# 一、容量调度器基于标签的设置

利用Yarn的标签调度机制，给其中某台机器打上Coordinator标签，剩下的打上Worker标签以保证每次Presto提交到Yarn上时，Coordinator这个组件都能运行在固定的机器上。

## （一）配置Yarn队列

修改yarn-site.xml文件：

|  |
| --- |
| <configuration>  <!-- 定义了root下有三个队列root.pro root.dev root.def -->  <property>  <name>yarn.scheduler.capacity.root.queues</name>  <value>pro,dev,def</value>  </property>  <!-- 定义了root队列默认分区的资源容量为100% -->  <property>  <name>yarn.scheduler.capacity.root.capacity</name>  <value>100</value>  </property>  <!-- 定义root队列的默认分区的资源最大容量为100% -->  <property>  <name>yarn.scheduler.capacity.root.maximum-capacity</name>  <value>100</value>  </property>  <!-- 定义了root队列可以访问的标签为所有分区，如果为空则表示只能访问默认分区 -->  <property>  <name>yarn.scheduler.capacity.root.accessible-node-labels</name>  <value>\*</value>  </property>  <!-- 定义了root队列可以访问的cpu分区的容量 -->  <property>  <name>yarn.scheduler.capacity.root.accessible-node-labels.cpu.capacity</name>  <value>100</value>  </property>  <!-- 定义了root队列可以访问的cpu分区的最大容量 -->  <property>  <name>yarn.scheduler.capacity.root.accessible-node-labels.cpu.maximum-capacity</name>  <value>100</value>  </property>  <!-- 定义了root队列可以访问的mem分区的容量 -->  <property>  <name>yarn.scheduler.capacity.root.accessible-node-labels.mem.capacity</name>  <value>100</value>  </property>  <!-- 定义了root队列可以访问的mem分区的最大容量 -->  <property>  <name>yarn.scheduler.capacity.root.accessible-node-labels.mem.maximum-capacity</name>  <value>100</value>  </property>  <!--队列root.pro使用默认分区的容量为40%-->  <property>  <name>yarn.scheduler.capacity.root.pro.capacity</name>  <value>40</value>  </property>  <!-- 队列root.pro使用默认分区的最大容量为100% -->  <property>  <name>yarn.scheduler.capacity.root.pro.maximum-capacity</name>  <value>100</value>  </property>  <!-- 队列root.pro可以访问的分区cpu和mem -->  <property>  <name>yarn.scheduler.capacity.root.pro.accessible-node-labels</name>  <value>cpu,mem</value>  </property>  <!-- 队列root.pro默认访问cpu分区 -->  <property>  <name>yarn.scheduler.capacity.root.pro.default-node-label-expression</name>  <value>cpu</value>  </property>  <!-- 队列root.pro访问cpu分区的容量为100% -->  <property>  <name>yarn.scheduler.capacity.root.pro.accessible-node-labels.cpu.capacity</name>  <value>100</value>  </property>  <!-- 队列root.pro访问cpu分区的最大容量为100% -->  <property>  <name>yarn.scheduler.capacity.root.pro.accessible-node-labels.cpu.maximum-capacity</name>  <value>100</value>  </property>  <!-- 队列root.pro访问50%的mem分区容量 -->  <property>  <name>yarn.scheduler.capacity.root.pro.accessible-node-labels.mem.capacity</name>  <value>50</value>  </property>  <!-- 队列root.pro可以访问100%的mem分区的容量 -->  <property>  <name>yarn.scheduler.capacity.root.pro.accessible-node-labels.mem.maximum-capacity</name>  <value>100</value>  </property>  <!-- 定义root.dev队列访问默认分区的比例为30% -->  <property>  <name>yarn.scheduler.capacity.root.dev.capacity</name>  <value>30</value>  </property>  <!-- 定义root.dev队列访问默认分区的最大比例为100% -->  <property>  <name>yarn.scheduler.capacity.root.dev.maximum-capacity</name>  <value>100</value>  </property>  <!-- 定义root.dev队列访问的分区为mem -->  <property>  <name>yarn.scheduler.capacity.root.dev.accessible-node-labels</name>  <value>mem</value>  </property>  <!-- 定义root.dev队列访问mem分区的容量为50% -->  <property>  <name>yarn.scheduler.capacity.root.dev.accessible-node-labels.mem.capacity</name>  <value>50</value>  </property>  <!-- 定义root.dev队列访问mem分区的最大容量为100%-->  <property>  <name>yarn.scheduler.capacity.root.dev.accessible-node-labels.mem.maximum-capacity</name>  <value>100</value>  </property>  <!-- 定义root.def队列访问默认分区30%的容量 -->  <property>  <name>yarn.scheduler.capacity.root.def.capacity</name>  <value>30</value>  </property>  <!-- 定义root.def队列访问默认分区最大容量为100%-->  <property>  <name>yarn.scheduler.capacity.root.def.maximum-capacity</name>  <value>100</value>  </property>  </configuration> |

