尚硅谷大数据技术之Oozie

(作者：尚硅谷大数据研发部)

# 第1章 Oozie简介

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。

# 第2章 Oozie的功能模块介绍

## 2.1 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（定时器）**

定时触发workflow

**3) Bundle Job （不同逻辑放到不同定时器中，用Bundle Job 绑定）**

绑定多个Coordinator

## 2.2 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。

# 第3章 Oozie的部署

## 3.1 部署Hadoop（CDH版本的）

### 3.1.2 修改Hadoop配置 /opt/module/cdh/hadoop-2.5.0-cdh5.3.6/etc/hadoop/ 在这个路径下

三个env 都是只改了JAVA\_HOME 的路径

**core-site.xm**

|  |
| --- |
| <!-- Oozie Server的Hostname -->  <property>  <name>hadoop.proxyuser.atguigu.hosts</name>  <value>\*</value>  </property>  <!-- 允许被Oozie代理的用户组 -->  <property>  <name>hadoop.proxyuser.atguigu.groups</name>  <value>\*</value>  </property>  从原来的hadoop core-site.xml文件中 找到这两个，并把dir修改为cdh路径  <property>  <name>fs.defaultFS</name>  <value>hdfs://hadoop102:9000</value>  </property>  <property>  <name>hadoop.tmp.dir</name>  <value>/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/data/tmp</value>  </property> |

**mapred-site.xml.template** 并把mapred-site.xml.template 文件名修改为mapred-site.xml

|  |
| --- |
| <!-- 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> |

**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 同步完之后修改102、103、104的slave文件

下面重启之前先格式化 /opt/module/cdh/hadoop-2.5.0-cdh5.3.6/bin/hadoop namenode -format

### 3.1.3 重启Hadoop集群

[atguigu@hadoop102 hadoop-2.7.2]$ /opt/module/cdh/hadoop-2.5.0-cdh5.3.6/sbin/start-dfs.sh

[atguigu@hadoop103 hadoop-2.7.2]$ /opt/module/cdh/hadoop-2.5.0-cdh5.3.6/sbin/start-yarn.sh

[atguigu@hadoop102 hadoop-2.7.2]$ /opt/module/cdh/hadoop-2.5.0-cdh5.3.6/sbin/mr-jobhistory-daemon.sh start historyserver

注意：需要开启JobHistoryServer, 最好执行一个MR任务进行测试。

## 3.2 部署Oozie

### 3.2.1 解压Oozie

[atguigu@hadoop102 software]$ tar -zxvf /opt/software/cdh/oozie-4.0.0-cdh5.3.6.tar.gz -C ./opt/module/cdh

### 3.2.2 在oozie根目录下解压oozie-hadooplibs-4.0.0-cdh5.3.6.tar.gz

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ tar -zxvf oozie-hadooplibs-4.0.0-cdh5.3.6.tar.gz -C ../

完成后Oozie目录下会出现hadooplibs目录。

### 3.2.3 在Oozie目录下创建libext目录

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ mkdir libext/

### 3.2.4 拷贝依赖的Jar包

1）将hadooplibs里面的jar包，拷贝到libext目录下：

[atguigu@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驱动包到libext目录下：

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -a /opt/software/mysql-connector-java-5.1.27/mysql-connector-java-5.1.27-bin.jar ./libext/

### 3.2.5 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：

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -a /opt/software/cdh/ext-2.2.zip libext/

### 3.2.6 修改Oozie配置文件

**oozie-site.xml**

|  |
| --- |
| 属性：oozie.service.JPAService.jdbc.driver  属性值：com.mysql.jdbc.Driver  解释：JDBC的驱动  属性：oozie.service.JPAService.jdbc.url  属性值：jdbc:mysql://hadoop102:3306/oozie  解释：oozie所需的数据库地址  属性：oozie.service.JPAService.jdbc.username  属性值：root  解释：数据库用户名  属性：oozie.service.JPAService.jdbc.password  属性值：000000  解释：数据库密码  属性：oozie.service.HadoopAccessorService.hadoop.configurations  属性值：\*=/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/etc/hadoop  解释：让Oozie引用Hadoop的配置文件 |

