

Oozie 工作流引擎

www.huawei.com



目标

- 学完本课程后，您将能够：
 - 掌握**Oozie**的基本概念
 - 掌握**Oozie**的使用场景
 - 掌握**Oozie**系统架构
 - 掌握**Oozie**的基本功能



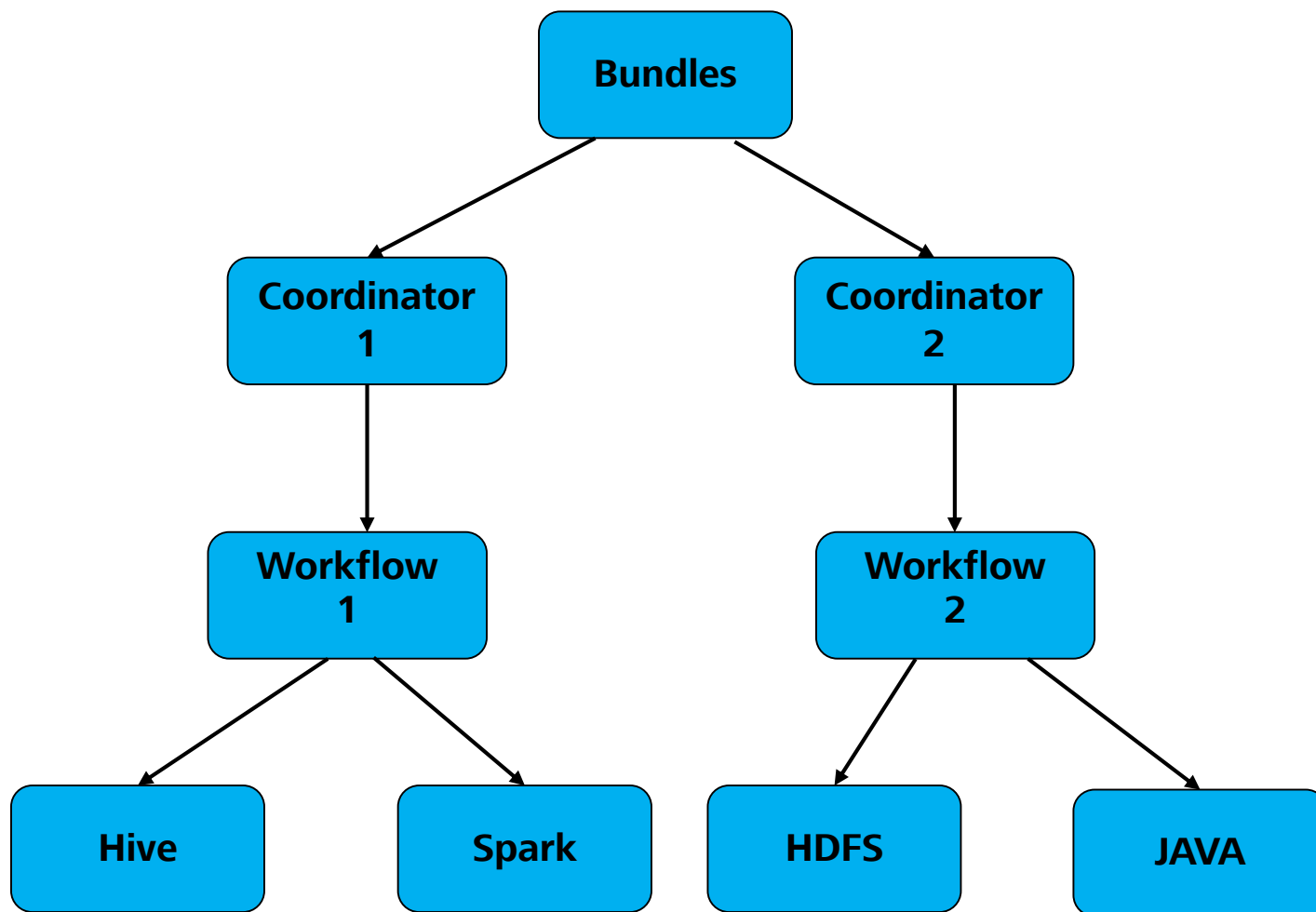
目录

1. **Oozie**基本介绍
2. **Oozie**架构与原理
3. 关键特性介绍
4. 应用举例

Oozie概述

- **Oozie**是一个**Hadoop**作业的工作流调度管理系统。
- **Oozie**工作流（**Workflow**）是放置在控制依赖**DAG**（有向无环图）中的一组动作（**Action**）集合，控制依赖可确保后续操作在前面的操作已成功完成后才会启动。
- **Oozie**的协调作业（ **Coordinator** ）是通过时间（频率）和有效数据来触发当前的**Oozie**工作流。
- **Oozie**支持多种**Hadoop**作业（包括： **HDFS**， **MapReduce**， **Hive**， **Streaming MR**， **Loader**， **Spark**， **Distcp**）以及系统类作业（例如**Java**与**Shell**）。

Oozie引擎介绍



Oozie引擎介绍

Oozie支持如下三种引擎：

- **Workflow Engine**

工作流引擎 以**Action**的方式运行工作流**Job**，用来执行 **Map/Reduce**与**Pig**等**Job**。

- **Coordinator Engine**

协调员引擎 基于时间和数据触发器运行工作流（**Workflow**），它可以定时调度工作流(例如每隔一小时运行一次)，也可以在数据可用的时候触发工作流(例如等到输入数据准备好的情况下再启动工作流)。