## （二）配置Yarn标签

### 1、添加存放标签的hdfs目录

|  |
| --- |
| hadoop fs -mkdir -p /yarn/node-labels  hadoop fs -chown -R cloudera-scm:cloudera-scm /yarn  hadoop fs -chmod -R 700 /yarn |

### 2、设置 yarn-site.xml

|  |
| --- |
| <property>  <name>yarn.node-labels.enabled</name>  <value>true</value>  </property>  <property>  <name>yarn.node-labels.fs-store.root-dir</name>  <value>hdfs://30.23.76.191:8020/yarn/node-labels</value>  </property> |

### 3、添加标签

1）在集群中注册标签

|  |
| --- |
| yarn rmadmin -addToClusterNodeLabels "presto\_coordinator"  yarn rmadmin -addToClusterNodeLabels "presto\_worker" |

2）给节点添加标签

|  |
| --- |
| yarn rmadmin -replaceLabelsOnNode \  "30.23.76.191:8041,presto\_coordinator,presto\_worker" |

3）删除节点上的标签

|  |
| --- |
| yarn rmadmin -replaceLabelsOnNode "30.23.76.191:8041" |

4）查看节点的标签状态

|  |
| --- |
| yarn node -status 30.23.76.191:8041 |

5）刷新队列配置

|  |
| --- |
| yarn rmadmin -refreshQueues |

6）删除集群标签

|  |
| --- |
| yarn rmadmin -removeFromClusterNodeLabels \  "presto\_coordinator, presto\_worker " |

# 二、利用Slider部署Presto on Yarn

## （一）编译Presto-yarn-1.5

1、下载地址：<https://github.com/prestodb/presto-yarn/>

2、解压，使用maven进行编译

|  |
| --- |
| # 自己指定Presto的版本  mvn clean package -Dpresto.version=0.220 |

3、编译完成后在/presto-yarn-package/target可找到编译好的zip包

## （二）编译Slider-0.90.2

1、下载地址：<https://archive.apache.org/dist/incubator/slider/>

2、解压，修改pom.xml中的java和hadoop版本，并添加cdh库

|  |
| --- |
| <project.java.src.version>1.8</project.java.src.version>  <enforced.java.version>${project.java.src.version}</enforced.java.version>  <groovy.version>2.4.5</groovy.version>  <hadoop.version>2.6.0-cdh5.15.2</hadoop.version>  <hbase.version>1.2.0-cdh5.15.2</hbase.version>  <accumulo.version>1.7.0</accumulo.version>  <repositories>  <repository>  <id>cloudera</id>  <url>https://repository.cloudera.com/artifactory/cloudera-repos/</url>  </repository>  </repositories> |

3、注释掉slider-core和slider-funtest下pom对hadoop-minicluster包的依赖

|  |
| --- |
| <!--<dependency>  <groupId>org.apache.hadoop</groupId>  <artifactId>hadoop-minicluster</artifactId>  <scope>test</scope>  </dependency>--> |

