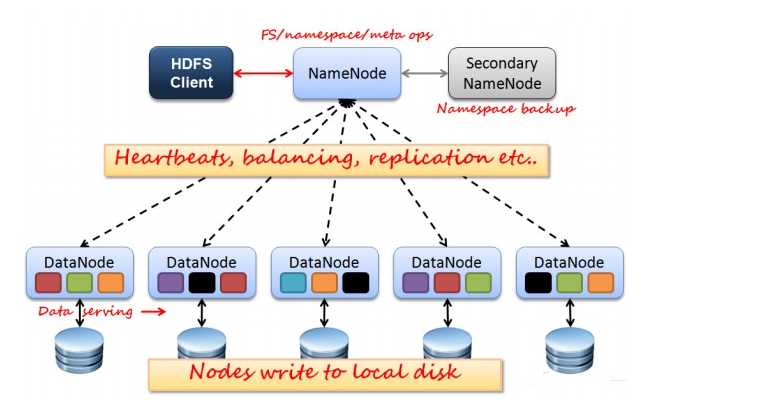
# HDFS结构

一.什么是HDFS (Hadoop Distributed File System Hadoop分布式文件系统) 基于Google的gfs

二.特点：高性能，故障容错，集中管理（主从结构Master/Slave），安全，分布式处理，可扩展性

三.架构



**a.NameNode**：是Master节点，是大领导。存储元数据信息；负责管理数据块映射；处理客户端的读写请求；配置副本策略；管理HDFS的名称空间；

metadata信息包括：

1:文件的权限，某一上传文件包含哪些Block块，Bolck块有保存在哪些DataNode上面；

2：DataNode在启动时候也会上报block信息；

3：NameNode的metadata信息在启动后会加载到内存

4：metadata存储到磁盘的文件为镜像形式fsimage,需要注意的是block的位置信息不保存在fsimage，block在DataNode启动时上报到DataNode加载到内存

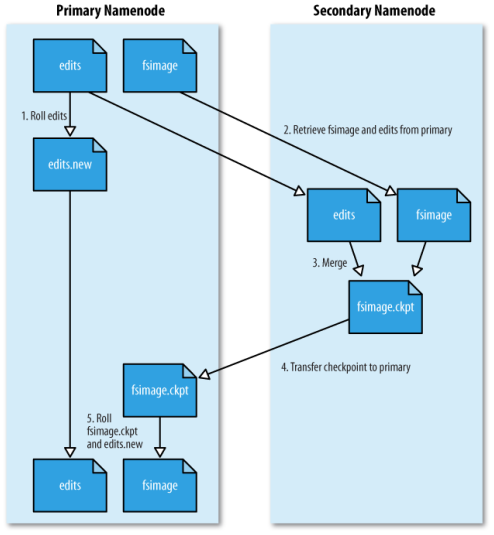
**b**.**SecondaryNameNode**：是一个小弟，分担大哥namenode的工作量；是NameNode的冷备份；合并fsimage和editlog然后再发给namenode。

