【课程内容回顾】

【本次课程内容】

**文件系统是什么？**

**数据是什么？实际的内容**

**元数据是什么？描述数据的数据。（数据的属性信息）**

**文件系统就相当于是字典，字典中有检字表，正文。**

**文件系统中有元数据和数据。**

**文件系统-->字典**

**元数据-->检字表**

**数据-->正文**

**文件系统中包含了数据和元数据，并且文件系统还是底层存储的组织者。它指定了数据块的使用方式和维护方式。**

**底层存储是块硬盘，上层存储就认为他是一个连续的存储空间。硬盘认为自己最小的空间是512B，文件系统认为它是多大？可以调节。**

**什么叫做格式化？**

**文件系统：NTFS FAT32/16 EXT2/3/4**

**HDFS认为硬件总是不可靠的。其实不仅是HDFS，整个Hadoop都认为硬件不可靠。在实际的统计中，IDC的数据丢失或者数据失效，57%是由于硬件故障造成的丢失。所以HDFS不能将数据安全完全的依托于硬件来实现，因为大数据中，数据丢失所造成的损失是无法弥补的。那么对于HDFS来说，所有的数据保护都是由自身软件完成的。**

**软件在通过测试后，修补好发现的bug，成熟的软件在运行过程当中是不会崩溃的，只占12%人为操作导致23%**

**HDFS的吞吐量是非常之高的。**

**HDFS的最终目的是什么？数据分析，数据挖掘。**

**分析，挖掘的是什么数据？是历史性数据，旧的**

**实时性的历史数据怎么写入？攒起来一起写的时间长**

**10G的文件写到你的电脑大概需要多久?1个小时左右Hadoop是2,3分钟**

**无论什么文件系统，都是需要读写数据的，那么HDFS在设计之初，就想到了一个问题，海量的数据写入都是为了做分析使用。所以对于历史性的分析数据来说，读的业务往往大于写的业务的，而且是超过多倍。写业务一般也以新写入为主，改写基本没有。所以HDFS再**

**设计读写进程的时候，只设计一个写进程，而设计了非常多的读进程，那么这样做，写进程对于读的影响占到最低，并且写进程的优先级最低。所以HDFS支持高吞吐量的数据读取。**

**这样设计目的就是以读取为主。**

**FAT32?**

**第一次用u盘的时候？最大4G 是规定元数据的大小**

**为什么HDFS可以支持tb-pb的级别的数据**

**HDFS在设定之初，就指定了一个准侧。不管数据文件有多大，元数据固定150B。那么这么做就打破了文件系统对文件维护时元数据的大小，固定为150B之后，我就可以存储大文件仍然使用很小的元数据开销**

**分析用的数据不会修改。（气象局的数据）**

**流式数据**

**为了能够快速的查找元数据，所以我们会将元数据存放在内存当中，而且内存的大小是固定有限的。所以大量的小文件会占用过多的内存空间，导致内存利用率飙升。**

**Hadoop的内存是非常宝贵，因为Hadoop有很多组件都是基于内存的**

**HDFS不适合随机写入，第一个原因是因为读写进程的设计问题，另一个就是大量的业务会长期占用带宽，而随机写入会出现很高的延迟。**

**大文件的存储和低延迟的读取本质上是鱼和熊掌的关系。所以我们为了保证数据的正常写入，我们就牺牲了低延迟的读取功能。**

**所以在实际的选择中，一个业务应该使用什么文件系统其实更多的是根据业务的实际需求。**

**HDFS是运行在通用硬件平台上的文件系统。指的就是HDFS不论底层什么系统，什么框架，都可以进行搭建。**

**数据量大，无需改动。**

**HDFS使用的是worm（write once read many）模型**

**这种类型的业务，数据一般在写入之后，就基本不再做更改了，那么HDFS根据Worm模型设计了一套操作，当数据写入完成之后，就不在允许更改。那么除此之外，HDFS还执行了一个操作叫追加写。追加写允许用户想当前的文件结尾添加数据，(华为)但是只能添加一次，之后就被锁死无法追加，无法改写，除非删除数据，重新输入。**

