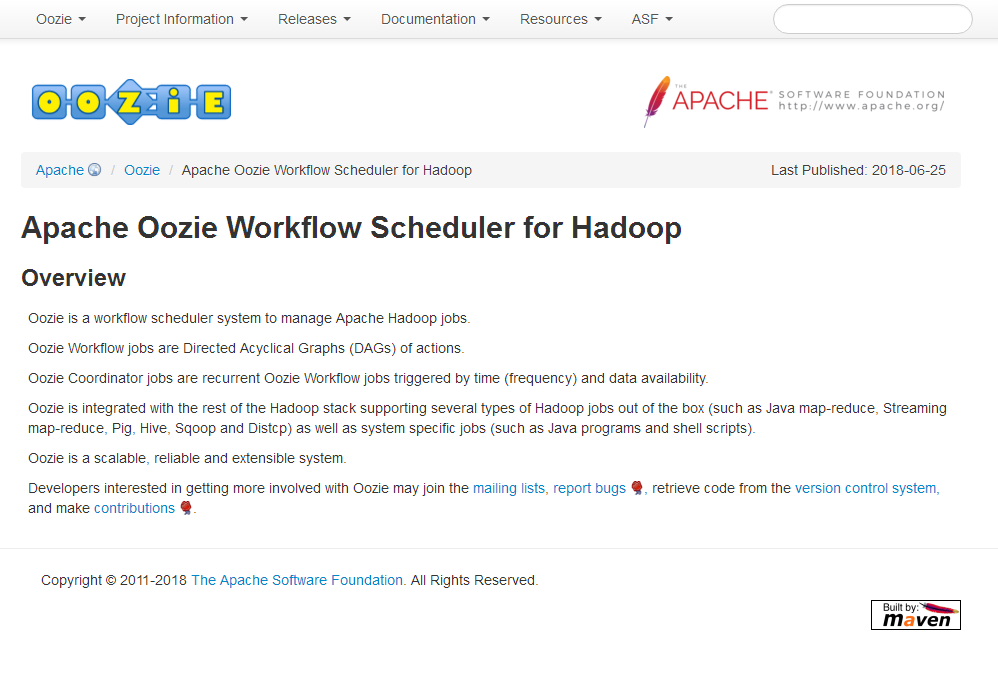
# 大数据技术之Oozie

讲师：刘峰吉

# 一、Oozie简介

http://oozie.apache.org/



Oozie英文翻译为：驯象人。一个基于工作流引擎的开源框架，由Cloudera公司贡献给Apache，提供对Hadoop Mapreduce、Pig Jobs的任务调度与协调。Oozie需要部署到Java Servlet容器中运行。主要用于定时调度任务，多任务可以按照执行的逻辑顺序调度。

# 二、Oozie的功能模块介绍

## 1、模块

1）Workflow

顺序执行流程节点，支持fork（分支多个节点），join（合并多个节点为一个）

2）Coordinator

定时触发workflow

3）Bundle Job

绑定多个Coordinator

## 2、常用节点

1）控制流节点（Control Flow Nodes）

控制流节点一般都是定义在工作流开始或者结束的位置，比如start,end,kill等。以及提供工作流的执行路径机制，如decision，fork，join等。

2）动作节点（Action Nodes）

负责执行具体动作的节点，比如：拷贝文件，执行某个Shell脚本等等。

# 三、Oozie的部署

## 1、解压Oozie

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf oozie-4.0.0-cdh5.3.6.tar.gz -C ../module |

## 2、修改Hadoop配置

oozie是有用户的，默认用户就是oozie

### 1）core-site.xml

|  |
| --- |
| <!-- Oozie Server的Hostname -->  ## 允许哪些框架被oozie 代理用户去操作hadoop修改成自己的用户名  <property>  <name>hadoop.proxyuser.alex.hosts</name>  <value>\*</value>  </property>  <!-- 允许被Oozie代理的用户组 -->  ## 允许oozie 代理用户去操作hadoop，修改成自己的用户名  <property>  <name>hadoop.proxyuser.alex.groups</name>  <value>\*</value>  </property> |

### 2）mapred-site.xml

使用oozie框架，一定要配置jobhistory server

|  |
| --- |
| <!-- 配置 MapReduce JobHistory Server 地址 ，默认端口10020 -->  <property>  <name>mapreduce.jobhistory.address</name>  <value>hadoop102:10020</value>  </property>  <!-- 配置 MapReduce JobHistory Server web ui 地址， 默认端口19888 -->  <property>  <name>mapreduce.jobhistory.webapp.address</name>  <value>hadoop102:19888</value>  </property> |

### 3）yarn-site.xml

|  |
| --- |
| <!-- 任务历史服务 -->  <property>  <name>yarn.log.server.url</name>  <value>http://hadoop102:19888/jobhistory/logs/</value>  </property> |

完成后：记得scp同步到其他机器节点

## 3、重启Hadoop集群

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/start-dfs.sh  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/start-yarn.sh  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/mr-jobhistory-daemon.sh \  start historyserver |

尖叫提示：需要开启JobHistoryServer, 最好执行一个MR任务进行测试

## 4、解压hadooplibs

在oozie根目录下解压hadooplibs

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ tar -xzvf \  oozie-hadooplibs-4.0.0-cdh5.3.6.tar.gz -C ../ |

