# 大数据技术复习

## 第一章、大数据技术概述

#### 1.大数据的概念、特征以及意义

（1）大数据的发展历程：

第一阶段（萌芽期）：上世纪90年代至本世纪初。

第二阶段（成熟期）：本世纪前十年（2000-2010）

第三阶段（大规模应用期）：2010年之后

1. 大数据的概念：大数据是指**无法在一定时间范围内用常规软件工具**进行捕捉、管理和处理的**数据集合**，是需要新处理模式才能具有更强的决策力和流程优化能力来适应海量、高增长率和多样化的**信息资产**。大数据技术的出现**主要解决海量数据的存储和分析计算问题**。
2. 大数据的特征（5V）：**大量（Volume）**、**数据类型多样（Variety）**、**处理速度快（Velocity）**、**价值密度低（Value）**、**真实性（Veracity）。**
3. 按数据格式划分大数据的数据类型：

**结构化信息**：如传统的关系型数据库，常用二维表表示。

**半结构化信息**：类似XML、HTML之类，自描述，数据结构和内容混杂在一起。

**非结构化信息**：例如各种文档、图片、视频/音频等。

1. 研究大数据的意义（**我估计了解一哈就行了**）：
   * 1. 大数据计算提高数据处理效率，增加人类认知盈余。
     2. 全局的大数据让人类了解事物背后的真相。有助于了解事物发展的客观规律，利于科学决策。
     3. 提供了同事物的连接，客观了解人类行为。
     4. 改变过去的经验思维，帮助人们建立数据思维。

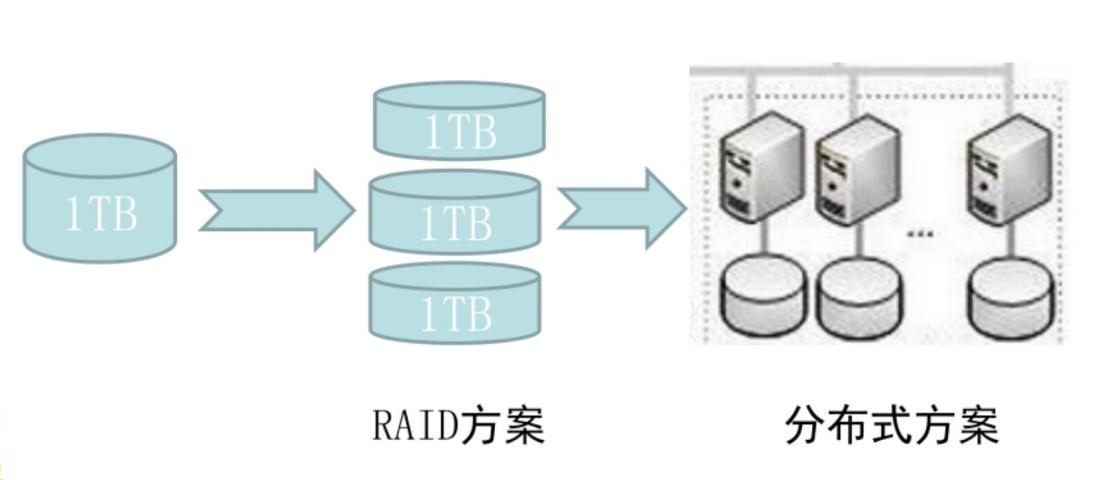
#### **大数据时代的思维变革**

**传统思维：因果逻辑思维，来自长久经验下的推理结论。**

1. 大数据的思维方式（**记关键字就行，至于解释由于很多说法不一到时候胡扯一点就行**）：
   * 1. **全样而非抽样**：数据太多，无法保存和分析，统计学采用抽样其结果误差放到全样上该误差会被放大。
     2. **效率而非精明**：大数据基础上的简单算法比小数据基础上的复杂算法更加有效。
     3. **相关而非因果**：在关联关系中，如果我们在知道A与B经常一起发生那么当B发生时，我们可以猜测A也发生了。
     4. **以数据为中心**：大数据技术的发展是以数据作为基本重要发展条件之一。
     5. **我为人人，人人为我**：在使用大数据提供的信息的时候，我们也可以将实时信息反馈回去，更新大数据的认识从而帮助其他人获取新的并且正确的大数据信息。（例如我们使用高德地图时我们将自己的实时信息反馈给服务端使得服务端对道路信息进行完善和更新）

#### **大数据的存储与计算模式**

大数据在存储方面所面对的问题：

1. 存储规模大，通常达到PB甚至是EB量级。（**大数据存储面临的主要问题：存储容量问题与读取速率问题**）
2. 管理复杂，需要兼容结构化、非结构化以及半结构化的数据。
3. 数据服务的种类和水平要求高。
4. 大数据存储面临的解决方法**：同时从多个硬盘上进行数据的读取与存储（即分布式储）**
5. 分布式存储问题与解决方法：

问题1：多个硬盘存储数据的时候难免会造成部分的硬盘损毁从而导致数据的丢失。

解决办法：进行**数据的复制**。（Hadoop中的HDFS便是采用这种方式）

问题2：在面对大多数分析需要结合大部分数据共同完成，面对不同的数据源其数据的正确性需要得到保证。

解决方法：MapReduce中有个编程模型抽象出这些硬盘读写问题并将其转换成对一个数据集的计算，**计算之前对数据集进行校验**。

1. 分布式数据库分类说明：

事务性数据库技术：采用NoSQL技术，主要包括BigTable、HBase、MongoDB

分析型数据库技术：针对Hadoop的SQL分析引擎，主要包括Hive，Impala等。

1. 大数据的计算模式：大数据计算模式就是指根据大数据的**不同数据特征**和**计算特征**，从多样性的大数据计算问题和需求中提炼并建立的各种抽象或模型。
2. 大数据计算的关键技术（了解一下就行）：

