**集群IP**

172.19.241.214 fzj

172.19.241.176 gxq

172.19.241.4 lcz

172.19.241.157 lh

**Web URI**

HDFS =====> http://fzj:50070/、http://lh:50070/

jobHistory =====> http://lcz:19888/

historyserver =====> http://gxq:8088/、http://lcz:8088/

SparkMaster =====> http://lcz:9999/

Spark History Server =====> http://lcz:18080/

**HA高可用集群架构**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | fzj | gxq | lcz | lh |
| NameNode | NameNode |  |  | NameNode |
| JournalNode | JournalNode | JournalNode |  | JournalNode |
| DataNode | DataNode | DataNode | DataNode | DataNode |
| ResourceManager |  | ResourceManager | ResourceManager |  |
| NodeManager | NodeManager | NodeManager | NodeManager | NodeManager |
| Zookeeper | QuorumPeerMain | QuorumPeerMain |  | QuorumPeerMain |
| DFSZKFailoverController | DFSZKFailoverController |  |  | DFSZKFailoverController |
| JobHistory |  |  | JobHistoryServer |  |
| Master |  |  | Master |  |
| Worker | Worker | Worker |  | Worker |
| HistoryServer |  |  | HistoryServer |  |

文本

描述已自动生成

**脚本**

所有脚本都在/usr/local/bin目录下，可以自己去查看

jpsall 查看集群java线程

xSource 集群集体source /etc/profile

xsync 集群分发

xHadoopStart 集群启动Hadoop（包含了Zookeeper）

xHadoopStop 集群关闭Hadoop（包含了Zookeeper）

xCloseFireWalld 集群群关防火墙

xZkStart 集群启动Zookeeper

xZkStop 集群关闭Zookeeper

xZkStatus 集群查看Zookeeper状态

xReformat 格式化所有的nn、sn、dn文件

**免密登陆**

所有人都操作

ssh-keygen -t rsa

ssh-copy-id fzj

fzj操作

scp /root/.ssh/authorized\_keys fzj:/root/.ssh

scp /root/.ssh/authorized\_keys gxq:/root/.ssh

scp /root/.ssh/authorized\_keys lcz:/root/.ssh

scp /root/.ssh/authorized\_keys lh:/root/.ssh

**系统配置文件**

/etc/profile

|  |
| --- |
| # set java environment  export JAVA\_HOME=/home/java/jdk1.8.0\_261  export PATH="$JAVA\_HOME/bin:$PATH"  # set zookeeper enviroment  export ZOOKEEPER\_HOME=/home/zookeeper/zookeeper-3.4.9  export PATH="$ZOOKEEPER\_HOME/bin:$PATH"  # set hadoop environment  export HADOOP\_HOME=/home/hadoop/hadoop-2.7.7  export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin  # set scala environment  export SCALA\_HOME=/home/scala/scala-2.13.3  export PATH="$SCALA\_HOME/bin:$PATH"  # set spark environment  export SPARK\_HOME=/home/spark/spark-3.0.1-bin-hadoop2.7  export PATH="$SPARK\_HOME/bin:$PATH"  # set spark lib  export JAVA\_LIBRARY\_PATH=$HADOOP\_HOME/lib/native/  unset i  unset -f pathmunge  unset MAILCHECK |

**Hadoop配置文件**

**1、core-site.xml**

|  |
| --- |
| <configuration>  <!--指定hadoop运行时产生文件的存储目录-->  <property>  <name>hadoop.tmp.dir</name>  <value>/home/hadoop/tmp</value>  <description>Abase for other temporary directories.</description>  </property>  <!--把两个NameNode的地址组装成一个集群mycluster-->  <property>  <name>fs.defaultFS</name>  <value>hdfs://mycluster</value>  </property>  <!-- 缓冲区大小, 实际工作中根据服务器性能动态调整 -->  <property>  <name>io.file.buffer.size</name>  <value>4096</value>  </property>  <!-- 配置HDFS-HA自动故障转移 QJM-->  <property>  <name>ha.zookeeper.quorum</name>  <value>fzj:2181,gxq:2181,lh:2181</value>  </property>  <!-- 开启hdfs的垃圾桶机制, 删除掉的数据可以从垃圾桶中回收, 单位分钟 -->  <property>  <name>fs.trash.interval</name>  <!-- 7天 -->  <value>10080</value>  </property>  </configuration> |

