上文中讲解如何配置WebHDFS以及WebHDFS的基本操作步骤，接下来我们将详细介绍WebHDFS命令的组织方式以及具体命令实现。

2.2.5.1  **启动Hadoop集群**

在进行实验之前，首先需要确保Hadoop集群已经正常启动。如果Hadoop没有启动，需要通过下列步骤启动Hadoop集群。

**在模板中，我们已经配置好了Hadoop伪分布式环境，同学们不需要再次配置，可以直接启动使用。**

**步骤1.**  **启动Hadoop**

打开一个终端模拟器，通过下列命令启动Hadoop：

start-all.sh

命令运行后的输出内容如下所示：

[root@master ~]# start-all.sh

This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh

18/11/08 14:42:12 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Starting namenodes on [master]

master: starting namenode, logging to /usr/cx/hadoop-2.7.1/logs/hadoop-root-namenode-master.out

localhost: starting datanode, logging to /usr/cx/hadoop-2.7.1/logs/hadoop-root-datanode-master.out

Starting secondary namenodes [master]

master: starting secondarynamenode, logging to /usr/cx/hadoop-2.7.1/logs/hadoop-root-secondarynamenode-master.out

18/11/08 14:42:30 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

starting yarn daemons

starting resourcemanager, logging to /usr/cx/hadoop-2.7.1/logs/yarn-root-resourcemanager-master.out

localhost: starting nodemanager, logging to /usr/cx/hadoop-2.7.1/logs/yarn-root-nodemanager-master.out

[root@master ~]#

由输出内容可以看到，每个进程在启动以及运行过程中，会将相关的日志信息记录到日志文件中，因此当进程出现问题时，一个有效的办法就是查看日志文件进行排错处理。

**步骤2.**  **验证Hadoop是否启动成功**

通过下列命令，查看相应的JVM进程确定Hadoop是否启动成功：

jps

当有以下5个进程启动时，则说明Hadoop已经成功启动，如果某个进程没有启动，则需要查看日志文件确定错误发生的原因：

[root@master ~]# jps

3631 NameNode                                     （注：HDFS元数据管理进程）

1679 VmServer.jar

3757 DataNode                                     （注：HDFS数据存储进程）

4488 Jps

3922 SecondaryNameNode                           （注：HDFS数据检查点进程）

4174 NodeManager                                  （注：任务执行进程）

4073 ResourceManager                             （注：资源管理进程）

[root@master ~]#

**步骤3.**  **检测WebHDFS是否可用**

Hadoop启动之后，使用下面的命令检测WebHDFS是否可用：

curl -i "http://master:50070/webhdfs/v1/?user.name=hadoop&op=LISTSTATUS"

命令运行后的返回结果如下所示（当出现类似于HTTP/1.1 200 OK的返回结果时，则说明WebHDFS测试成功）：

[root@master ~]# curl -i "http://master:50070/webhdfs/v1/?user.name=hadoop&op=LISTSTATUS"

HTTP/1.1 200 OK

Cache-Control: no-cache

Expires: Mon, 12 Nov 2018 01:50:29 GMT

Date: Mon, 12 Nov 2018 01:50:29 GMT

Pragma: no-cache

Expires: Mon, 12 Nov 2018 01:50:29 GMT

Date: Mon, 12 Nov 2018 01:50:29 GMT

Pragma: no-cache

Content-Type: application/json

Set-Cookie: hadoop.auth="u=hadoop&p=hadoop&t=simple&e=1542023429812&s=RNeyxX6k2N1vxOby9MVGF5e28Yw="; Path=/; Expires=???, 12-???-2018 11:50:29 GMT; HttpOnly

Transfer-Encoding: chunked

Server: Jetty(6.1.26)

（注：下面的内容为当前HDFS中的相关文件属性信息）

{"FileStatuses":{"FileStatus":[

{"accessTime":0,"blockSize":0,"childrenNum":2,"fileId":16680,"group":"supergroup","length":0,"modificationTime":1541729530634,"owner":"root","pathSuffix":"archive","permission":"755","replication":0,"storagePolicy":0,"type":"DIRECTORY"},