①批处理计算模式：主要用于针对大规模数据的批量处理，代表产品：**MapReduce**、**Spark**

②流计算：主要用于针对流数据的实现计算，代表产品：Storm、S4、Flume、Strems、Puma、DStream、Super Mario 银河流数据处理平台

③图计算：针对大规模图结构数据的处理，代表产品：PwerGraphx

④查询分析计算：大规模数据的存储管理和查询分析，代表产品：**HBase**、**Hive**

1. 大数据两大核心技术：**分布式存储（例如：HDFS）与分布式处理（MapReduce、Spark）等。**

#### 作业部分（根据上次考试经验来看这个部分一定要记住，我的答案不一定准确，记住自己的答案就行）

（1）说出大数据与人工智能的联系

答大数据的出现是人工智能发展的一个不可缺少的前提条件，人共智能的发展离不开大数据的支持，可以说二者相辅相成，互相促进发展，人工智能需要海量的数据作为思考决策的基础。

（2）简述一个或者多个大数据的应用案例

答：①医疗行业：Seton Healthcare 是采用IBM最新沃森医疗包间内容分析预测的首个客户，该技术允许企业找到大量病人相关的临床医疗信息，通过大数据处理，更好地分析病人的信。

②保险行业：保险行业并非技术创新的指示灯，然而MetLife保险公司已经投资3亿美金建立一个新式系统其中的第一款产品是一个基于MongoDB的应用程序，它将所有的客户信息放在同一个地方。

（3）**大数据的含义以及特征**

答：大数据的含义：大数据是指在一定时间范围内无法用常规工具进行捕捉、管理、和处理的数据集合，是需要新处理模式才能具有更强的决策力、洞察发现力、洞察发现力和流程优化能力来适应海量、高增长和多样化的信息资产。

大数据的特征（5V）：数据量大、数据种类繁多、处理速度快、价值密度低、真实性

1. **大数据的思维方式包括哪些内容**：

答：①全样而非抽样②效率而非精确③相关而非因果④以数据为中心⑤我为人人，人人为我

1. **大数据不能做什么**

答：①无法替代决策力②不能替代有效的商业模式③不能无目的发现知识④不能替代专家⑤不能一次建模终身受益

1. **谈谈大数据、云计算、物联网三者关系**

答：三者相辅相成云计算是大数据与物联网发展的基础，根据市场数据统计可知，大数据技术需要将长期发生，记录下来的海量数据进行分析，才能得出具有价值的结论。但是巨量的数据需要分析就不得不需要更强大的计算资源，否则同样多的数据如果采用计算能力较为低的服务器进行处理那么就有很大可能会失去市场竞争力，故此云计算为大数据的发展提供了莫大的支持，除此之外未来物联网的发展将万物互联，这之中产生的数据量将会越来越大，如果靠老一套的计算技术将不能适配这一切，然而大数据的分析工作只能交给云计算来解决才能最大化地实现其价值，故此我的出的结论是云计算是大数据与物联网发展的一个必然条件和基础。

1. **大数据中的计算模式有哪些**？

答：①批计算模式：用于解决大规模数据批量处理

②流计算模式：针对流数据的实时计算

③图计算模式：针对大规模图结构的计算

④查询分析计算模式：大规模数据的存储管理和查询分析

1. **大数据分析流程包含那几个步骤？**

答：①**数据获取**②**数据清洗**③**数据管理**④**数据分析**⑤**数据呈现**

## 数据采集与预处理

#### 大数据分析流程

**数据分析流程大致分为如下**：**数据的获取**——》**数据的清洗**——》**数据管理**——》**数据分析**——》**数据呈现** 五个步骤。

数据获取：指从生活环境之中或是互联网之中获取的数据、图片、文本等信息。

数据清洗：补充部分数据缺失的属性值，或是统一数据格式、编码和度量监测与删除异常数据等。

数据管理：对数据进行分类、编码、存储、索引和查询等

数据分析：一般的统计与查询。对数据进行深层次的理解

数据呈现：建立从输入数据到符合认知规律的可视化表征，使得用户可以交互的视图。

#### 大数据的采集

大数据采集：又称“数据获取”，是数据分析的入口，也是数据分析过程中相当重要的一个环节，他通过各种技术手段把外部各种数据源产生的数据实时地采集并加以利用。

（1）数据采集的三大特点：**全面性**、**多维性**、**高效性**

（2）数据采集的数据源：传感器数据、互联网数据、日志文件、企业业务系统数据

（3）数据采集的方法：系统日志采集、分布式消息订阅分发、ETL、网络数据采集

（4）网络数据采集：网络数据采集是指通过**网络爬虫或网站公开应用程序编程接口**等方式从网站上获取数据信息。

#### 大数据预处理

**大数据预处理的说明与处理方式**：数据预处理是指在数据进行分析挖掘之前，对原始数据进行转换、清洗与集成等一系列操作。通过数据预处理工作，可以使残缺的数据完整，并将错误的数据纠正、多余的数据祛除，有效提高质量。

1. 预处理的重要性：**没有高质量的数据就没有高质量的数据挖掘结果。所以目的是可以提高数据的质量**。
2. 数据清洗的内容：1.缺失值的处理、2.异常值的处理、3.数据类型转换、4.重复值处理