- **Bundle Engine**

管道引擎 提供了更高级别的**Oozie**抽象，用户可以批量设置**Coordinator**应用，并且能够 启动/停止/暂停/恢复/重新运行 一组**Coordinator**应用，从而提高易用性与灵活性。

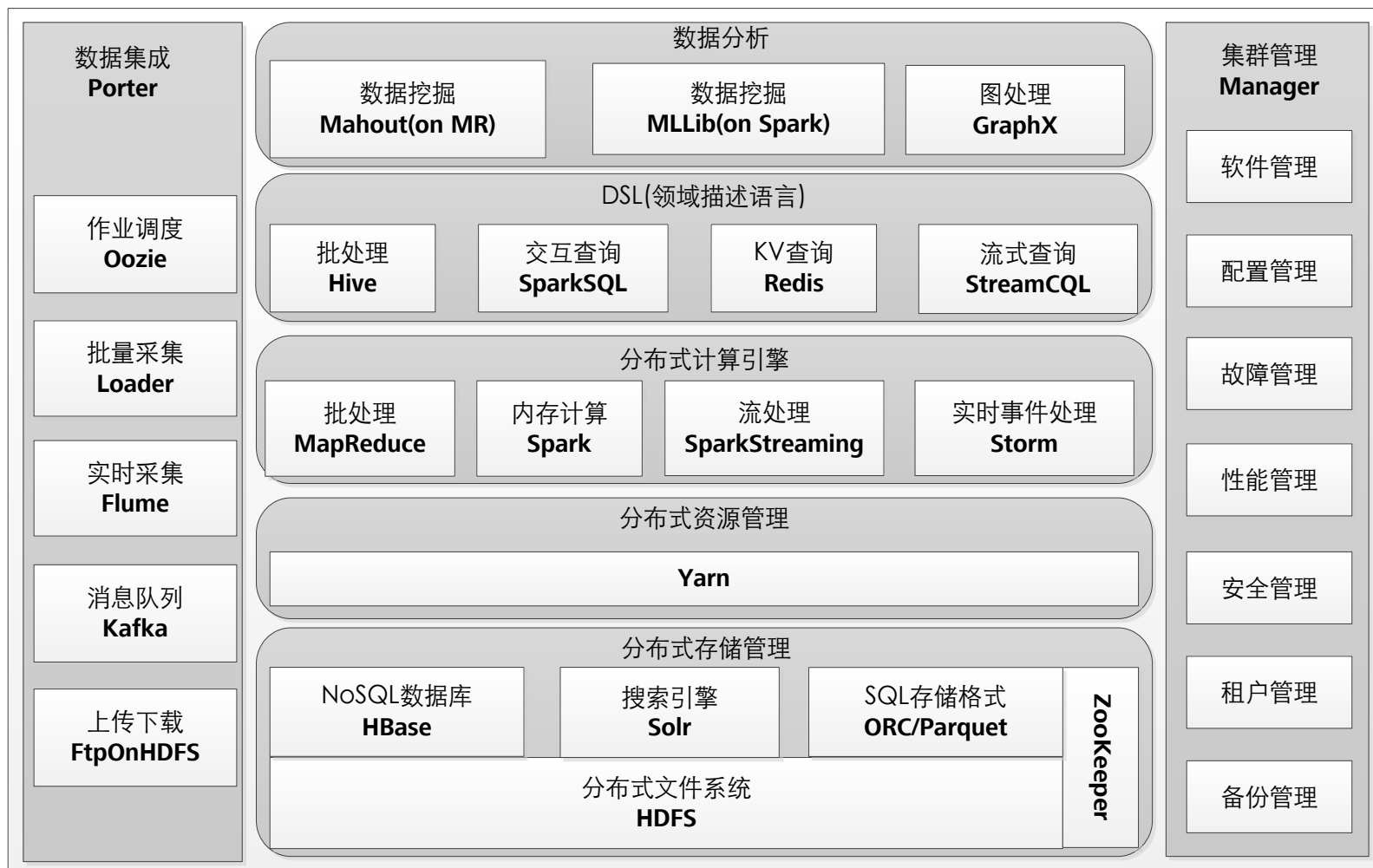
应用场景

Oozie是一个工作流调度引擎，对各种类型的**Hadoop**作业进行编排与调度。

主要应用于以下几种场景：

- 编排与管理逻辑复杂的多种类型的**Hadoop**作业，按照指定的顺序协同运行，例如**ETL**任务
- 基于时间(频率)触发工作流，例如每天/小时执行一次的重复任务或者指定时间执行的任务
- 基于数据有效性触发工作流，当依赖的**HDFS**数据产生之后才触发下一步动作，可用于数据管道处理
- 实时监控与管理集群的工作流，快速定位问题；掌握集群的资源使用情况，并根据需要调整工作流的调度，最大化利用集群资源