尖叫提示:完成后Oozie目录下会出现hadooplibs目录,目的是把解压后的hadooplibs文件夹，放到oozie目录下

## 5、创建libext

在Oozie目录下创建libext目录

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ mkdir libext |

## 6、拷贝一些依赖的Jar包

### 1）拷贝hadooplib jar包

将hadooplibs里面的jar包，拷贝到libext目录下

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -ra \  hadooplibs/hadooplib-2.5.0-cdh5.3.6.oozie-4.0.0-cdh5.3.6/\* libext/ |

### 2）拷贝mysql驱动包

拷贝Mysql驱动包到libext目录下

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -ra \  mysql-connector-java-5.1.27-bin.jar libext/ |

oozie 的元数据保存在mysql中，默认derby

## 7、拷贝ext-2.2.zip

将ext-2.2.zip拷贝到libext/目录下，ext是一个js框架，用于展示oozie前端页面

oozie是一个定时的任务调度框架，是有web页面的，依赖ext-2.2.zip

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -r ext-2.2.zip libext/ |

## 8、修改Oozie配置文件

oozie-site.xml

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim conf/oozie-site.xml  <!-- JDBC的驱动-->  <property>  <name>oozie.service.JPAService.jdbc.driver</name>  <value>com.mysql.jdbc.Driver</value>  </property>  <!-- oozie所需的数据库地址-->  <property>  <name>oozie.service.JPAService.jdbc.url</name>  <value>jdbc:mysql://hadoop102:3306/oozie</value>  </property>  <!-- oozie所需的数据库用户名-->  <property>  <name>oozie.service.JPAService.jdbc.username</name>  <value>root</value>  </property>  <!-- oozie所需的数据库密码-->  <property>  <name>oozie.service.JPAService.jdbc.password</name>  <value>123456</value>  </property>  <!--让Oozie引用Hadoop的配置文件\*=不能删-->  <property>  <name>oozie.service.HadoopAccessorService.hadoop.configurations</name>  <value>\*=/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/etc/hadoop</value>  </property> |

## 9、在Mysql中创建Oozie的数据库

进入Mysql并创建oozie数据库

|  |
| --- |
| [alex@hadoop102 ~]$ mysql -uroot -p123456  mysql > create database oozie; |

## 10、初始化Oozie

### 1）上传yarn.tar.gz

上传Oozie目录下的yarn.tar.gz文件到HDFS

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie-setup.sh sharelib \  create -fs \  hdfs://hadoop102:9000 -locallib oozie-sharelib-4.0.0-cdh5.3.6-yarn.tar.gz |

尖叫提示：yarn.tar.gz文件会自行解压

命令的意思是把这个oozie-sharelib-4.0.0-cdh5.3.6-yarn.tar.gz 包解压出来，放到hdfs上面

以wordcount案例，oozie作为定时调度框架，必须先把wc.jar上传到hdfs，oozie才有能力去调度它，放到本地是不行的，执行成功之后，去50070检查对应目录有没有文件生成。

### 2）创建oozie.sql文件

操作的时候mysql一定要有oozie这个数据库

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie-setup.sh db \  create -run -sqlfile oozie.sql |

尖叫提示：这个操作是在oozie这个库里面，创建了一系列的表

### 3）打包项目，生成war包

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie-setup.sh prepare-war |

## 11、启动Oozie服务（附关闭Oozie服务）

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozied.sh start  如需正常关闭Oozie服务，请使用  [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozied.sh stop |

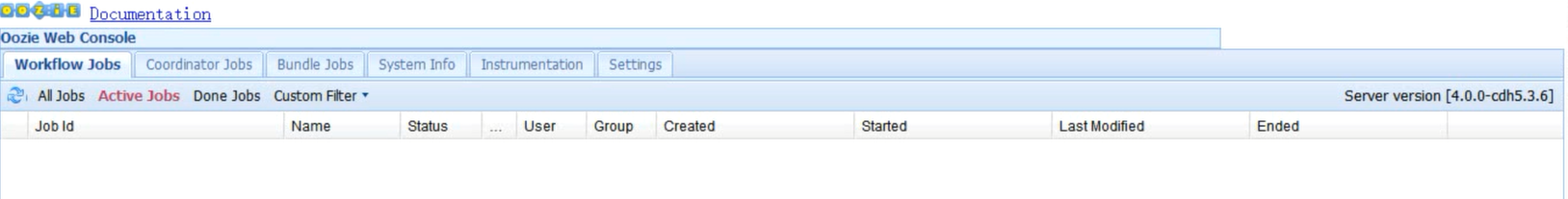
尖叫提示：[alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie-start.sh

WARN: Use of this script is deprecated; use 'oozied.sh start' instead

## 12、访问Oozie的Web页面

看到的web页面， 依赖js框架

|  |
| --- |
| http://hadoop102:11000/oozie |



# 四、Oozie的使用

## 1、案例一

目标：使用Oozie调度Shell脚本

分步实现

### 1）解压官方案例模板

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ tar -xzvf oozie-examples.tar.gz |

### 2）创建工作目录

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ mkdir oozie-apps/ |

### 3）拷贝任务模板到oozie-apps/目录

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -r examples/apps/shell oozie-apps/ |

### 4）随意编写一个脚本p1.sh

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ whereis ifconfig  [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim oozie-apps/shell/p1.sh  #!/bin/bash  /usr/sbin/ifconfig > /tmp/p1.log  [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ chmod 755 oozie-apps/shell/p1.sh |