#### 大数据集成

数据集成是把不同来源格式的数据有机地集中起来，通过一致的、精确的表示法，对同一种实体对象的不同数据做整合的过程。

1. **数据继承解决的主要问题**：**实体识别**、**冗余问题**、**数据值冲突的检测与处理**。

#### 作业部分

1. 额呃呃呃貌似没有这一部分的作业，说明这一部分基本以理论为主看看就是了

## Hadoop大数据平台

#### Hadoop的发展过程

（1）**Hadoop的概念**：Hadoop是一个**开源分布式计算平台**，在分布式环境下提供了海量数据的处理能力，能在大量廉价计算机组成的集群中完成海量数据的存储与计算。Hadoop中提供了一个**共享存储系统（HDFS）**与**分析系统（MapReduce**）实现数据的分析与处理。

#### Hadoop的优势（重点内容）

1. **高可靠性**：Hadoop中HDFS分布式系统采用了**备份恢复机制**以及MapReduce中的任务**监控机制**，保证了分布式处理的可靠性。
2. **高容错性**：Hadoop采用自动保存数据的多个副本方式，并能自动为失败的任务进行重新分配。
3. **高可扩展性**：Hadoop是在可用的计算机集群间进行数据的分配与计算，这些集群能扩展到数以千计的节点中。
4. **高效性**：表现在Hadoop能够在节点之间进行动态的移动数据。同时能保证个个节点的数据动态平衡。使得数据时速度非常快。

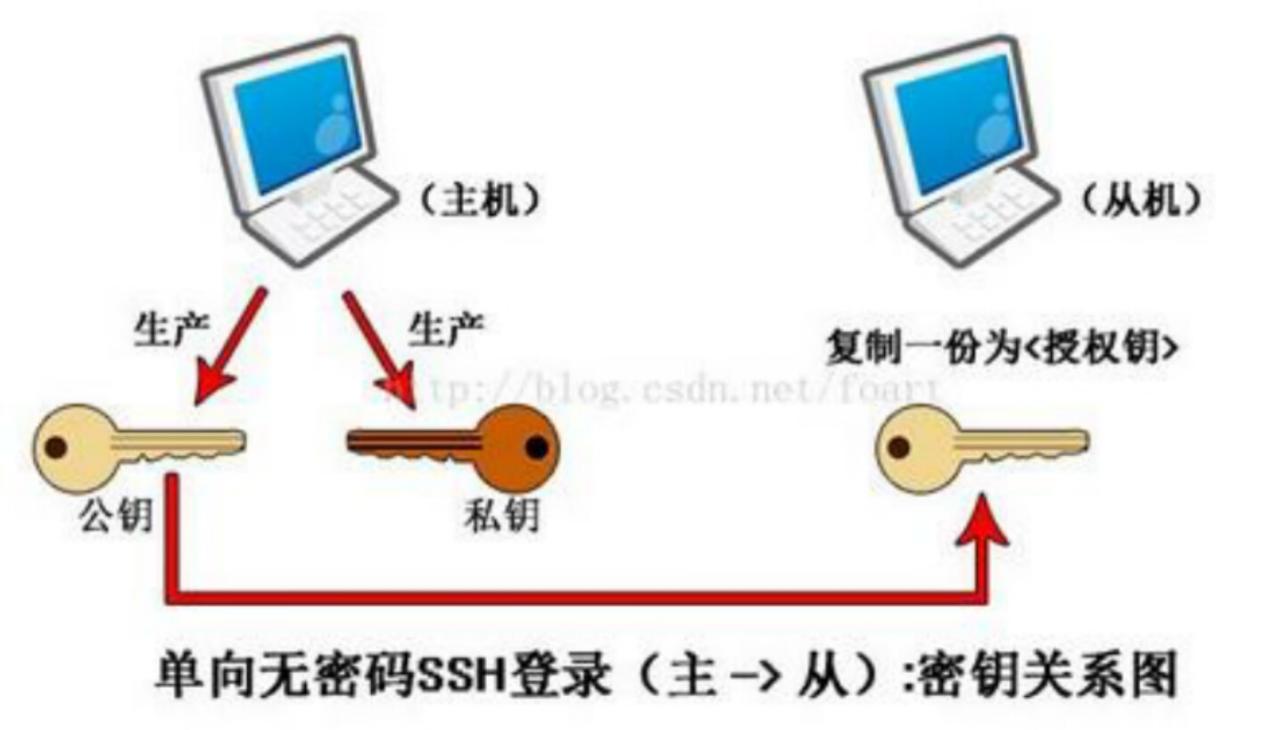
#### Hadoop的生态系统

Hadoop是由许多子项目组成，其中**HDFS**和**MapReduce**是两个最基础、最重要的成员。

1. Hadoop中的成员介绍：
   * 1. **HDFS**：分布式存储系统，HDFS提供了高可靠性、高扩展性和高吞吐率的数据存储服务，是GFS克隆版。
     2. **MapRdduce**：分布式计算框架，具有易于编程、高容错性和高扩展性等优点。
     3. **YARN**：Hadoop的资源管理系统、负责集群资源的统一管理和调度，是Hadoop2.0新增加的系统组件，它允许多种计算框架运行在一个集群中。
     4. **Tez**：运行在YARN之上的下一代Hadoop查询和处理框架。
     5. **Spark**：类似于Hadoop MapReduce的通用并行框架，基于内存实现。
     6. **Hive**：基于MapReduce的数据仓库有着独特的类SQL查询语言——HQL。
     7. **Pig**：构建在Hadoop之上的数据仓库，极简化了Hadoop常见的工作任务。
     8. **Oozie**：作业流调度系统，可以把多个MapReduce作业组合到一个逻辑单元中。
     9. **HBase**：一种分布式数据库。
     10. **Zookeeper**：一个分布式协作服务组件，解决分布式环境下数据管理问题。
     11. **Sqoop**：数据同步工具，是连接Hadoop与传统数据库之间的桥梁。
     12. **Flume**：日志搜集工具。