**Worm 使用最多的地方，公安局，法院，政府部门**

**HDFS架构：**

**NameNode：名称节点，元数据节点（HDFS中一个进程），其主要工作是用于存储元数据和维护元数据，并且要记录用户对于元数据的更改信息（操作日志）。NameNode作为一个软件进程。平时是运行在内存中的，那么在开机的时候，Hadoop底层系统会将该进程调用运行在内存中。NameNode分为主备俩个进程。平时由主来保证业务的正常执行。备用于做持久化操作。**

**DataNode：数据节点，其作用是维护和存储数据，并且还可以存储namenode所维护的元数据和日志信息。（namenode在内存，关机就会没了，所以要把日志和元数据储存在DataNode里）DataNode需要周期性的上报自己所维护数据的摘要信息给namenode，namenode负责及时更新元数据。（无数个）**

**Client:client是HDFS中的一个基本的进程，用于提供外部访问的接口，并且对内部来说，用于向DataNode和namenode发起请求，相当于，用户提交一个请求给HDFS，client 受理该请求在之后，会代理该用户进行HDFS的内部交互，这样做就可以保证HDFS的数据安全。（数量根据业务部署自定义）**

**Hdfs认为硬盘不可靠的**

**Hdfs -HA**

**Hdfs认为硬件永远是不可靠的。**

**底层组件的安全性由上层组件保护，上层组件安全性有 zookeeper保护、**

**DataNode作为一个进程，负责维护数据，并且上报数据信息给namenode，为了保证DataNode的安全性，hdfs设计DataNode必须 定时上报心跳数据包给namenode进行包活。（定时是否允许改变）但是DataNode周期性还要上报自己所维护的数据信息，所以周期性上报俩个数据包，不利于带宽并且过于繁杂。那么我们将心跳信息和维护的数据信息合并成为一个信息，其实就是所维护的数据信息。那么namenode收到了维护的数据信息包，就相当于收到了心跳数据。（GFS(有心跳数据包)）**

**Namenode维护元数据，存在有主备进程，namenode作为 HDFs的顶级进程会周期性的通过ZKFC进程（（zookeeper failover controller）上传自己的心跳到Zookeeper，一旦Zookeeper没有周期性的获取到主NameNode的心跳，则会下发故障切换的命令通过ZKFC到达备NameNode。备就变成主。**

**备节点平时不参与业务，但是备节点参与了日志合并的元数据持久化的操作，备节点通过JN进程联系主节点获取用户对元数据的操作日志，执行数据持久化。**

**三副本机制**

**数据副本机制：**

**由于HDFS认为硬件不可靠，所以对于数据的安全性和可靠性都是由HDFS自身软件机制实现的，但是软件机制实现数据的保护本身就没有很多的方法去实现，出于绝对安全的考虑，HDFS采用的就是副本机制来实现相关的安全性保证、所谓副本机制就是指将数据直接复制多份，每个副本都是完全可用的相同的数据，HDFS默认将数据复制3份，那么4份数据就一起保证了数据的绝对安全性，我们也称该副本机制为HDFS3副本。**