**尖叫提示：一定使用vim编辑器，centos6 和 centos7 目录地址不一样，一定要whereis一下**

### 5）修改 job.properties

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim oozie-apps/shell/job.properties  #HDFS地址9000端口  nameNode=hdfs://hadoop102:9000  #ResourceManager地址， 8032默认端口不用改  jobTracker=hadoop103:8032  #队列名称  queueName=default  标红是hdfs的上传目录  examplesRoot=oozie-apps  # hdfs://hadoop102:9000/user/alex/oozie-apps/shell  oozie.wf.application.path=${nameNode}/user/${user.name}/${examplesRoot}/shell  EXEC=p1.sh |

### 6）修改 workflow.xml

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim oozie-apps/shell/workflow.xml  <workflow-app xmlns="uri:oozie:workflow:0.4" name="shell-wf">  <start to="shell-node"/>  <action name="shell-node">  <shell xmlns="uri:oozie:shell-action:0.2">  <job-tracker>${jobTracker}</job-tracker>  <name-node>${nameNode}</name-node>  <configuration>  <property>  <name>mapred.job.queue.name</name>  <value>${queueName}</value>  </property>  </configuration>  <exec>${EXEC}</exec>  <!-- <argument>my\_output=Hello Oozie</argument> -->  <file>/user/alex/oozie-apps/shell/${EXEC}#${EXEC}</file>  <capture-output/>  </shell>  <ok to="end"/>  <error to="fail"/>  </action>  <decision name="check-output">  <switch>  <case to="end">  ${wf:actionData('shell-node')['my\_output'] eq 'Hello Oozie'}  </case>  <default to="fail-output"/>  </switch>  </decision>  <kill name="fail">  <message>Shell action failed, error message[${wf:errorMessage(wf:lastErrorNode())}]</message>  </kill>  <kill name="fail-output">  <message>Incorrect output, expected [Hello Oozie] but was [${wf:actionData('shell-node')['my\_output']}]</message>  </kill>  <end name="end"/>  </workflow-app> |

### 7）上传任务配置

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/hdfs dfs -put \  ../oozie-4.0.0-cdh5.3.6/oozie-apps /user/alex |

### 8）执行任务

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie job \  -oozie http://hadoop102:11000/oozie \  -config oozie-apps/shell/job.properties \  -run |

### 9）杀死某个任务

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie job \  -oozie http://hadoop102:11000/oozie \  -kill 0000004-170425105153692-oozie-z-W |

### 10）查看脚本结果

|  |
| --- |
| [alex@hadoop102 ~]$ cat > /tmp/p1.log |

## 2、案例二

目标：使用Oozie执行多个Job调度

oozie 有时候因为err，你给它kill 掉， oozie 的web还显示，这是一个oozie 的bug,不要紧，放着就行了， 强迫症的怎么办呢？？ 去mysql oozie库 相应表把相应数据删除了就行了

分步执行

### 1）解压官方案例模板

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ tar -xzvf oozie-examples.tar.gz |

尖叫提示：案例一已经解压过了，这里就不用再解压了

### 2）编写脚本

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim oozie-apps/shell/p2.sh  #!/bin/bash  /usr/bin/date -R > /tmp/p2.log  [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ sudo chmod 755 oozie-apps/shell/p2.sh  [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ whereis date  [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim oozie-apps/shell/p1.sh  #!/bin/bash  /usr/bin/date -R > /tmp/p1.log  [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ sudo chmod 755 oozie-apps/shell/p1.sh  [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ whereis date |

**尖叫提示：一定使用vim编辑器，标黄的使用绝对路径一定whereis一下**

### 3）修改job.properties文件

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim oozie-apps/shell/job.properties  nameNode=hdfs://hadoop102:9000  jobTracker=hadoop103:8032  queueName=default  examplesRoot=oozie-apps  oozie.wf.application.path=${nameNode}/user/${user.name}/${examplesRoot}/shell  EXEC1=p1.sh  EXEC2=p2.sh |

### 4）修改workflow.xml文件

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim oozie-apps/shell/workflow.xml  <workflow-app xmlns="uri:oozie:workflow:0.4" name="shell-wf">  <start to="p1-shell-node"/>  <action name="p1-shell-node">  <shell xmlns="uri:oozie:shell-action:0.2">  <job-tracker>${jobTracker}</job-tracker>  <name-node>${nameNode}</name-node>  <configuration>  <property>  <name>mapred.job.queue.name</name>  <value>${queueName}</value>  </property>  </configuration>  <exec>${EXEC1}</exec>  <file>/user/alex/oozie-apps/shell/${EXEC1}#${EXEC1}</file>  <!-- <argument>my\_output=Hello Oozie</argument>-->  <capture-output/>  </shell>  <ok to="p2-shell-node"/>  <error to="fail"/>  </action>  <action name="p2-shell-node">  <shell xmlns="uri:oozie:shell-action:0.2">  <job-tracker>${jobTracker}</job-tracker>  <name-node>${nameNode}</name-node>  <configuration>  <property>  <name>mapred.job.queue.name</name>  <value>${queueName}</value>  </property>  </configuration>  <exec>${EXEC2}</exec>  <file>/user/alex/oozie-apps/shell/${EXEC2}#${EXEC2}</file>  <!-- <argument>my\_output=Hello Oozie</argument>-->  <capture-output/>  </shell>  <ok to="end"/>  <error to="fail"/>  </action>  <decision name="check-output">  <switch>  <case to="end">  ${wf:actionData('shell-node')['my\_output'] eq 'Hello Oozie'}  </case>  <default to="fail-output"/>  </switch>  </decision>  <kill name="fail">  <message>Shell action failed, error message[${wf:errorMessage(wf:lastErrorNode())}]</message>  </kill>  <kill name="fail-output">  <message>Incorrect output, expected [Hello Oozie] but was [${wf:actionData('shell-node')['my\_output']}]</message>  </kill>  <end name="end"/>  </workflow-app> |