#### Hadoop的安装与配置（省略）

1. SSH配置：一种安全协议防止远程管理过程中的信息泄露问题。默认使用RSA算法创建公钥/私钥对并让整个Hadoop集群共享该秘钥对。



SSH使用流程：由远程主机收到用户从机发送登录请求连接，主机将自己的公钥发送给用户，用户使用这个公钥将登录密码加密后发送回来。远程主机使用自己的私钥在对这个密码进行解密，如果校验正确那么统一对方的登录。

#### 作业部分

1. 华为大数据平台的名字：**FusionlnSight**。
2. **简述SSH工作原理**：

答：SSH用于两台机器之间传输数据的时候保证安全所用的，其采用的是非对称加密的方式，所谓非对称加密方式是指为了保证密文的安全性使用公钥对其进行加密，若要解密则需要对应的私钥进行解密，这就达成了密文不易泄露的目的。其运作流程如下：①远程主机收到用户登录请求，将自己的公钥发给用户。②用户使用这个公钥将登录密码加密后发送回来。③远程主机用自己的私钥再对这个密码进行解密如果密码校验正确则同意对方的登录。

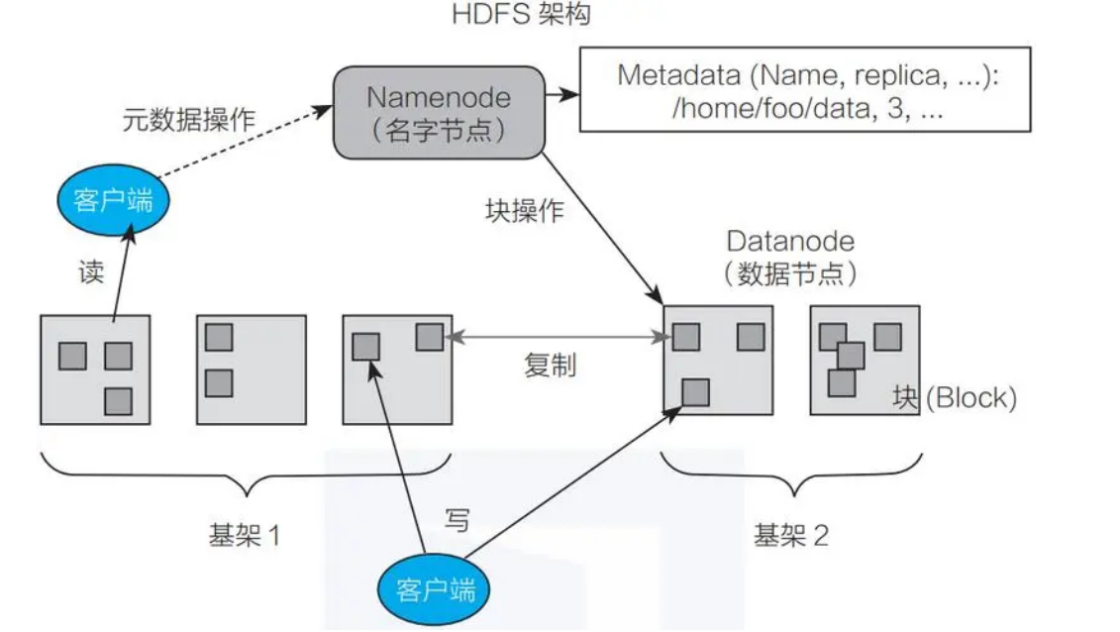
1. 简述Hadoop的生态系统：

答：Hadoop是一个用在通用计算机设备上组成干的大型集群上执行分布式应用的框架，其由诸多子项目组成，其中HDFS与MapReduce是两个最基础、最重要的成员。HDFS提供了高可靠性、高扩展性以及高吞吐量的数据存储服务。MapReduce是一个分布式计算框架，具有易于编程、高容错性和高扩展性等优点。此外还有YARN：这是Hadoop的资源管理系统，负责集群资源的统一管理与调度。Hive基于MapReduce的数据仓库。Pig:构建在Hadoop之上的数据仓库，用于简化Hadoop的常见工作任务。Zookeeper:用于解决分布式环境下的数据管理问题，包括统一命名、状态、同步等集群管理问题。Sqoop是数据同步工具，是连接Hadoop与传统数据库之间的桥梁。Flume：日志收集工具。

## （重点）Hadoop分布式文件系统

#### HDFS概述

**HDFS是一个主从（master/slave）结构模型，一个HDFS集群是由一个NameNode和若干个DataNode组成的。**



**NameNode**：主节点（元数据节点），**管理文件系统的命名**和**客户端对文件的访问操作**。

**DataNode**：数据节点，集群中的一般节点，**负责节点数据的存储**。

数据块：每个磁盘都有默认的数据块大小，这是磁盘进行数据读/写的最小单位。普通文件系统通常以簇为单位分配磁盘空间，一个簇就是一个数据块。HDFS支持文件的“**一次写入，多次读取**”。

