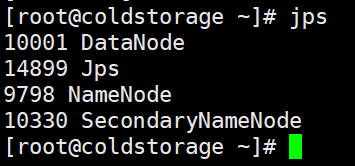
###### hdfs\_C\_API调用实现过程

## API环境搭建过程

### 确认hadoop**伪分布式**环境部署正确



注：一定要保证NameNode、DataNode、SecondaryName都存在，否则hadoop启动失败。

进入到hadoop目录中，初始化、启动、关闭hadoop操作对应的命令：

cd /usr/local/hadoop

./bin/hdfs namenode -format

./sbin/start-dfs.sh

./sbin/stop-dfs.sh

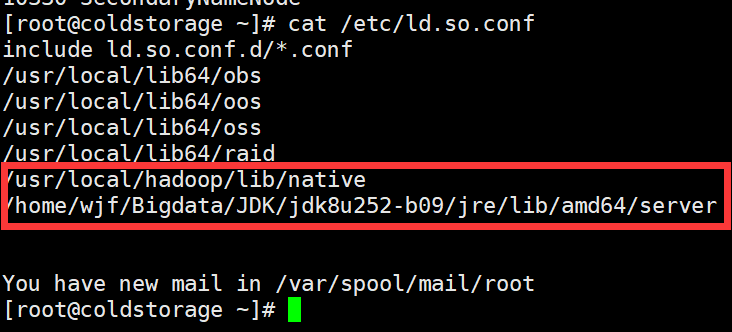
### 确认库环境正确

cat /etc/ld.so.conf

ldconfig -v | grep hdfs

ldconfig -v | grep lvm

find / -name libjvm.so



注：这里两个文件目录分别存放着libhdfs.so 和 liblvm.so文件。

### 确认~/.bashrc环境正确

1. java环境

#export JAVA\_HOME=/usr/local/jdk-11.0.3

export JAVA\_HOME=/home/wjf/Bigdata/JDK/jdk8u252-b09

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

export PATH=${JAVA\_HOME}/bin:$PATH

【2】hadoop和hadoop依赖库环境

export HADOOP\_HOME=/usr/local/hadoop

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export PATH=${HADOOP\_HOME}/bin:$PATH

export CLASSPATH=${CLASSPATH}":"`find ${HADOOP\_HOME}/share/hadoop | awk '{path=path":"$0}END{print path}'`

### 编写 hdfs.c 测试代码

#include "hdfs.h"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main(int argc, char \*\*argv)

{

hdfsFS fs = hdfsConnect("**192.168.226.11**", **9200**);

const char\* writePath = "**/user/wjf/testfile1.txt**";

hdfsFile writeFile = hdfsOpenFile(fs, writePath, O\_WRONLY|O\_CREAT, 0, 0, 0);

if(!writeFile)

{

fprintf(stderr, "Failed to open %s for writing!\n", writePath);

exit(-1);

}

char\* buffer = "Hello, World!\n";

tSize num\_written\_bytes = hdfsWrite(fs, writeFile, (void\*)buffer, strlen(buffer)+1);

if (hdfsFlush(fs, writeFile))

{

fprintf(stderr, "Failed to 'flush' %s\n", writePath);

exit(-1);

}

hdfsCloseFile(fs, writeFile);

return 0;

}

注：该程序会写入信息到一个文件中，并将该文件上传hdfs。

### 创建Makefile文件

INCLUDES = -I /usr/local/hadoop/include

SRCS = $(wildcard \*.c)

OBJS = $(SRCS:.c=.o)

CC = gcc

CFLAGS = -Wall -O -g

LIBS = -L /usr/local/hadoop/lib/native -lhdfs -lpthread -lm

LIB\_RAID = c\_hdfs.so

EXE = c\_hdfs

all: $(LIB\_RAID) $(EXE)

$(EXE):$(OBJS)

$(CC) $^ -o $@ $(LIBS)

$(LIB\_RAID): $(OBJS)

$(CC) $(CFLAGS) $(INCLUDES) $(LIBS) $^ -o $@ -fPIC -shared

$(OBJS):%.o:%.c

$(CC) $(CFLAGS) $(INCLUDES) -c $< -o $@ -fPIC -shared

install:

# install libraid.so ../lib

# install libraid.so ../../../lib64/libraid.so

# install libraid.so /usr/local/lib64/raid/libraid.so

# ldconfig 2> /dev/null

clean:

rm -rf $(LIB\_RAID) $(OBJS) $(EXE)