### 3.2.7 在Mysql中创建Oozie的数据库

进入Mysql并创建oozie数据库：

|  |
| --- |
| $ mysql -uroot -p000000  mysql> create database oozie; |

### 3.2.8 初始化Oozie

**1) 上传Oozie目录下的yarn.tar.gz文件到HDFS：**

提示：yarn.tar.gz文件会自行解压

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie-setup.sh sharelib create -fs hdfs://hadoop102:8020（9000） -locallib oozie-sharelib-4.0.0-cdh5.3.6-yarn.tar.gz

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**

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/ooziedb.sh create -sqlfile oozie.sql -run

**3) 打包项目，生成war包**

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

### 3.2.9 Oozie的启动与关闭

启动命令如下：

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

关闭命令如下：

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

### 3.2.10 访问Oozie的Web页面

<http://hadoop102:11000/oozie>

# 第4章 Oozie的使用

## 4.1 案例一：Oozie调度shell脚本

目标：使用Oozie调度Shell脚本

分步实现：

1）解压官方案例模板

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ tar -zxvf oozie-examples.tar.gz

2）创建工作目录

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

3）拷贝任务模板到oozie-apps/目录

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

4）编写脚本p1.sh

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ vi oozie-apps/shell/p1.sh

内容如下：

|  |
| --- |
| #!/bin/bash  /sbin/ifconfig > /opt/module/p1.log |

5）修改job.properties和workflow.xml文件

**job.properties**

|  |
| --- |
| #HDFS地址 注意nameNode 默认是hdfs://locahost:8020 改成Hadoop102:9000  nameNode=hdfs://hadoop102:8020（9000）  #ResourceManager地址  jobTracker=hadoop103:8032  #队列名称  queueName=default  examplesRoot=oozie-apps  oozie.wf.application.path=${nameNode}/user/${user.name}/${examplesRoot}/shell  EXEC=p1.sh |

**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/atguigu/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> |

6）上传任务配置

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ /opt/module/cdh/hadoop-2.5.0-cdh5.3.6/bin/hadoop fs -put oozie-apps/ /user/atguigu

7）执行任务

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

8）杀死某个任务

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

## 4.2 案例二：Oozie逻辑调度执行多个Job

目标：使用Oozie执行多个Job调度

分步执行：

1. 解压官方案例模板

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ tar -zxf oozie-examples.tar.gz

1. 编写脚本

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ vi oozie-apps/shell/p2.sh

内容如下：

|  |
| --- |
| #!/bin/bash  /bin/date > /tmp/p2.log |

3）修改job.properties和workflow.xml文件

**job.properties**

|  |
| --- |
| nameNode=hdfs://hadoop102:8020  jobTracker=hadoop103:8032  queueName=default  examplesRoot=oozie-apps  oozie.wf.application.path=${nameNode}/user/${user.name}/${examplesRoot}/shell  EXEC1=p1.sh  EXEC2=p2.sh |

**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/atguigu/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/admin/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> |

1. 上传任务配置

$ bin/hadoop fs -rmr /user/atguigu/oozie-apps/

$ bin/hadoop fs -put oozie-apps/mapreduce /user/atguigu/oozie-apps

1. 执行任务

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie job -oozie http://hadoop102:11000/oozie -config oozie-apps/shell2/job.properties -run

## 4.3 案例三：Oozie调度MapReduce任务

目标：使用Oozie调度MapReduce任务

分步执行：

1）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

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -r /opt/module/cdh/oozie-4.0.0-cdh5.3.6/examples/apps/map-reduce/ oozie-apps/

1. **测试一下wordcount在yarn中的运行**

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ /opt/module/cdh/hadoop-2.5.0-cdh5.3.6/bin/yarn jar /opt/module/cdh/hadoop-2.5.0-cdh5.3.6/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.5.0-cdh5.3.6.jar wordcount /input/ /output/

**4) 配置map-reduce任务的job.properties以及workflow.xml**

**job.properties**

|  |
| --- |
| nameNode=hdfs://hadoop102:8020  jobTracker=hadoop103:8032  queueName=default  examplesRoot=oozie-apps  #hdfs://hadoop102:8020/user/admin/oozie-apps/map-reduce/workflow.xml  oozie.wf.application.path=${nameNode}/user/${user.name}/${examplesRoot}/map-reduce/workflow.xml  outputDir=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="${nameNode}/output/"/>  </prepare>  <configuration>  <property>  <name>mapred.job.queue.name</name>  <value>${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>/output/</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>  <property>  <name>mapred.map.tasks</name>  <value>1</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> |