1. **HDFS采用抽象的块概念可以带来以下几个明显的好处**：**1、支持大规模的文件存储2、简化系统设计3、适合数据备份**。
2. **HDFS主要组件的功能**：

**NameNode**：①**存储元数据**②**元数据保存在内存中**③**保存文件，block，DataNode之间的映射关系。**

**DataNode**：①**存储文件内容**②**文件内容保存在磁盘**③**维护了block id到DataNode本地文件的映射关系**。

1. **元数据节点（NameNode）**：元数据节点的作用是管理分布式文件系统的命名空间，并将所有的文件和目录的元数据保存在Linux本地文件系统的目录中。
2. **元数据（Metadata）**：元数据就是指数据的数据，HDFS的元数据就是指维护HDFS文件系统中文件和目录所需要的信息。从形式上讲元数据可分为**内存元数据**与**元数据文件**两种。
3. 辅助元数据节点（**Secondary NameNode**）：辅助元数据**并非**NameNode出现问题时的**备用节点**，它负责周期性地将NameNode的命名空间镜像文件和修改日志合并，以防日志文件过大。它会保存合并后的文件以防**NameNode**失效时进行恢复。
4. **数据节点（DataNode）**：数据节点的**作用是保存HDFS文件的数据内容**，当客户端想HDFS写入文件的时候，NameNode将这些数据块的存储任务指派给不同的DataNode。每一个**DataNode在接受到任务后会直接从客户端接收数据**。DataNode周期性地向NameNode发送其存储的数据快的信息。每个数据节点的数据会被保存到各自节点的本地Linux文件系统中。
5. HDFS特点：HDFS与其他分布式文件系统一个明显的不同之处是，HDFS采用“**一次写入，多次读取**”模型，该模型降低了并发控制的要求能支持高吞吐量的访问。
6. **HDFS的优点**：

①**支持超大文件的存储**

②**支持流式的访问数据**

③**运行于廉价的商用机器集群**。

（9）**HDFS的缺点**：1.**不适合低延迟数据访问**2.**无法高效存储大量的小文件**3.**不支持多用户写入和任意修改文件**。

#### HDFS的体系结构

1. **HDFS设计目标**：

1.**能检测和快速恢复硬件故障** 2.**支持流式的数据访问**3.**支持超大规模的数据集**4.**简化一致性模型**5.**移动计算逻辑代价比移动数据代价低**。6.**具备良好的异构软硬件平台间的可移植性**。

（2）**HDFS的结构模型**：HDFS采用主从架构。一个HDFS集群包含一个**NameNode**（用于主控制服务器，管理文件系统名称空间并规范客户端对文件的访问）；多个**DataNode**节点（数据节点将数据以块为单位进行存储，一般一个物理节点上部署一个数据节点）

（3）HDFS结构的局限性：

①**命名空间的限制**：元数据在内存中能够容纳的对象的个数收到内存空间大小的限制。

②**性能的瓶颈**：整个分布式文件系统的吞吐量，受限于单个名称节点的吞吐量。

③**隔离问题**：由于集群中只有一个元数据节点与一个命名空间，无法对不同应用程序进行隔离。

④**集群的可用性**：一旦这个唯一的元数据节点发生故障，会导致整个集群变得不可用。

（4）**HDFS数据读取**：读文件时用户端会**向NameNode发送数据读操作请求**，并通过NameNode获得组成该文件的数据块的位置列表，然后客户端直接从这些DataNode读取文件数据，**在读数据过程中NameNode不参与文件的传输**，

（5）HDFS的数据写入：首先**向NameNode发送数据“写”操作与操作请求，包括文件名和目录路径等部分元数据信息**，然后，NameNode告诉客户端到哪个机架（Rack）的哪个DataNode进行具体的数据写入操作。最后，客户端直接将文件数据纯属给DataNode，**在数据写入过程中NameNode节点不参与文件的传输**。

（6）HDFS高可靠性（HA）：**存在的目的是为了解决单点故障问题**，当活跃点（Active）出现故障的时候，将立即切换到待命元数据节点状态（Standby），使得

（7）HDFS联邦（Federation）：在HDFS Federation之中设计了多个相互独立的元数据节点，使用HDFS的命名服务能够水平扩展，这些元数据节点分别进行各自命名空间和块的管理，相互之间是联盟（Federation）关系，不需要彼此协调。在HDFS Federation中，所有元数据节点会共享底层的数据节点存储资源，数据节点向所有的元数据节点汇报，属于同一个命名空间的块构成一个“块池”。

#### HDFS Shell操作（看PPT吧全是代码）

#### HDFS API编程（还是看PPT吧代码太多不想复制）

## 作业部分

1. **简述NameNode、DataNode、Secondary NameNode的功能**？

答：NameNode:用于管理分布式文件系统的命名空间，即用于存储数据的元数据的。  
DataNode:用于存储HDFS数据文件的内容，并周期性地想NameNode发送其存储的数据块信息。  
Secondary NameNode:辅助元数据，负责周期性地将NameNode的命名空间镜像文件和修改日志合并，以防止日志文件过大，会保存合并后的文件以防NameNode失效时进行恢复。