### 5）上传任务配置

|  |
| --- |
| 删除  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/hdfs dfs -rm -r /user/alex/oozie-apps  上传  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/hdfs dfs -put \  ../oozie-4.0.0-cdh5.3.6/oozie-apps /user/alex/ |

### 6）执行任务

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie job \  -oozie http://hadoop102:11000/oozie \  -config oozie-apps/shell/job.properties \  -run |

## 3、案例三

目标：使用Oozie调度MapReduce任务

分步执行

### 1）MapRreduce Jar

找到一个可以运行的mapreduce任务的jar包

（可以用官方的，也可以是自己写的）

### 2）拷贝官方模板到oozie-apps

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -r examples/apps/map-reduce \  oozie-apps/  [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ rm -rf oozie-apps/map-reduce/lib/\* |

### 3）wordcount

测试一下wordcount在yarn中的运行

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/yarn jar \  share/hadoop/mapreduce/hadoop-mapreduce-examples-2.5.0-cdh5.3.6.jar \  wordcount \  /wcinput/ \  /wcoutput/ |

### 4）修改job.properties

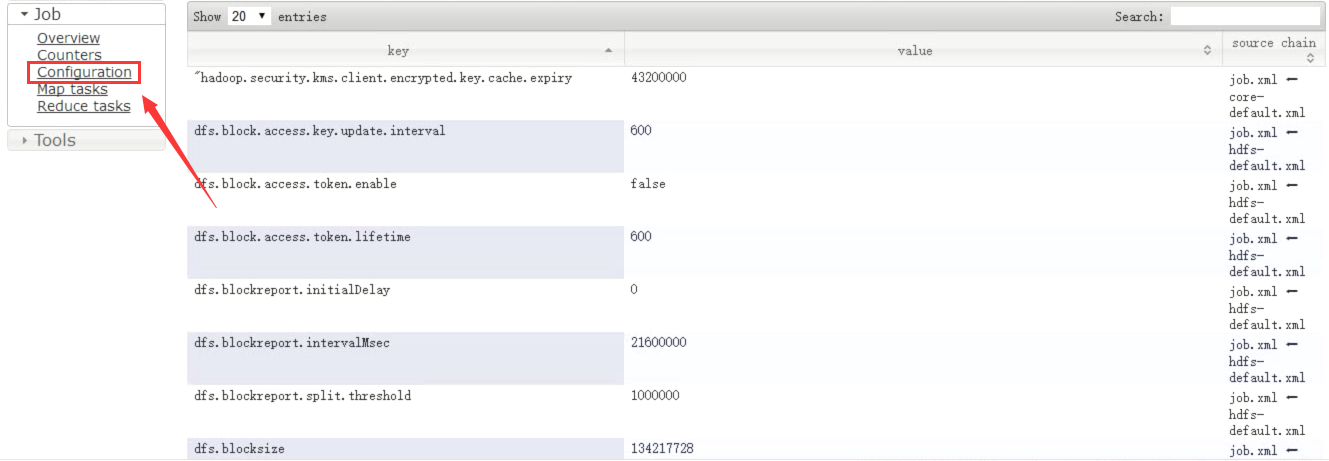
配置map-reduce任务的job.properties

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim oozie-apps/map-reduce/job.properties  nameNode=hdfs://hadoop102:9000  jobTracker=hadoop103:8032  queueName=default  examplesRoot=oozie-apps  #hdfs://hadoop102:9000/user/alex/oozie-apps/map-reduce/workflow.xml  oozie.wf.application.path=${nameNode}/user/${user.name}/${examplesRoot}/map-reduce/workflow.xml  outputDir=output |

### 5）修改workflow.xml

配置map-reduce任务的workflow.xml

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim oozie-apps/map-reduce/workflow.xml  <workflow-app xmlns="uri:oozie:workflow:0.2" name="map-reduce-wf">  <start to="mr-node"/>  <action name="mr-node">  <map-reduce>  <job-tracker>${jobTracker}</job-tracker>  <name-node>${nameNode}</name-node>  <prepare>  <delete path="${nameNode}/${outputDir}/"/>  </prepare>  <configuration>  <property>  <name>mapred.job.queue.name</name>  <value>${queueName}</value>  </property>  <!-- 配置调度MR任务时，使用新的API -->  <property>  <name>mapred.mapper.new-api</name>  <value>true</value>  </property>  <property>  <name>mapred.reducer.new-api</name>  <value>true</value>  </property>  <!-- 指定Job Key输出类型 -->  <property>  <name>mapreduce.job.output.key.class</name>  <value>org.apache.hadoop.io.Text</value>  </property>  <!-- 指定Job Value输出类型 -->  <property>  <name>mapreduce.job.output.value.class</name>  <value>org.apache.hadoop.io.IntWritable</value>  </property>  <!-- 指定输入路径 -->  <property>  <name>mapred.input.dir</name>  <value>/input/</value>  </property>  <!-- 指定输出路径 -->  <property>  <name>mapred.output.dir</name>  <value>/${outputDir}/</value>  </property>  <!-- 指定Map类 -->  <property>  <name>mapreduce.job.map.class</name>  <value>org.apache.hadoop.examples.WordCount$TokenizerMapper</value>  </property>  <!-- 指定Reduce类 -->  <property>  <name>mapreduce.job.reduce.class</name>  <value>org.apache.hadoop.examples.WordCount$IntSumReducer</value>  </property>  <!-- 设置map tasks 最大数量 -->  <property>  <name>mapreduce.tasktracker.map.tasks.maximum</name>  <value>2</value>  </property>  </configuration>  </map-reduce>  <ok to="end"/>  <error to="fail"/>  </action>  <kill name="fail">  <message>Map/Reduce failed, error message[${wf:errorMessage(wf:lastErrorNode())}]</message>  </kill>  <end name="end"/>  </workflow-app> |



