

Bug报告

小组成员：孔德桐、雷玺霖、陈朴炎

报告人：陈朴炎

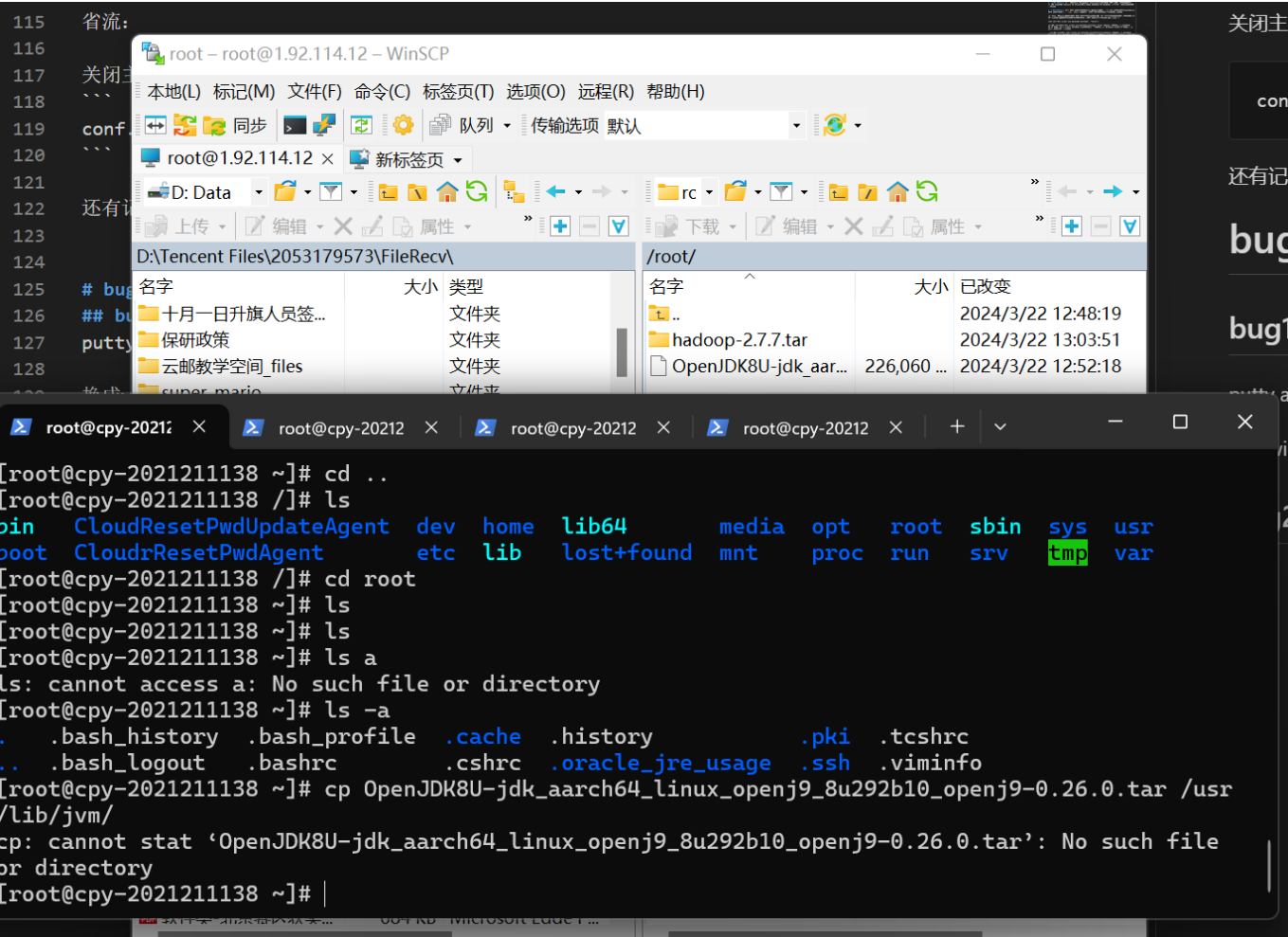
bug1

bug:putty access denied

解决方法：换成windows的powershell，输入ssh root@"ip"，"ip"为服务器外网ip

bug2

用WinSCP传入的两个文件夹都在node1里找不到



关机重启当前服务器，才能看到

bug3

主节点运行 start-all.sh 之后，显示namenode写入log，但是输入jps命令之后却没有显示出namenode

```
[root@cpy-2021211138 ~]# cd /home/modules/hadoop-2.7.7
[root@cpy-2021211138 ~]# start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
24/03/22 17:28:30 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [node1]
node1: starting namenode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-namenode-cpy-2021211138.out
node2: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
node3: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
node4: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
Starting secondary namenodes [node1]
node1: starting secondarynamenode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-secondarynamenode-cpy-2021211138.out
24/03/22 17:28:46 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 /home/modules/hadoop-2.7.7/logs/yarn-root-resourcemanager-cpy-2021211138.out
node2: starting nodemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-nodemanager-cpy-2021211138.out
node4: starting nodemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-nodemanager-cpy-2021211138.out
node3: starting nodemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-nodemanager-cpy-2021211138.out
[root@cpy-2021211138 ~]# jps
2050 SecondaryNameNode
2373 ResourceManager
2682 Jps
```

原来是我忘记执行上一条命令 `hadoop namenode -format`，重新执行后，再start-all.sh，jps输出结果就符合预期

bug4

```
name: hdfs://1.92.114.12/upload_2021211138.txt, folder: false, size: 0
Upload file:
org.apache.hadoop.ipc.RemoteException(java.io.IOException): File /upload_2021211138.txt could only be replicated to 0 nodes instead of minReplicationFactor=1
    at org.apache.hadoop.hdfs.server.blockmanagement.BlockManager.chooseTarget4NewBlock(BlockManager.java:1620)
