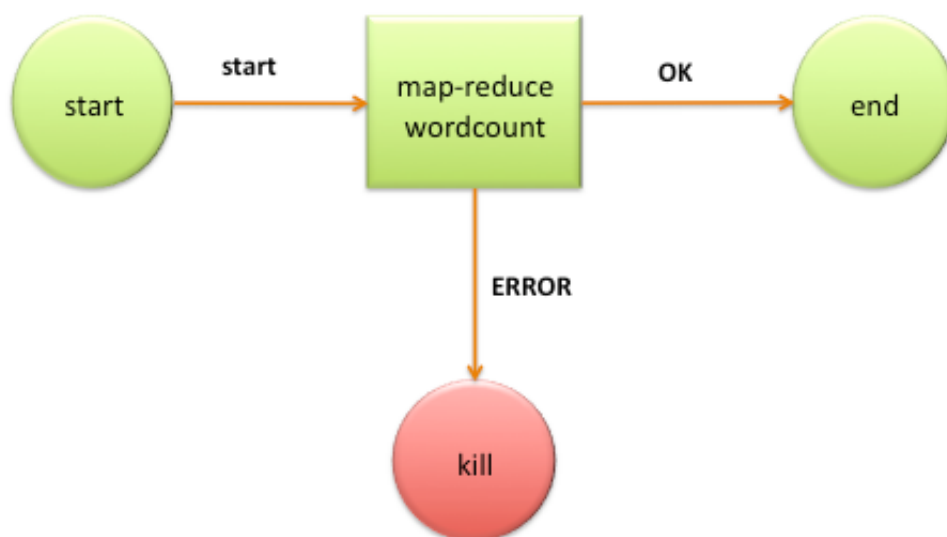


2018-03-15 16:13:26 星期四

Overview

1. Oozie工作流使用hPDL来定义；
2. Oozie工作流在Remote Host上启动，执行完成后会返回执行结果；
3. Oozie工作流中包含：control flow nodes和 action nodes:
 - control flow nodes 包括start、stop、fail、decision、fork、join
 - action nodes 包括 Hadoop map-reduce, Hadoop file system, Pig, SSH, HTTP, eMail , Oozie sub-workflow
4. Oozie需要依赖hadoop的proxy的代理功能



例子

运行 oozie-examples.tar.gz包中的一个example：

解压oozie-examples.tar.gz，修改examples/apps/map-reduce/job.properties文件

```
1 nameNode=hdfs://nameservice1 #只要配置HDFS的集群名即可（配置文件Cloudera已经为我们
2 jobTracker=yarnrm #改成这个
3 queueName=default
4 examplesRoot=examples #这里配置的是HDFS上的路径，绝对路径是/user/${user.name}/${e
5
```

```
6 oozie.wf.application.path=\${nameNode}/user/\${user.name}/\${examplesRoot}/app
7 outputDir=map-reduce #HDFS的输出路径
```

上传整个examples文件夹到指定的examplesRoot目录

```
1 # 启动
2 oozie job -oozie http://cdh1:11000/oozie -config examples/apps/map-reduce/job.
3 # 查看状态 (PS:通过OOZIE_URL环境变量可以省略--oozie 参数)
4 export OOZIE_URL="http://cdh1:11000/oozie"
5 oozie job -info 0000001-180315040600623-oozie-oozi-W
6
```

Oozie Specification

Definitions

1. Workflow (工作流) 可以协调以下类型的action : Hadoop,Pig,sub-workflows。
2. 完整Workflow application包含 : workflow definition、jar包文件、native libraries、以及其他资源文件。
3. 当前Oozie不支持action循环 , 所有定义的workflows必须是严格的有向无环图。
4. Workflow有以下类型的控制值节点 : start、end、kill、decision、fork、join ([写法参考](#))。
5. 关于Action节点的重试 , 当Action启动失败时 , Oozie可以配置重试 (如网络故障、传输故障等)。但是运行过程中发生ERROR时Oozie不会进行重试 (将重试交给了Hadoop)。