**副本的存放遵循有相关的方法，是根据副本距离来判定的。**

**我们认为如果一个数据和其副本存储在了一台服务器上，那么他们的距离为0**

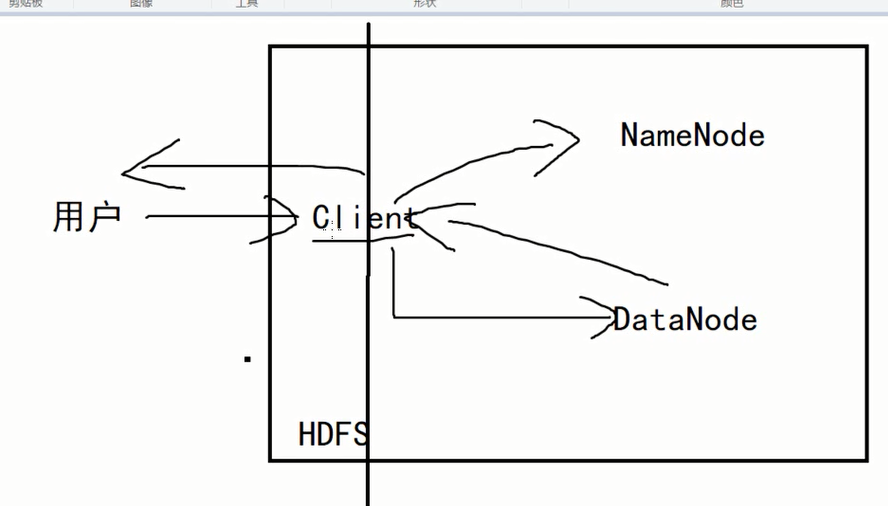
**我们认为如果一个数据和其副本存储在了相同机架的不同服务器上，其距离为2**

**我们认为如果一个数据和其副本存储在了不同的机架上，其距离为4.**

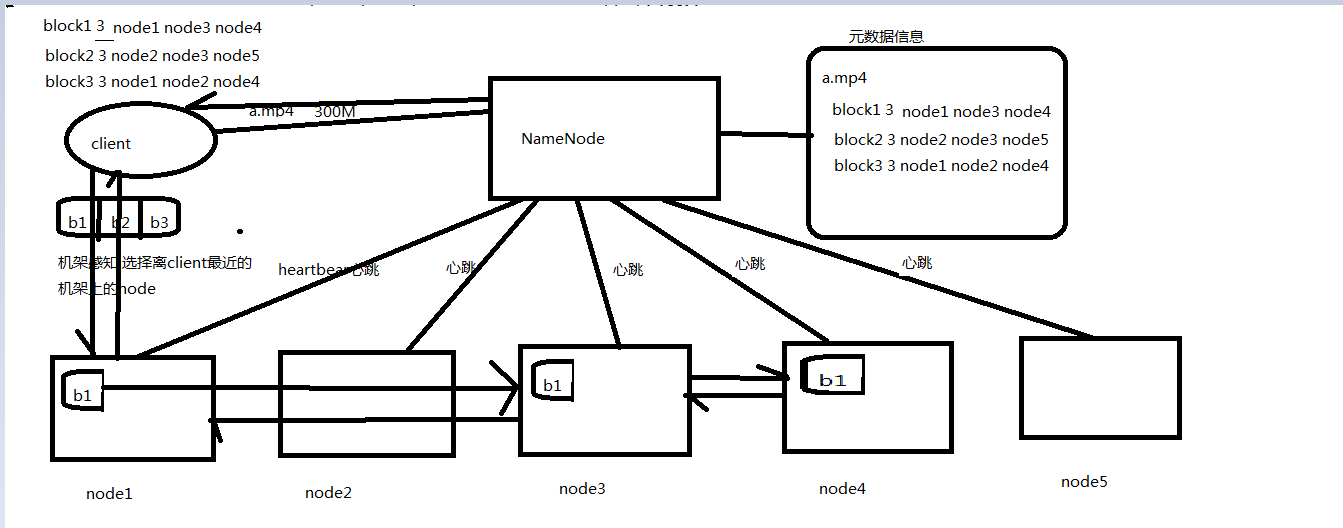
**第一份副本要和源数据放在一起，则距离为0**

**第二份副本随机进行存储，其实也就是指存储在除了源数据所在的服务器以外的任意的服务器上。**

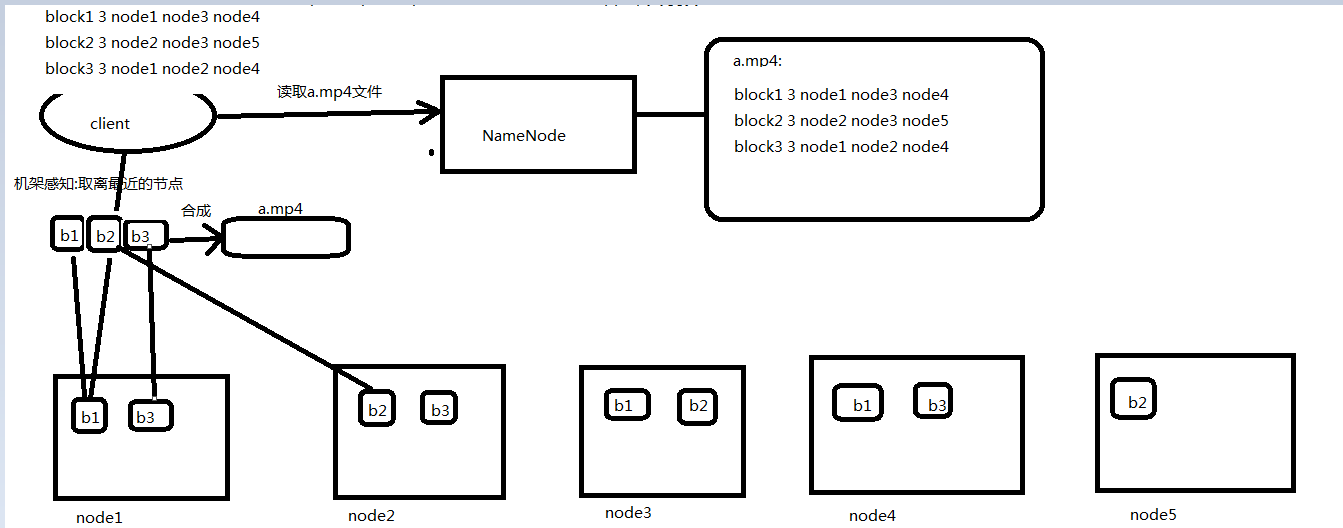
**第三份副本，如果第二份副本的距离为4，那么第三份副本的距离为2，如果第二份副本的距离为2，那么第三份副本的距离为4**



**HDFS写入流程**



**HDFS读流程：**

****

**重点：元数据持久化**

**元数据为什么要持久化？**

**元数据加载在内存当中，更改元数据改变内存当中的数据。如果关机，内存当中的数据就会丢失。**

**Editlog上记载的对元数据的操作。**

**Fsimage元数据镜像**

**8点开始 9点关机 10点再开机**

**开机时HDFS会把fsimage加载到内存当中**

**9点关机 10点再开机加载的是8的fsimage没有8点到9点之前的对元数据的操作。**

**-9点的操作在editlog之中，在10开机后根据editlog进行还原。**

**但是时间间隔过长，这样会让还原时间非常长，**

**元数据持久化：**

**1.当Editlog每满64M或者时间到达1h的时候，就启动一次元数据持久化工作。**

**2.