HDFS 分布式系统

W.J.Z

分布式文件系统是一种通过网络实现文件在多台主机进行分布式储存的文件系统，分布式文件系统的设计一般采用“客户机/服务器”模式，客户端通过特定的通信协议与服务器建立连接，提出文件访问请求。

1 hdfs 的特点

1. 兼容廉价的硬件设备：hdfs 可运行在廉价服务器上，并设计了快速检测硬件故障和进行自动恢复机制，使得在硬件出错情况下也能实现数据的完整性。
2. 流数据读写：普通文件系统主要用于随机读写及与用户交互，而分布式系统 HDFS 是为了满足批量数据处理的要求而设计的。
3. 大数据集：HDFS 中的文件通常可以达到 GB 或 TB 级别。
4. 简单文件模型：HDFS 采用一次写入、多次读取的模型，写入时只支持单个写入者且写操作是以“只添加”方式在文件末尾写数据。
5. 跨品台兼容性;HDFS 采用 java 语言实现，支持 JVM 的机器都可以运行 HDFS。
6. 不适合低延迟数据访问：HDFS 主要针对大规模数据批量处理而设计的，延迟较高.
7. 不适用于存储小文件：小文件过多会造成存储在 namenode 节点内存中的元数据过多，造成元数据检索效率降低，同时内存空间需求量会大大增加；使用 MapReduce 处理大量小文件时，会产生过多的 Map 任务，线程管理开销会增加。

2 HDFS 相关概念

2.1 块

HDFS 块的大小为 128MB(以后随着硬盘容量的变大还会变)，比磁盘块大，其目的是为了最小化寻址开销。如果块足够大，数据传输时间会明显的大于定位该块起始位置所需的时间。

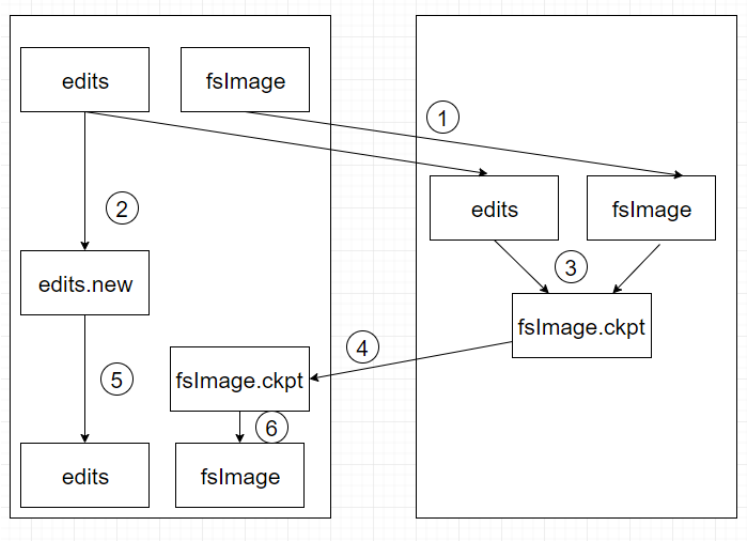
HDFS 上的文件也被划分为块大小的多个分块，这样的好处：(1)、一个文件的大小可以大于网络中任意一个磁盘的大小;(2)、使用抽象块而非整个文件作为储存单元，大大简化存储子系统的设计。(3)、块适用于数据备份而提供数据容错能力和提高可用性。

2.2 namenode 和 datanode

namenode 负责管理分布式文件系统的命名空间，保存了两个核心的数据结构 FsImage 和 EditLog，FsImage 用于维护文件系统树以及文件树中所有的文件和文件夹的元数据，EditLog 记录所有针对文件的创建、删除、命名等操作，这些信息以两个文本的形式永久的保存在本地磁盘上。namenode 还记录每个文件中的各个块所在的数据节点信息，这些信息不会持久化保存，而在系统每次启动时扫描所有数据节点重构得到这些信息。

2.3 secondarynamenode

一旦 namenode 失效，文件系统将无法工作，因此对 namenode 实现容错非常重要。secondarynamenode 负责定期合并并编辑日志与命名空间镜像，以防止编辑日志过大。



1. 使用 http get 复制 edits 和 fsimage 文件。
2. 生成新的 edits
3. 将 fsimage 导入内存，应用 edits 中的操作，生成新的 fsimage.ckpt。
4. 使用 http post 发送 fsimage.ckpt。
5. 将 edits 替换为新的 edits.new。
6. 将 fsimage 替换为 fsimage.ckpt。

secondarynamenode 请求 namenode 停止使用 edits, 暂时将新写操作放入一个新的文件中（edits.new）. 通过 http get 获取 edits 和 fsimage，将它们加载到内存中完成一系列合并操作生成新的 fsiamge，然后通过 http post 方式将新的 fsimage 发送给 namenode。namenode 获取到新的 fsimage 并代替原有的 fsimage，并把 edits.new 变为 edits。namenode 和 secondarynamenode 一般放在不同的机器上，fs.checkpoint.period：默认是一个小时。fs.checkpoint.size: 默认 64MB。

2.4 联邦 HDFS

为了实现系统横向扩展，联邦 HDFS 允许系统通过添加 nadenode 实现扩展，每个 namenode 管理文件系统命名空间的一部分，比如一个 namenode 复杂管理/usr 目录下的所有文件，一个 namenode 负责管理/share 目录下的文件。

3 常用命令

1. hadoop fs -ls：显示指定目录信息
2. hadoop fs -cat：将内容输出到标准输出
3. hadoop fs -chgrp: 改变文件所属的组
4. hadoop fs -chown: 改变文件的所有者
5. hadoop fs -tail: 将文件末尾内容输出到标准输出
6. hadoop fs -mkdir: 创建一个文件夹
7. hadoop fs -copyFromLocal<local><dis>: 将本地文件上传到分布式系统中
8. hadoop fs -copyToLocal: 将目标文件复制到本地系统中
9. hadoop fs -cp<src><dis>：复制文件
10. hadoop fs -get<src><local>: 复制分布式系统中文件到本地系统
11. hadoop fs -put<local><dis>：将本地系统文件复制到分布式系统中
12. hadoop fs -mv: 移动文件
13. hadoop fs -rm: 删除文件
14. hadoop fs -test: 检查文件相关信息,-e 文件是否存在，存在返回 0，否则 1;-z 文件是否为 0 字节，是返回 0 否则返回 1;-d 路径是否是一个目录，是返回 0 否则返回 1;

4 HDFS 常用 JAVA API 及应用实例

1. org.apache.hadoop.fs.FileSystem: 一个通用文件系统的抽象基类，可以被分布式系统继承。
2. org.apache.hadoop.fs.FileStatus: 一个接口，向客户端展示系统中文件和目录的元数据。
3. org.apache.hadoop.fs.FSDataInputStream: 文件输入流，用于读取 hadoop 文件。
4. org.apache.hadoop.fs.FSDataOutputStream: 文件输出流，用于写 hadoop 文件。
5. org.apache.hadoop.conf.Configuration: 访问配置项。
6. org.apache.hadoop.fs.Path: 用于表示 hadoop 文件系统中的一个文件或一个目录。

7. org.apache.hadoop.fs.PathFilter: 一个接口，实现方法 PathFilter.accpt(Path path) 来决定是否接受路径 path 表示的文件或目录。