{"accessTime":0,"blockSize":0,"childrenNum":1,"fileId":16709,"group":"supergroup","length":0,"modificationTime":1541729737327,"owner":"root","pathSuffix":"distcpdir","permission":"755","replication":0,"storagePolicy":0,"type":"DIRECTORY"},

{"accessTime":0,"blockSize":0,"childrenNum":2,"fileId":16761,"group":"supergroup","length":0,"modificationTime":1541729820526,"owner":"root","pathSuffix":"distcpdir1","permission":"755","replication":0,"storagePolicy":0,"type":"DIRECTORY"},

{"accessTime":0,"blockSize":0,"childrenNum":4,"fileId":16693,"group":"supergroup","length":0,"modificationTime":1541729620103,"owner":"root","pathSuffix":"files.har","permission":"755","replication":0,"storagePolicy":0,"type":"DIRECTORY"},

{"accessTime":0,"blockSize":0,"childrenNum":2,"fileId":16393,"group":"supergroup","length":0,"modificationTime":1541664344842,"owner":"root","pathSuffix":"tmp","permission":"755","replication":0,"storagePolicy":0,"type":"DIRECTORY"},

{"accessTime":0,"blockSize":0,"childrenNum":2,"fileId":16386,"group":"supergroup","length":0,"modificationTime":1475131797317,"owner":"root","pathSuffix":"user","permission":"755","replication":0,"storagePolicy":0,"type":"DIRECTORY"}

]}}

[root@master ~]#

2.2.5.2  **创建文件并写入内容**

**步骤1.**  **创建输入源文件**

1.   通过下面的命令在本地创建输入数据源文件：

echo "this is input file" > /home/webfile1.txt

2.   通过下面的命令测试数据源文件是否创建成功：

cat /home/webfile1.txt

命令运行后的返回结果如下所示（可见文件中已经成功写入相关的内容）：

[root@master home]# cat /home/webfile1.txt

this is input file

[root@master home]#

**步骤2.**  **创建webhdfsFile文件**

通过下面的命令，使用root用户在HDFS中创建文件webhdfsFile：

curl -i -X PUT "http://master:50070/webhdfs/v1/webhdfsFile?user.name=root&op=CREATE"

上述命令运行后的返回结果如下所示：

[root@master ~]# curl -i -X PUT "http://master:50070/webhdfs/v1/webhdfsFile?user.name=root&op=CREATE"

HTTP/1.1 307 TEMPORARY\_REDIRECT

Cache-Control: no-cache

Expires: Mon, 12 Nov 2018 02:31:36 GMT

Date: Mon, 12 Nov 2018 02:31:36 GMT

Pragma: no-cache

Expires: Mon, 12 Nov 2018 02:31:36 GMT

Date: Mon, 12 Nov 2018 02:31:36 GMT

Pragma: no-cache

Content-Type: application/octet-stream

Set-Cookie: hadoop.auth="u=root&p=root&t=simple&e=1542025896530&s=YcVRVT46udRU0jHwngZNW40XdCc="; Path=/; Expires=???, 12-???-2018 12:31:36 GMT; HttpOnly

Location: <http://master:50075/webhdfs/v1/webhdfsFile?op=CREATE&user.name=root&namenoderpcaddress=master:9000&overwrite=false>

（注：Location位置信息中包括已创建文件所在的DataNode地址及创建路径，需要通过此路径写入内容）

Content-Length: 0

Server: Jetty(6.1.26)

[root@master ~]#

使用上述命令之后，会返回一个Location位置信息，它包括已创建文件所在的DataNode地址及创建路径。

**步骤3.**  **文件内容写入**

结合上文返回的Location位置信息，通过下面的命令就可以将本地文件/home/webfile1.txt中的内容写入到DataNode对应路径下的文件内：

curl -i -X PUT -T /home/webfile1.txt "http://master:50075/webhdfs/v1/webhdfsFile?op=CREATE&user.name=root&namenoderpcaddress=master:9000&overwrite=false"

命令运行后的返回结果如下所示：

[root@master ~]# curl -i -X PUT -T /home/webfile1.txt "http://master:50075/webhdfs/v1/webhdfsFile?op=CREATE&user.name=root&namenoderpcaddress=master:9000&overwrite=false"