注：gcc hdfs.c -o c\_hdfs -I /usr/local/hadoop/include -L /usr/local/hadoop/lib/native -lhdfs

hdfs.h ： /usr/local/hadoop/include/hdfs.h

libhdfs.so ： /usr/local/hadoop/lib/native/libhdfs.so

### run.sh -- 用于查看和读取文件

#!/bin/bash

echo "hdfs文件目录:"

hdfs dfs -ls /user/wjf

echo "hdfs文件内容:"

hdfs dfs -cat /user/wjf/testfile1.txt

### delete.sh -- 用于删除文件

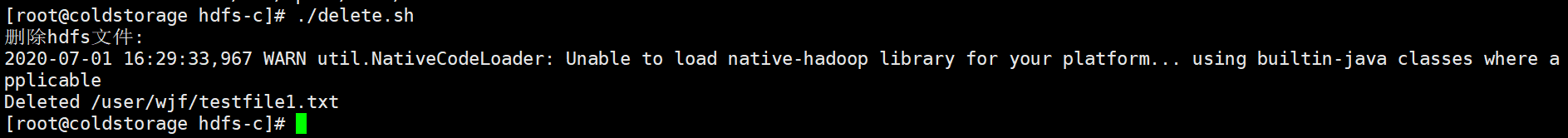
#!/bin/bash

echo "删除hdfs文件:"