    at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.getNewBlockTargets(FSNamesystem.java:3135)
    at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.getAdditionalBlock(FSNamesystem.java:3059)
    at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.addBlock(NameNodeRpcServer.java:725)
    at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.addBlock(ClientNamenodeProtocolServerSideTranslatorPB.java:49)
```

上传文件的时候告诉我可用结点个数为0

出错原因分析：

将 127.0.0.1 的部分注释掉，然后将四台服务器 ip 修改为局域网 ip（可在华为云上查看）
然后设置电脑与服务器的 ssh 免密登陆：↵
打开终端，输入下面命令↵

```
ls ~/.ssh↵
```

如果存在 id_rsa 和 id_rsa.pub 文件，说明之前生成过密钥，无需操作；↵
如果不存在上述两个文件，则命令行输入↵

```
ssh-keygen -t rsa↵
```

即可生成上述两个文件。↵

将公钥文件 id_rsa.pub 传送到服务器到 ~/.ssh 目录下↵

```
scp ~/.ssh/id_rsa.pub user-name@10.10.10.6:~/id_rsa.pub↵
```

服务器 ~/.ssh 目录已存在 authorized_keys，则将上传的 id_rsa.pub 添加到文件内容的后面↵
确保本地电脑可以直接 ssh 连通服务器↵

修改本地 hosts 文件，在本地终端输入：↵

```
vim /etc/hosts↵
```

添加四台服务器局域网 ip 以及服务器名称↵

这部分没有做好，重新做了一遍

是因为我没有在 windows 上安装 vim，并且当时 ls 指令它没有匹配到报错了

而且找不到 /etc/hosts 就跳过了

后来发现它在 C:/Windows/System32/drivers/etc/hosts 下面

找到了但是改不了

用 VSCode 打开 -> 用管理员权限更改就行了

最后成功了

bug5

```

[[root@ecs-62b7-0001 ~]# netstat -lptn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address   Foreign Address State      PID/Program name
tcp        0      0 0.0.0.0:22      0.0.0.0:*      LISTEN    3907/sshd
tcp        0      0 0.0.0.0:50070   0.0.0.0:*      LISTEN    1739/java
tcp        0      0 192.168.0.132:50090 0.0.0.0:*      LISTEN    1956/java
tcp6       0      0 :::22          :::*           LISTEN    3907/sshd
tcp6       0      0 192.168.0.132:8088 :::*           LISTEN    2141/java
tcp6       0      0 :::1:25        :::*           LISTEN    1039/master
tcp6       0      0 192.168.0.132:8030 :::*           LISTEN    2141/java
tcp6       0      0 192.168.0.132:8031 :::*           LISTEN    2141/java
tcp6       0      0 192.168.0.132:8032 :::*           LISTEN    2141/java
tcp6       0      0 192.168.0.132:8033 :::*           LISTEN    2141/java

