大厂学苑-大数据&人工智能
概论

版本：V1.0

第1章 大数据基础

1.1 大数据概念

所谓的大数据就是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。大数据主要解决海量数据采集，存储，分析计算的问题。

1.1.1 大数据特点

- Volume(大量)

数据存储单位：bit, Byte, KB, MB, GB, TB, PB, EB, ZB, YB, BB, NB, DB 单位之间的转换为 1024 进制。

为了能更直接体会这些单位的大小，我们列举一些生活中的案例：截至目前，人类生产的所有印刷材料的数据量是 200PB，而历史上全人类总共说过的话的数据量大约是 5EB。当前，典型个人计算机硬盘的容量为 TB 量级，而一些大企业的数据量已经接近 EB 量级。

- Velocity(高速)

处理速度快是大数据区分传统数据挖掘的最显著特征。根据 IDC 的“数字宇宙”的报告，预计到 2025 年，全球数据使用量将达到 163ZB，在如此海量的数据面前，处理数据的效率就是企业的生命。

- Variety(多样)

生活中数据的多样性，让数据分为了结构化数据和非结构化数据两大类，相对于以往便于存储的以数据库、文本为主的结构化数据，非结构化的数据越来越多，包括网络服务器日志，音、视频，图片，地理位置信息。这些多类型的数据对数据的处理能力提出了更高的要求。

- Value(低价值密度)

数据的价值密度可以认为是：单位数据所产生的有价值的信息量。

传统数据基本都是结构化数据，每个字段都是有用的，价值密度非常高。大数据时代，越来越多数据都是半结构化和非结构化数据，比如网站访问日志，里面大量内容都是没价值的，真正有价值的比较少，虽然数据量比以前大了 N 倍，但价值密度确实低了很多。

如果有海量的结构化数据，需要大数据技术才能处理得了，当然也可以称之为大数据，但价值密度并不低。举个例子，银联、VISA 等清算组织有海量的交易数据，不仅数据量大，而且很有价值。所以“数据量越大，数据价值密度越低”是常见情况，但不是必然情况。

1.1.2 大数据应用场景

- 生活

- 工作

- 未来

1.1.3 大数据发展前景

- 党的十九大提出“**推动互联网，大数据，人工智能和实体经济深度融合**”
- 2020 年初，中央推出 34 万亿“新基建”投资计划

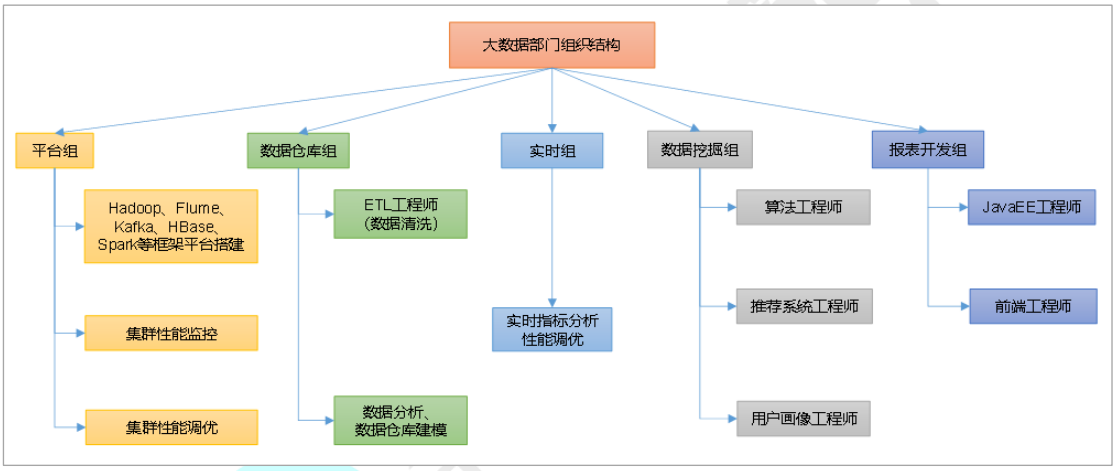
"新基建"投资规模拆分	
项目	2020年投资规模（亿元）
5G	3000
特高压	600
轨道交通	5000
充电桩	100
数据中心	1000
人工智能	350
工业互联网	100
合计	10150

- 2020 年是 5G 的元年，国家大力铺设 5G 设备，2021 年就是 5G 手机应用的开始，也是大数据要爆发的一年，海量数据所带来的就是下一个风口
- 大数据待遇还是不错的