HTTP/1.1 100 Continue

HTTP/1.1 201 Created

Location: hdfs://master:9000/webhdfsFile

Content-Length: 0

Connection: close

[root@master ~]#

**步骤4.**  **查看文件内容**

使用下面的命令验证webhdfsFile中的内容是否写入成功：

hadoop fs -cat /webhdfsFile

命令运行后的返回结果如下所示：

[root@master ~]# hadoop fs -cat /webhdfsFile

18/11/12 10:36:01 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

this is input file

[root@master ~]#

由返回结果可以看到，通过上述操作已经成功完成文件的创建以及文件内容的写入。

2.2.5.3  **文件内容追加**

如果希望向文件中追加内容，首先需要获取待追加内容的文件所在地址，然后根据获取到的地址信息实现向文件中追加内容。

**步骤1.**  **创建输入源文件**

1.   通过下面的命令在本地创建输入数据源文件：

echo "this is append file" > /home/webfile2.txt

2.   通过下面的命令测试数据源文件是否创建成功：

cat /home/webfile2.txt

命令运行后的返回结果如下所示：

[root@master ~]# cat /home/webfile2.txt

this is append file

[root@master ~]#

**步骤2.**  **获取文件位置**

通过下面的命令获取待追加内容的文件/webhdfsFile所在地址：

curl -i -X POST "http://master:50070/webhdfs/v1/webhdfsFile?user.name=root&op=APPEND"

命令运行后的返回结果如下所示：

[root@master ~]# curl -i -X POST "http://master:50070/webhdfs/v1/webhdfsFile?user.name=root&op=APPEND"

HTTP/1.1 307 TEMPORARY\_REDIRECT

Cache-Control: no-cache

Expires: Mon, 12 Nov 2018 08:25:42 GMT

Date: Mon, 12 Nov 2018 08:25:42 GMT

Pragma: no-cache

Expires: Mon, 12 Nov 2018 08:25:42 GMT

Date: Mon, 12 Nov 2018 08:25:42 GMT

Pragma: no-cache

Content-Type: application/octet-stream

Set-Cookie: hadoop.auth="u=root&p=root&t=simple&e=1542047142091&s=aVnv+JOvvO4aINvXBcEIAkad7tc="; Path=/; Expires=???, 12-???-2018 18:25:42 GMT; HttpOnly

Location: http://master:50075/webhdfs/v1/webhdfsFile?op=APPEND&user.name=root&namenoderpcaddress=master:9000

（注：Location位置信息中包括文件所在的DataNode地址及创建路径，需要通过此路径写入内容）

Content-Length: 0

Server: Jetty(6.1.26)

[root@master ~]#

**步骤3.**  **文件内容追加**

通过下面的命令，结合返回内容的Location信息实现文件内容追加：

curl -i -X POST -T /home/webfile2.txt "<http://master:50075/webhdfs/v1/webhdfsFile?op=APPEND&user.name=root&namenoderpcaddress=master:9000>"

命令运行后的返回结果如下所示：

[root@master ~]# curl -i -X POST -T /home/webfile2.txt "http://master:50075/webhdfs/v1/webhdfsFile?op=APPEND&user.name=root&namenoderpcaddress=master:9000"

HTTP/1.1 100 Continue

HTTP/1.1 200 OK

Content-Length: 0

Connection: close

[root@master ~]#

**步骤4.**  **查看文件内容**

使用下面的命令验证webhdfsFile中的内容是否追加成功：

hadoop fs -cat /webhdfsFile

命令运行后的返回结果如下所示：

[root@master ~]# hadoop fs -cat /webhdfsFile

18/11/12 16:27:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

this is input file

this is append file

[root@master ~]#

由返回结果可以看到，通过上述操作已经成功完成了文件内容的追加。

2.2.5.4  **打开并读取文件内容**

通过下面的命令，可以打开并读取/webhdfsFile文件中的内容：

curl -i -L "http://master:50070/webhdfs/v1/webhdfsFile?op=OPEN"

命令运行后的返回结果如下所示：

[root@master ~]# curl -i -L "http://master:50070/webhdfs/v1/webhdfsFile?op=OPEN"

