Apache Oozie Workflow Scheduler for Hadoop

Overview

Apache Oozie is a scheduler system to run and manage Hadoop jobs in a distributed environment. It allows to combine multiple complex jobs to be run in a sequential order to achieve a bigger task. Within a sequence of task, two or more jobs can also be programmed to run parallel to each other.

One of the main advantages of Oozie is that it is tightly integrated with Hadoop stack supporting various Hadoop jobs like Hive, Pig, Sqoop as well as system-specific jobs like Java and Shell.

Oozie is an Open Source Java Web-Application available under Apache license 2.0. It is responsible for triggering the workflow actions, which in turn uses the Hadoop execution engine to actually execute the task. Hence, Oozie is able to leverage the existing Hadoop machinery for load balancing, fail-over, etc.

Oozie detects completion of tasks through callback and polling. When Oozie starts a task, it provides a unique callback HTTP URL to the task, and notifies that URL when it is complete. If the task fails to invoke the callback URL, Oozie can poll the task for completion.

Following three types of jobs are common in Oozie −

- **Oozie Workflow Jobs** − These are represented as Directed Acyclic Graphs (DAGs) to specify a sequence of actions to be executed.

- **Oozie Coordinator Jobs** − These consist of workflow jobs triggered by time and data availability.

- **Oozie Bundle** − These can be referred to as a package of multiple coordinator and workflow jobs.

How does OOZIE work?

Oozie runs as a service in the cluster and clients submit workflow definitions for immediate or later processing.

Oozie workflow consists of **action nodes** and **control-flow nodes**.

An action node represents a workflow task, e.g., moving files into HDFS, running a MapReduce, Pig or Hive jobs, importing data using Sqoop or running a shell script of a program written in Java.

A control-flow node controls the workflow execution between actions by allowing constructs like conditional logic wherein different branches may be followed depending on the result of earlier action node.

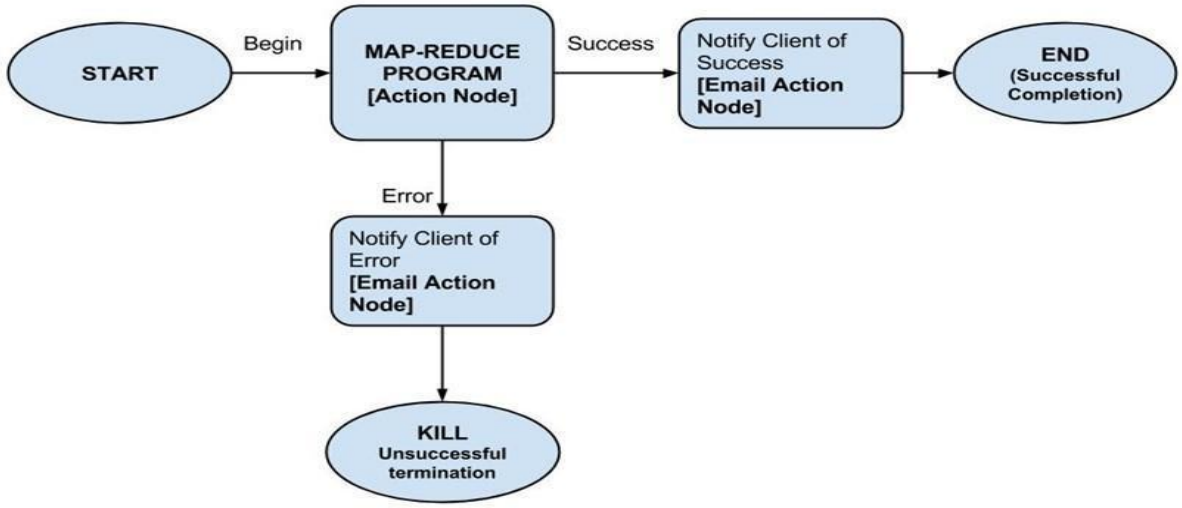
Start Node, End Node and Error Node fall under this category of nodes.

Start Node, designates start of the workflow job.

End Node, signals end of the job.

Error Node, designates an occurrence of error and corresponding error message to be printed.

At the end of execution of workflow, HTTP callback is used by Oozie to update client with the workflow status. Entry-to or exit-from an action node may also trigger callback.



Why use Oozie?

Main purpose of using Oozie is to manage different type of jobs being processed in Hadoop system.

Dependencies between jobs are specified by user in the form of Directed Acyclic Graphs. Oozie consumes this information and takes care of their execution in correct order as specified in a workflow. That way user's time to manage complete workflow is saved. In addition, Oozie has a provision to specify frequency of execution of a particular job.

Example Workflow

Consider we want to load a data from external hive table.