**2、hdfs-site.xml**

|  |
| --- |
| <configuration>  <!--完全分布式集群名称-->  <property>  <name>dfs.nameservices</name>  <value>mycluster</value>  </property>  <!--集群中NameNode节点都有哪些-->  <property>  <name>dfs.ha.namenodes.mycluster</name>  <value>nn1,nn2</value>  </property>  <!--nn1的RPC通讯地址-->  <property>  <name>dfs.namenode.rpc-address.mycluster.nn1</name>  <value>fzj:9000</value>  </property>  <!--nn2的RPC通讯地址-->  <property>  <name>dfs.namenode.rpc-address.mycluster.nn2</name>  <value>lh:9000</value>  </property>  <!-- 指定nn1的访问地址和端口 -->  <property>  <name>dfs.namenode.http-address.mycluster.nn1</name>  <value>fzj:50070</value>  </property>  <!-- 指定nn2的访问地址和端口 -->  <property>  <name>dfs.namenode.http-address.mycluster.nn2</name>  <value>lh:50070</value>  </property>  <!-- 指定NameNode元数据在JournalNode上的存放位置 -->  <property>  <name>dfs.namenode.shared.edits.dir</name>  <value>qjournal://fzj:8485;gxq:8485;lh:8485/mycluster</value>  </property>  <!--配置隔离机制, 即同一时刻只能有一台服务器对外相应-->  <property>  <name>dfs.ha.fencing.methods</name>  <value>sshfence</value>  </property>  <!--使用隔离机制时需要ssh无密钥登陆-->  <property>  <name>dfs.ha.fencing.ssh.private-key-files</name>  <value>/root/.ssh/id\_rsa</value>  </property>  <property>  <name>dfs.replication</name>  <value>3</value>  </property>  <!-- 指定namenode的访问地址和端口  <property>  <name>dfs.namenode.http-address</name>  <value>fzj:50070</value>  </property>  -->  <!--secondarynamenode的ip路径-->  <property>  <name>dfs.namenode.secondary.http-address</name>  <value>lcz:50090</value>  </property>  <!--namenode存放路径-->  <property>  <name>dfs.namenode.name.dir</name>  <value>file:///home/hadoop/tmp/dfs/name</value>  </property>  <!--datanode存放路径-->  <property>  <name>dfs.datanode.data.dir</name>  <value>file:///home/hadoop/tmp/dfs/data</value>  </property>  <!--journalnode服务器存储目录-->  <property>  <name>dfs.journalnode.edits.dir</name>  <value>/home/hadoop/tmp/dfs/jn</value>  </property>  <!-- 指定namenode日志文件的存放目录 -->  <property>  <name>dfs.namenode.edits.dir</name>  <value>file:///home/hadoop/tmp/dfs/nn/edits</value>  </property>  <property>  <name>dfs.namenode.checkpoint.dir</name>  <value>file:///home/hadoop/tmp/dfs/snn/name</value>  </property>  <property>  <name>dfs.namenode.checkpoint.edits.dir</name>  <value>file:///home/hadoop/tmp/dfs/snn/edits</value>  </property>  <!-- 关闭权限检查-->  <property>  <name>dfs.permissions.enable</name>  <value>false</value>  </property>  <!--访问代理类:client、mycluster、active配置失败自动切换实现方式-->  <property>  <name>dfs.client.failover.proxy.provider.mycluster</name>  <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>  </property>  <!-- 配置HDFS-HA自动故障转移 -->  <property>  <name>dfs.ha.automatic-failover.enabled</name>  <value>true</value>  </property>  </configuration> |

**3、mapred-site.xml**

|  |
| --- |
| <configuration>  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  <!-- 开启MapReduce小任务模式 -->  <property>  <name>mapreduce.job.ubertask.enable</name>  <value>true</value>  </property>  <!-- 设置历史任务的主机和端口 -->  <property>  <name>mapreduce.jobhistory.address</name>  <value>lcz:10020</value>  </property>  <!-- 设置网页访问历史任务的主机和端口 -->  <property>  <name>mapreduce.jobhistory.webapp.address</name>  <value>lcz:19888</value>  </property>  </configuration> |

