1.hdfs不适合大量小文件存储的原因是什么？

答：（1）NameNode将文件系统的元数据储存在内存中，存储的文件数受限于NameNode的内存大小，如果存放的文件数目过多会占用大量的内存。

（2）hdfs适用于高吞吐量，不适合低时间延迟的访问，同时存入大量小文件会花费很长时间。

（3）流式读取的方式不适合多用户写入。如果访问小文件，则必须从一个datanode跳转到另一个datanode，就大大降低了读取性能。

1. 简述client、namenode、sancondarynamenode、datanode的作用。

答：client：即客户端。

（1）文件的切分。文件上传到HDFS的时候，Client将文件切分成一个一个的Block，然后存储。

（2）与NameNode交互，获取文件的位置信息。

（3）与Datanode交互，读取或者写入数据。

（4）Client对HDFS进行访问与管理。

Namenode：即master，是一个管理者。

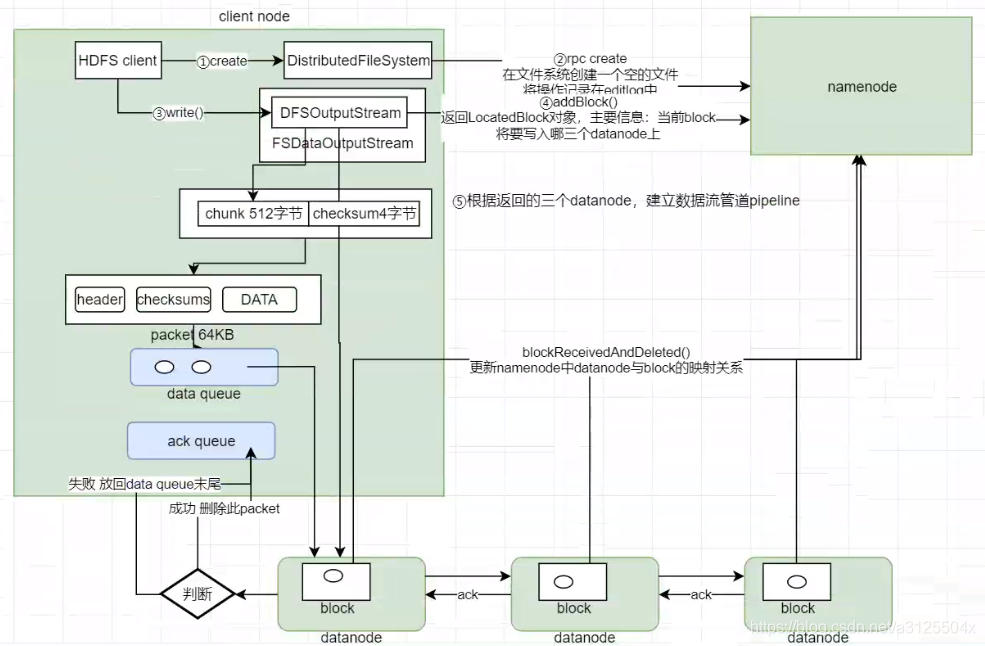
1. 管理HDFS的名称空间。
2. 管理数据块映射信息。
3. 配置副本策略。
4. 处理客户端读写请求。

Secondary namenode: 当namenode挂掉的时候，它并不替换掉namenode提供服务。

1. 辅助namenode，分担工作量。
2. 定期合并fsimage和fsedits，并推送给namenode。
3. 可在紧急情况下辅助恢复namenode。