Oozie产品定位

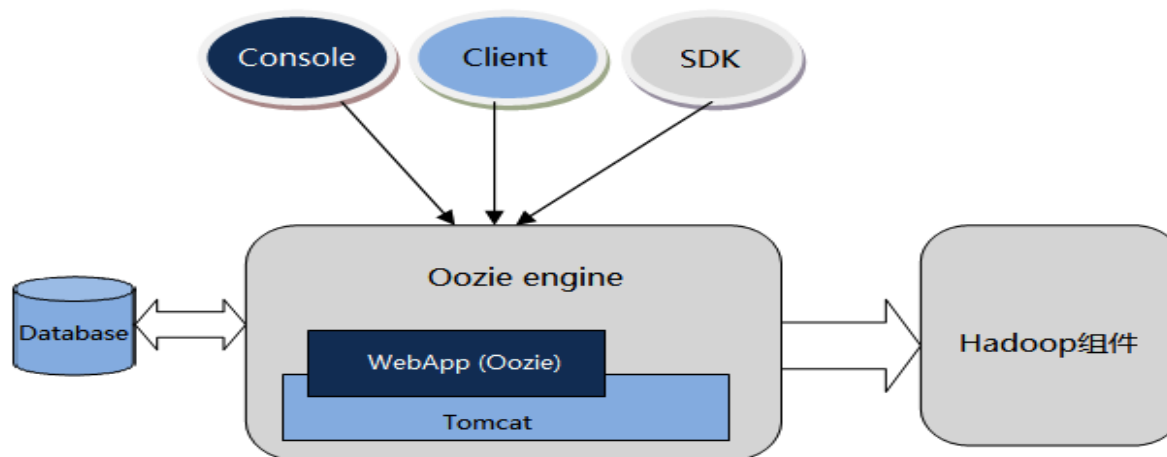




目录

1. **Oozie**基本介绍
2. **Oozie**架构与原理
3. 关键特性介绍
4. 应用举例

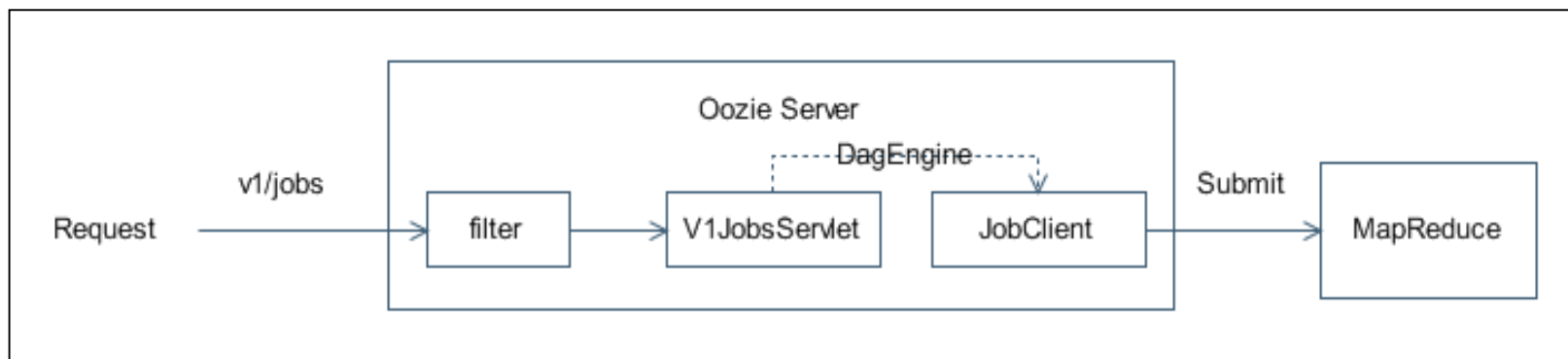
Oozie架构



Console	提供对 Oozie 流程的查看和监控功能。
Client	通过接口控制 workflow 流程：可以执行提交流程，启动流程，运行流程，终止流程，恢复流程等操作。 Hue 界面上的 Workflow 与 JobDesign 就属于 client 范畴
SDK	软件开发工具包 SDK （ SoftwareDevelopmentKit ）是被软件工程师用于为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件的开发工具的集合。
Database	PG 数据库，用于存储作业信息。
WebApp (Oozie)	webApp (Oozie) 即 Oozie server ，可以用内置的 Tomcat 容器，也可以用外部的，记录的信息比如日志等放在 PG 数据库中。
Tomcat	Tomcat 服务器是免费的开放源代码的 Web 应用服务器。
HaDoop组件	底层执行 Oozie 编排流程的各个组件，包括 MapReduce 、 Hive 、 Spark 等。

