一 HDFS 概述

1.1 HDFS 产生背景

随着数据量越来越大，在一个操作系统管辖的范围内存不下了，那么就分配到更多的操作系统管理的磁盘中，但是不方便管理和维护，迫切需要一种系统来管理多台机器上的文件，这就是分布式文件管理系统。HDFS 只是分布式文件管理系统中的一种。

1.2 HDFS 概念

HDFS，它是一个文件系统，用于存储文件，通过目录树来定位文件；其次，它是分布式的，由很多服务器联合起来实现其功能，集群中的服务器有各自的角色。

HDFS 的设计适合一次写入，多次读出的场景，且不支持文件的修改。适合用来做数据分析，并不适合用来做网盘应用。

1.3 HDFS 优缺点

1.3.1 优点

1）高容错性

（1）数据自动保存多个副本。它通过增加副本的形式，提高容错性；

（2）某一个副本丢失以后，它可以自动恢复。

2）适合大数据处理

（1）数据规模：能够处理数据规模达到 GB、TB、甚至 PB 级别的数据；

（2）文件规模：能够处理百万规模以上的文件数量，数量相当之大。

3）流式数据访问，它能保证数据的一致性。

4）可构建在廉价机器上，通过多副本机制，提高可靠性。

1.3.2 缺点

1）不适合低延时数据访问，比如毫秒级的存储数据，是做不到的。

2）无法高效的对大量小文件进行存储。

（1）存储大量小文件的话，它会占用 NameNode 大量的内存来存储文件、目录和块信息。这样是不可取的，因为 NameNode 的内存总是有限的；

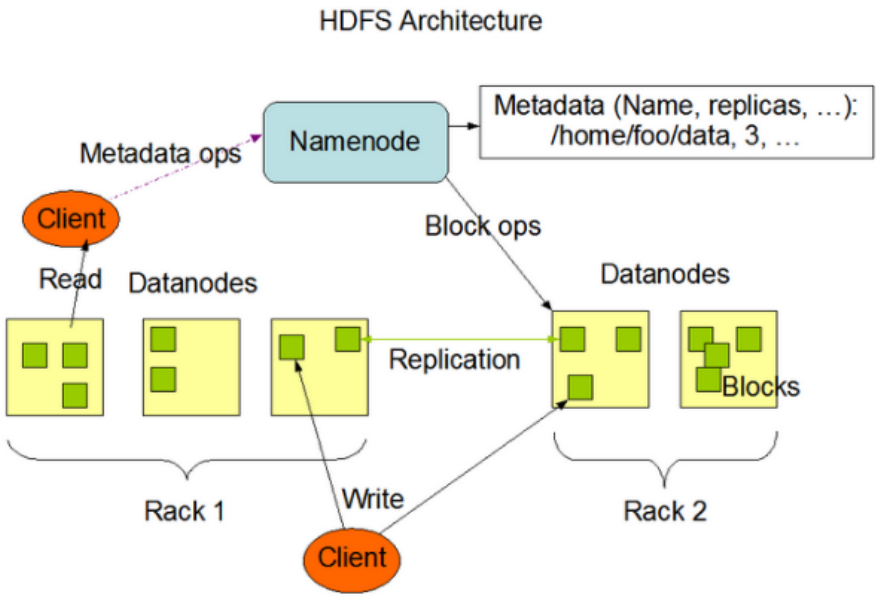
（2）小文件存储的寻址时间会超过读取时间，它违反了 HDFS 的设计目标。

3）并发写入、文件随机修改。

（1）一个文件只能有一个写，不允许多个线程同时写；

（2）仅支持数据 append（追加），不支持文件的随机修改。

1.4 HDFS 组成架构



HDFS 的架构图

这种架构主要由四个部分组成，分别为 HDFS Client、NameNode、DataNode 和 Secondary NameNode。下面我们分别介绍这四个组成部分。

1）Client：就是客户端。

（1）文件切分。文件上传 HDFS 的时候，Client 将文件切分成一个一个的 Block，然后进行存储；

（2）与 NameNode 交互，获取文件的位置信息；

（3）与 DataNode 交互，读取或者写入数据；

（4）Client 提供一些命令来管理 HDFS，比如启动或者关闭 HDFS；

（5）Client 可以通过一些命令来访问 HDFS；

2）NameNode：就是 Master，它是一个主管、管理者。

（1）管理 HDFS 的名称空间；

（2）管理数据块（Block）映射信息；

（3）配置副本策略；

（4）处理客户端读写请求。

3） DataNode：就是 Slave。NameNode 下达命令，DataNode 执行实际的操作。

（1）存储实际的数据块；

（2）执行数据块的读/写操作。

4） Secondary NameNode：并非 NameNode 的热备。当 NameNode 挂掉的时候，它并不能马上替换 NameNode 并提供服务。

（1）辅助 NameNode，分担其工作量；

（2）定期合并 Fsimage 和 Edits，并推送给 NameNode；

（3）在紧急情况下，可辅助恢复 NameNode。

1.5 HDFS 文件块大小

HDFS 中的文件在物理上是分块存储（block），块的大小可以通过配置参数(dfs.blocksize)来规定，默认大小在 hadoop2.x 版本中是 128M，老版本中是 64M。

HDFS 的块比磁盘的块大，其目的是为了最小化寻址开销。如果块设置得足够大，从磁盘传输数据的时间会明显大于定位这个块开始位置所需的时间。因而，传输一个由多个块组成的文件的时间取决于磁盘传输速率。

如果寻址时间约为 10ms，而传输速率为 100MB/s，为了使寻址时间仅占传输时间的 1%，我们要将块大小设置约为 100MB。默认的块大小 128MB。

块的大小：10ms*100*100M/s = 100M