HTTP/1.1 307 TEMPORARY\_REDIRECT

Cache-Control: no-cache

Expires: Mon, 12 Nov 2018 08:32:53 GMT

Date: Mon, 12 Nov 2018 08:32:53 GMT

Pragma: no-cache

Expires: Mon, 12 Nov 2018 08:32:53 GMT

Date: Mon, 12 Nov 2018 08:32:53 GMT

Pragma: no-cache

Content-Type: application/octet-stream

Location: http://master:50075/webhdfs/v1/webhdfsFile?op=OPEN&namenoderpcaddress=master:9000&offset=0

Content-Length: 0

（注：首先会打印出文件的Location信息）

Server: Jetty(6.1.26)

HTTP/1.1 200 OK

Access-Control-Allow-Methods: GET

Access-Control-Allow-Origin: \*

Content-Type: application/octet-stream

Connection: close

Content-Length: 39

this is input file

this is append file

（注：最后会输出文件的内容信息）

[root@master ~]#

由返回结果可以看出，使用命令进行文件内容读取时，首先会返回文件所在的Location信息，然后会打印文件的具体内容。

2.2.5.5  **创建文件夹**

通过下面的命令可以进行文件夹创建（我们在此创建一个名称为webhdfsDir的文件夹）：

curl -i -X PUT "<http://master:50070/webhdfs/v1/webhdfsDir?user.name=root&op=MKDIRS>"

命令运行后的返回结果如下所示：

[root@master ~]# curl -i -X PUT "http://master:50070/webhdfs/v1/webhdfsDir?user.name=root&op=MKDIRS"

HTTP/1.1 200 OK

Cache-Control: no-cache

Expires: Mon, 12 Nov 2018 08:44:02 GMT

Date: Mon, 12 Nov 2018 08:44:02 GMT

Pragma: no-cache

Expires: Mon, 12 Nov 2018 08:44:02 GMT

Date: Mon, 12 Nov 2018 08:44:02 GMT

Pragma: no-cache

Content-Type: application/json

Set-Cookie: hadoop.auth="u=root&p=root&t=simple&e=1542048242023&s=x2e43XjlMPY9E4qEtUOlGBNFtF8="; Path=/; Expires=???, 12-???-2018 18:44:02 GMT; HttpOnly

Transfer-Encoding: chunked

Server: Jetty(6.1.26)

{"boolean":true}[root@master ~]#

2.2.5.6  **文件重命名**

**步骤1.**  **将文件夹重新命名**

通过下面的命令可以实现将webhdfsDir文件夹重命名为webhdfsDir1：

curl -i -X PUT "http://master:50070/webhdfs/v1/webhdfsDir?user.name=root&op=RENAME&destination=/webhdfsDir1"

命令运行后的返回结果如下所示：

[root@master ~]# curl -i -X PUT "http://master:50070/webhdfs/v1/webhdfsDir?user.name=root&op=RENAME&destination=/webhdfsDir1"

HTTP/1.1 200 OK

Cache-Control: no-cache

Expires: Mon, 12 Nov 2018 08:48:43 GMT

Date: Mon, 12 Nov 2018 08:48:43 GMT

Pragma: no-cache

Expires: Mon, 12 Nov 2018 08:48:43 GMT

Date: Mon, 12 Nov 2018 08:48:43 GMT

Pragma: no-cache

Content-Type: application/json

Set-Cookie: hadoop.auth="u=root&p=root&t=simple&e=1542048523890&s=gn+jr35+oWW3PHWkGqXXnb8154A="; Path=/; Expires=???, 12-???-2018 18:48:43 GMT; HttpOnly

Transfer-Encoding: chunked

Server: Jetty(6.1.26)

