HDFS数据读写流程

# 1.数据读流程

下图描述了数据读取过程中客户端与之交互的HDFS、namenode和datanode之间的数据流：



图1 客户端读取HDFS中的数据

如上图所示，文件系统客户端在步骤1中用FileSystem对象的open函数打开文件，文件系统使用RPC通信机制与管理节点通信获取该文件的数据块信息(步骤2)。文件的读取通过FSDataInputstream输入流来完成，在步骤1，2后文件系统会返回一个输入流的对象。在步骤3中，客户端通过输入流对象调用read()方法读取数据，在文件系统中每个文件有多个副本，输入流对象会根据系统信息选择连接距离最近的datanode读取文件。

1. client发起open请求

通过DistributedFileSystem.open创建一个DFSInputStream并返回，在DFSInputStream构造函数中，openInfo函数被调用，其主要从Namenode中得到要打开的文件的blocks信息。

1. client向datanode发送连接请求，获取blocks信息

# 数据写流程

Write操作一般是先create文件然后write内容，下面介绍用户调用FileSystem API向HDFS写入数据。下图是从HDFS文件系统中写入数据的流程图：



图1 客户端将数据写入HDFS

文件系统的写入过程相对复杂，而且时间耗费大。在步骤1，2中，Client通过文件系统提供的API，调用分布式文件系统，通过文件系统对象的create方法启动创建文件过程，要完成的工作包括在命名空间创建文件元数据，并返回一个FSDataOutputStream输出流对象。为了保证文件的可靠性，写入多个副本，在输出流中根据副本数选择多个数据节点构成管线，步骤3通过输出流对象将数据其中的一个数据节点，然后数据节点之间进行数据发送（步骤4）。各数据节点在数据写入完成时向管道返回确认信息，在步骤5中管道确认数据写入完毕后，从队列中删除数据包。客户端完成数据的写入后，通过步骤6，7结束数据写入流程。

1) client发起create请求

通过DistributedFileSystem.create方法创建文件，其实是通过DistributedFileSystem内部Wrapper的DFSClient向Namenode发送RPC请求。与元数据相关的FileSystem API都转移到dfsclient上，然后通过与Namenode的RPC完成。

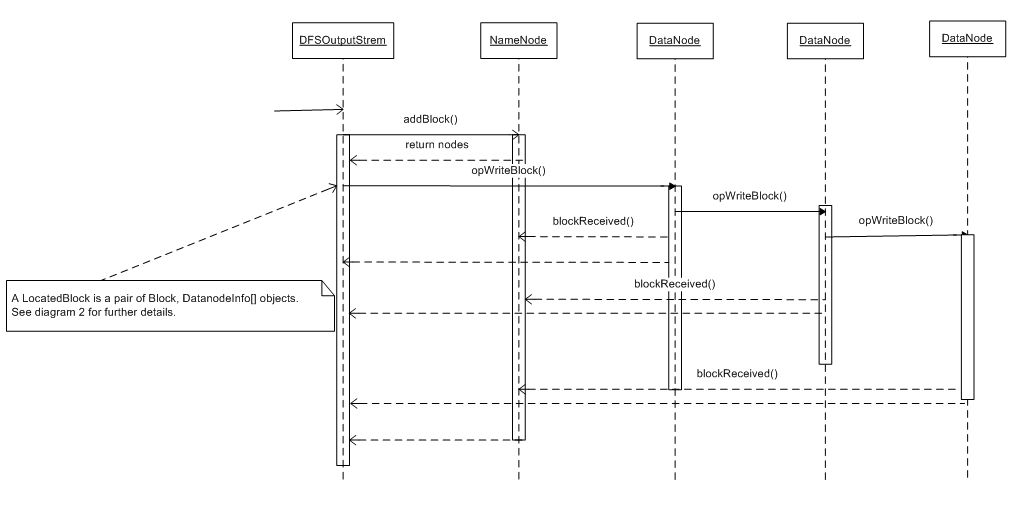
create操作生成DFSOutputStream对象，构造这个对象首先是向 Namenode发送create RPC，然后构造并启动DataStreamer线程，最后开启Lease定时Renew机制LeaseRenewer。

2) NameNode处理Create请求，然后给client发送response

Create RPC到达NameNode之后依次走NameNodeRpcServer.create()->

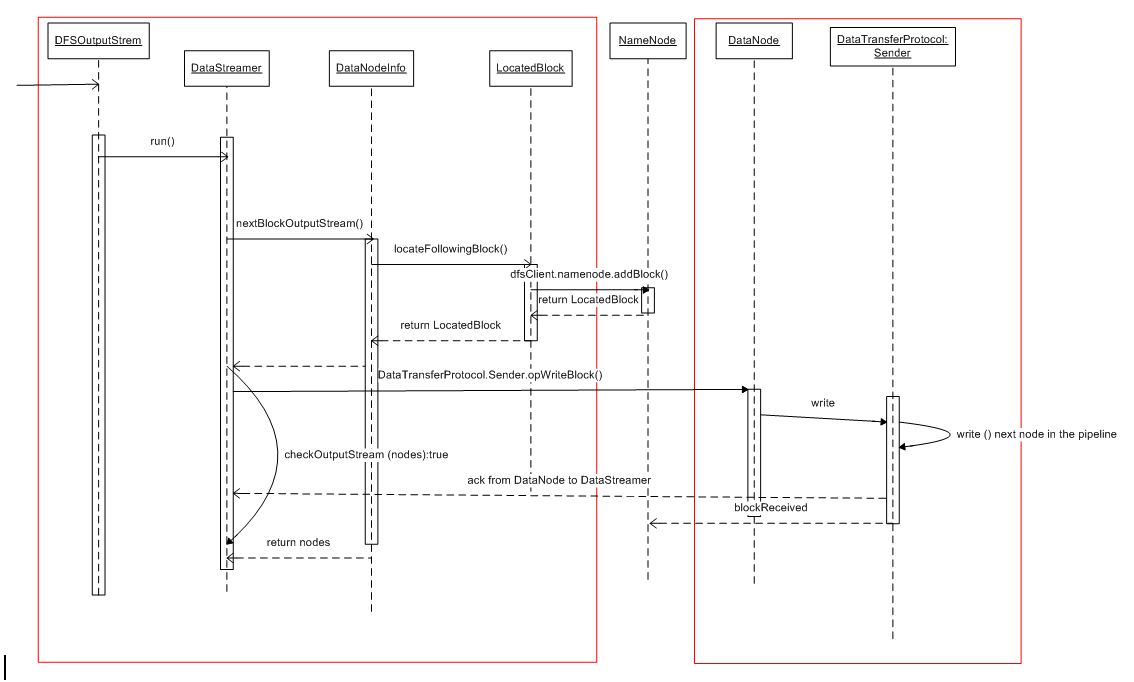
FSNamesystem.startFile()->startFileInt()->startFileInternal()

3)Client 向Namenode请求分配Block，并向多个Datanode建立pipeline传输数据



HDFS文件以一个一个Block的形式写入HDFS, Client首先向NameNode发送addBlock RPC请求，NameNode返回一个LocatedBlock对象包含该Block的元数据信息和对应的三副本所在的DataNode。然后client向pipelinr中的第一个DataNode发起TCP连接。

DFSoutputStream中的内部类DataStreamer是写入数据的操作逻辑，是整个write操作的核心，如下图所示：



在DFSoutputstream中有两个队列，dataQueue和ackQueue，整个写数据是以BlOCK为单位。