Step 1 − DDL for Hive table user_rating_table_creation.hive

use movie_lens_data;

CREATE TABLE IF NOT EXISTS user_ratings (user_id bigint, movie_id bigint, max_rating float, avg_rating float, min_rating float) COMMENT 'Max min avg rating of any movie by the user.' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE;

Step 2 − DDL for Hive table user_ratings_data_truncate.hive

use movie_lens_data;
truncate table user_ratings;

Step 3 − Hive script to insert data (user_ratings_data_population.hive)

use movie_lens_data;

insert overwrite table user_ratings select user_id,movie_id,max(rating) as max,avg(rating) as avg,min(rating) as min from ratings where movie_id=20 group by user_id, movie_id;

Step 4 − Create a workflow to execute all the above three steps- user_ratings_wf.xml

```
<workflow-app xmlns="uri:oozie:workflow:0.2.5" name="Movie-Rating-Max-Min-Avg">
  <credentials>
        <credential name='hive_auth' type='hcat'>
              <property>
                    <name>hcat.metastore.uri</name>
                    <value>${hiveMetaStoreUri}</value>
              </property>
              <property>
                    <name>hcat.metastore.principal</name>
                    <value>${hiveMetaStorePrincipal}</value>
              </property>
        </credential>
  </credentials>
  <start to="table_creation" />
  <action name="table_creation" cred='hive_auth'>
    <hive xmlns="uri:oozie:hive-action:0.2">
        <job-tracker>${jobTracker}</job-tracker>
        <name-node>${nameNode}</name-node>
```

```
        <script>${table_creation_script_file}</script>
      </hive>
    <ok to="data_cleanup" />
    <error to="kill_job" />
  </action>

  <action name="data_cleanup" cred='hive_auth'>
    <hive xmlns="uri:oozie:hive-action:0.2">
        <job-tracker>${jobTracker}</job-tracker>
        <name-node>${nameNode}</name-node>
        <script>${table_cleanup_script_file}</script>
      </hive>
    <ok to="data_population" />
    <error to="kill_job" />
  </action>

  <action name="data_population" cred='hive_auth'>
    <hive xmlns="uri:oozie:hive-action:0.2">
        <job-tracker>${jobTracker}</job-tracker>
        <name-node>${nameNode}</name-node>
        <script>${data_population_script_file}</script>
      </hive>
    <ok to="end" />
    <error to="kill_job" />
  </action>
  <kill name="kill_job">
    <message>Job failed</message>
  </kill>
  <end name = "end" />
</workflow-app>
```

Explanation of the Above Example

Action Nodes in the above example defines the type of job that the node will run. Hive node inside the action node defines that the action is of type hive. This could also have been a pig, java, shell action, etc. as per the job you want to run.

Each type of action can have its own type of tags. In the above job we are defining the job tracker to us, name node details, script to use and the param entity. The Script tag defines the script we will be running for that hive action. The Param tag defines the values which we will pass into the hive script.

Running the Workflow

The workflow and hive scripts should be placed in HDFS path before running the workflow.

Oozie job --oozie http://impetus-i0161.impetus.co.in:11000/oozie -Doozie.wf.application.path=hdfs://EETeamJ1/user/oozie/oozie-jobs/job-3/user_ratings_wf.xml -config /home/IMPETUS/rupika.mahajan/ActualWF/UserRating-WF/job.properties -run

Apache Oozie - Property File

Oozie workflows can be parameterized. The parameters come from a configuration file called as property file. We can run multiple jobs using same workflow by using multiple .property files (one property for each job).

Suppose we want to change the jobtracker url or change the script name or value of a param.

We can specify a config file (.property) and pass it while running the workflow.

Variables like **${nameNode}** can be passed within the workflow definition. The value of this variable will be replaced at the run time with the value defined in the '.properties' file.0

Following is an example of a property file we have used in our workflow example.

```
oozie.action.sharelib.for.pig=hive,pig,hcatalog
oozie.action.sharelib.for.hive=hive,hcatalog,sqoop
oozie.use.system.libpath=true
oozie.libpath=hdfs://EETeamJ1/user/oozie/share/lib
jobTracker=impetus-i0161.impetus.co.in:8050
nameNode=hdfs://EETeamJ1
hiveMetaStoreUri=thrift://impetus-i0161.impetus.co.in:9083
hiveMetaStorePrincipal=hive/_HOST@IMPETUS.CO.IN
table_creation_script_file=/user/oozie/oozie-jobs/job-3/user_rating_table_creation.hive
table_cleanup_script_file=/user/oozie/oozie-jobs/job-3/user_ratings_data_truncate.hive
data_population_script_file=/user/oozie/oozie-jobs/job-/user_ratings_data_population.hive
```

At run time, all the parameters in ${} will be replaced by its corresponding value in the .properties file.

WorkFlow execution screenshots

```
oozie@impetus-i0161:~$ oozie job --oozie http://impetus-i0161.impetus.co.in:11000/oozie -Doozie.wf.application.path=hdfs://EETeamJ1/user/oozie/oozie-jobs/job-3/user_ratings_wf.xml -config /home/IMPETUS/rupika.mahajan/ActualWF/UserRating-WF/job.properties -run
job: 0000004-160721220716093-oozie-oozi-W
oozie@impetus-i0161:~$
```