### 6）拷贝jar包

拷贝待执行的jar包到map-reduce的lib目录下

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -r \  ../hadoop-2.5.0-cdh5.3.6/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.5.0-cdh5.3.6.jar \  oozie-apps/map-reduce/lib |

### 7）上传配置好的app文件夹到HDFS

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/hdfs dfs -put \  ../oozie-4.0.0-cdh5.3.6/oozie-apps/map-reduce /user/alex/oozie-apps |

### 8）执行任务

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie job \  -oozie http://hadoop102:11000/oozie \  -config oozie-apps/map-reduce/job.properties \  -run |

## 4、案例四

目标：Coordinator周期性调度任务

分步实现：

### 1）配置Linux时区以及时间服务器

检查系统当前时区

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ date -R |

注意这里，如果显示的时区不是+0800，你可以删除localtime文件夹后，再关联一个正确时区的链接过去，命令如下

|  |
| --- |
| [root@hadoop102 ~]# rm -rf /etc/localtime  [root@hadoop102 ~]# ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime |

同步时间

|  |
| --- |
| [root@hadoop102 ~]# ntpdate pool.ntp.org |

修改NTP配置文件

|  |
| --- |
| [root@hadoop102 ~]# vim /etc/ntp.conf  去掉下面这行前面的# ,并把网段修改成自己的网段  restrict 192.168.2.0 mask 255.255.255.0 nomodify notrap  注释掉以下几行  #server 0.centos.pool.ntp.org  #server 1.centos.pool.ntp.org  #server 2.centos.pool.ntp.org  把下面两行前面的#号去掉,如果没有这两行内容,需要手动添加  server 127.127.1.0 # local clock  fudge 127.127.1.0 stratum 10 |

重启NTP服务

|  |
| --- |
| [root@hadoop102 ~]# systemctl start ntpd.service |

注意，如果是centOS7以下的版本，使用命令：service ntpd start

开机启动NTP服务

|  |
| --- |
| [root@hadoop102 ~]# systemctl enable ntpd.service |

注意，如果是centOS7以下的版本，使用命令：chkconfig ntpd on

集群其它节点去同步这台时间服务器时间

首先需要关闭这两台计算机的ntp服务

|  |
| --- |
| [root@hadoop103 ~]# systemctl stop ntpd.service  [root@hadoop104 ~]# systemctl stop ntpd.service  [root@hadoop103 ~]# systemctl disable ntpd.service  [root@hadoop104 ~]# systemctl disable ntpd.service  [root@hadoop103 ~]# systemctl status ntpd.service  [root@hadoop104 ~]# systemctl status ntpd.service  查看ntp服务进程id  [root@hadoop103 ~]# pgrep ntpd  [root@hadoop104 ~]# pgrep ntpd |

centOS7以下，则：service ntpd stop

centOS7以下，则：chkconfig ntpd off

同步第一台服务器linux01的时间

|  |
| --- |
| [root@hadoop103 ~]# ntpdate hadoop102  [root@hadoop104 ~]# ntpdate hadoop102 |

使用root用户制定计划任务,周期性同步时间

|  |
| --- |
| [root@hadoop103 ~]# crontab -e  \*/10 \* \* \* \* /usr/sbin/ntpdate hadoop102  [root@hadoop104 ~]# crontab -e  \*/10 \* \* \* \* /usr/sbin/ntpdate hadoop102 |

重启定时任务

|  |
| --- |
| [root@hadoop103 ~]# systemctl restart crond.service  [root@hadoop104 ~]# systemctl restart crond.service |

centOS7以下使用：service crond restart

### 2）配置oozie-site.xml文件

|  |
| --- |
| <!--修改时区为东八区区时-->  <property>  <name>oozie.processing.timezone</name>  <value>GMT+0800</value>  <description>  Oozie server timezone. Valid values are UTC and GMT(+/-)####, for example 'GMT+0530' would be India  timezone. All dates parsed and genered dates by Oozie Coordinator/Bundle will be done in the specified  timezone. The default value of 'UTC' should not be changed under normal circumtances. If for any reason  is changed, note that GMT(+/-)#### timezones do not observe DST changes.  </description>  </property> |

尖叫提示：该属性去oozie-default.xml中找到即可

### 3）修改js框架中的关于时间设置的代码

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim \  oozie-server/webapps/oozie/oozie-console.js \  function getTimeZone() {  Ext.state.Manager.setProvider(new Ext.state.CookieProvider());  return Ext.state.Manager.get("TimezoneId","**GMT+0800**");  } |

### 4）重启oozie服务

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozied.sh stop  [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozied.sh start |

尖叫提示：并重启浏览器（一定要注意清除缓存）

### 5）拷贝官方模板配置定时任务

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -r examples/apps/cron oozie-apps/ |