Action

map-reduce

包括以下可用字段 :

1. job-xml : 用来指定Hadoop Conf文件 (一般放在HDFS上)
2. configuration : 指定Hadoop Conf的配置项
3. config-class : 指定一个自定义的Class , 我们可以通过这个Class配置Job的一些相关配置 (一般而言我们通过

org.apache.oozie.action.hadoop.OozieActionConfigurator实现自己的config-class)

4. streaming : 包括mapper和reducer两个属性, 这个两个属性中我们可以定义可执行脚本 (如果在job-xml或者configuration中定义了mapred.mapper.class和mapred.reducer.class , streaming会被忽略)。这个配置主要是为了Hadoop Streaming准备的 (workflow.xml里可能需要配置ShareLib)。
5. pipes:和streaming类似, 但是似乎多了一个program 属性? (主要Hadoop pipes 用的, 即通过C++设计的MR任务)

关于hadoop MR任务我们可以定义在以下几个地方:

streaming , job-xml , configuration , config-class (按这个顺序加载, 传统java写的mr任务用后三个定义, hadoop streaming用第一个) !

pig

略

fs

1. 支持的操作有: move , delete , mkdir , chmod , touchz , chgrp
2. fs操作是同步操作, 当操作确实返回后, 线程才会退出
3. fs操作不支持回滚, 因此需要每次操作时需要检查路径的有效性
4. 支持使用HCatalog地址来操作HBase的分区文件
5. delete操作递归删除所有文件夹的文件 (skipTrash)
6. action里的所有路径需要带system URI

```
1 <workflow-app name="[WF-DEF-NAME]" xmlns="uri:oozie:workflow:0.5">
2     ...
3     <action name="[NODE-NAME]">
4         <fs>
5             <delete path='[PATH]' skip-trash='[true/false]'/>
6             ...
7             <mkdir path='[PATH]'/>
8             ...
9             <move source='[SOURCE-PATH]' target='[TARGET-PATH]'/>
10            ...
11            <chmod path='[PATH]' permissions='[PERMISSIONS]' dir-files='false'
12            ...
```

```

13         <touchz path='[PATH]' />
14         ...
15         <chgrp path='[PATH]' group='[GROUP]' dir-files='false' />
16     </fs>
17     <ok to="[NODE-NAME]"/>
18     <error to="[NODE-NAME]"/>
19 </action>
20 ...
21 </workflow-app>

```

ssh

0.2版本之后已经被删除了，主要功能是执行远程机器上的脚本。

sub-workflow

运行一个子workflow，workflow会等待子工作流完成。

```

1 <workflow-app name="sample-wf" xmlns="uri:oozie:workflow:0.1">
2     ...
3     <action name="a">
4         <sub-workflow>
5             <app-path>child-wf</app-path>
6             <configuration>
7                 <property>
8                     <name>input.dir</name>
9                     <value>${wf:id()}/second-mr-output</value>
10                </property>
11            </configuration>
12        </sub-workflow>
13        <ok to="end"/>
14        <error to="kill"/>
15    </action>
16    ...
17 </workflow-app>

```

java

1. java action 在hadoop集群中作为一个单mapper task运行。

2. Java主程序需要使用System.exit(int n)来退出。
3. oozie将hadoop的环境变量注入到java的环境变量中，通过以下方式可以加载环境变量：

```
1 // loading action conf prepared by Oozie
2 Configuration actionConf = new Configuration(false);
3 actionConf.addResource(new Path("file:///", System.getProperty("oozie.action.c
```

```
1 <workflow-app name="sample-wf" xmlns="uri:oozie:workflow:0.1">
2     ...
3     <action name="myfirstjavajob">
4         <java>
5             <job-tracker>foo:8021</job-tracker>
6             <name-node>bar:8020</name-node>
7             <prepare>
8                 <delete path="${jobOutput}"/>
9             </prepare>
10            <configuration>
11                <property>
12                    <name>mapred.queue.name</name>
13                    <value>default</value>
14                </property>
15            </configuration>
16            <main-class>org.apache.oozie.MyFirstMainClass</main-class>
17            <java-opts>-Dblah</java-opts>
18            <arg>argument1</arg>
19            <arg>argument2</arg>
20        </java>
21        <ok to="myotherjob"/>
22        <error to="errorcleanup"/>
23    </action>
24    ...
25 </workflow-app>
```

