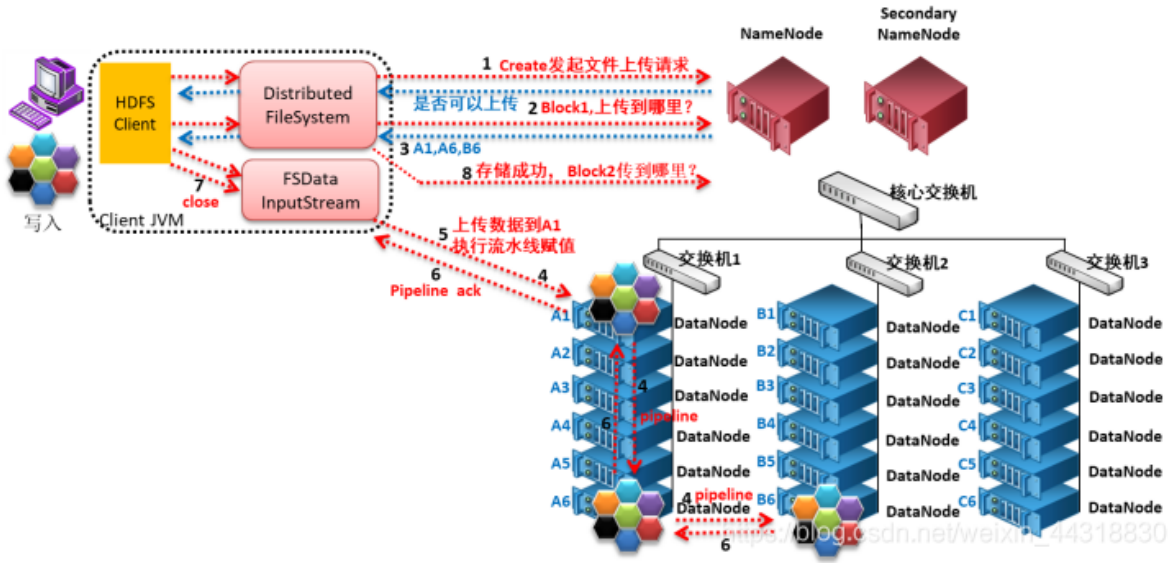
HDFS3 文件读写框架

文件写入和读出得过程是理解HDFS框架的重点。

文件写入过程（重点）



1. client 发起文件上传请求，通过RPC与NameNode建立通讯，NameNode检查目标文件是否存在，父目录是否存在，返回是否可以上传。（RPC是指 远程过程调用，在RPC模块有着专门的讲解。）;NameNode返回结果

2. client 请求第一个block该传输到哪些DataNode服务器上

3. NameNode根据配置文件中指定的备份数量以及机架感知原理进行文件分配，返回可用的DataNode的地址：A,B,C（机架感知原理会有专门的讲解）

4. client请求3台DataNode中的A上传数据（建立pipeline也是一个RPC通道），A收到请求会继续调用B，然后B调用C，将整个pipeline建立起来，然后逐级返回client

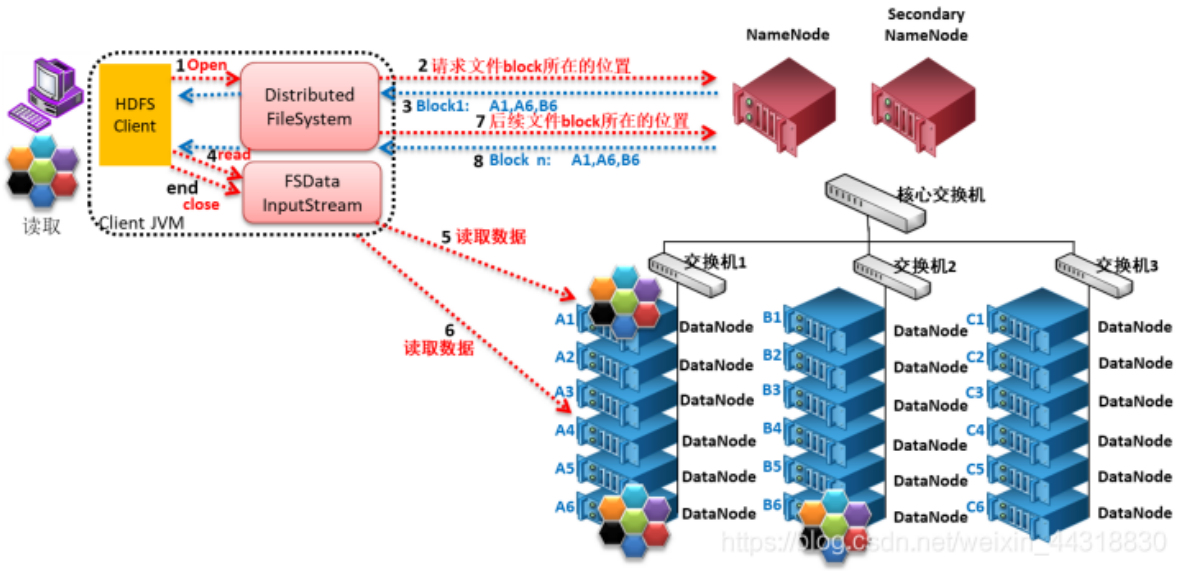
5. client开始往A机器传第一个block（先从磁盘读取数据到一个本地内存缓存），一packet为单位（默认64k），A收到一个packet就会传给B，B传给C，A每传一个packet会放入一个应答队列等待应答