Datanode：即slave，执行namenode分配的操作。

1. 存储实际的数据块。
2. 执行数据块的读/写操作。
3. hdfs的读写流程。
4. 写数据流程。



1.客户端通过Distributed FileSystem模块向NameNode请求上传文件，NameNode检查目标文件是否已存在，父目录是否存在

2.NameNode返回是否可以上传

3.客户端请求第一个 Block上传到哪几个DataNode服务器上（就近原则，离当前client近的优先）

4.NameNode返回3个DataNode节点，分别为dn1、dn2、dn3

5.数据切割，先按照chunk和checksum组成小包（512bytes+4bytes），然后塞入packet（64KB）。

HDFS传输数据生成的crc文件实际上就是校验文件。

6.HDFS以packet为单位传输文件。此时分两个流向：

- datanode01->datanode02->datanode03

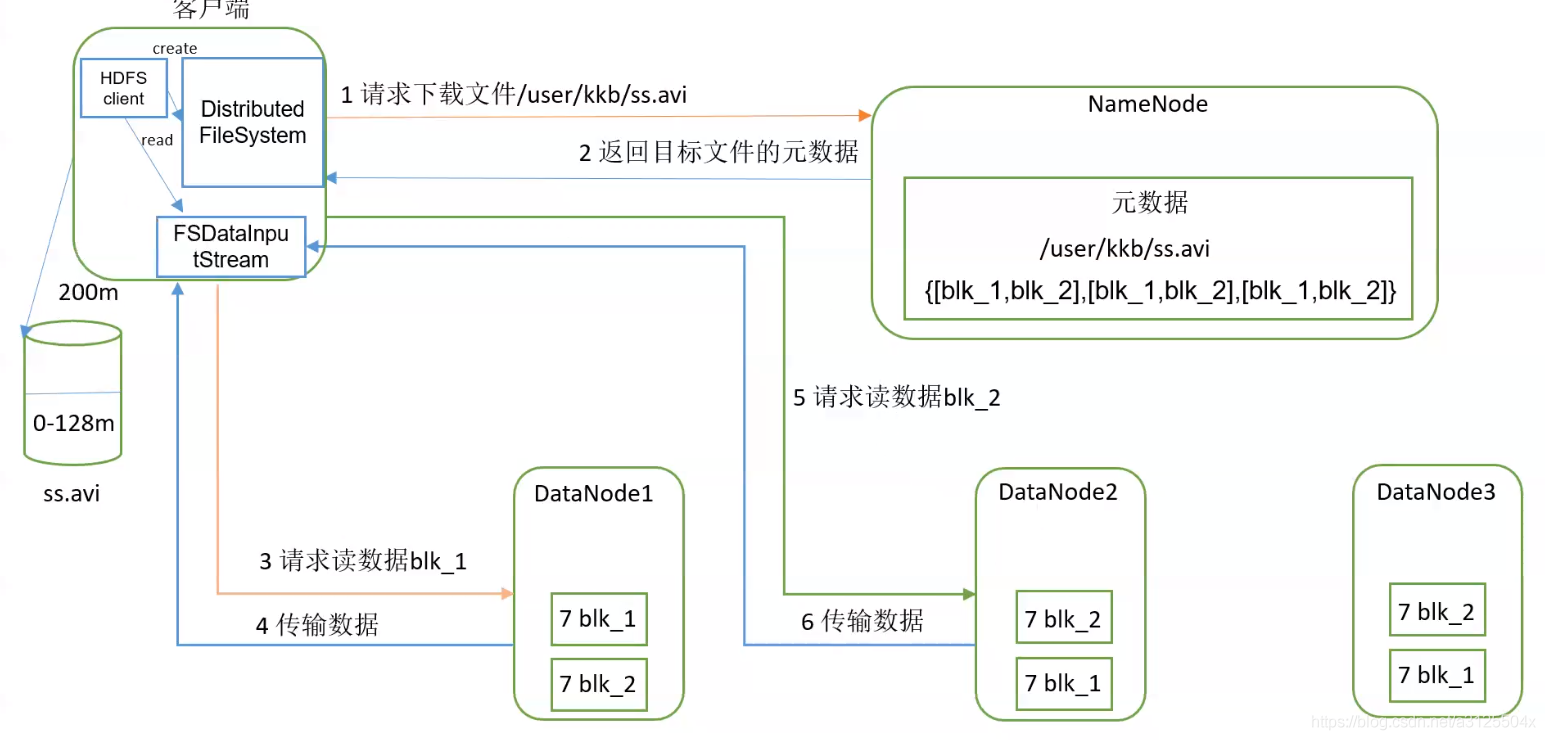
- ack queue

7.当传送成功一个packet后，逆方向返回结果datanode03->datanode02->datanode01。然后删除ack queue中的备份

8.当向datanode数据传出完成后，更新NameNode中的映射关系。

9.开始传输下一个block，重复5-8

1. 读数据流程。



1.客户端通过Distributed FileSystem向NameNode请求下载文件，NameNode通过查询元数据，找到文件块所在的DataNode地址

2.挑选一台DataNode（就近原则，然后随机）服务器，请求读取数据

3.DataNode开始传输数据给客户端（从磁盘里面读取数据输入流，以Packet为单位来做校验）

4.客户端以Packet为单位接收，先在本地缓存，然后写入目标文件。最终整合成block

5.读完一个block之后，再读取下一个block