### 6）修改模板job.properties

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim oozie-apps/cron/job.properties  nameNode=hdfs://hadoop102:9000  jobTracker=hadoop103:8032  queueName=default  examplesRoot=oozie-apps  oozie.coord.application.path=${nameNode}/user/${user.name}/${examplesRoot}/cron  **#start：必须设置为未来时间，否则任务失败**  start=2020-10-19T17:00+0800  end=2020-10-30T17:00+0800  workflowAppUri=${nameNode}/user/${user.name}/${examplesRoot}/cron  EXEC1=p1.sh  EXEC2=p2.sh |

### 7）修改模板coordinator.xml（定时器）

尖叫提示：最小5分钟

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim oozie-apps/cron/coordinator.xml  <coordinator-app name="cron-coord" frequency="${coord:minutes(5)}" start="${start}" end="${end}" timezone="GMT+0800" xmlns="uri:oozie:coordinator:0.2">  <action>  <workflow>  <app-path>${workflowAppUri}</app-path>  <configuration>  <property>  <name>jobTracker</name>  <value>${jobTracker}</value>  </property>  <property>  <name>nameNode</name>  <value>${nameNode}</value>  </property>  <property>  <name>queueName</name>  <value>${queueName}</value>  </property>  </configuration>  </workflow>  </action>  </coordinator-app> |

### 8）修改模板workflow.xml

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim workflow.xml  <workflow-app xmlns="uri:oozie:workflow:0.5" name="one-op-wf">  <start to="p1-shell-node"/>  <action name="p1-shell-node">  <shell xmlns="uri:oozie:shell-action:0.2">  <job-tracker>${jobTracker}</job-tracker>  <name-node>${nameNode}</name-node>  <configuration>  <property>  <name>mapred.job.queue.name</name>  <value>${queueName}</value>  </property>  </configuration>  <exec>${EXEC1}</exec>  <file>/user/alex/oozie-apps/shell/${EXEC1}#${EXEC1}</file>  <!-- <argument>my\_output=Hello Oozie</argument>-->  <capture-output/>  </shell>  <ok to="p2-shell-node"/>  <error to="fail"/>  </action>  <action name="p2-shell-node">  <shell xmlns="uri:oozie:shell-action:0.2">  <job-tracker>${jobTracker}</job-tracker>  <name-node>${nameNode}</name-node>  <configuration>  <property>  <name>mapred.job.queue.name</name>  <value>${queueName}</value>  </property>  </configuration>  <exec>${EXEC2}</exec>  <file>/user/alex/oozie-apps/shell/${EXEC2}#${EXEC2}</file>  <!-- <argument>my\_output=Hello Oozie</argument>-->  <capture-output/>  </shell>  <ok to="end"/>  <error to="fail"/>  </action>  <decision name="check-output">  <switch>  <case to="end">  ${wf:actionData('shell-node')['my\_output'] eq 'Hello Oozie'}  </case>  <default to="fail-output"/>  </switch>  </decision>  <kill name="fail">  <message>Shell action failed, error message[${wf:errorMessage(wf:lastErrorNode())}]</message>  </kill>  <kill name="fail-output">  <message>Incorrect output, expected [Hello Oozie] but was [${wf:actionData('shell-node')['my\_output']}]</message>  </kill>  <end name="end"/>  </workflow-app> |

### 9）上传配置

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/hdfs dfs -put \  ../oozie-4.0.0-cdh5.3.6/oozie-apps/cron/ /user/alex/oozie-apps |

### 10）启动任务

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie job \  -oozie http://hadoop102:11000/oozie \  -config oozie-apps/cron/job.properties \  -run |

尖叫提示：oozie允许的最小执行任务的频率是5分钟

# 五、可能遇到的问题总结

1、Mysql权限配置

授权所有主机可以使用root用户操作所有数据库和数据表

mysql> grant all on \*.\* to root@'%' identified by '123456';

mysql> flush privileges;

mysql> exit;

2、workflow.xml

workflow.xml配置的时候不要忽略file属性

3、jps查看进程时

注意有没有bootstrap

4、关闭oozie

如果bin/oozied.sh stop无法关闭，则可以使用kill -9 [pid]，之后oozie-server/temp/xxx.pid文件一定要删除。

5、Oozie重新打包时

一定要注意先关闭进程，删除对应文件夹下面的pid文件。（可以参考第4条目）

6、配置文件一定要生效

起始标签和结束标签无对应则不生效，配置文件的属性写错了，那么则执行默认的属性。

7、libext下边的jar存放于某个文件夹中，导致share/lib创建不成功。

8、调度任务时，找不到指定的脚本，可能是oozie-site.xml里面的Hadoop配置文件没有关联上。

9、修改Hadoop配置文件，需要重启集群。一定要记得scp到其他节点。

10、JobHistoryServer必须开启，集群要重启的。

11、Mysql配置如果没有生效的话，默认使用derby数据库。

12、在本地修改完成的job配置，必须重新上传到HDFS。

13、将HDFS中上传的oozie配置文件下载下来查看是否有错误。

14、Linux用户名和Hadoop的用户名不一致。

# 六、Apache Oozie编译源码

## 1、概述

oozie 的使用，最好去使用cdh版本，各个框架兼容性好，如果使用apache 版本，需要逛逛apache社区，看看各个框架的版本依赖，最好不要跨版本号依赖，解决兼容性问题是很头疼的问题

大数据框架中，oozie 是比较复杂的，尤其是部署的时候，编译啊….