作业提交流程

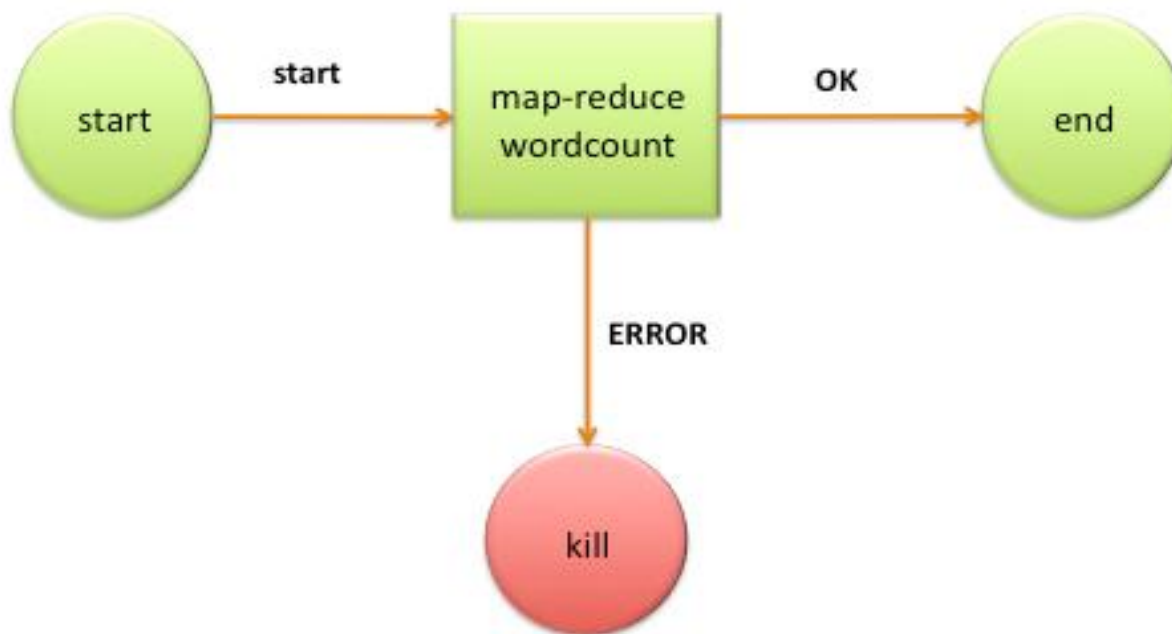
- Hue或其他组件通过Oozie提供的Rest接口来访问Oozie服务。
- Rest请求进入Oozie Server后, 经filter和servlet的识别, DagEngine将输入转化成MapReduce能识别的数据, 提交给MapReduce执行。



基本原理

原理

Oozie对工作流的编排, 是基于**workflow.xml**文件来完成的。用户预先将工作流执行规则定制于**workflow.xml**文件中, 并在**job.properties**配置上相关的参数, 然后由**Oozie Server**向**MR**提交一个**Job**来启动工作流。



Map-Reduce Action Node示例

```
<action name="myfirstHaDooJob">
  <map-reduce>
    <job-tracker>foo:9001</job-tracker>
    <name-node>bar:9000</name-node>
    <prepare>
      <delete
path="hdfs://foo:9000/usr/tucu/output-data"/>
    </prepare>
    <job-xml>/myfirstjob.xml</job-xml>
    <configuration>
      <property>
        <name>mapred.input.dir</name>
        <value>/usr/tucu/input-data</value>
      </property>
      <property>
        <name>mapred.output.dir</name>
        <value>/usr/tucu/input-data</value>
      </property>
      <property>
        <name>mapred.reduce.tasks</name>
        <value>${firstJobReducers}</value>
      </property>
    </configuration>
  </map-reduce>
</action>
```

接左侧代码：

```
<name>oozie.action.external.stats.write</name>
  <value>true</value>
</property>
</configuration>
</map-reduce>
<ok to="myNextAction"/>
<error to="errorCleanup"/>
</action>
```

- **map-reduce:** action类型
- **job-tracker :** Mapreduce jobtracker地址
- **name-node :** HDFS namenode地址
- **configuration:** action需要的参数
- **ok:** 标注当前action执行成功后跳转的下个节点
- **error :** 标注当前action执行失败后跳转的下一个节点

Java Action Node示例

```
<action name="myfirstjavajob">
  <java>
    <job-tracker>foo:9001</job-tracker>
    <name-node>bar:9000</name-node>
    <prepare>
      <delete path="${jobOutput}"/>
    </prepare>
    <configuration>
      <property>
        <name>mapred.queue.name</name>
        <value>default</value>
      </property>
    </configuration>
    <main-class>org.apache.oozie.MyFirstMainClass
    </main-class>
    <java-opts>-Dblah</java-opts>
    <arg>argument1</arg>
    <arg>argument2</arg>
  </java>
  <ok to="myotherjob"/>
  <error to="errorcleanup"/>
</action>
```

- **java** : action类型
- **job-tracker** : Mapreduce jobtracker地址
- **name-node** : HDFS namenode地址
- **configuration**: action需要的参数
- **main-class** : java代码的main函数入口
- **ok** : 标注当前action执行成功后跳转的下个节点
- **error** : 标注当前action执行失败后跳转的下个节点

Hive Action Node示例

```
<action name="myfirsthivejob">
  <hive xmlns="uri:oozie:hive-action:0.2">
    <job-tracker>foo:9001</job-tracker>
    <name-node>bar:9000</name-node>
    <prepare>
      <delete path="${jobOutput}"/>
    </prepare>
    <configuration>
      <property>

<name>mapred.queue.name</name>
      <value>default</value>
    </property>
    </configuration>
    <script>myscript.q</script>
    <param>InputDir=/home/tucu/input-
data </param>
```

接左侧代码：

```
<param>OutputDir=${jobOutput}</param>
  </hive>
  <ok to="myotherjob"/>
  <error to="errorcleanup"/>
</action>
```