```

确保 8020 端口监听的不是本地 IP（上图为正确情况，可跳过 hosts 文件修改步骤；若 127.0.0.1:8020 则需要修改 hosts 文件）

修改 hosts 文件（四台服务器都需要操作），输入：

`vim /etc/hosts`

```

::1      localhost        localhost.localdomain  localhost6        localhost6.localdomain6
#127.0.0.1      localhost        localhost.localdomain  localhost4        localhost4.localdomain4
#127.0.0.1      localhost        localhost
#121.36.99.86   ecs-62b7-0001
#119.3.190.191 ecs-62b7-0002
#121.36.80.207  ecs-62b7-0003
#121.36.39.159  ecs-62b7-0004
#127.0.0.1      ecs-62b7-0001   ecs-62b7-0001
192.168.0.132   ecs-62b7-0001
192.168.0.83    ecs-62b7-0002
192.168.0.62    ecs-62b7-0003
192.168.0.104   ecs-62b7-0004
192.168.0.132   ecs-62b7-0001   ecs-62b7-0001
127.0.0.1       ecs-62b7-0001   ecs-62b7-0001

```

不理解host文件中的配置信息以及作用

解决：host这个东西你可以理解为本地的DNS解析，就是最终还是会解析成ip地址的，这个一个ip对应两个名字，就是这两个名字都能解析成前面这个ip

bug6

配置免密登录的时候，总是失败。可以添加-vvv来打印ssh调试信息。最后发现一切正常，错误是因为复制的时候，vim会吞掉第一个字符，不知道为什么。

bug7

```

[root@lxl-2021211146-0002 ~]# hadoop namenode -format
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

/home/modules/hadoop-2.7.7/bin/hdfs: line 304: /usr/lib/jvm/jdk8u292-b10/bin/java:
No such file or directory

```

```
cd /usr/lib/ rm -rf jvm mkdir jvm
```

```
scp OpenJDK8U-jdk_aarch64_linux_openj9_8u292b10_openj9-0.26.0.tar root@1.92.137.22:/root/download scp  
OpenJDK8U-jdk_aarch64_linux_openj9_8u292b10_openj9-0.26.0.tar root@124.70.57.108:/root/download scp  
OpenJDK8U-jdk_aarch64_linux_openj9_8u292b10_openj9-0.26.0.tar root@1.92.139.33:/root/download scp  
OpenJDK8U-jdk_aarch64_linux_openj9_8u292b10_openj9-0.26.0.tar root@1.92.124.93:/root/download
```

```
mv /root/download/OpenJDK8U-jdk_aarch64_linux_openj9_8u292b10_openj9-0.26.0.tar /usr/lib/jvm/
```

```
cd /usr/lib/jvm/ tar -vxf OpenJDK8U-jdk_aarch64_linux_openj9_8u292b10_openj9-0.26.0.tar
```

bug8

```
start-all.sh  
starting resourcemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-  
resourcemanager-lxl-2021211146-0002.out  
node3: Error: JAVA_HOME is not set and could not be found.  
node1: Error: JAVA_HOME is not set and could not be found.  
node4: Error: JAVA_HOME is not set and could not be found.
```