定时调度任务框架有

Azkaban Azkaban部署起来非常简单，是最简单的

crontab 滴滴公司就是用的crontab轻量级

oozie 是最强大的,部署的时候,细节很重要

## 2、安装jdk 1.8.44

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf jdk1.8.44.tar.gz -C /opt/module  [alex@hadoop102 software]$ sudo vim /etc/profile  ##JAVA\_HOME  export JAVA\_HOME=/opt/module/jdk1.8.0\_144  export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar  export PATH=$PATH:$JAVA\_HOME/bin  [alex@hadoop102 software]$ source /etc/profile  [alex@hadoop102 module]$ javac -version  javac 1.8.0\_144 |

## 3、安装maven 3.3.9

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf apache-maven-3.3.9.tar.gz -C /opt/module  [alex@hadoop102 apache-maven-3.3.9]$ vim conf/settings.xml  <localRepository>/opt/module/apache-maven-3.3.9/m2/repository</localRepository>  <mirror>  <id>alimaven-new</id>  <mirrorOf>central</mirrorOf>  <name>aliyun maven</name>  <url>https://maven.aliyun.com/repository/central/</url>  </mirror>    <mirror>  <id>nexus-aliyun</id>  <name>Nexus aliyun</name>  <url>http://maven.aliyun.com/nexus/content/groups/public/</url>  <mirrorOf>central</mirrorOf>  </mirror>    <mirror>  <id>aliyun-repos</id>  <name>aliyun-repos</name>  <url>https://maven.aliyun.com/repository/public</url>  <mirrorOf>central</mirrorOf>  </mirror>    <mirror>  <id>sonatype-repos-s</id>  <name>sonatype-repos-s</name>  <url>https://oss.sonatype.org/content/repositories/snapshots</url>  <mirrorOf>central</mirrorOf>  </mirror>  <mirror>  <id>repo2</id>  <name>Mirror from Maven Repo2</name>  <url>http://repo2.maven.org/maven2/</url>  <mirrorOf>central</mirrorOf>  </mirror>  <mirror>  <id>centor</id>  <name>Mirror from Maven central</name>  <url>http://central.maven.org/maven2/</url>  <mirrorOf>central</mirrorOf>  </mirror>  <profile>  <id>jdk-1.8</id>  <activation>  <activeByDefault>true</activeByDefault>  <jdk>1.8</jdk>  </activation>  <properties>  <maven.compiler.source>1.8</maven.compiler.source>  <maven.compiler.target>1.8</maven.compiler.target>  <maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>  </properties>  </profile>  [alex@hadoop102 software]$ sudo vim /etc/profile  ##MAVEN\_HOME  export MAVEN\_HOME=/opt/module/apache-maven-3.3.9  export MAVEN\_REPO=/opt/module/apache-maven-3.3.9/m2/repository  export PATH=$PATH:$MAVEN\_HOME/bin  [alex@hadoop102 software]$ source /etc/profile  [alex@hadoop102 module]$ mvn -v  Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)  Maven home: /opt/module/apache-maven-3.3.9  Java version: 1.8.0\_144, vendor: Oracle Corporation  Java home: /opt/module/jdk1.8.0\_144/jre  Default locale: zh\_CN, platform encoding: UTF-8  OS name: "linux", version: "3.10.0-514.el7.x86\_64", arch: "amd64", family: "unix" |

## 4、安装pig

http://pig.apache.org/

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf pig-0.16.0.tar.gz -C /opt/module/  [alex@hadoop102 software]$ sudo vim /etc/profile  ##PIG\_HOME  export PIG\_HOME=/opt/module/pig-0.16.0  export PATH=$PATH:$PIG\_HOME/bin  [alex@hadoop102 software]$ source /etc/profile  [alex@hadoop102 pig-0.16.0]$ pig -i  [alex@hadoop102 module]$ pig -version  Apache Pig version 0.16.0 (r1746530)  compiled Jun 01 2016, 23:10:49  [alex@hadoop102 pig-0.16.0]$ pig -info |

## 5、下载oozie 源码

http://oozie.apache.org/

oozie-4.3.1.tar.gz

## 6、解压oozie

|  |
| --- |
| [root@hadoop102 software]# tar -xzvf oozie-4.3.1.tar.gz -C /opt/module/ |

## 7、修改Oozie的pom.xml

尖叫提示: 慎重修改，不要一下子改很多，有的jar在中央maven仓，没有

|  |
| --- |
| [root@hadoop102 oozie-4.3.1]# vim oozie-4.3.1/pom.xml  <targetJavaVersion>1.8</targetJavaVersion>  <sourceJavaVersion>1.8</sourceJavaVersion>  <minJavaVersion>1.8</minJavaVersion>  <hadoop.version>2.7.2</hadoop.version>  <profile>  <id>hadoop-2</id>  <activation>  <activeByDefault>true</activeByDefault>  </activation>  <properties>  <hadoop.version>2.7.2</hadoop.version>  <hadoop.majorversion>2</hadoop.majorversion>  <pig.classifier>h2</pig.classifier>  <sqoop.classifier>hadoop200</sqoop.classifier>  <jackson.version>1.9.13</jackson.version>  </properties>  </profile>  <profile>  <id>spark-1</id>  <activation>  <activeByDefault>false</activeByDefault>  </activation>  <properties>  <spark.version>1.6.1</spark.version>  <spark.streaming.kafka.version>1.6.1</spark.streaming.kafka.version>  <spark.bagel.version>1.6.1</spark.bagel.version>  </properties>  </profile>  <profile>  <id>spark-2</id>  <activation>  <activeByDefault>true</activeByDefault>  </activation>  <properties>  <spark.version>2.1.1</spark.version>  <spark.streaming.kafka.version>1.6.2</spark.streaming.kafka.version>  <spark.bagel.version>1.6.2</spark.bagel.version>  </properties>  </profile>  <hadoop.majorversion>2</hadoop.majorversion>  <pig.version>0.16.0</pig.version>  <maven.javadoc.opts>-Xdoclint:none</maven.javadoc.opts>  <sqoop.version>1.4.7</sqoop.version>  <hive.version>1.2.1</hive.version>  <hbase.version>0.94.7</hbase.version> |