4、使用maven编译

|  |
| --- |
| mvn clean package -Dmaven.test.skip=true -DskipTests |

5、编译完成在/slider-assembly/target可找到编译好的tar包

6、Slider最新版本为0.92，此版本代码匹配Apache Hadoop2.7版本，与cdh5.15的Hadoop2.6版本不兼容

## （三）利用Slider搭建Presto-yarn

### 1、配置Slider

1）解压编译得到的tar包

2）修改conf下slider-env.sh中的环境变量

|  |
| --- |
| export JAVA\_HOME=${JAVA\_HOME}  export HADOOP\_CONF\_DIR=${HADOOP\_CONF\_DIR} |

3）修改conf下slider-client.xml，添加zookeeper信息

|  |
| --- |
| <property>  <name>slider.zookeeper.quorum</name>  <value>30.23.76.191:2181</value>  </property>  <property>  <name>fs.defaultFS</name>  <value>hdfs://30.23.76.191:8020</value>  </property> |

### 2、配置Presto

1）解压编译得到的zip包

2）修改appConfig-default

|  |
| --- |
| {  "schema": "http://example.org/specification/v2.0.0",  "metadata": {},  "global": {  "site.global.app\_user": "hive",  "site.global.user\_group": "hadoop",  "site.global.data\_dir": "/var/lib/presto/data",  "site.global.config\_dir": "/var/lib/presto/etc",  "site.global.app\_name": "presto-server-0.227",  "site.global.app\_pkg\_plugin": "${AGENT\_WORK\_ROOT}/app/definition/package/plugins/",  "site.global.singlenode": "true",  "site.global.coordinator\_host": "${COORDINATOR\_HOST}",  "site.global.presto\_query\_max\_memory": "8GB",  "site.global.presto\_query\_max\_memory\_per\_node": "800MB",  "site.global.presto\_server\_port": "8090",  "site.global.catalog": "{'hive': ['connector.name=hive-iriskmaster','hive.metastore.uri=thrift://30.23.76.191:9083'],'tpch': ['connector.name=tpch']}",  "site.global.jvm\_args": "['-server', '-Xmx10240M', '-XX:+UseG1GC', '-XX:G1HeapRegionSize=160M', '-XX:+UseGCOverheadLimit', '-XX:+ExplicitGCInvokesConcurrent', '-XX:+HeapDumpOnOutOfMemoryError', '-XX:OnOutOfMemoryError=kill -9 %p']",  "site.global.log\_properties": "['com.facebook.presto=INFO']",  "application.def": ".slider/package/PRESTO/presto-yarn-package-1.5-0.227.zip",  "java\_home": "/usr/openv/cdh-5.15.2/jdk1.8.0\_181"  },  "components": {  "slider-appmaster": {  "jvm.heapsize": "128M"  }  }  } |

！！！注意：/var/lib/presto/data和/var/lib/presto/etc需要提前创建，且owner必须为yarn

3）修改resources-default.json

|  |
| --- |
| {  "schema": "http://example.org/specification/v2.0.0",  "metadata": {},  "global": {  "yarn.vcores": "1"  },  "components": {  "slider-appmaster": {},  "COORDINATOR": {  "yarn.role.priority": "1",  "yarn.component.instances": "1",  "yarn.component.placement.policy": "1",  "yarn.memory": "8192",  "yarn.label.expression": "presto\_cooridinator"  },  "WORKER": {  "yarn.role.priority": "2",  "yarn.component.instances": "10",  "yarn.component.placement.policy": "1",  "yarn.memory": "8192",  "yarn.label.expression": "presto\_worker"  }  }  } |

4）用修改后的json文件替换掉压缩包中的文件

### 3、启动Presto集群

1）部署Presto

|  |
| --- |
| sudo -u hive ./bin/slider package --install --name PRESTO --package /usr/local/apache-slider-0.92.0-incubating-all/presto-yarn-package-1.5-0.227.zip |

2）启动yarn任务

|  |
| --- |
| sudo -u hive ./bin/slider create presto-app --template appConfig-default.json --resources resources-default.json |