- **hive** : action类型
- **job-tracker** : Mapreduce jobtracker地址
- **name-node** : HDFS namenode地址
- **configuration**: action需要的参数
- **script** : 执行的sql文件
- **ok** : 标注当前action执行成功后跳转的下一个节点
- **error** : 标注当前action执行失败后跳转的下一个节点

流程编排举例

```
<workflow-app xmlns="uri:oozie:workflow:0.2" name="wf">
  <start to="first"/>
  <action name="first">
    <map-reduce>
      ...
    </map-reduce>
    <ok to="second"/>
    <error to="fail"/>
  </action>
  <action name="second">
    <java>
      ...
    </java>
    <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>
```

左侧为流程编排的一个示例。

工作流是从**start**节点开始的, 由**start**进入**first**, 执行**map-reduce action**计算, 执行成功则进入**second**, 失败则进入**kill**节点, **fail**并打印错误信息。

Second执行成功后, 进入**end**节点, 表示流程结束。

Oozie 使用**HPDL**来构造工作流, 只有当上一个节点执行完成, 才会进入下一个节点。

所有“**Action Nodes**”均以有向无环图（**DAG Direct Acyclic Graph**）的模式部署, 不存在闭环流程。



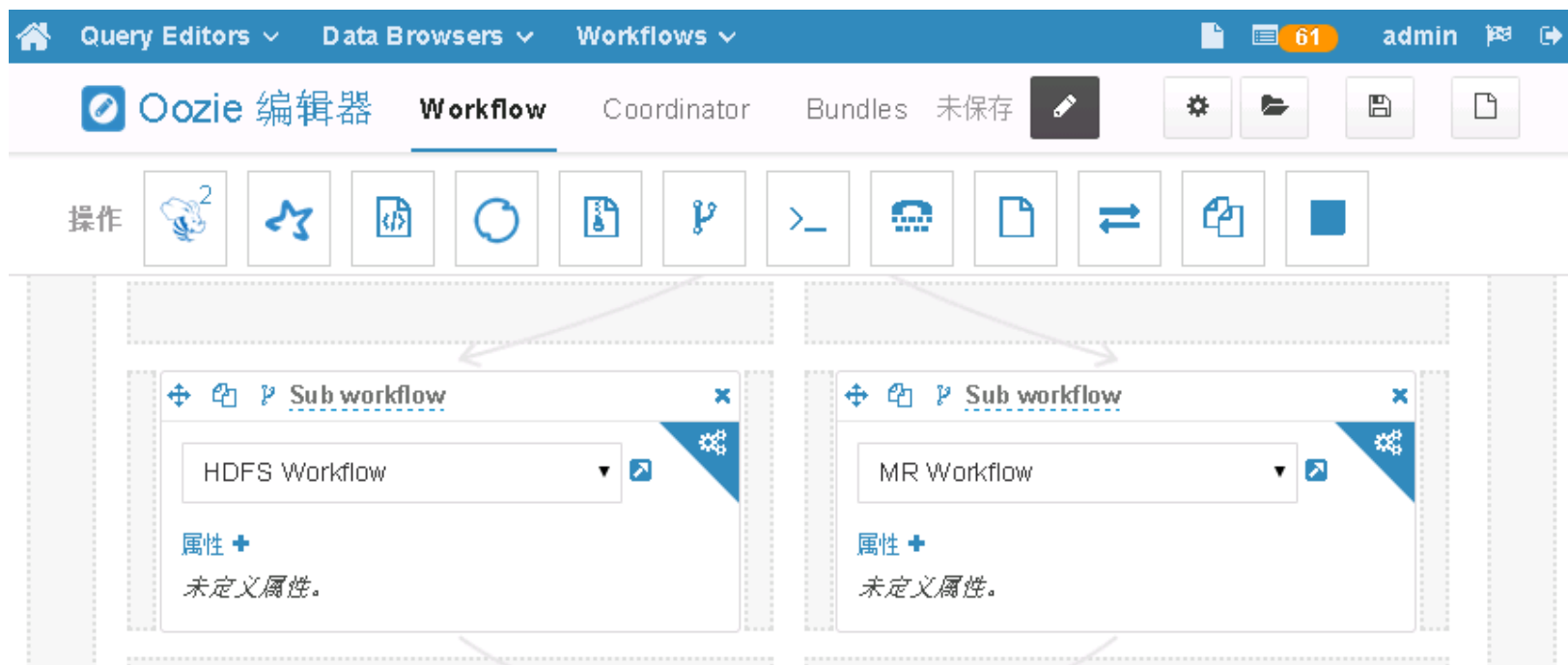
目录

1. Oozie基本介绍
2. Oozie架构与原理
3. 关键特性介绍
4. 应用举例

工作流编排

通过可视化界面对如下几种作业类型进行拖拽式工作流编排：

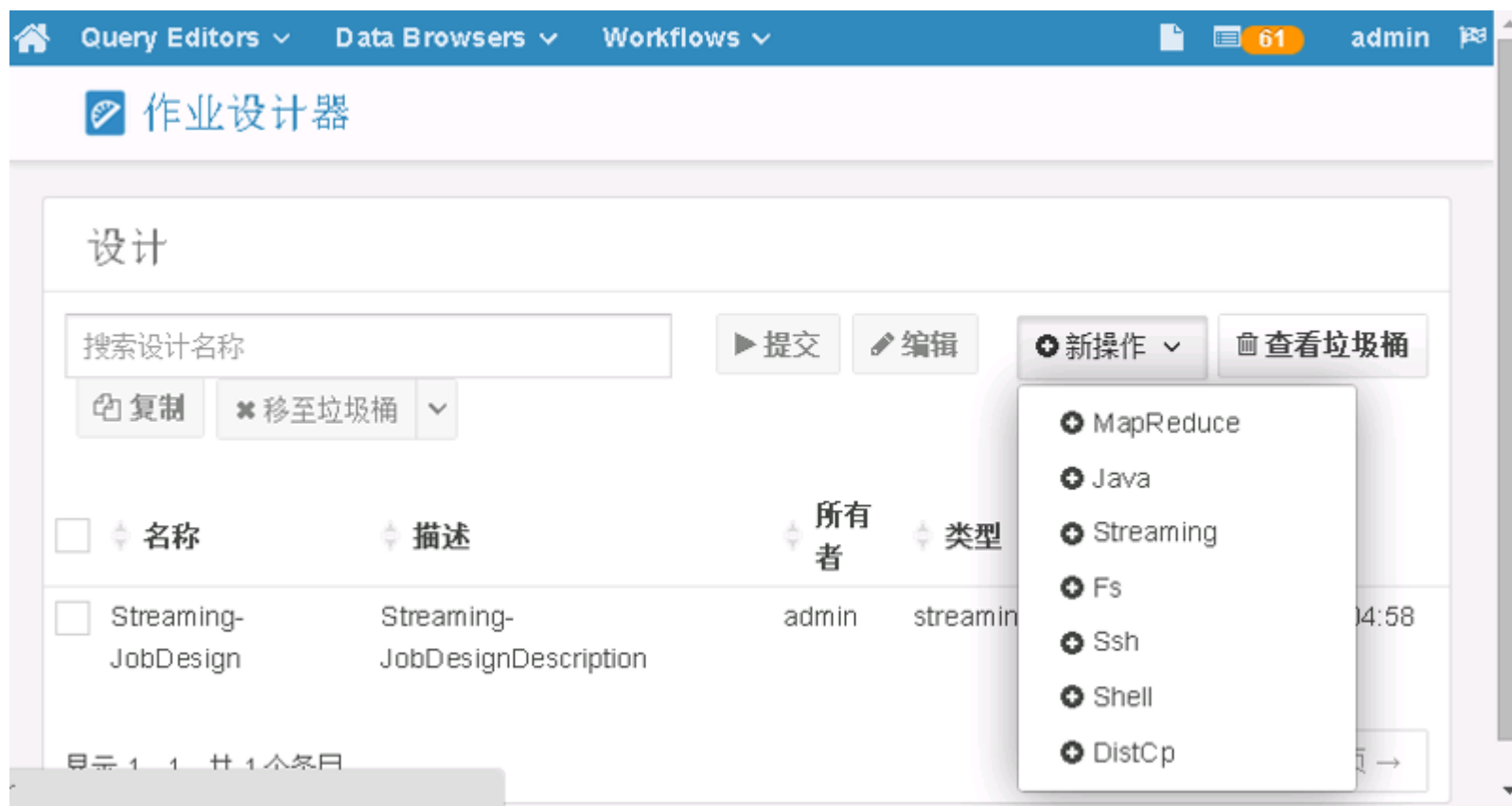
Hive、Spark、Java、Loader、MapReduce、Shell、Ssh、HDFS、Distcp、Streaming-MR



作业设计

通过可视化界面对如下几种作业类型进行作业设计：

MapReduce, Java, Streaming-MR, HDFS, SSH, Shell, Distcp



工作流调度

Coordinator调度工作流，可指定时间范围，调用频率；**Bundles**可以批量管理一组**Coordinator**。

