Hdfs基本操作

1. 创建目录 bin/hadoop fs -mkdir /zzy
2. 删除目录 bin/hadoop fs -rm -r /zzy
3. 向hadoop的目录传文件：bin/hadoop fs -put 源 目标
4. 从hadoop系统中下载文件:bin/hadoop fs -get 源 目标
5. 查看某一文件内容 bin/hadoop fs -cat 文件
6. 查看文件的详细信息 hdfs fsck 文件名 -files -blocks -locations
7. 检测块丢失 hdfs fsck -list-corruptfileblocks
8. 有时候datanode 加载不上 先删除current 再格式化

Hdfs API操作

1. 两个基本类型：

struct hdfsBuilder \*pbld = NULL; hdfsFS my\_hdfsfs;

1. hdfsBuilderSetNameNode(pbld,masterip.c\_str());
2. hdfsBuilderSetNameNodePort(pbld,masterport);
3. hdfsBuilderConfSetStr(pbld,"fs.hdfs.impl","org.apache.hadoop.hdfs.DistributedFileSystem");
4. my\_hdfsfs = hdfsBuilderConnect(pbld);
5. int CheckHDFS = hdfsExists(my\_hdfsfs,"/");
6. int filenum = 0;

hdfsFileInfo \* fileinfo = hdfsListDirectory(my\_hdfsfs, "/data/pictures", &filenum);

1. hdfsOpenFile(my\_hdfsfs, filename.c\_str(), O\_RDONLY, 0, 2, 0);
2. hdfsDisconnect(my\_hdfsfs);
3. struct hdfsStreamBuilder \*temp;

temp = hdfsStreamBuilderAlloc(my\_hdfsfs,filefullpath.c\_str(),O\_RDONLY);

hdfsFile my\_hdfsfile = hdfsStreamBuilderBuild(temp);

1. hdfsFile hdfstmpfile = hdfsOpenFile(my\_hdfsfs, filename.c\_str(), O\_RDONLY, 0, 2, 0);

tSize result = hdfsRead(my\_hdfsfs, hdfstmpfile, (void\*)buffer, length);

1. 确定指针偏移量和文件当前偏移量

int hdfsSeek(hdfsFS fs, hdfsFile file, tOffset desiredPos);

tOffset hdfsTell(hdfsFS fs, hdfsFile file);

1. 从mblockSize位置开始，往后查看一个mblockSize大小的内容属于什么host

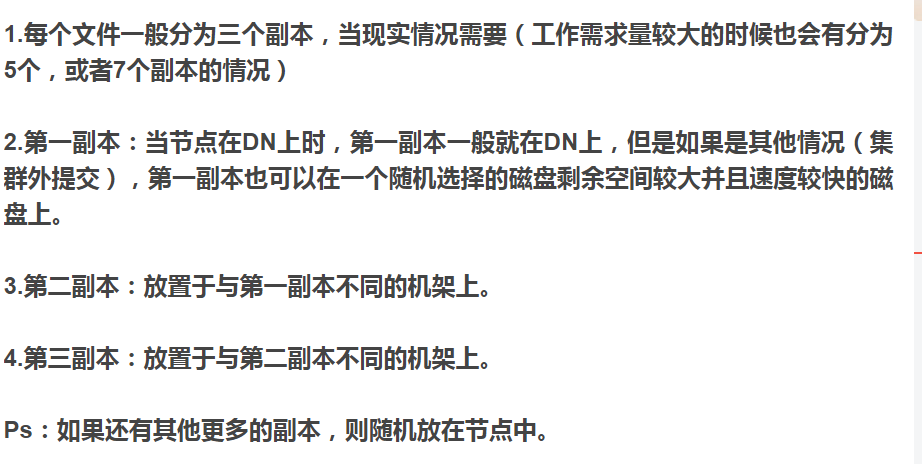
char\*\*\* filehost;

Filehost = hdfsGetHosts(my\_hdfsfs, filefullpath.c\_str(),fileinfo->mBlockSize, fileinfo->mBlockSize);

cout << "machine[0][0] :" << filehost[0][0]<<endl;

心得体会

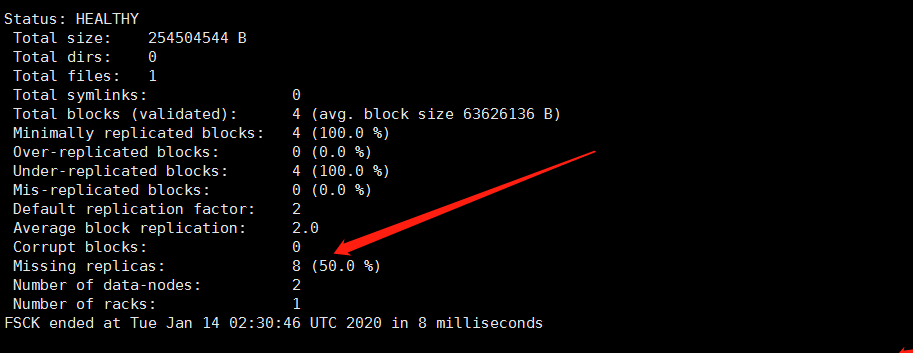
能通过HDFSC的接口得到副本所在机器了。另外，按照hdfs的副本放置规则，第一个副本在本机，第二个副本在和第一副本不一样的机器。第三副本放置在与第二副本不一样的机器。我测试了一下project-master-nostream.tar.gz （250M,块大小64M）分别是4个副本和两个副本的时候，因为集群是slave06和slave07两台机器，所以当设置4个副本的时候，保留了slave06 slave07两个机器的副本，同时丢失了另外两个副本，副本丢失率也就是50%,当有2个副本的时候，保留了slave06 slave07两个机器的副本，丢失副本是0。感觉hdfs是一台机器放一个副本。如果副本数超过机器数，就自动丢失多余的副本。



执行:

hdfs fsck /006zzy/project5-master-nostream.tar.gz -files -blocks -locations

4个副本的时候：



2个副本的时候：