{"boolean":true}[root@master ~]#

**步骤2.**  **验证重命名是否成功**

通过下面的命令可以进行HDFS文件查看，以确定文件夹是否重新命名：

hadoop fs -ls /

命令运行后的返回结果如下所示（由返回结果可以看出文件夹已经成功被重新命名）：

[root@master ~]# hadoop fs -ls /

18/11/12 16:49:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 8 items

drwxr-xr-x   - root supergroup          0 2018-11-09 10:12 /archive

drwxr-xr-x   - root supergroup          0 2018-11-09 10:15 /distcpdir

drwxr-xr-x   - root supergroup          0 2018-11-09 10:17 /distcpdir1

drwxr-xr-x   - root supergroup          0 2018-11-09 10:13 /files.har

drwxr-xr-x   - root supergroup          0 2018-11-08 16:05 /tmp

drwxr-xr-x   - root supergroup          0 2016-09-29 14:49 /user

drwxr-xr-x   - root supergroup          0 2018-11-12 16:44 /webhdfsDir1

-rwxr-xr-x   1 root supergroup         39 2018-11-12 16:26 /webhdfsFile

[root@master ~]#

2.2.5.7  **删除文件**

**步骤1.**  **文件删除**

通过下面的命令，可以删除文件/webhdfsFile：

curl -i -X DELETE "http://master:50070/webhdfs/v1/webhdfsFile?user.name=root&op=DELETE"

命令运行后的返回结果如下所示：

[root@master ~]# curl -i -X DELETE "http://master:50070/webhdfs/v1/webhdfsFile?user.name=root&op=DELETE"

HTTP/1.1 200 OK

Cache-Control: no-cache

Expires: Mon, 12 Nov 2018 09:00:51 GMT

Date: Mon, 12 Nov 2018 09:00:51 GMT

Pragma: no-cache

Expires: Mon, 12 Nov 2018 09:00:51 GMT

Date: Mon, 12 Nov 2018 09:00:51 GMT

Pragma: no-cache

Content-Type: application/json

Set-Cookie: hadoop.auth="u=root&p=root&t=simple&e=1542049251846&s=F0W/zi3f7eXmOSEp2dPQdJ5WdEE="; Path=/; Expires=???, 12-???-2018 19:00:51 GMT; HttpOnly

Transfer-Encoding: chunked

Server: Jetty(6.1.26)

{"boolean":true}[root@master ~]#

**步骤2.**  **验证删除命令是否执行成功**

通过下面的命令可以进行HDFS文件查看，以确定文件是否被删除：

hadoop fs -ls /

命令运行后的返回结果如下所示（由返回结果可以看出文件已经成功被删除）：

[root@master ~]# hadoop fs -ls /

18/11/12 17:01:10 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 7 items

drwxr-xr-x   - root supergroup          0 2018-11-09 10:12 /archive

drwxr-xr-x   - root supergroup          0 2018-11-09 10:15 /distcpdir

drwxr-xr-x   - root supergroup          0 2018-11-09 10:17 /distcpdir1

drwxr-xr-x   - root supergroup          0 2018-11-09 10:13 /files.har

drwxr-xr-x   - root supergroup          0 2018-11-08 16:05 /tmp

drwxr-xr-x   - root supergroup          0 2016-09-29 14:49 /user

drwxr-xr-x   - root supergroup          0 2018-11-12 17:00 /webhdfsDir1

[root@master ~]#

2.2.5.8  **查看文件属性**

通过下面的命令可以查看/webhdfsDir1的属性信息：

curl -i "http://master:50070/webhdfs/v1/webhdfsDir1?op=GETFILESTATUS"

命令运行后的返回结果如下所示：

[root@master ~]# curl -i "http://master:50070/webhdfs/v1/webhdfsDir1?op=GETFILESTATUS"

HTTP/1.1 200 OK

Cache-Control: no-cache

Expires: Mon, 12 Nov 2018 09:02:01 GMT

Date: Mon, 12 Nov 2018 09:02:01 GMT

Pragma: no-cache

Expires: Mon, 12 Nov 2018 09:02:01 GMT

Date: Mon, 12 Nov 2018 09:02:01 GMT

Pragma: no-cache

Content-Type: application/json

Transfer-Encoding: chunked

Server: Jetty(6.1.26)

（注：下面的内容为对应的属性信息）

{"FileStatus":{"accessTime":0,"blockSize":0,"childrenNum":0,"fileId":16826,"group":"supergroup","length":0,"modificationTime":1542013227957,"owner":"root","pathSuffix":"","permission":"755","replication":0,"storagePolicy":0,"type":"DIRECTORY"}}[root@master ~]#

2.2.5.9  **查看文件夹内容**

通过下面的命令可以列举文件夹的内容：

curl -i "http://master:50070/webhdfs/v1/webhdfsDir1?op=LISTSTATUS"