The screenshot shows the HUE Oozie Coordinator configuration page. The top navigation bar includes 'Query Editors', 'Data Browsers', and 'Workflows'. The main toolbar has 'Oozie 编辑器', 'Workflow', 'Coordinator' (selected), and 'Bundles'. The page title is 'HDFS Coordinator'. Below the title, there is a section '要计划哪个 Workflow?' with a dropdown menu showing 'HDFS Workflow'. The '频率?' section has a text input for frequency, a '高级语法' checkbox, and a date range selector from '2016-04-04 16:39Z' to '2016-04-12 16:39Z'.

HUE Query Editors Data Browsers Workflows 61 admin

Oozie 编辑器 Workflow Coordinator Bundles 未保存

HDFS Coordinator

添加说明...

要计划哪个 Workflow?

HDFS Workflow

频率? 仅限 UTC 时间。(例如, 如果您需要 10pm PST (UTC+8), 则将其设置为比次日 6 am 晚 8 个小时。)

每 小时 在 30 超过小时的分钟数

隐藏

高级语法

自 2016-04-04 16:39Z

至 2016-04-12 16:39Z

The screenshot shows the HUE Oozie Bundles configuration page. The top navigation bar is the same as the previous screenshot. The main toolbar has 'Oozie 编辑器', 'Workflow', 'Coordinator', and 'Bundles' (selected). The page title is 'My Bundle'. Below the title, there is a section '要包含哪些计划?' with a '+ 添加 Coordinator' button. The 'HDFS Coordinator' entry is expanded, showing 'start_date' (2016-04-04T16:39Z) and 'end_date' (2016-04-12T16:39Z). The 'Shell Coordinator' entry is also expanded, showing 'start_date' (2016-04-06T12:07Z) and 'end_date' (2016-04-13T12:07Z).