1.1.4 大数据部门组织结构

- 大数据业务流程分析

- 大数据部门组织结构

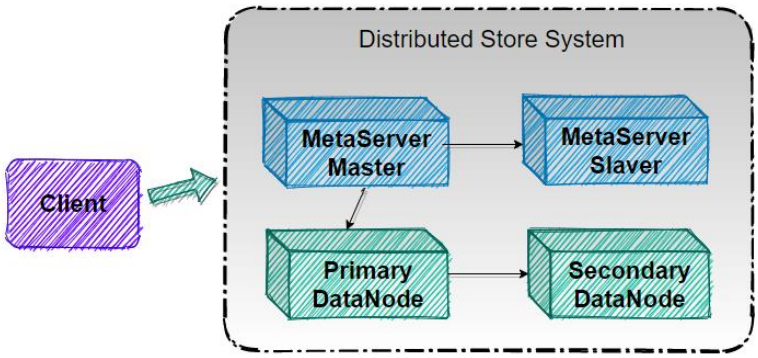


1.2 分布式数据存储

分布式存储系统可以理解为多台单机存储系统的各司其职、协同合作，统一的对外提供存储的服务。所以无论是存储非结构化数据的分布式文件系统，存储结构化数据的分布式数据库，还是半结构化数据的分布式KV，在系统的设计上主要需要满足以下需求(但不仅限于):基本读写功能、性能、可扩展性、可靠性、可用性。

从基本的读写功能来说，分布式文件系统主要提供文件的读/写/删功能，那么从哪里读写？怎么读写？以什么形式存储？在系统架构的设计上主要区别在于两点：

- 有无中心管理节点
- 存储节点是否有主从之分。



➢ **Master**

在有中心管理节点的分布式文件系统，从哪里读写的任务基本是由中心管理节点完成的，即图中的 Master 节点。为了能完成这一项核心的任务，Master 节点需要做三件比较重要的事情：

1) 存储节点信息和状态

对于分布式文件系统，从文件到某一台单机上的一块磁盘的某个位置，会有一个逻辑的拓扑结构便于分区扩展或者数据隔离等等。拓扑结构从上至下存储池、分区、服务器、磁盘、文件目录等。

Master 节点需要保存整个集群的全局视图，为了提高性能，这个逻辑拓扑结构一般会缓存在内存中，定期地持久化在 Master 节点的磁盘上。

Master 节点需要监听系统的所有数据节点和磁盘状态，如节点的上下线等，并对事件作出相应的处理来保证系统状态的正确性。

同时为了使得系统的数据分布和资源使用更均衡，Master 节点可以获取数据节点的容量、负荷等状态，供读写调度模块有策略的去分配可写的资源。

2) 文件读写的调度策略

由于没有采用类似 Hash 算法这种静态计算读写位置的方式，中心管理节点就需要担任起调度的角色。当客户端发起写请求时，第一步会到 Master 节点获取文件 ID，Master 节点根据客户端读写文件的大小、备份数等参数以及当前系统节点的状态和权重，选择合适的节点和备份，返回给客户端一个文件 ID，而这个文件 ID 包含了该文件多个副本位置信息。这样的好处是不用将每个文件的映射关系都存储下来。而对于对象存储系统因为功能的需要，这个对象和文件的映射关系是必需要保存的。

3) 存储节点选主

对于存储节点有主从之分的系统，每个备份的主从节点的选取也需要 Master 节点来控制。

- **DataNode**

数据节点除了负责文件在单机系统上如何进行存储，对于有主从之分的存储节点，各自还承担如何保持备份数据一致性的任务。归纳为以下几个要点：

1) 单机存储引擎的实现

主从存储节点在单机存储引擎的实现上几乎没有差异，解决的是文件如何存储的问题。对于不同的系统还有个区别是一个存储引擎负责一台机器所有磁盘的存储，还是一个磁盘一个存储引擎，我们这里以一个存储引擎负责一个磁盘的设计说明

分布式文件系统的存储引擎大多都是在单机文件系统之上，数据最终以文件的形式存在。而单机文件系统以目录的形式将文件组织起来。该系统基本沿用了这一思想，是以目录为单位进行副本划分，而不是节点，和 Swift 一样称作为 Partition，是备份的基本单位。以三备份为例，中心节点会根据策略选择三个不同的物理磁盘上创建同一个 ID 的 Partition，在不同物理磁盘的同一个 PartitionID 下存储的文件是一样的。