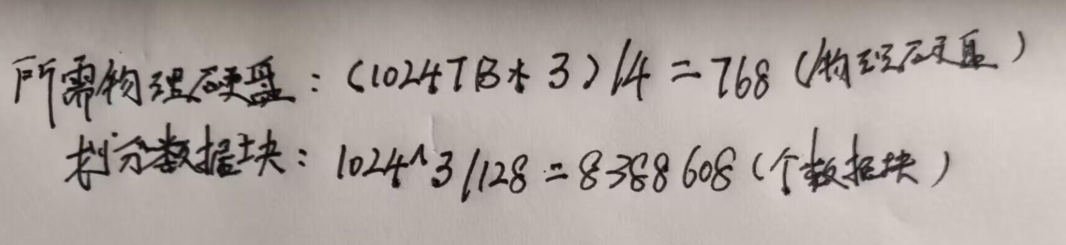
1. **HDFS的优点？**

答：①**支持超大文件的存储，通常直属局在TB级以上的文件。**

②**支持流式的访问数据，“一次写入多次读取”**

③**运用廉价的计算机集群**

1. 假设HDFS数据块为128MB，HDFS副本系数为3，现在需要设计一个HDFS存储系统来存储1PB的视频文件，若所用物理硬盘的容量统一为4TB，则至少需要多少个物理硬盘？该视频文件再上传至HDFS时，市集划分了多少个数据块？



### 简述HDFS的HA和Federation

HA：HDFS中的HA是高可靠性机制，其存在的**目的是为了解决单点故障问题**，在HA集群之中设置了两个元数据节点，“**活跃（Active）与待命（Standby**）”当两个状态同步的时候可以借助一个共享存储系统来实现，**一旦活跃元数据节点（Active）出现问题就立即切换到待命元数据节点。**

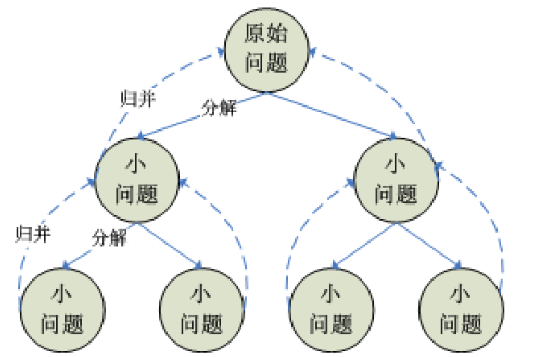
1. **简述HDFS体系结构的局限性**
   1. **命名空间的限制**：元数据在内存中能够容纳的对象的个数收到内存空间大小的限制。
   2. **性能的瓶颈**：整个分布式文件系统的吞吐量，受限于单个名称节点的吞吐量。
   3. **隔离问题**：集群中只有一个元数据节点与一个命名空间，无法对不同应用程序进行隔离。
   4. **集群的可用性**：一旦这个唯一的元数据节点发生故障，会导致整个集群变得不可用。

## Hadoop的计算框架

#### MapReduce概述

**MapReduce介绍**：是**一种处理海量数据的并行编程模型和计算框架**，**用于对大规模数据集的并行计算。**

1. 相对于传统MapReduce的解决方案：MapReduce采用**分而治之**的思想。



**分而治之的步骤（简单了解）：**

第一步**、**进行**数据预处理**。

第二步、**生成初始key-value键值对**。

第三步、**映射转换**

第四步、进行**合并映射结果并排序**

第五步、**归于最终结果**

1. **MapReduce**的基本概念：MapReduce包括两部分操作：**Map**与**Reduce**。

**Map**：**将一组数据一对一映射为另外一组数据**，起映射的规则由一个函数来指定。

如：对[1,2,3,4]进行乘2的映射就变成了[2,4,6,8]