<targetJavaVersion>1.8</targetJavaVersion>

<hadoop.version>2.7.2</hadoop.version>

<hadoop.majorversion>2</hadoop.majorversion>

<pig.version>0.16.0</pig.version>

<maven.javadoc.opts>-Xdoclint:none</maven.javadoc.opts>

<spark.version>2.1.1</spark.version>

<sqoop.version>1.99.7</sqoop.version>

<hive.version>1.2.1</hive.version>

<hbase.version>1.3.1</hbase.version>

## 8、下载依赖

|  |
| --- |
| [alex@hadoop102 module]# echo ${MAVEN\_REPO}  /opt/module/apache-maven-3.3.9/m2/repository  pentaho-aggdesigner-algorithm-5.1.5-jhyde.jar  mkdir -p ${MAVEN\_REPO}/org/pentaho/pentaho-aggdesigner-algorithm/5.1.5-jhyde  cd ${MAVEN\_REPO}/org/pentaho/pentaho-aggdesigner-algorithm/5.1.5-jhyde  wget http://repo.spring.io/plugins-release/org/pentaho/pentaho-aggdesigner-algorithm/5.1.5-jhyde/pentaho-aggdesigner-algorithm-5.1.5-jhyde.jar  doxia-module-twiki-1.0-alpha-9.2y.jar  mkdir -p ${MAVEN\_REPO}/org/apache/maven/doxia/doxia-module-twiki/1.0-alpha-9.2y  cd ${MAVEN\_REPO}/org/apache/maven/doxia/doxia-module-twiki/1.0-alpha-9.2y  wget https://repository.cloudera.com/content/repositories/releases/org/apache/maven/doxia/doxia-module-twiki/1.0-alpha-9.2y/doxia-module-twiki-1.0-alpha-9.2y.jar --no-check-certificate  doxia-core-1.0-alpha-9.2y.jar  mkdir -p ${MAVEN\_REPO}/org/apache/maven/doxia/doxia-module-twiki  cd ${MAVEN\_REPO}/org/apache/maven/doxia/doxia-module-twiki  wget https://repository.cloudera.com/content/repositories/releases/org/apache/maven/doxia/doxia-core/1.0-alpha-9.2y/doxia-core-1.0-alpha-9.2y.jar --no-check-certificate |

## 9、编译oozie

注意编译命令后面有很多组件版本，请注意版本匹配问题和默认的版本

很容易在中央仓库中找不到对应版本的jar包

|  |
| --- |
| [root@hadoop102 oozie-4.3.1]# bin/mkdistro.sh -Phadoop-2 -DskipTests(成功) |

bin/mkdistro.sh -DskipTests -Puber -Phadoop-2

-Dhadoop.version=2.7.2

-Dhadoop.auth.version=2.7.2

-Ddistcp.version=2.7.2

-Dsqoop.version=1.4.7

-Dhive.version=1.2.1

-Dhbase.version=0.94.7

bin/mkdistro.sh -DskipTests -Puber -Phadoop-2 \

-Dhadoop.version=2.7.2 \

-Dhadoop.auth.version=2.7.2 \

-Ddistcp.version=2.7.2 \

-Dsqoop.version=1.4.3 \

-Dhive.version=1.2.0 \

-Dhbase.version=0.94.27

[root@hadoop102 oozie-4.3.1]# mvn clean package assembly:single -DskipTests

<tomcat.version>7.0.69</tomcat.version>

<hadoop.version>2.6.0</hadoop.version>

<targetJavaVersion>1.7</targetJavaVersion>

<pig.version>0.15.0</pig.version>

## 10、编译后jar包

|  |
| --- |
| /opt/module/oozie-4.3.1/distro/target/oozie-4.3.1-distro.tar.gz |

## 11、可能出现的错误

[ERROR] Failed to execute goal org.apache.maven.plugins:maven-antrun-plugin:1.6:run (default) on project\

oozie-distro: An Ant BuildException has occured: java.net.ConnectException: 连接超时 (Connection timed out) -> [Help]

这里需要下载tomcat，可能因为公司的网有代理导致java连接超时，那就手动下载一下吧。。。

|  |
| --- |
| [root@hadoop102 oozie-4.3.1]# cd distro/downloads  [root@hadoop102 downloads]# wget \  http://archive.apache.org/dist/tomcat/tomcat-6/v6.0.47/bin/apache-tomcat-6.0.47.tar.gz  [root@hadoop102 downloads]# mv apache-tomcat-6.0.47.tar.gz tomcat-6.0.47.tar.gz  需要重命名下文件名，然后修改oozie-distro.xml,删了下面三行代码，然后再编译一次吧  [root@hadoop102 oozie-4.3.1]# vim distro/oozie-distro.xml  <!--<mkdir dir="downloads"/>  <get src="http://archive.apache.org/dist/tomcat/tomcat-6/v${tomcat.version}/bin/apache-tomcat-${tomcat.version}.tar.gz"  dest="downloads/tomcat-${tomcat.version}.tar.gz" verbose="true" skipexisting="true"/>--> |