6.数据被分割成一个个packet数据包在pipeline上依次传输，在pipeline反方向上，逐个发生ack（命令正确应答），最终pipeline中第一个DataNode节点A将pipeline ack（管道正确发送命令）发送给client

7. 关闭写入流

8. 当一个block传输完成之后，client再次请求NameNode上传第二个block到服务器。

文件读取过程（重点）



1. 客户端通过调用FileSystem对象的open()来读取希望打开的文件

2. Client 向NameNode发送RPC请求，来确定请求的文件中block所在的位置

3. NameNode会根据情况返回文件的部分或者全部的block列表
对于每个block，NameNode都会返回含有该block副本的DataNode地址；对于这些返回的DataNode地址，会按照集群的拓扑结构得出DataNode与客户端的距离，然后进行排序，排序两个规则：
（1）网络拓扑中距离Client靠得较前；
（2）心跳机制中超时汇报得DataNode为STALE，这样的靠后

4. Client 选取排序靠前的DataNode来读取block，如果客户端本身就是DataNode，那么将直接从本地直接获取数据（短路读取特性）

5. 底层上实质上建立 Socket Stream（FSDataInputStream），重复调用父类的DataInputStream的read方法，直到这个块的数据读取完毕

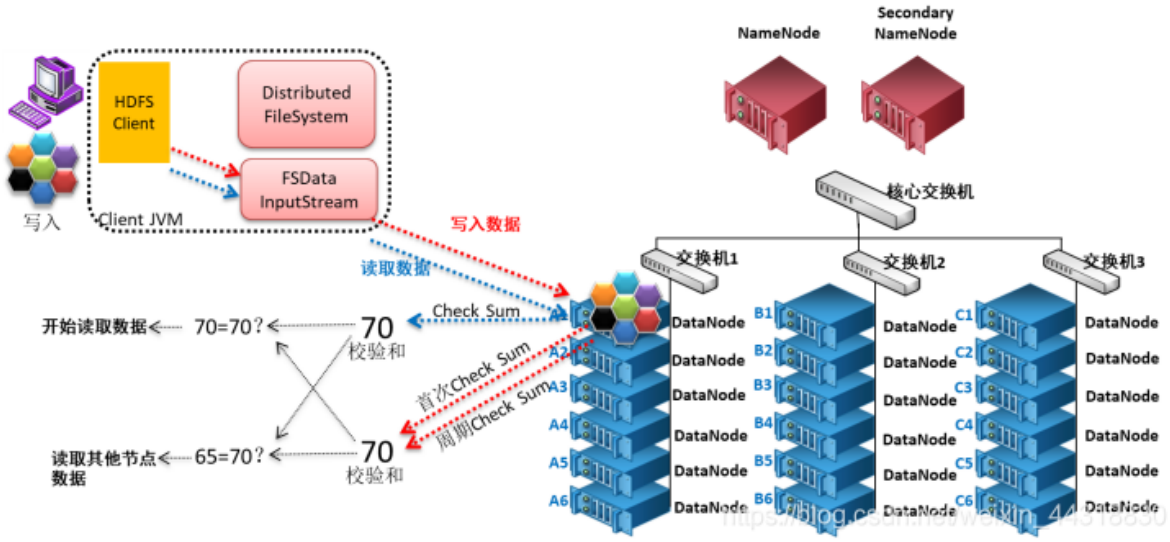
6. 并行读取，如果读取失败就重新读取

7. 当读完列表的block后，若文件读取还没有结束，客户端会继续向NameNode获取下一批的block列表
(注意：
1. 读取完一个block都会进行checksum验证，如果读取DataNode时出现错误，客户端就通知NameNode，然后再从下一个拥有该block的副本DataNode继续读取。
2. read方法是并行读取，不是一块一块读取；NameNode返回Client请求包含的DataNode地址，并不是返回请求块的数据
)

8. 返回后续的block列表，重复执行456操作

9. 最终关闭读流，并将读取来的所有block会合并成一个完整的最终文件。

HDFS数据的完整性（校验和）



1. 数据再写入之后会进行校验和计算，DataNode周期信进行校验和计算，将计算结果与第一次结果进行对比。
（如果相同就没有数据丢失，如果不同就是数据块丢失，然后进行数据恢复操作）
2. 数据读取之前对数据进行校验，与第一次的结果进行对比。若相同表示数据没有丢失，可以读取。若不相同表示数据，有所丢失。需要到其他副本读取。