首先第一步备NameNode通知主NameNode开始执行元数据持久化，主NameNode收到信息之后会创建一个新的文件editlog.new文件用于存储最新的元数据操作日志。这么做的目的主要有两个，第一个是因为备NameNode读取Editlog的时候会对文件加锁，主无法写入。第二个原因是因为editlog本身记录的就是对元数据的操作，其主要存在的目的就是要在HDFS重启之后做元数据的回滚，而现在我们将editlog中记录的操作和镜像文件做了合并，那么本身editlog就没有意义了。新的editlog中记录的是本次持久化开始，到下一次持久化开始之前的所有的对元数据的操作日志。**

**3.备NameNode获取旧的editlog和元数据镜像文件。**

**4.备NameNode将editlog中记录的对元数据的操作合并到fsimage元数据镜像文件中。生成一个文件叫做Fsimage.ckpt（checkpoint）。**

**5.备NameNode生成完成之后，下一步会将该文件传输到主NameNode，然后使用最新的Fsimage.ckpt将旧的Fsimage文件回滚**

**6.回滚完成之后元数据镜像被更新，之后旧的Editlog文件直接删除，Editlog.new文件被直接重命名成Editlog。**

**联邦和统一名空间非常相像的。**

**一个HDFS的集群需要几个节点来做？3个节点**

**Namenode可以在三个节点上进行混合部署。主备不能同时布置1台**

**Datanode提供存储空间。**

**RAM\_DISK(内存盘)把内存虚拟成硬盘。**

**Archive 磁带光盘**

**安全模式**

**当HDFS中有Datanode出现故障，那么这个时候，HDFS就自动的进入到了安全模式中，在安全模式里HDFS是不允许用户进行写操作的，只可以进行只读操作。那么这样做就可以保证在故障出现的时候，故障不会影响到其他的节点，并且故障不会进行扩散（坏两块）。**

【作业】

【预习内容】