**4、yarn-site.xml**

|  |
| --- |
| <configuration>  <!-- Site specific YARN configuration properties -->  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  <!--启动resourcemanager ha-->  <property>  <name>yarn.resourcemanager.ha.enabled</name>  <value>true</value>  </property>  <!--声明两台resourcemanager的地址-->  <property>  <name>yarn.resourcemanager.cluster-id</name>  <value>cluster-yarn1</value>  </property>  <property>  <name>yarn.resourcemanager.ha.rm-ids</name>  <value>rm1,rm2</value>  </property>  <property>  <name>yarn.resourcemanager.hostname.rm1</name>  <value>gxq</value>  </property>  <property>  <name>yarn.resourcemanager.hostname.rm2</name>  <value>lcz</value>  </property>  <!--指定zookeeper集群的地址-->  <property>  <name>yarn.resourcemanager.zk-address</name>  <value>fzj:2181,gxq:2181,lh:2181</value>  </property>  <!--启动自动恢复-->  <property>  <name>yarn.resourcemanager.recovery.enabled</name>  <value>true</value>  </property>  <!--指定resourcemanager的状态信息存储在zookeeper集群-->  <property>  <name>yarn.resourcemanager.store.class</name>  <value>org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateStore</value>  </property>  <!--  <property>  <name>yarn.resourcemanager.hostname</name>  <value>gxq</value>  </property>  -->  <!-- 开启日志聚合功能 -->  <property>  <name>yarn.log-aggregation-enable</name>  <value>true</value>  </property>  <!-- 设置聚合日志在hdfs上的保存时间, 单位为秒 -->  <property>  <name>yarn.log-aggregation.retain-seconds</name>  <value>604800</value>  </property>  <!-- 设置yarn集群的内存分配方案 -->  <property>  <name>yarn.nodemanager.resource.memory-mb</name>  <value>20480</value>  </property>  <property>  <name>yarn.scheduler.minimum-allocation-mb</name>  <value>2048</value>  </property>  <property>  <name>yarn.nodemanager.vmem-pmem-ratio</name>  <value>2.1</value>  </property>  </configuration> |

**ZK配置文件**

zoo.cfg

|  |
| --- |
| dataDir=/home/zookeeper/zkdatas  # 保留多少个快照  autopurge.snapRetainCount=3  # 日志多少小时清理一次  autopurge.purgeInterval=1  # 集群中服务器地址  server.1=fzj:2888:3888  server.2=gxq:2888:3888  server.3=lh:2888:3888 |

**Spark配置文件**

主机spark-env.sh的配置

export SPARK\_LOCAL\_IP=172.19.241.4 # 主机ip, 此处不能用127.0.0.1，要不然也访问不了，亲测

export HADOOP\_CONF\_DIR=/home/hadoop/hadoop-2.7.7/etc/hadoop

#缺少该配置,会找不到worker节点

export SPARK\_MASTER\_HOST=172.19.241.4 # 主机ip

export SPARK\_WORKER\_MEMORY=1g

export MASTER=spark://172.19.241.4:7077 # 主机master访问端口，web默认访问端口为8080

export SPARK\_MASTER\_WEBUI\_PORT=9999 # web ui访问端口9999

export LD\_LIBRARY\_PATH=$JAVA\_LIBRARY\_PATH

从机配置，不能跟主机配一致

export SPARK\_LOCAL\_IP=127.0.0.1

export HADOOP\_CONF\_DIR=/home/hadoop/hadoop-2.7.7/etc/hadoop

#缺少该配置,会找不到worker节点

export SPARK\_MASTER\_HOST=172.19.241.4

export SPARK\_WORKER\_MEMORY=1g

export MASTER=spark://172.19.241.4:7077

export SPARK\_MASTER\_WEBUI\_PORT=9999

export LD\_LIBRARY\_PATH=$JAVA\_LIBRARY\_PATH