**Reduce**：**是对一组数据进行归约**，归约的规则由一个函数来指定。

如：对[1,2,3,4,5]进行求和的归约得到结果是15，而对它进行秋季的归约的结果是120。

**总结**：**Map负责把任务分解成的多个任务**，**Reduce负责把分解后多任务处理的结果汇总起来**。

1. **MapReduce v1的运行机制**：**参与MapReduce作业执行涉及4个独立的实体。**

①**客户端**：编写MapReduce程序，配置作业并提交作业。

②**TaskTracker**：保持与JobTracker的通信，在分配的数据片段上执行任务。

③**JobTracker**：初始化作业，与Tracker通信，协调整个作业的执行。

④**HDFS**：保存作业数据、配置信息以及计算结果等。

1. MapReduce v1的局限性（了解即可，重点是YARN）：

①**单点故障**：所有Job的完成都依赖与JobTracker的调度和分配，一旦此节点宕机就意味着整个平台的瘫痪。

②**可扩展性差**：随着节点数量的增加，JobTracker的任务就会越来越多，官方给出了限制节点数。

③**资源浪费**：对于每一个job，都是通过节点资源的数量进行分配，显然这种分配方式不能动态的实现负载均衡。

④**版本耦合**：MapReduce v1框架如果有变更，不管用户是否同意，都会强制用户进行系统级别的升级。

（5）JobTracker在MapReduce中的详细职责：①管理集群中的计算资源②协调集群中的运行任务。

#### YARN运行机制

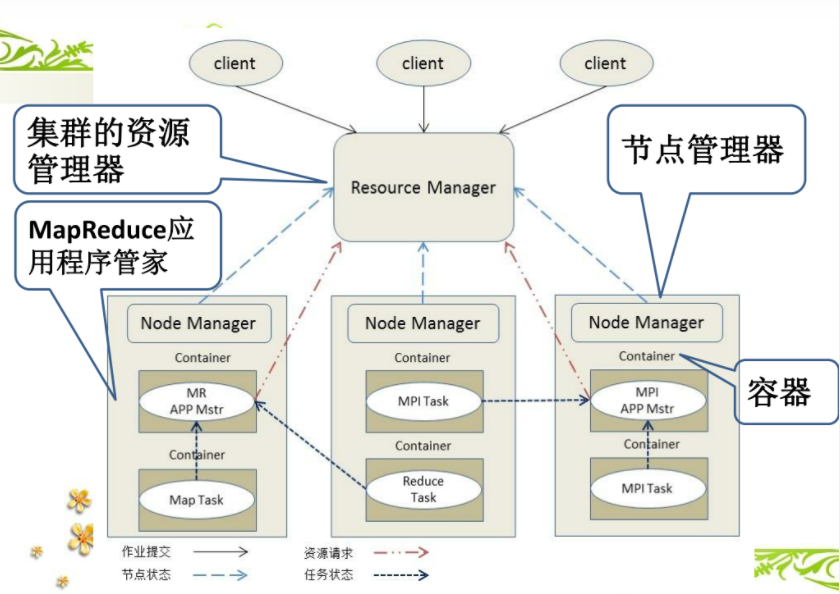
**YARN是一种新的MapReduce架构，也是一种新的Hadoop资源管理器**，它是一个通用资源管理系统，可为上层应用提供统一的资源管理和调度。将之前JobTracker的职责更改为整个集群的资源管理和分配，在YARN中，名称变为**ResourceManager（资源管理器）**，而协调任务则交给了**ApplicationMaster（应用程序管家）**。

ResourceManager将可用的集群资源分配给各个应用程序。在用户提交一个应用程序的时候ApplicationMaster实例会启动以协调应用程序内的所有任务的执行。

新的YARN解决了ManReduce v1中的哪些问题？

①更高的集群利用率 ②可扩展性的提高 ③在YARN中通过加入ApplicationMaster这个可变更的部分，用户可以针对不同的编程模型编写自己的ApplicationMaster。 ④JobTracker一个很大的负担就是监控Job的tasks运行情况，使用YARN后这个部分下方的到了AoolicationMater。

1. YARN的组成结构：YARN总体上来说还是一个**主从结构（master/slave）**，在整个资源管理框架之中，ResourceManager充当着master的角色，NodeManager为slave，ResourceManager负责对各个NodeManager上的资源进行统一管理和调度。



1. ResourceManager介绍：ResourceManager是一个全局的资源管理器，负责整个系统的资源管理与分配，它主要由两个组件构成，调度器和应用程序管理器。

①调度器：根据容量、队列等限制条件，将系统中的资源分配给各个正在运行的应用程序。

②应用程序管理器：负责整个系统中所有的应用程序。

1. ApplicationMaster：ApplicationMaster管理在YARN中运行的每个应用程序实例。它负责向ResourceManager申请资源，并通过NodeManager监视容器的执行和资源的使用情况，其功能如下：

①与ResourceManager调度器协商以获取资源。

②将得到的作业进一步划分为多个任务

③与NodeManager通信以启动/停止任务

④监控所有任务运行状态，并在任务运行失败时从新为任务申请资源以重启任务。

1. NodeManager：NodeManager管理YARN集群中的每个节点，它提供了针对急群众每个节点的服务，从管理一个容器的生命周期到监视资源以跟踪节点的状态。主要功能如下：

①管理单个节点上的资源。

②处理来自ResourceManager的命令。

③处理来自ApplicationMaster的命令。

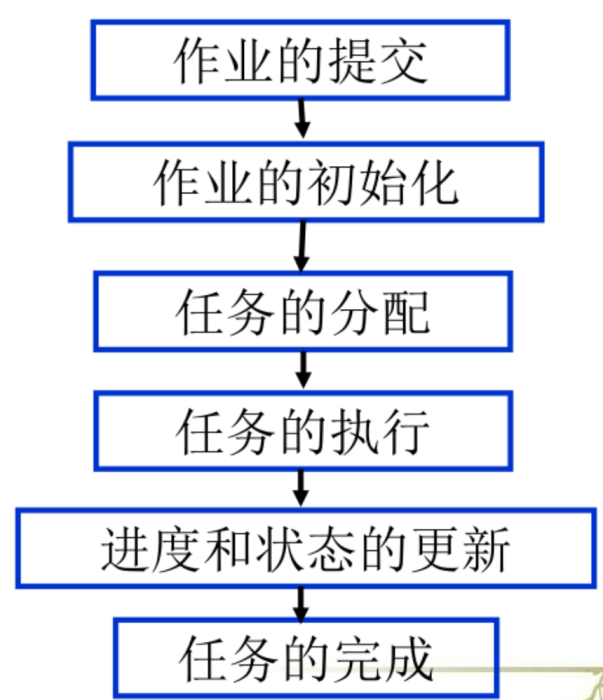
1. Container：是YARN中的资源抽象，它封装了某个节点上的多维度资源，如内存、CPU、磁盘、网络等。当ApplicationMaster向ResourceManager申请资源时，ResourceManager为ApplicationMaster返回的资源便是用Container。
2. YARN的通信协议：