HUE Query Editors Data Browsers Workflows 未保存

Oozie 编辑器 Workflow Coordinator Bundles

My Bundle

添加说明...

要包含哪些计划?

+ 添加 Coordinator

HDFS Coordinator

start_date 2016-04-04T16:39Z

end_date 2016-04-12T16:39Z

+ 添加参数

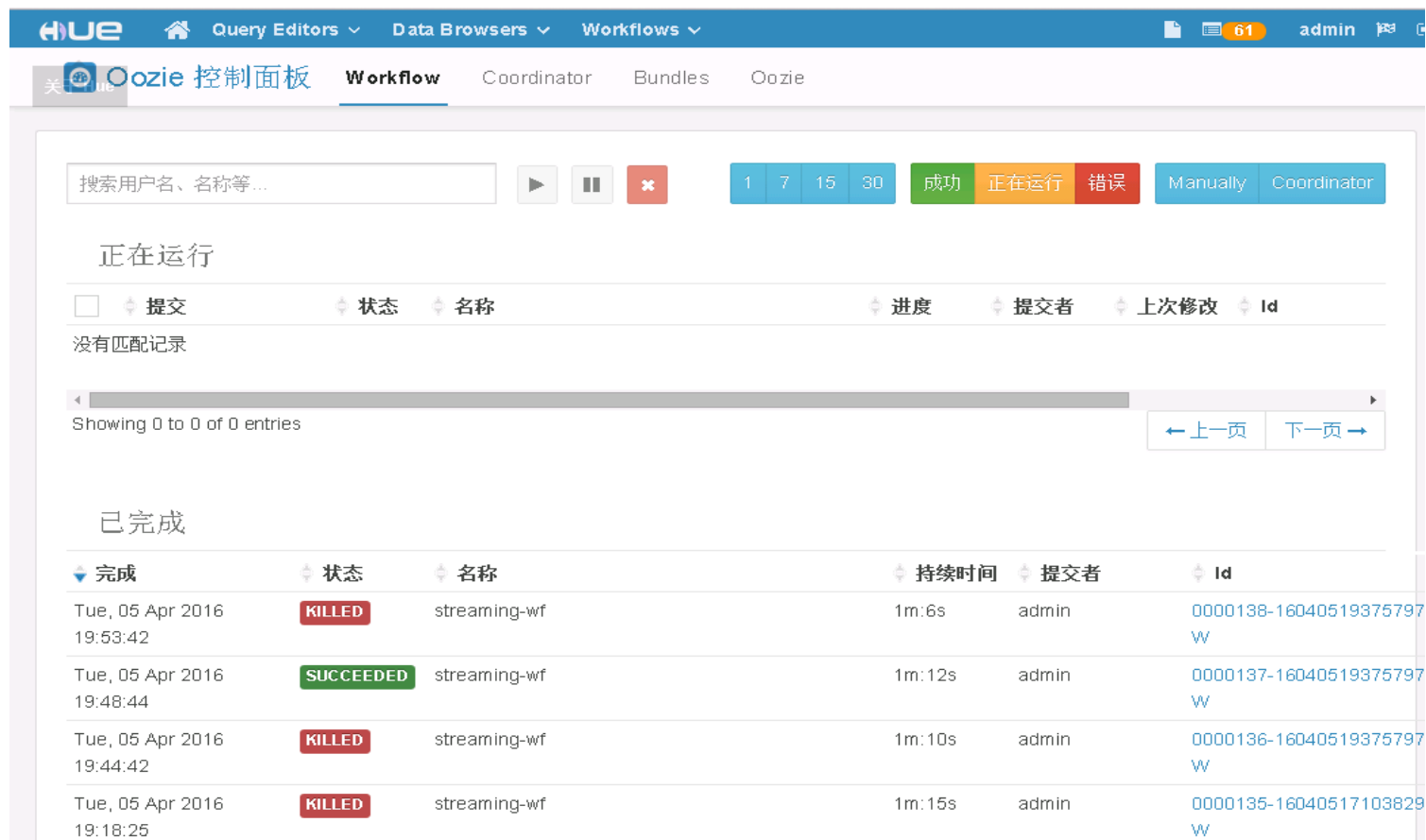
Shell Coordinator

start_date 2016-04-06T12:07Z

end_date 2016-04-13T12:07Z

作业监控

监控作业的执行状态，查看运行日志，作业配置等



搜索用户名、名称等...