SecondaryNameNode作用

1：主要工作保存namenode中对HDFS metadata的信息的备份

2：主要工作帮助NameNode合并edit log，减少NameNode的启动时间

3：Secondary Name合并Nodeedit log过程如图所示



  解释：SecondaryNameNode会周期性的将EditLog进行合并，合并原因：

               1：editlog到达某一值时对其进行合并；

               2：每隔一段时间对其进行合并；

          将记录HDFS的editlog与其上一次合并后存在的fsimage进行合并到fsimage.checkpoint，然后清空

          editlog，将checkpoint拷贝到NameNode

4：secondary NameNode 可以作为NameNode的冷备份：

        将本地保存的fsimage（镜像）导入；

        修改所有cluster的所有DN的master地址；

        修改所有client端NameNode地址；

热备份：b是a的热备份，如果a坏掉。那么b马上运行代替a的工作。

冷备份：b是a的冷备份，如果a坏掉。那么b不能马上代替a工作。但是b上存储a的一些信息，减少a坏掉之后的损失。

**c.DataNode**：Slave节点，奴隶干活的，负责存储client发来的数据块block，执行数据块的读写操作。（hadoop最新版本block已达到128M）

1：保存Block块信息；

2：DataNode启动时，DataNode向NameNode报告block信息

3：通过向NameNode发送心跳与其保持与其联系（三秒一次），如果10分钟没有收到心跳，则将其block拷贝到

   其他节点。

4：分布式情况下，一台服务器都会运行一个DataNode

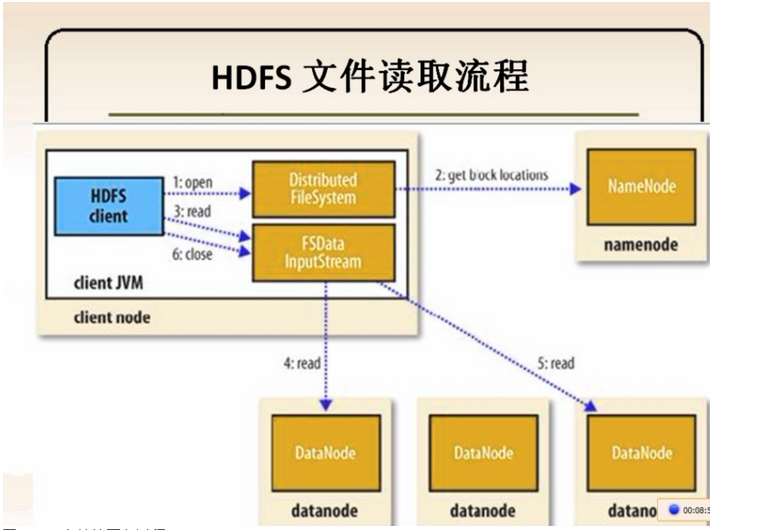
5：替换Block操作，如果一个DataName接受了一个Block替换请求，那么这个请求包含了这个Block所在源

   NameNode，当前节点向源节点发送拷贝请求，进行拷贝

**d. Client：:**文件切分，与NameNode交互，获取文件位置信息;与DataNode交互,读取或者写入数据;管理和访问HDFS

四.工作原理

**a.读原理**：



1.首先调用FileSystem对象的open方法，其实是一个DistributedFileSystem的实例。

2.DistributedFileSystem通过rpc获得文件的第一批block的locations，同一个block按照重复数会返回多个locations，这些locations按照hadoop拓扑结构排序，距离客户端近的排在前面。

3.前两步会返回一个FSDataInputStream对象，该对象会被封装DFSInputStream对象，DFSInputStream可 以方便的管理datanode和namenode数据流。客户端调用read方法，DFSInputStream最会找出离客户端最近的datanode 并连接。

4.数据从datanode源源不断的流向客户端。

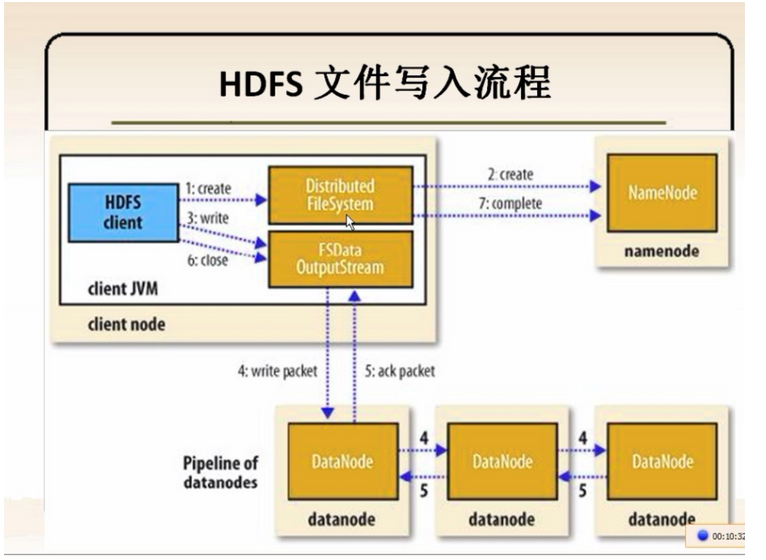
5.如果第一块的数据读完了，就会关闭指向第一块的datanode连接，接着读取下一块。这些操作对客户端来说是透明的，客户端的角度看来只是读一个持续不断的流。

6.如果第一批block都读完了， DFSInputStream就会去namenode拿下一批block的locations，然后继续读，如果所有的块都读完，这时就会关闭掉所有的流。

如果在读数据的时候， DFSInputStream和datanode的通讯发生异常，就会尝试正在读的block的排序第二近的datanode,并且会记录哪个 datanode发生错误，剩余的blocks读的时候就会直接跳过该datanode。 DFSInputStream也会检查block数据校验和，如果发现一个坏的block,就会先报告到namenode节点，然后 DFSInputStream在其他的datanode上读该block的镜像。

该设计就是客户端直接连接datanode来检索数据并且namenode来负责为每一个block提供最优的datanode， namenode仅仅处理block location的请求，这些信息都加载在namenode的内存中，hdfs通过datanode集群可以承受大量客户端的并发访问。

**b.写原理：**



1.客户端通过调用DistributedFileSystem的create方法创建新文件。

2.DistributedFileSystem通过RPC调用namenode去创建一个没有blocks关联的新文件，创建前， namenode会做各种校验，比如文件是否存在，客户端有无权限去创建等。如果校验通过， namenode就会记录下新文件，否则就会抛出IO异常。

3.前两步结束后，会返回FSDataOutputStream的对象，与读文件的时候相似， FSDataOutputStream被封装成DFSOutputStream。DFSOutputStream可以协调namenode和 datanode。客户端开始写数据到DFSOutputStream，DFSOutputStream会把数据切成一个个小的packet，然后排成队 列data quene。