命令运行后的返回结果如下所示（由于当前文件夹下没有子文件，因此返回的结果是空的）：

[root@master ~]# curl -i "http://master:50070/webhdfs/v1/webhdfsDir1?op=LISTUS"T

HTTP/1.1 200 OK

Cache-Control: no-cache

Expires: Mon, 12 Nov 2018 09:02:36 GMT

Date: Mon, 12 Nov 2018 09:02:36 GMT

Pragma: no-cache

Expires: Mon, 12 Nov 2018 09:02:36 GMT

Date: Mon, 12 Nov 2018 09:02:36 GMT

Pragma: no-cache

Content-Type: application/json

Transfer-Encoding: chunked

Server: Jetty(6.1.26)

{"FileStatuses":{"FileStatus":[

]}}

[root@master ~]#

**2.2.6  常见问题**

2.2.6.1  **RPC通信错误**

当执行WebHDFS命令时候，出现类似Server IPC version 9 cannot communicate with client version 470错误信息，如下所示：

[root@master ~]# curl -i -X PUT "http://master:9000/webhdfs/v1/wenhdfsFile?op=CREATE"

org.apache.hadoop.ipc.RPC$VersionMismatch\*>Server IPC version 9 cannot communicate with client version 470:@[root@master ~]#

此时应该着重考虑是否是请求的协议写错了，以及通信端口是否填写错误。在上文的报错信息中，由于错误的将[http://master:50070](http://master:50070/)写成了[http://master:9000](http://master:9000/)，因此会出现错误信息。

2.2.6.2  **权限错误**

当使用WebHDFS命令向HDFS文件系统中写入数据的时候，首先需要确保所使用的用户是否拥有HDFS的写入权限。

如果在WebHDFS命令中没有显示的指定用户，则会使用默认的dr.who作为访问用户，而如果没有在配置文件中进行配置，此用户是没有权限对HDFS中的文件进行写操作的，如果用户权限错误，往往会出现类似于下列的输出信息：

[root@master ~]# curl -i -X PUT "http://master:50070/webhdfs/v1/webhdfsDir?op=MKDIRS"

HTTP/1.1 403 Forbidden

Cache-Control: no-cache

Expires: Mon, 12 Nov 2018 08:42:59 GMT

Date: Mon, 12 Nov 2018 08:42:59 GMT

Pragma: no-cache

Expires: Mon, 12 Nov 2018 08:42:59 GMT

Date: Mon, 12 Nov 2018 08:42:59 GMT

Pragma: no-cache

Content-Type: application/json

Transfer-Encoding: chunked

Server: Jetty(6.1.26)

{"RemoteException":{"exception":"AccessControlException","javaClassName":"org.apache.hadoop.security.AccessControlException","message":"Permission denied: user=dr.who, access=WRITE, inode=\"/webhdfsDir\":root:supergroup:drwxr-xr-x"}}[root@master ~]#

[root@master ~]# curl -i -X POST -T /home/webfile2.txt "http://master:50075/webhdfs/v1/webhdfsFile?op=APPEND&namenoderpcaddress=master:9000"