JAVA_HOME is not set and could not be found

在JDK配置无错的情况下，可能是没有配置hadoop-env.sh文件。这个文件里写的是hadoop的环境变量,主要修改hadoop的JAVA_HOME路径。

切到 [hadoop]/etc/hadoop目录 执行：vim hadoop-env.sh 修改java_home路径和hadoop_conf_dir路径为具体的安装路径 例如： export JAVA_HOME=/usr/jdk1.8.0_65 export HADOOP_CONF_DIR=/usr/hadoop-3.1.3/etc/hadoop 重新加载使修改生效：source hadoop-env.sh

```
vim /home/modules/hadoop-2.7.7/etc/hadoop/hadoop-env.sh
```

bug9

```
can't find module  
Could not find artifact arg.apache.hadoop:hadoop-common:pom:2.7.7 in central  
(https://repo.maven.apache.org/maven2)
```

mave->reload project

maven下载dependency失败

发现应该是打字打错了

正确配置如下

```
<dependencies>  
  <dependency>  
    <groupId>log4j</groupId>
```

```

        <artifactId>log4j</artifactId>
        <version>1.2.17</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-client</artifactId>
        <version>${hadoop.version}</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-common</artifactId>
        <version>${hadoop.version}</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-hdfs</artifactId>
        <version>${hadoop.version}</version>
    </dependency>

```

bug10

```
org.apache.hadoop.ipc.RemoteException(java.io.IOException): File /2021211146.txt
could only be replicated to 0 nodes instead of minReplication (=1). There are 3
datanode(s) running and 3 node(s) are excluded in this operation.
```

要查看 HDFS 是否正常工作，你可以采取以下几种方法来检查：

1. **HDFS Web UI**：通过浏览器访问 HDFS 的 Web 用户界面可以获取有关 HDFS 集群健康状态的信息。通常，你可以通过以下地址访问 HDFS Web UI：
 - NameNode Web UI: http://<namenode_hostname>:50070 或 http://<namenode_hostname>:9870
 - DataNode Web UI: http://<datanode_hostname>:50075
2. **HDFS fsck 命令**：在命令行中运行 `hdfs fsck /` 命令，可以对整个 HDFS 文件系统进行一致性检查，并获取有关块位置、副本数量等信息。
3. **JMX 监控**：通过 JMX (Java Management Extensions) 可以监控 HDFS 的运行状况。你可以使用 JMX 工具（如 JConsole 或 JVisualVM）连接到 NameNode 或 DataNode 进程，以检查各个组件的运行状态。
4. **日志文件**：查看 HDFS 的日志文件可以帮助你了解系统运行过程中的任何错误或警告信息。通常，HDFS 的日志文件位于 `logs` 目录中，你可以检查这些日志文件来查看集群的运行情况。

通过以上几种方法，你可以监控 HDFS 的健康状态、文件系统一致性以及集群各组件的运行状况。如果发现任何异常情况，你就可以更快地解决问题并恢复 HDFS 的正常工作状态。祝你顺利！🔧🔍

```
conf.set("dfs.client.use.datanode.hostname", "false");
```

当设置为false时: "dfs.client.use.datanode.hostname= false" Hadoop 客户端会使用 DataNode 的 IP 地址而不是主机名与 DataNode 通信。这意味着客户端会使用 DataNode 上报的 IP 地址进行通信, 而不会依赖于主机名。

而当设置为true时: "dfs.client.use.datanode.hostname=true", Hadoop 客户端将使用 DataNode 的主机名而不是 IP 地址与 DataNode 进行通信。这意味着客户端会使用 DataNode 主机名进行通信, 这有助于在集群中更好地使用主机名进行通信, 而不仅仅依赖于 IP 地址。这种设置更有利于集群环境的稳定性和可靠性, 因为主机名更不容易发生变化。

因此, "dfs.client.use.datanode.hostname=false" 和 "dfs.client.use.datanode.hostname=true" 的区别主要在于客户端与 DataNode 进行通信时使用的是 IP 地址还是主机名。选择哪种设置取决于你的环境和需求, 但通常建议使用主机名来提高稳定性和可维护性。希望这能帮助理解两者之间的区别! 🐞 😊

notepad C:\Windows\System32\drivers\etc\hosts

ipconfig /flushdns

省流:

关闭主机名解析模式, 使用IP解析

```
conf.set("dfs.client.use.datanode.hostname", "false");
```

还有记得安全组中添加50010端口开放, 否则无法解析datanode

bug11

Bug: 连续format两次之后, slave和master不能通信, slave没有datanode进程;

解决方法: 删除所有节点的hadoop.tmp.dir配置中的目录中的文件 重新format