4.DataStreamer会去处理接受data quene，它先询问namenode这个新的block最适合存储的在哪几个datanode里（比如重复数是3，那么就找到3个最适合的 datanode），把他们排成一个pipeline。DataStreamer把packet按队列输出到管道的第一个datanode中，第一个 datanode又把packet输出到第二个datanode中，以此类推。

5.DFSOutputStream还有一个对列叫ack quene，也是由packet组成，等待datanode的收到响应，当pipeline中的所有datanode都表示已经收到的时候，这时akc quene才会把对应的packet包移除掉。

如果在写的过程中某个datanode发生错误，会采取以下几步：

1) pipeline被关闭掉；

2)为了防止防止丢包ack quene里的packet会同步到data quene里；

3)把产生错误的datanode上当前在写但未完成的block删掉；

4)block剩下的部分被写到剩下的两个正常的datanode中；

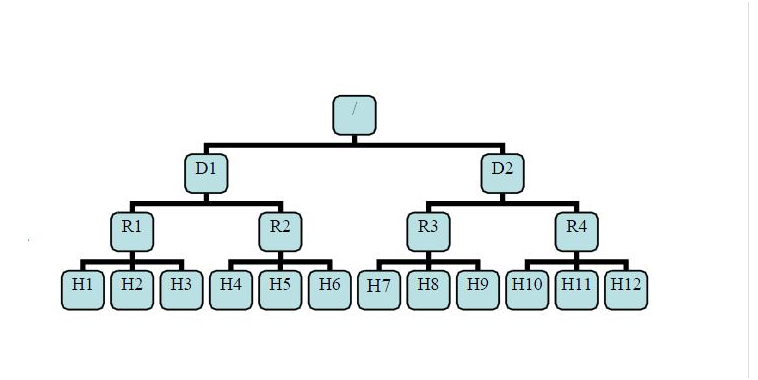
5)namenode找到另外的datanode去创建这个块的复制。当然，这些操作对客户端来说是无感知的。

6.客户端完成写数据后调用close方法关闭写入流。

7.DataStreamer把剩余得包都刷到pipeline里，然后等待ack信息，收到最后一个ack后，通知datanode把文件标视为已完成。

注意：客户端执行write操作后，写完的block才是可见的，正在写的block对客户端是不可见的，只有 调用sync方法，客户端才确保该文件的写操作已经全部完成，当客户端调用close方法时，会默认调用sync方法。是否需要手动调用取决你根据程序需 要在数据健壮性和吞吐率之间的权衡。

**c.机架感知**（Rack）大规模集群



distance(/D1/R1/H1,/D1/R1/H1)=0  相同的datanode

distance(/D1/R1/H1,/D1/R1/H2)=2  同一rack下的不同datanode

distance(/D1/R1/H1,/D1/R1/H4)=4  同一IDC下的不同datanode

distance(/D1/R1/H1,/D2/R3/H7)=6  不同IDC下的datanode

(1)副本存放是HDFS可靠性和性能的关键.优化的副本存放策略正是HDFS某一重要特性之一.

(2)支架的作用:

         ①改进数据的可靠性,可用性,和网络带宽的利用率;

         ②副本存放在不同的机架上,有效防止整个机架失效时数据丢失,并且允许读数据充分利用多个机架的带宽

         ③分布式的集群通常包含非常多的机器，由于受到机架槽位和交换机网口的限制，通常大型的分布式集群都会跨好几个机架，由多个机架上的机器共同组成一个分布式

            集群。机架内的机器之间的网络速度通常都会高于跨机架机器之间的网络速度，并且机架之间机器的网络通信通常受到上层交换机间网络带宽的限制。

 (3)hadoop集群中副本存放准则

        具体到Hadoop集群，由于hadoop的HDFS对数据文件的分布式存放是按照分块block(128M)存储，每个block会有多个副本(默认为3)，并且为了数据的安全和高效，所以

         hadoop默认对3个副本的存放策略为：

            第一个block副本放在和client所在的node里（如果client不在集群范围内，则这第一个node是随机选取的）。

            第二个副本放置在与第一个节点不同的机架中的node中（随机选择）。

            第三个副本似乎放置在与第一个副本所在节点同一机架的另一个节点上

            如果还有更多的副本就随机放在集群的node里。

       注:( 这样的策略可以保证对该block所属文件的访问能够优先在本rack下找到，如果整个rack发生了异常，也可以在另外的rack上找到该block的副本。这样足够的高效，并

       且同时做到了数据的容错。但是，hadoop对机架的感知并非是自适应的，亦即，hadoop集群分辨某台slave机器是属于哪个rack并非是只能的感知的，而是需要hadoop的管理者人为的告知hadoop

       哪台机器属于哪个rack，这样在hadoop的namenode启动初始化时，会将这些机器与rack的对应信息保存在内存中，用来作为对接下来所有的HDFS的写块操作分配

       datanode列表时（比如3个block对应三台datanode）的选择datanode策略，做到Hadoop allocate block的策略：尽量将三个副本分布到不同的rack。)