HTTP/1.1 100 Continue

HTTP/1.1 403 Forbidden

Content-Type: application/json; charset=utf-8

Content-Length: 2109

Connection: close

{"RemoteException":{"exception":"AccessControlException","javaClassName":"org.apache.hadoop.security.AccessControlException","message":"Permission denied: user=dr.who, access=WRITE, inode=\"/webhdfsFile\":root:supergroup:-rwxr-xr-x\n\tat org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.check(FSPermissionChecker.java:319)\n\tat org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkPermission(FSPermissionChecker.java:219)\n\tat org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkPermission(FSPermissionChecker.java:190)\n\tat org.apache.hadoop.hdfs.server.namenode.FSDirectory.checkPermission(FSDirectory.java:1698)\n\tat org.apache.hadoop.hdfs.server.namenode.FSDirectory.checkPermission(FSDirectory.java:1682)\n\tat org.apache.hadoop.hdfs.server.namenode.FSDirectory.checkPathAccess(FSDirectory.java:1656)\n\tat org.apache.hadoop.hdfs.server.namenode.FSNamesystem.appendFileInternal(FSNamesystem.java:2668)\n\tat org.apache.hadoop.hdfs.server.namenode.FSNamesystem.appendFileInt(FSNamesystem.java:2985)\n\tat org.apache.hadoop.hdfs.server.namenode.FSNamesystem.appendFile(FSNamesystem.java:2952)\n\tat org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.append(NameNodeRpcServer.java:653)\n\tat org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.append(ClientNamenodeProtocolServerSideTranslatorPB.java:421)\n\tat org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.java)\n\tat org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:616)\n\tat org.apache.hadoop.ipc.RPC$Server.call(RPC.java:969)\n\tat org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2049)\n\tat org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2045)\n\tat java.security.AccessController.doPrivileged(Native Method)\n\tat javax.security.auth.Subject.doAs(Subject.java:422)\n\tat org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1657)\n\tat org.apache.hadoop.ipc.Server$Handler.run(Server.java:2043)\n"}}[root@master ~]# curl -i -X POST "http://master:500r.name=root&op=APPEND"ile?use

HTTP/1.1 307 TEMPORARY\_REDIRECT

Cache-Control: no-cache

Expires: Mon, 12 Nov 2018 08:25:37 GMT

Date: Mon, 12 Nov 2018 08:25:37 GMT

Pragma: no-cache

Expires: Mon, 12 Nov 2018 08:25:37 GMT

Date: Mon, 12 Nov 2018 08:25:37 GMT

Pragma: no-cache

Content-Type: application/octet-stream

Set-Cookie: hadoop.auth="u=root&p=root&t=simple&e=1542047137389&s=emkW9j8r6EANJW60pc4ShuFy/VQ="; Path=/; Expires=???, 12-???-2018 18:25:37 GMT; HttpOnly

Location: http://master:50075/webhdfs/v1/webhdfsFile?op=APPEND&user.name=root&namenoderpcaddress=master:9000

Content-Length: 0

Server: Jetty(6.1.26)

解决权限错误的问题往往是比较简单的，可以在命令中通过显示的指定user.name=root，来设定以root用户的权限进行WebHDFS命令操作。

2.2.6.3  **HTTP/1.1 400 Bad Request**

出现HTTP/1.1 400 Bad Request错误，往往是由于命令行拼写错误所造成的，如下所示：

[root@master ~]# curl -i -X PUT "http://master:50070/webhdfs/v1/webhdfsDir?user.name=root&op= RENAME&destination=/webhdfsDir1"

HTTP/1.1 400 Bad Request

Connection: close

Server: Jetty(6.1.26)

在上述错误中，由于在op= RENAME拼写中多写了一个空格，因此导出命令行识别失败，从而出现Bad Request错误。

2.2.6.4  **其他错误**

在执行写入或者追加等操作时，需要结合命令运行所返回的Location地址来进行内容写入，同时需要将返回的Location地址用双引号（""）括起来，否则可能会出现类似下面的错误：

[root@master /]# curl -i -X POST -T /home/webfile2.txt http://master:50075/webhdfs/v1/webhdfsFile?op=APPEND&user.name=root&namenoderpcaddress=master:9000

[1] 3096

[2] 3097

[root@master /]# bash: user.name=root: command not found

HTTP/1.1 100 Continue

HTTP/1.1 400 Bad Request

Content-Type: application/json; charset=utf-8

Content-Length: 161

Connection: close

{"RemoteException":{"exception":"IllegalArgumentException","javaClassName":"java.lang.IllegalArgumentException","message":"java.net.UnknownHostException: null"}}

在上述错误中，由于在没有将Location地址/home/webfile2.txt <http://master:50075/webhdfs/v1/webhdfsFile?op=APPEND&user.name=root&namenoderpcaddress=master:9000>用双引号括起来，所以会出现命令解析错误。

**2.2.7  课后思考**

1.   请思考WebHDFS的作用及应用场景。 提供集群外部访问的接口，连接不同的hadoop版本集群。

2.   在上文所述内容中，由于权限管理限制，需要手动指定user.name才能进行HDFS文件写操作，请查阅相关资料实现以默认的用户进行HDFS文件写操作。