这里还有另外一个问题，文件存储到单机文件系统上是否合并的问题。简单直接的做法，就是以一个个文件直接存储在指定的某个 Partition 目录下。这样的方式带来的问题也是直接可见的，直接受到文件系统随机读写的性能的制约，对于小文件读写比较多的场景来说，磁盘常常会成为整个系统的瓶颈，然后不得不通过增加节点来进行吞吐能力的扩展。

所以该系统将多个文件追加合并写在一个相对较大且大小固定的文件里，将随机写转为了顺序写，实验表明可以大大提高单机存储的吞吐能力。这样的解决方式带来的性能提升是

明显，但是在实现上增加了一定的复杂性，合并必然就会带来定位文件的最终位置需要二次映射，即是在合并文件中的偏移。

2) 保证备份一致性

在保证备份一致性上，主从存储节点的角色就有一些区别了。对于 Master 节点选出的主存储节点，它需要根据主从一致性协议将数据推送到其它从节点，一般采用存储节点分主从的系统都会选择强一致协议，即主节点采用将数据发送给从节点收到响应成功后，才会将数据持久化到本地，返回用户成功，这样用户读到的数据始终是一致的。当然整个过程不是那么简单的，后续再一致性协议部分再展开。

3) 汇报消息，听从调度

对于存储节点需要保持和 Master 节点的心跳信息，同时将自己当前的容量和资源使用情况汇报给 Master 节点。从控制系统的角度来说，这才形成了一个负反馈系统。

同时，存储节点处于待命状态，等待 Master 节点的派遣任务，比如说数据备份的恢复迁移等。

- **Client**

Client 一般作为分布式文件系统的接入层，对写操作，接受用户数据流，将数据写入存储节点；对于读操作，从多副本中随机选择副本来读取。同时为了提高系统整体性能和可用性，该系统的 Client 一般还会负责额外的功能:

- 集群信息缓存:主要为了减少与 Master 节点的交互，提高写的性能。可以从 Master 节点获取副本位置信息，缓存在本地，设置缓存过期时间。
- 异常处理：Client 在提高系统可用性上扮演这重要的角色，在性能损耗可容忍的情况下，通过简单的重试超时方式即可解决 Master、Data 节点不可用的异常，最大限度地保证系统可用性。

上面的系统架构从性能角度来看，Master 节点是不会成为系统瓶颈的，毕竟现在的服务器处理能力是很高的，无论是 Master 节点缓存的集群信息占用的内存，还是对于一个几万台机器的集群的调度，单机 Master 节点都是可以扛得住的。

那么对于有中心管理节点，数据节点有主从之分的系统，它的性能瓶颈是在哪里？根据理论分析，假设单机存储系统参数不当或代码实现导致的性能问题都已排除解决，即纵向的性能优化基本符合预期。这样一个系统，对于小文件的读写场景，在有限的磁盘数量和文件数量，

如果数据分布的不是很分散，那么主要的性能瓶颈在于集群中某些磁盘的吞吐能力，而在大文件的读写场景，系统的瓶颈在主节点的出口网卡流量。以三备份为例，主节点需要往两个从节点写数据，这时候主节点的出口网卡流量是入口网卡的两倍，如果出口入口均为千兆网卡那么入口网卡的流量始终只能跑满到网卡最大流量的一半。所以在设计时才会以 Partition 为单位作为备份的基本单位，这样一来每个节点上面都有主从，每个节点的流量基本都可以跑满。

从可用性的角度来看，如果不做高可用的设计，Master 节点就是系统的单点。消除系统单点的方法很多，一般分布式系统中常直接使用 Zookeeper 来保证节点的高可用。所以其实中心管理节点的单点问题并没有那么严重的。相比而言，中心管理节点的调度策略显得更为重要，因为数据分布的是否均衡直接影响到系统对外服务的性能。

1.3 分布式数据计算

分布式计算(Distributed computing)是一种把需要进行大量计算的工程数据分割成小块，由多台计算机分别计算，数据在各个计算机节点上流动，同时各个计算机节点都能以某种方式访问共享数据，最终分布式计算后的输出结果被持久化存储和输出。

分布式计算比起其它算法具有以下几个优点：

1、稀有资源可以共享。

2、通过分布式计算可以在多台计算机上平衡计算负载。

3、可以把程序放在最适合运行它的计算机上。

其中，共享稀有资源和平衡负载是计算机分布式计算的核心思想之一。