在YARN中，**各组件的通信都依靠RPC协议来实现**。RPC（Remote Procedure Call Protocol），一个RPC协议通信有两端，一端是Client，另一端是Server，**且Client总是主动连接Server的**。

1. **YARN工作流程（重点一定要记下来）**：当用户向YARN提交一个应用程序 后YARN将分为两个阶段运行该应用程序。

①第一阶段：启动ApplciationMaster

②第二阶段：是由ApplicationMaster创建应用程序，为它申请资源，并监控它的整个运行过程，直到运行完成。

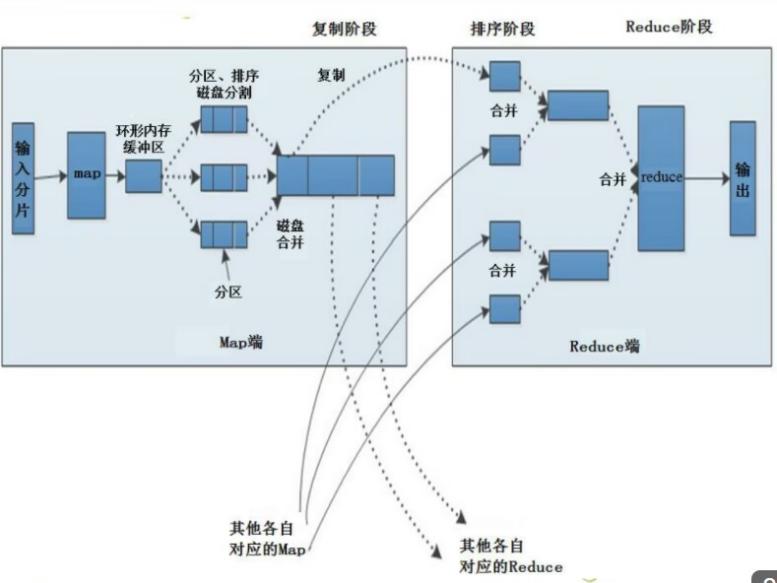


**YARN的执行流程**

#### 数据的混洗处理

**MapReduce**框架会确保每个reduce的输入都是按key排序的，系统执行排序的过程（将map输出作为输入传给reduce）称为**shuffle**（即混洗）

1. **Shuffle的职责**：**把map的输出结果有效地传送到reduce端**。
2. Shuffle在执行过程中需要满足的要求;①完整地从map端提取到数据到reduce端。②在跨节点拉取数据时，尽可能地减少对带宽的不必要消耗。③减少磁盘IO对任务执行的影响。



#### 作业部分

1. **Shuffle的职责是什么？描述Shuffle的过程**。

职责：对Map任务输出结果进行分区、合并、等处理后交给reduce端

过程：Shuffle过程分为两个不同角度，分别为map端和reduce端

map端：Map任务的输出结果首先被写入缓存，**当缓存满时，便启动“溢写”操作**，将缓存中的操作写入磁盘文件，并清空缓存，当启动“溢写”操作时，首先要对缓存中数据进行分区，然后对每个分区的数据进行排序与合并，再写入磁盘文件，每次的“溢写”操作会生成一个新的磁盘文件，随着Map任务执行磁盘中会产生多个溢写文件。

reduce端：reduce端从map端获取属于自己处理的那部分数据，然后对数据进行归并后交给reduce进行处理。

## Spark部分（直接背作业）

### （1）**什么是Spark，它有哪些特点**？

答：Spark是**基于内存计算的大数据分布式计算框架**，具有在大数据环境下的实时性， 具有高容错性和高伸缩性，可以允许用户将Spark部署在大量廉价硬件之上形成集群。

特点：①**速度快** ②**易用性** ③**通用性** ④**跨平台** 。

1. Spark有哪些组件分别用于哪些场景？

答：四大组件分别有：

①**Spark Streaming**  ：将输入的数据流以时间片为单位进行拆分，然后以类似批处理的方式处理每个时间片数据。

②**Spark SQL**：提供结构化数据的交互式操作，类似于SQL操作，进行复杂的数据分析操作

③**Spark MLlib**：用于迭代式计算。

④**Spark GraphX**：是一个分布式图处理框架，用于社交网络、社区发现中，提供基于Spark平台对图计算和图挖掘简介易用而丰富多彩的接口，极大地方便了大家对分布式图处理的需求。

### （3）Spark的核心数据结构是什么？它的特征是什么？

答：核心数据结构：**RDD**

特征：①**带有分区的集合类型** ②**可以并行操作** ③**并且是容错的**。

（4）Spark中的RDD有哪两类算子？分别起到什么作用？

答：两类算子分别是：①**Transformation**：可以从已有的数据集转换成一个新的数据集②**Action**：当rdd触发Action算子时，并执行Transformation算子时，每个Action会生成一个job。

### （5）什么是窄依赖？什么是宽依赖？它们在DAG划分Stage时有什么作用？

答：窄依赖：父RDD的分区和子RDD的分区关系：**一对一**，窄依赖不会产生Shuffle，**执行 效率高。**

宽依赖：父RDD的分区和子RDD的分区关系是：**一对多**，宽依赖会产生shuffle，**会产 生磁盘读写无法优化**。

作用：DAG会根据RDD之间的依赖关系进行Stage划分，流程是：以Action为基准，向前回溯，遇到宽依赖就形成一个Stage，遇到窄依赖则执行流水线优化（将多个连续的窄依赖放到一起执行）