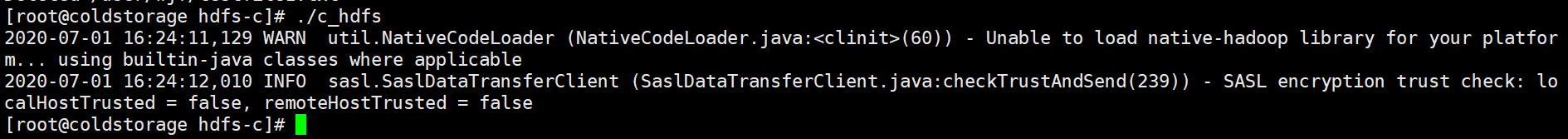
hdfs dfs -rm /user/wjf/testfile1.txt

## 执行结果

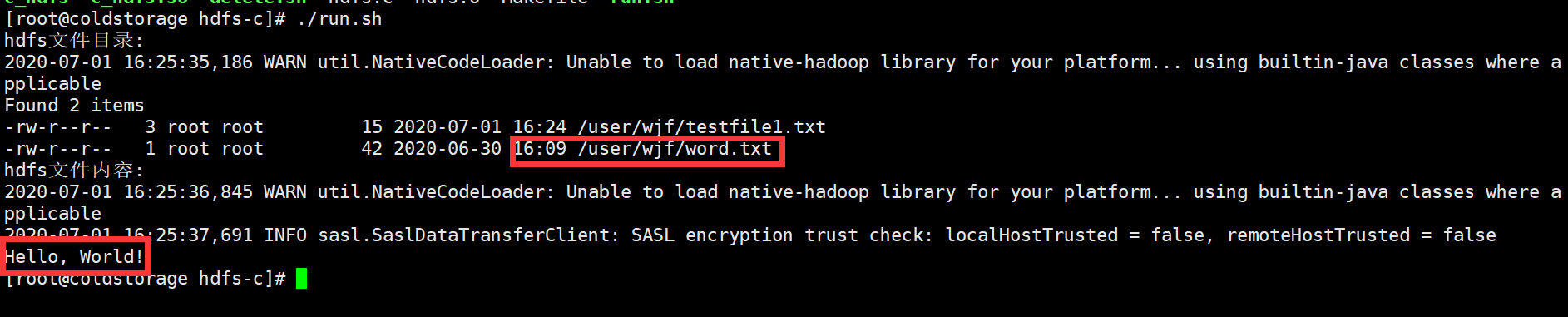
### 执行delete.sh脚本



### 执行生成的可执行程序

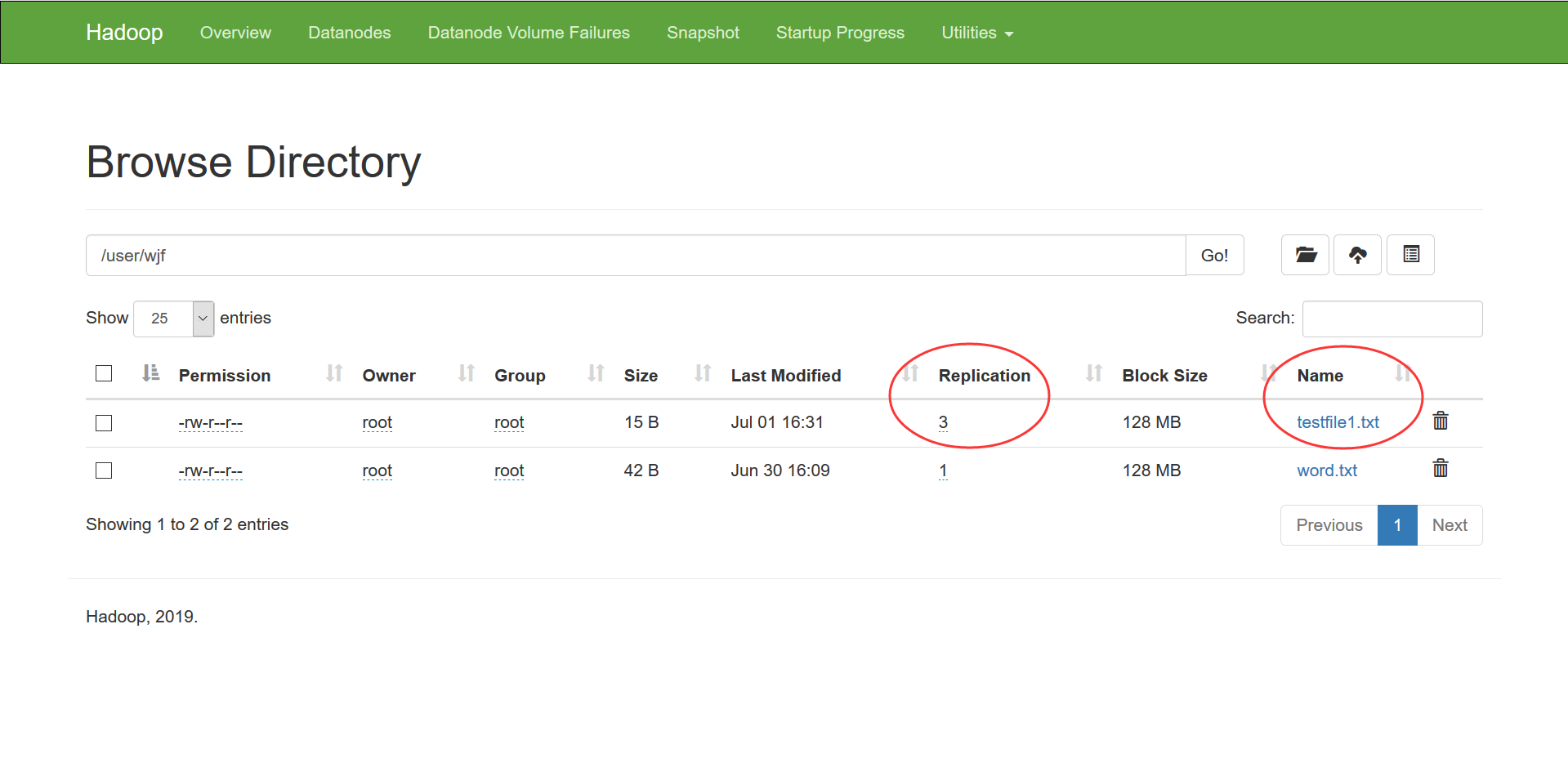


### 执行run.sh脚本



### 通过浏览器查看

<http://192.168.226.11:9870/explorer.html#/user/wjf>



## 参考资料

【1】C-API-libhdfs

https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/LibHdfs.html

【2】Hadoop通过C的API访问HDFS

https://blog.csdn.net/deqingguo/article/details/7004898

【3】使用Hadoop的C API操作HDFS

https://blog.csdn.net/cjf\_wei/article/details/76374389

【4】大数据Hadoop Streaming编程实战之C、Php、Python

https://cloud.tencent.com/developer/news/129802

【5】Hadoop通过C API访问HDFS文件系统

https://blog.csdn.net/weixin\_38308151/article/details/79541623

【6】libhdfs 的C API

https://blog.csdn.net/cjfeii/article/details/19203731?utm\_medium=distribute.pc\_relevant.none-task-blog-BlogCommendFromBaidu-9.nonecase&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-BlogCommendFromBaidu-9.nonecase

【7】基于libhdfs库操作HDFS文件

https://blog.csdn.net/byxdaz/article/details/103216540?utm\_medium=distribute.pc\_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase

【8】HDFS简介及用C语言访问HDFS接口操作实践

<https://blog.csdn.net/zhouzhaoxiong1227/article/details/47448227>