5）拷贝待执行的jar包到map-reduce的lib目录下

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -a /opt /module/cdh/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

6）上传配置好的app文件夹到HDFS

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ /opt/module/cdh/hadoop-2.5.0-cdh5.3.6/bin/hdfs dfs -put oozie-apps/map-reduce/ /user/admin/oozie-apps

7）执行任务

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

## 4.4 案例四：Oozie定时任务/循环任务

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
2. 检查系统当前时区：

# date -R

注意：如果显示的时区不是+0800，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：

|  |
| --- |
| # rm -rf /etc/localtime  # ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime |

同步时间：

|  |
| --- |
| # ntpdate pool.ntp.org |

修改NTP配置文件：

|  |
| --- |
| # vi /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 192.168.122.0 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：

|  |
| --- |
| # systemctl start ntpd.service，  注意，如果是centOS7以下的版本，使用命令：service ntpd start  # systemctl enable ntpd.service，  注意，如果是centOS7以下的版本，使用命令：chkconfig ntpd on |

集群其他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  # systemctl stop ntpd.service，  centOS7以下，则：service ntpd stop  # systemctl disable ntpd.service，  centOS7以下，则：chkconfig ntpd off  # systemctl status ntpd，查看ntp服务状态  # pgrep ntpd，查看ntp服务进程id  同步第一台服务器linux01的时间：  # ntpdate linux01 |

使用root用户制定计划任务,周期性同步时间：

|  |
| --- |
| # crontab -e  \*/10 \* \* \* \* /usr/sbin/ntpdate hadoop102 |

重启定时任务：

|  |
| --- |
| # systemctl restart crond.service，  centOS7以下使用：service crond restart， |

其他台机器的配置同理。

3）配置oozie-site.xml文件 从defult配置文件中 拷贝过来

|  |
| --- |
| 属性：oozie.processing.timezone  属性值：GMT+0800  解释：修改时区为东八区区时 |

注：该属性去oozie-default.xml中找到即可

4）修改js框架中的关于时间设置的代码

|  |
| --- |
| $ vi /opt/module/cdh/oozie-4.0.0-cdh5.3.6/oozie-server/webapps/oozie/oozie-console.js  修改如下：  function getTimeZone() {  Ext.state.Manager.setProvider(new Ext.state.CookieProvider());  return Ext.state.Manager.get("TimezoneId","GMT+0800");  } |

5）重启oozie服务，并重启浏览器（一定要注意清除缓存）

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

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

6）拷贝官方模板配置定时任务\

$ cp -r examples/apps/cron/ oozie-apps/

7）修改模板job.properties和coordinator.xml以及workflow.xml

**job.properties**

|  |
| --- |
| nameNode=hdfs://hadoop102:8020  jobTracker=hadoop103:8032  queueName=default  examplesRoot=oozie-apps  oozie.coord.application.path=${nameNode}/user/${user.name}/${examplesRoot}/cron  #start：必须设置为未来时间，否则任务失败  start=2017-07-29T17:00+0800  end=2017-07-30T17:00+0800  workflowAppUri=${nameNode}/user/${user.name}/${examplesRoot}/cron  EXEC3=p3.sh |

**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> |

**workflow.xml**

|  |
| --- |
| <workflow-app xmlns="uri:oozie:workflow:0.5" name="one-op-wf">  <start to="p3-shell-node"/>  <action name="p3-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>${queueName}</value>  </property>  </configuration>  <exec>${EXEC3}</exec>  <file>/user/atguigu/oozie-apps/cron/${EXEC3}#${EXEC3}</file>  <!-- <argument>my\_output=Hello Oozie</argument>-->  <capture-output/>  </shell>  <ok to="end"/>  <error to="fail"/>  </action>  <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> |

8）上传配置

[atguigu@hadoop102 oozie-4.0.0-cdh5.3.6]$ /opt/module/cdh/hadoop-2.5.0-cdh5.3.6/bin/hdfs dfs -put oozie-apps/cron/ /user/admin/oozie-apps

9）启动任务

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

注意：oozie允许的最小执行任务的频率是5分钟

# 第5章 常见问题总结

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 '000000';  mysql> flush privileges;  mysql> exit; |

2）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。