

2、HDFS

- HDFS的构成

- namenode

管理命名空间。其中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，操作日志文件记录了所有针对文件的创建、删除、重命名等操作。名称节点在启动时，会将FSImage的内容加载到内存当中，然后执行EditLog文件中的各项操作，使得内存中的元数据保持最新。这个操作完成之后，就会创建一个新的FSImage和一个空的editlog。名称节点启动成功并进入正常运行状态以后，HDFS中的更新操作都会被写入到editlog，而不是直接写入FSImage。名称节点在启动的过程中处于“安全模式”，只能对外提供读操作，无法提供写入操作（写入EditLog）。启动过程结束后，系统就会退出安全模式，进入正常运行状态，对外提供写操作。

- 命名空间镜像文件（FSImage）
 - 编辑日志文件（EditLog）

- secondary namenode

定期合并namenode传来的命名空间镜像文件以及编辑日志文件，并在namenode故障的情况下，成为新的namenode（冷启动）

- 功能1：editLog与FsImage的合并操作

每隔一段时间，第二名称节点会和名称节点通信，请求其停止使用EditLog文件，暂时将新达到namenode的写操作添加到一个新的文件editlog.new中。然后，第二名称节点把名称节点中的FsImage和editlog拉回本地，在加载到内存中，对二者进行合并操作。合并结束后，第二名称节点会把合并后的FsImage文件发送到名称节点。名称节点收到后，会用收到的FsImage替换原来的FsImage，同时用EditLog.new替换掉EditLog。

- 功能2：称为名称节点的checkpoint

从合并过程可以看出，第二名称节点会定期和名称节点通信，从名称节点获取FsImage文件和EditLog文件，执行合并操作得到新的FsImage文件。从这个角度来讲，第二名称节点相当于为名称节点设置了一个“检查点”。

- datanode

- 基本概念

- 数据块

- 块缓存

对于访问频繁的文件，其对应的块可能被显示地存放于datanode的内存中。

- 联邦HDFS

多个namenode，每个namenode管理命名空间的一部分

- 高可用性

- 传统方法

- namenode在多个文件系统中备份元数据
 - 备用namenode创建监测点

- 活动-备用（active-standby）namenode（Hadoop2、热备）

- 主备namenode通过高可用共享存储实现编辑日志的共享
 - datanode向两个namenode发送数据库处理报告

- HDFS操作常用shell命令

这些命令就是一个Java应用（指的是dfs -ls这种命令，并非sh结尾的那种）

- 查看命令使用方法

- 启动Hadoop
 - cd /usr/local/hadoop
 - ./sbin/start-dfs.sh(单机模式下)
- 查看hdfs dfs总共支持哪些操作
 - ./bin/hdfs dfs
 - 可以看出，hdfs dfs 命令的统一格式是类似hdfs dfs -ls这种形式，即在-后面跟上具体的操作
- 查看某个命令的作用
 - ./bin/hdfs dfs -help put
- HDFS目录操作
 - /user/hadoop目录即为hadoop用户对应的用户目录
 - ...
- 利用HDFS的web管理界面
 - 在浏览器中输入<http://localhost:50070>就可以看到HDFS的web管理界面
- HDFS编程实践
 - Hadoop采用Java语言开发，提供了Java API与HDFS进行交互。上面介绍的Shell命令，在执行的时候实际上会被系统转换为Java API调用。（貌似HDFS支持随机读，不支持随机写，随机读可以通过Java api做到）
 - FileSystem类
 - 是一个通用文件系统的抽象基类，可以被分布式文件系统继承。Hadoop为FileSystem这个抽象类提供了多种具体的实现，DistributedFileSystem就是FileSystem在HDFS文件系统中的实现。FileSystem的open方法返回的是一个输入流FSDataInputStream对象，在HDFS文件系统中具体的输入流就是DFSInputStream；FileSystem的create方法返回的是一个输出流FSDataOutputStream对象，在HDFS文件系统中具体的输出流就是DFSOutputStream。
 - 具体步骤