业务流程管理JBPM（Java Business Process Management）是覆盖了业务流程管理、工作流、服务协作等领域的一个开源的、灵活的、易扩展的可执行流程语言框架

Oozie是类似JBPM的用于 Hadoop 平台的开源的工作流调度引擎，用来管理Hadoop作业，属于web应用程序，由Oozie client和Oozie Server两个组件构成，Oozie Server运行于Java Servlet容器（Tomcat）中的web程序

Oozie统一调度hadoop系统中常见的MapReduce任务启动、hdfs操作、shell调度、hive操作等，使得复杂的依赖关系、时间触发、事件触发使用xml语言进行表达，开发效率提高，一组任务使用一个有向无环图DAG（Directed Acyclic Graph）来表示，使用图形表达流程逻辑更加清晰，支持很多种任务调度，能完成大部分hadoop任务处理，程序定义支持EL常量和函数，表达更加丰富

Oozie Web控制台需要解压ext-2.2（固定的控制台版本，Oozie不兼容其他控制台版本）到/var/lib/oozie目录下命令unzip ext-2.2.zip -d /var/lib/oozie/，Oozie服务中配置启用web控制台，保存后重启oozie服务

Oozie相关配置：节点内存配置

oozie.service.callablequeueservice.callable.concurrency（节点并发）

oozie.service.callablequeueservice.queue.size（队列大小）

oozie.service.ActionService.executor.ext.classes（扩展）

Oozie共享库默认在HDFS上的/user/oozie/share/lib目录下

Oozie的web控制台地址：http://oozie\_host\_ip:11000/oozie/

oozie管理：任务列表查看、任务状态查看、流程返回信息、节点信息查看、流程图信息、日志查看、系统信息查看和配置

Oozie的一个Bundle任务可以管理多个Coordinator任务，一个Coordinator任务可以管理多个Workflow任务

Oozie CLI 命令：

oozie job -oozie http://ip:11000/oozie/ -config job.properties（Oozie任务的配置文件） –run 启动任务

oozie job -oozie http://ip:11000/oozie/ -kill Oozie的任务ID 停止任务

oozie job -oozie http://ip:11000/oozie/ -config job.properties -submit提交任务，提交任务后Oozie会为任务指定唯一表示的任务ID

oozie job -oozie http://ip:11000/oozie/ -config job.properties -start Oozie的任务ID 开始任务

oozie job -oozie http://ip:11000/oozie/ -config job.properties -info Oozie的任务ID 查看任务执行情况

Job配置job.properties一般格式：

nameNode=hdfs地址

jobTracke=jobTracker地址，Hadoop2.x之后没有jobTracker配置为yarn的resourceManager:8032

queueName=oozie队列（default）

examplesRoot=全局目录（examples）

oozie.usr.system.libpath=是否加载用户lib目录（true/false）

oozie.libpath=用户lib库

oozie.wf.appication.path=oozie流程所在HDFS地址（workflow.xml）

user.name=当前用户

oozie.coord.application.path=coordinator.xml地址

oozie.bundle.application.path=bundle.xml地址

WorkFlow配置 workflow.xml，workflow.xml中的标签称为节点，节点分为流程控制节点和动作节点，${变量名或者函数名}引用job.properties或workflow.xml的变量和函数

流程控制节点（标签）：

start 定义workflow开始

end 定义workflow结束

decision 实现switch功能

sub-workflow 调用子workflow

kill 杀死workflow

fork 并发执行workflow

join 并发执行结束（与fork一起使用）

动作节点：shell、java、fs、MR、hive、sqoop

workflow.xml的一般格式：

<workflow-app xmlns="uri:oozie:workflow:0.3" name="shell-wf">版本信息，需要与Oozie兼容的版本一致

<start to="shell-node"/>

<action name="shell-node">

<shell xmlns="uri:oozie:shell-action:0.1">shell节点

<job-tracker>${jobTracker}</job-tracker>

<name-node>${nameNode}</name-node>

<configuration>

<property>

<name>mapred.job.queue.name</name>

<value>${queueName}</value>

</property>

</configuration>

<exec>echo</exec>

[<argument>hi shell in oozie</argument>]

<exec>impala-shell</exec>Impala节点

[<argument>-i</argument>]Impala参数

<capture-output/>

</shell>

<fs>HDFS节点

<delete path='[PATH]'/>

<mkdir path='[PATH]'/>

<move source='[SOURCE-PATH]' target='[TARGET-PATH]'/>

<chmod path='[PATH]' permissions='[PERMISSIONS]' dir-files='false' />

<touchz path='[PATH]' />

<chgrp path='[PATH]' group='[GROUP]' dir-files='false' />

</fs>

<java>Java节点

<job-tracker>${jobTracker}</job-tracker>

<name-node>${nameNode}</name-node>

<prepare>准备节点

<delete path="${nameNode}/user/path" />

<mkdir path="${nameNode}/user/path" />

</prepare>

<configuration>

<property>

<name>mapred.job.queue.name</name>

<value>${queueName}</value>

</property>

</configuration>

<main-class>com.pagename.classname</main-class>

[<arg>args1</arg>]

</java>

<ok to="end"/>

<error to="fail"/>

</action>

<decision name="[NODE-NAME]">

<switch>

[<case to="[NODE\_NAME]">[PREDICATE]</case>]

<default to="[NODE\_NAME]" />

</switch>

</decision>

<fork name="[FORK-NODE-NAME]">

[<path start="[NODE-NAME]" />]

</fork>

<join name="[JOIN-NODE-NAME]" to="[NODE-NAME]" />

<kill name="fail">

<message>Map/Reduce failed, error message[${wf:errorMessage(wf:lastErrorNode())}]</message>

</kill>

<end name="end"/>

</workflow-app>

EL函数：

基本的EL函数：

String firstNotNull(String value1, String value2)

String concat(String s1, String s2)

String replaceAll(String src, String regex, String replacement)

String appendAll(String src, String append, String delimeter)

String trim(String s)

String urlEncode(String s)

String timestamp()

String toJsonStr(Map) (since Oozie 3.3)

String toPropertiesStr(Map) (since Oozie 3.3)

String toConfigurationStr(Map) (since Oozie 3.3 )

WorkFlow EL：

String wf:id() 返回当前workflow作业ID

String wf:name() 返回当前workflow作业NAME

String wf:appPath() 返回当前workflow的路径

String wf:conf(String name) 获取当前workflow的完整配置信息

String wf:user() 返回启动当前job的用户

String wf:callback(String stateVar) 返回结点的回调URL，其中参数为动作指定的退出状态

int wf:run() 返回workflow的运行编号，正常状态为0

Map wf:actionData(String node) 返回当前节点完成时输出的信息

int wf:actionExternalStatus(String node) 返回当前节点的状态

String wf:lastErrorNode() 返回最后一个ERROR状态推出的节点名称

String wf:errorCode(String node) 返回指定节点执行job的错误码，没有则返回空

String wf:errorMessage(String message) 返回执行节点执行job的错误信息，没有则返回空

HDFS EL：

boolean fs:exists(String path)

boolean fs:isDir(String path)

long fs:dirSize(String path) 目录则返回目录下所有文件字节数；否则返回-1

long fs:fileSize(String path) 文件则返回文件字节数；否则返回-1

long fs:blockSize(String path) 文件则返回文件块的字节数；否则返回-1