shell action

参

考：http://oozie.apache.org/docs/3.3.1/DG_ShellActionExtension.html#Shell_Action

shell action似乎是cloudera扩展的一个action。

1. shell action会随机选择一台机器运行shell脚本
2. 通过\$OOZIE_ACTION_CONF_XML可以获取hadoop conf配置
3. 本质上是执行了一个hadoop Single Mapper Task
4. 如果想提交Spark，可以用shell结合spark-submit，这样能够非阻塞的提交spark任务。另一个spark action，提交job时会发生阻塞。

其他扩展action

参考：http://oozie.apache.org/docs/4.2.0/index.html#Action_Extensions

Action Extensions

- [Email Action](#)
- [Shell Action](#)
- [Hive Action](#)
- [Hive 2 Action](#)
- [Sqoop Action](#)
- [Ssh Action](#)
- [DistCp Action](#)
- [Spark Action](#)
- [Writing a Custom Action Executor](#)

Parameterization of Workflows

1. workflow中的definitions基于JSP Expression Language syntax，支持变量定义、以及函数定义；
2. 通过config-default.xml和job.properties设定变量；
3. 带有"."的变量定义不能直接访问，可以通过\${wf:conf('user.name')}
4. 通过内置的el函数可以获得：workflow的信息、hadoop集群的信息、HDFS中的文件信息、HCatalog信息、以及其他常用参数（[参考](#)）；

Workflow Notifications

包括：

oozie.wf.workflow.notification.url -- wf状态变更通知

oozie.wf.action.notification.url -- action状态变更通知

Share Libraries

1. 正常情形下app依赖的jar，被放在\${oozie.wf.application.path}/lib/目录下。通过

配置，`oozie.use.system.libpath`和`oozie.libpath=`，可以将HDFS上的共享目录加入到`path`中。

2. `oozie admin -shareliblist`打印当前HDFS上的所有共享Jar包，一般`sharelib`按照`action`的类型放在`${oozie.service.WorkflowAppService.system.libpath}`目录下。
3. 使用`oozie.action.sharelib.for.#ACTIONTYPE#`可以覆盖某个`action`的 Share Library地址

User-Retry for Workflow Actions

1. 重方式可以为`periodic`或者`exponential`，通过`oozie.service.LiteWorkflowStoreService.user.retry.policy`（在`oozie.site.xml`）中定义；
2. 通过`oozie.service.LiteWorkflowStoreService.user.retry.error.code.ext`可以定义允许重试的错误码

```
1 <workflow-app xmlns="uri:oozie:workflow:0.5" name="wf-name">
2 <action name="a" retry-max="2" retry-interval="1" retry-policy="exponential">
3 </action>
```

Suspend On Nodes

使用`oozie.suspend.on.nodes`（在`job.properties`指定），会使`oozie`运行到指定`node`时挂起，这个主要用来`debug`。

Oozie Web Services API

参考：<http://oozie.apache.org/docs/4.3.1/WebServicesAPI.html>

Workflow Jobs Recovery

1. 通过`oozie.wf.rerun.skip.nodes`指定要跳过的`node`，通过`oozie.wf.rerun.failnodes`（`true` or `false`）指定重新运行失败的节点。
2. `recovery workflow job`使用原来的`workflow job ID`。
3. 重新运行时可以使用新的 `job parameters`和`workflow application path`。

最佳实践：Oozie工作流属性配置的三种方式

Oozie有三种方法可以给工作流提供属性属性配置：

1. App部署文件夹根目录下的：config-default.xml
对于静态的，不会因每次启动而发生改变的配置项，全部配置在config-default.xml。
2. 作业属性文件:job.properties
避免使用job.properties。
3. 在命令行中指定属性:-Dkey=value
对于动态的，会因为每次启动而变化的配置项（典型的例子是coordinator的start time), 通过命令行的方式指定是最合适的。

