HDFS中的block、packet、chunk

block

这个大家应该知道，文件上传前需要分块，这个块就是block，一般为128MB，当然你可以去改，不顾不推荐。因为块太小：寻址时间占比过高。块太大：Map任务数太少，作业执行速度变慢。它是最大的一个单位。

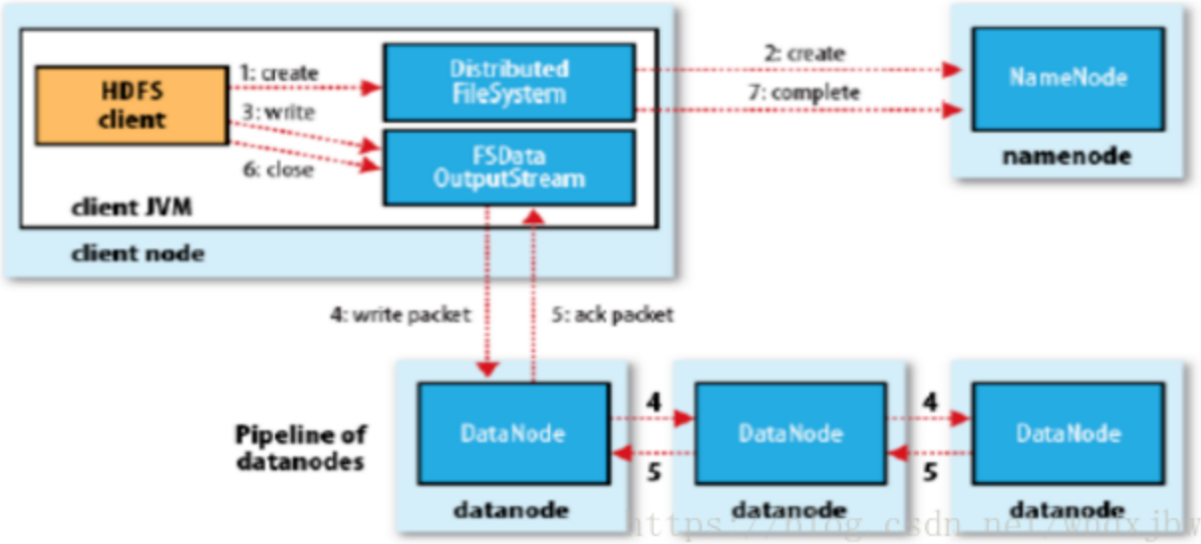
packet

packet是第二大的单位，它是client端向DataNode，或DataNode的PipLine之间传数据的基本单位，默认64KB。

chunk

chunk是最小的单位，它是client向DataNode，或DataNode的PipLine之间进行数据校验的基本单位，默认512Byte，因为用作校验，故每个chunk需要带有4Byte的校验位。所以实际每个chunk写入packet的大小为516Byte。由此可见真实数据与校验值数据的比值约为128 : 1。（即64*1024 / 512）

HDFS写入



1. 客户端向NameNode发出写文件请求。
2. 检查是否已存在文件、检查权限。若通过检查，直接先将操作写入EditLog，并返回输出流对象。
3. client端按块切分文件。
4. client将NameNode返回的分配的可写的DataNode列表和Data数据一同发送给最近的第一个DataNode节点，此后client端和NameNode分配的多个DataNode构成pipeline管道，client端向输出流对象中写数据。client每向第一个DataNode写入

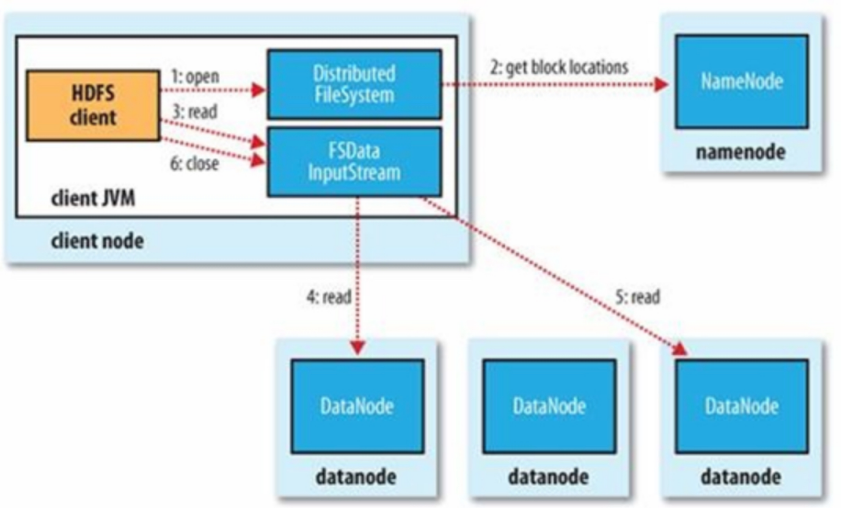
一个packet，这个packet便会直接在pipeline里传给第二个、第三个...DataNode。（注：并不是写好一个块或一整个文件后才向后分发）

5. 每个DataNode写完一个块后，会返回确认信息。

6. 写完数据，关闭输输出流。

7. 发送完成信号给NameNode。

HDFS读取



1.初始化FileSystem，然后客户端(client)用FileSystem的open()函数打开文件

2.FileSystem用RPC调用元数据节点，得到文件的数据块信息，对于每一个数据块，元数据节点返回保存数据块的数据节点的地址。

3.FileSystem返回FSDataInputStream给客户端，用来读取数据，客户端调用stream的read()函数开始读取数据。

4.DFSInputStream连接保存此文件第一个数据块的最近的数据节点，data从数据节点读到客户端(client)

5.当此数据块读取完毕时，DFSInputStream关闭和此数据节点的连接，然后连接此文件下一个数据块的最近的数据节点。

6.当客户端读取完毕数据的时候，调用FSDataInputStream的close函数。

7.在读取数据的过程中，如果客户端在与数据节点通信出现错误，则尝试连接包含此数据块的下一个数据节点。

8. 失败的数据节点将被记录，以后不再连接。【注意：这里的序号不是一一对应的关系】