一、基础概念

1、是什么

HDFS是Hadoop Distribute File System 的简称，也就是Hadoop的一个分布式文件系统。

分布式文件系统（Distributed File System）是指文件系统管理的物理存储资源不一定直接连接在本地节点上，而是通过计算机网络与节点相连。分布式文件系统的设计基于客户机/服务器模式。一个典型的网络可能包括多个供多用户访问的服务器。另外，对等特性允许一些系统扮演客户机和服务器的双重角色

2、相关概念

- **block（数据块）**

提供真实文件数据的存储服务。是最基本的存储单位。对于文件内容而言，一个文件的长度大小是size，那么从文件的 0 偏移开始，按照固定的大小，顺序对文件进行划分并编号，划分好的每一个块称一个Block。每一个block会在多个datanode上存储多份副本，默认是3份。

- **namenode（元数据节点）**

namenode负责管理文件目录、文件和block的对应关系以及block和datanode的对应关系。

文件结构

- 1

fsimage:元数据镜像文件。存储某一时段NameNode内存元数据信息。
edits:操作日志文件。
fstime:保存最近一次checkpoint的时间

处理过程

- 1

Namenode始终在内存中保存metedata，用于处理“读请求”到有“写请求”到来时，namenode会首先写editlog到磁盘，即向edits文件中写日志，成功返回

后，才会修改内存，并且向客户端返回

Hadoop会维护一个fsimage文件，也就是namenode中metedata的镜像，但是fsimage不会随时与namenode内存中的metedata保持一致，而是每隔一段时间通过合并edits文件来更新内容。Secondary namenode就是用来合并fsimage和edits文件来更新NameNode的metedata的。

- **datanode（数据节点）**

datanode就负责存储了，当然大部分容错机制都是在datanode上实现的。

- **secondarynamenode（从元数据节点）**

不是我们所想象的元数据节点的备用节点，其实它主要的功能是主要功能就是周期性将元数据节点的命名空间镜像文件和修改日志合并，以防日志文件过大。

处理过程：

1、secondary通知namenode切换edits文件

- 1

2、secondary从namenode获得fsimage和edits(通过http)

- 1

3、secondary将fsimage载入内存，然后开始合并edits

- 1

4、secondary将新的fsimage发回给namenode

- 1

5、namenode用新的fsimage替换旧的fsimage

- 1

3、优缺点

- **优点**

处理超大文件

- 1

这里的超大文件通常是指百MB、设置数百TB大小的文件。目前在实际应用中，HDFS已经能用来存储管理PB级的数据了。

流式的访问数据

- 1

HDFS的设计建立在更多地响应”一次写入、多次读写”任务的基础上。这意味着一个数据集一旦由数据源生成，就会被复制分发到不同的存储节点中，然后响应各种各样的数据分析任务请求。在多数情况下，分析任务都会涉及数据集中的大部分数据，也就是说，对HDFS来说，请求读取整个数据集要比读取一条记录更加高效。

运行于廉价的商用机器集群上

- 1

Hadoop设计对硬件需求比较低，只须运行在低廉的商用硬件集群上，而无需昂贵的高可用性机器上。廉价的商用机也就意味着大型集群中出现节点故障情况的概率非常高。这就要求设计HDFS时要充分考虑数据的可靠性，安全性及高可用性。

- **缺点**

不适合低延迟数据访问

- 1

如果要处理一些用户要求时间比较短的低延迟应用请求，则HDFS不适合。HDFS是为了处理大型数据集分析任务的，主要是为达到高的数据吞吐量而设计的，这就可能要求以高延迟作为代价。

改进策略：对于那些有低延时要求的应用程序，HBase是一个更好的选择。通过上层数据管理项目来尽可能地弥补这个不足。在性能上有了很大的提升，它的口号就是goes real time。使用缓存或多master设计可以降低client的数据请求压力，以减少延时。还有就是对HDFS系统内部的修改，这就得权衡大吞吐量与低延时了，HDFS不是万能的银弹。

无法高效存储大量小文件

- 1

因为Namenode把文件系统的元数据放置在内存中，所以文件系统所能容纳的文件数目是由Namenode的内存大小来决定。一般来说，每一个文件、文件夹和Block需要占据150字节左右的空间，所以，如果你有100万个文件，每一个占据一个Block，你就至少需要300MB内存。当前来说，数百万的文件还是可行的，当扩展到数十亿时，对于当前的硬件水平来说就没法实现了。还有一个问题就是，因为Map task的数量是由

splits来决定的，所以用MR处理大量的小文件时，就会产生过多的Maptask，线程管理开销将会增加作业时间。举个例子，处理10000M的文件，若每个split为1M，那就会有10000个Maptasks，会有很大的线程开销；若每个split为100M，则只有100个Maptasks，每个Maptask将会有更多的事情做，而线程的管理开销也将减小很多。

不支持多用户写入及任意修改文件

- 1

在HDFS的一个文件中只有一个写入者，而且写操作只能在文件末尾完成，即只能执行追加操作。目前HDFS还不支持多个用户对同一文件的写操作，以及在文件任意位置进行修改。

总结：

这次我们知道了HDFS是一个分布式的文件存储系统，它的一些基本的概念和优缺点我们已经知道了，下次我们将给大家分享一下HDFS的运行原理。