1 7 15 30 成功 正在运行 错误 Manually Coordinator

正在运行

提交	状态	名称	进度	提交者	上次修改	Id
没有匹配记录						

Showing 0 to 0 of 0 entries

← 上一页 下一页 →

已完成

完成	状态	名称	持续时间	提交者	Id
Tue, 05 Apr 2016 19:53:42	KILLED	streaming-wf	1m:6s	admin	0000138-16040519375797W
Tue, 05 Apr 2016 19:48:44	SUCCEEDED	streaming-wf	1m:12s	admin	0000137-16040519375797W
Tue, 05 Apr 2016 19:44:42	KILLED	streaming-wf	1m:10s	admin	0000136-16040519375797W
Tue, 05 Apr 2016 19:18:25	KILLED	streaming-wf	1m:15s	admin	0000135-16040517103829W

项目分类与共享

按照不同的项目对保存的工作流或作业设计进行分类；并且可以共享给其他用户

🏠 我的文档

操作

➕ 新文档

🗑 trash 0

我的项目 + -

📁 admin_test 7















📁 default 21

与我共享

👤 oozie_admin

📁 test 4

Search for name, description, etc...

名称	描述	上次修改	项目	共享
 Hive2 Workflow	None	04/05/16 16:27:27	default admin_test	
 SSH Workflow	None	04/05/16 16:27:27	default admin_test	
 Sqoop Workflow	None	04/05/16 16:27:27	default admin_test	
 Hive2 Workflow-NewQueue	None	04/05/16 17:05:46	default admin_test	
 Spark Workflow-NewQueue	None	04/05/16 16:27:27	default admin_test	
 Shell Workflow-NewQueue	None	04/05/16 16:27:27	default admin_test	
 DistcpWorkflow	None	04/05/16 16:27:27	default admin_test	

Oozie其他特性说明

- 访问方式多样化
 - 提供可视化作业管理的同时**Oozie**也提供**Rest**接口与**Shell**接口，方便与第三方对接。
- 查看**Oozie**配置
 - 提供可视化界面查看**Oozie**配置。
- 操作方式多样化
 - 对作业，工作流、定时调度，批量调度的启动，暂停，停止，恢复，查看等操作，提高易用性与灵活性。

Oozie增强特性

- 安全增强
 - 支持**Oozie**权限管理，提供管理员与普通用户两种角色。
 - 支持单点登录登出，**HTTPS**方式访问以及审计日志
- 可靠性增强
 - 支持**Oozie HA**
- 可维护性增强
 - 支持告警，日志压缩，健康检查。



目录

1. Oozie基本介绍
2. Oozie架构与原理
3. 关键特性介绍
4. 应用举例

命令行提交作业

- 需要先下载**FusionInsight**的客户端，并且先将 workflow 需要的文件（例如 **Workflow.xml**, **job.properties**, 以及依赖的 **jar** 包等）上传至 **HDFS** 指定目录，同时每次本地配置项发生更新时，需要同步更新 **HDFS** 上的文件。
- 命令行示例：
 - `hdfs dfs -mkdir /user/admin` //在HDFS上创建用户工作目录
 - `hdfs dfs -put -f /opt/client/Oozie/oozie-client-4.2.0/examples /user/admin/` //上传 workflow 需要的配置文件与 **jar** 包到工作目录
 - `hdfs dfs -put -f /opt/client/Oozie/oozie-client-4.2.0/share/lib /user/oozie/share/` //上传公共 **jar** 包
 - `oozie job -oozie https://192-18-0-118:21003/oozie -config job.properties -run` //提交 workflow

Hue (UI) 提交作业

在**Hue**上的**Workflow**编辑器中编辑工作流，点击提交按钮即可

应用举例

例如，有这样一个需求，某个业务系统每天产生**20G**的原始数据，我们每天都要对其进行处理，处理步骤如下：

- 先将原始数据同步到**HDFS**上（**Loader Action**）。
- 借助**MapReduce**对原始数据进行转换，生成的数据以分区表的形式存储到多张**Hive**表中（**MapReduce Action**）。
- 对**Hive**中多个表的数据进行**Join**处理，得到一个明细数据**Hive**大表(**Hive Action**)。
- 将明细数据进行复杂的统计分析，得到排序后的报表信息（**Hive Action**）。
- 将统计分析得到的结果数据同步到业务系统中，供业务调用（ **Loader Action** ）。

上述过程可以通过工作流系统来编排任务，最终生成一个工作流实例，然后每天定时启动这个实例即可。



思考题

1. **Oozie**的作用是什么？
2. 请描述**Oozie**提交作业的流程？
3. **Oozie**有哪些增强特性？



练习题

如下哪项不属是**Oozie**的引擎？（ ）

A.Workflow

B.Coordinator

C.Action

D.Bundles

练习题

下列些描述是正确的

- A.运行Oozie作业前需要先将依赖文件上传到HDFS**
- B.Oozie的工作流中包含控制流节点和动作节点**
- C.Oozie是在Tomcat中运行的Java Web应用程序**
- D.Oozie支持时间(频率)和数据有效性触发 workflow**



总结

- 本章首先介绍了**Oozie**的概念，及其使用场景；然后讲解**Oozie**的架构原理，最后讲解了**Oozie**关键特性及增强特性，并使用例子展示了如何提交**Oozie**工作流。



更多信息

- 华为官方网站
 - 企业业务: <http://enterprise.huawei.com/cn/>
 - 技术支持: <http://support.huawei.com/enterprise/>
 - 在线学习: <http://learning.huawei.com/cn/>
- Oozie开源社区参考
 - <https://oozie.apache.org/>
 - <http://oozie.apache.org/docs/4.2.0/index.html>
 - <http://blog.cloudera.com/blog/category/oozie/>

Thank you

www.huawei.com