例子：

```
1 oozie job -run \  
2 -Doozie.wf.application.path=hdfs://your-namenode:8020/your/app/path \  
3 -DPARAM1=${PARAM1} \  
4 -DPARAM1=${PARAM2}
```

Coordinator

Overview

1. 虽然Oozie进程所处的时区不支持夏令时，但是可以在coordinator applications中提供夏令时能力？
2. 配置时区：

```
1 <property>  
2   <name>oozie.processing.timezone</name>  
3   <value>GMT+0800</value>  
4 </property>
```

1. 时间间隔（frequency属性）的配置：由于可能需要处理DST，因此尽量不要使用常数，而是使用内置的EL表达式；如`coord: endOfDays(intn)`、`{coord:days(int n)}`
2. frequency属性支持使用crontab的语法进行配置（[参考](#)）

Synchronous Datasets

例子：

```
1  ```.xml
2  <datasets>
3      <!-- 用include可以定义一个外部数据集 -->
4      <include>hdfs://foo:8020/app/dataset-definitions/globallogs.xml</include>
5      <dataset name="logs" frequency="${coord:days(1)}"
6          initial-instance="2009-01-01T10:30Z" timezone="America/Los_Angeles"
7          <uri-template>
8              <!-- ${YEAR}这些变量是oozie在指定时间生成的 -->
9              hdfs://foo:8020/usr/app/logs/${YEAR}/${MONTH}/${DAY}/${HOUR}/${MIN
10          </uri-template>
11          <!-- 省略done-flag时会检查目录下的_SUCCESS文件，为空时只要文件夹存在就触发
12          <done-flag></done-flag>
13      </dataset>
14 </datasets>
```

特点：

1. 支持hdfs路径和HCatalog路径；
2. 几个比较有用的函数：`${coord:current(0)}`、`${coord:dataIn('dataset name')}`、`${coord:dataOut('dataset name')}`、`${coord:offset(int n, String timeUnit)}`，[参考](#)。
3. 可以定义数据集的输入的逻辑关系，包括：OR、AND、COMBINE(指同一个instance可以来自A或者B，但是最后的范围需要和之前定义的相同)、MIN、WAIT (in minutes)

```
1 <input-logic>
2     <or>
3         <and name="AorB">
4             <data-in dataset="A"/>
5             <data-in dataset="B"/>
6         </and>
7         <and name="CorD">
8             <data-in dataset="C"/>
```

```
9         <data-in dataset="D"/>
10     </and>
11 </or>
12 </input-logic>
```

夏令时的处理，参考

Asynchronous Datasets

To be defined !!

Coordinator Application

1. 触发条件包括：DataSet就绪，时间周期，外部事件触发
2. Execution Policies可以进行以下设定：
 - Timeout：coordinator actions可以等待的最长时间；
 - Concurrency：action的并发度；
 - Execution strategy：oldest first、newest first、none、last one only
 - Throttle：允许激活的action数目
1. 关于Coordinator Job和Coordinator Action的状态；

Oozie Bundle

参考：<https://oozie.apache.org/docs/4.3.1/BundleFunctionalSpec.html>

通过oozie bundle可以约束多个coordinator job成为一个bundle Job，单个bundle Job可以进行start、stop、suspend、resume、rerun操作

JMS通知和SLA

参考：https://oozie.apache.org/docs/4.3.1/DG_JMSNotifications.html

参考：https://oozie.apache.org/docs/4.3.1/DG_SLAMonitoring.html

FAQ

- 使用Cloudera安装完之后没法打开页面

Oozie web console is disabled.

To enable Oozie web console install the Ext JS library.

Refer to [Oozie Quick Start](#) documentation for details.



[Documentation](#)

Oozie Web Console

下载ExtJS 2.2 , 放到/opt/cloudera/parcels/CDH/lib/oozie/libext目录