套并行化编程模型和接口。
- MapReduce包括**Map和Reduce两个阶段**。
- MapReduce**以键值对方式**来处理数据，并能**自动完成**数据划分，调度程序的执行，监控程序的运行状态，负责程序执行时各计算节点的同步以及中间结果的收集整理。

## 2. 分布式文件系统HDFS

- HDFS (Hadoop Distributed File System) 是一个类似于Google GFS的开源的分布式文件系统。
- 基于物理上分布在各个数据存储节点的本地Linux文件系统，向上层应用程序提供一个**逻辑上**成为整体的大规模数据存储文件系统。
- 可扩展、高可靠、高可用。例如，多副本（默认：3个）、数据出错检验和恢复机制。



# Hadoop平台的基本组成与生态系统

## 3. 分布式数据库管理系统HBase

- 为了克服HDFS难以管理结构化/半结构化海量数据的缺点。
- 建立在HDFS之上的分布式可扩展NoSQL数据库。
- 提供了一个基于行、列和时间戳的三维数据管理模型。HBase中每张表的记录数（行数）可以多达几十亿甚至更多，每条记录可以拥有多达上百万的字段。

## 4. 公共服务模块Common

- Common是一套为Hadoop系统提供**底层支撑服务**和**常用工具**的类库和API编程接口。
- 在Hadoop 0.2及以前的版本中，Common包含HDFS、MapReduce和其它公共的项目内容；从0.21版本开始，HDFS和MapReduce被分离为独立的子项目，其余部分内容构成Hadoop Common。



# Hadoop平台的基本组成与生态系统

## 5. 数据序列化系统Avro

- Avro是一个序列化系统，用于将数据结构或数据对象转换成便于存储和网络传输的格式。

## 6. 分布式协调服务框架Zookeeper

- Zookeeper是一个分布式协调服务框架，主要用于**解决分布式环境中的一致性问题**。
- 可靠性维护、数据状态同步、统一命名服务、分布式应用配置项管理等功能。
- 在分布式环境下，维护系统运行管理中的一些数据量不大的重要状态数据，并提供检测数据状态变化的机制。

## 7. 分布式数据仓库处理工具Hive

- Hive是一个建立在Hadoop之上的数据仓库，用于管理存储在HDFS或HBase中的结构化/半结构化数据。
- 2008年Facebook将Hive贡献给Apache，成为Hadoop开源项目。
- 最主要的目的是为了熟悉SQL的数据库使用者能够基于Hadoop系统进行数据查询分析（HiveSQL->MapReduce）。



# Hadoop平台的基本组成与生态系统

## 8. 数据流处理工具Pig

- Pig是Yahoo!贡献给Apache成为开源项目。
- 提供了一个面向领域的高层抽象语言Pig Latin，将复杂的数据分析任务转换为Pig操作上的数据流脚本（Pig数据流脚本->MapReduce任务链）。
- Yahoo!上有大量MapReduce作业是通过Pig实现的。

## 9. 键值对数据库系统Cassandra

- Cassandra是一套基于Key-Value的分布式数据库系统。
- Cassandra最初由Facebook开发，后被贡献给Apache成为开源项目。
- Cassandra以Dynamo为基础，结合Google BigTable（Column Family）数据模型，提供了一套高度可扩展、最终一致、分布式的结构化键值存储系统。



# Hadoop平台的基本组成与生态系统

## 10. 日志数据处理系统Chukwa

- Chukwa是一个由Yahoo! 贡献的开源数据收集系统，主要用于日志的收集和数据的监控，并与MapReduce协调处理数据。
- Chukwa使用HDFS来存储数据，MapReduce来处理数据，以及灵活强大的辅助工具用以分析、显示、监视数据结果。

## 11. 科学计算基础工具库Hama

- Hama是一个基于BSP并行计算模型（Bulk Synchronous Parallel）的计算框架，用于支持**大规模科学或具有复杂数据关联性的图计算**。
- 矩阵计算、排序、PageRank、BFS等。

## 12. 数据分析挖掘工具库Mahout

- Mahout来源于Apache Lucene子项目，其主要目标是创建并提供经典的机器学习和数据挖掘并行化算法类库。
- Mahout包含了分类、聚类、推荐、频繁项集挖掘等广泛使用的算法。



# Hadoop平台的基本组成与生态系统

## 13. 关系数据交换工具Sqoop

- Sqoop (SQL-to-Hadoop)是一个在**关系型数据库和Hadoop平台间进行快速批量数据交换**的工具。
- Sqoop可以将一个关系型数据库中的数据批量导入Hadoop的HDFS、HBase、Hive中，也可以反过来将Hadoop平台中的数据导入关系型数据库中。

## 14. 日志数据收集工具Flume

- Flume是由Cloudera开发并维护的一个分布式、高可靠、高可用、适合复杂环境的大规模日志数据采集系统。
- Flume将数据从产生、传输、处理、输出的过程抽象为数据流，并允许在数据源中定义数据的发送方，从而支持收集基于不同传输协议的数据，并提供对日志数据的过滤、格式转换等处理。

