Session​ ​ 10:​ ​ Oozie​ ​ and​ ​ Sqoop

Assignment​ ​ 1​ ​ Question

*Session*​ ​ *10:*​ ​ *Assignment*​ ​ *1*

**Table**​ ​ **of**​ ​ **Contents**

1. Introduction

2. Problem​ ​ Statement

3. Output

**1. Introduction**

This​ ​ assignment​ ​ will​ ​ help​ ​ you​ ​ to​ ​ consolidate​ ​ the​ ​ concepts​ ​ learnt​ ​ in​ ​ the​ ​ session​ .

**2. Problem**​ ​ **Statement**

Explain​ ​ in​ ​ Brief:

● The​ ​ workflow​ ​ of​ ​ Oozie​ ​ and​ ​ its​ ​ Benefits

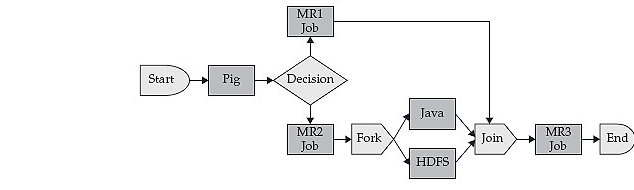
In the modern IT world, Hadoop is an open source framework that is highly popular for its effectiveness and ease-of-use in handling tasks related to big data analysis. As the big data analysis tasks require multiple jobs to be created in the process of analysis, it becomes essential to use an efficient process of job handling. This is where OOZIE comes in handy. Similar to the Hadoop framework it usually works on, OOZIE is also an open source project with the aim of making the workflow simpler and coordination between multiple jobs convenient. By using OOZIE project, users of Hadoop can define different actions/jobs and the inter-dependency between the jobs. After this, OOZIE takes over the control of the job scheduling process and starts executing different actions when the respective necessary dependencies are met.

DAG (Directed Acyclical Graph) is an in-built process in the OOZIE structure that is used by the program to define the actions. In the DAG process of OOZIE, the Acyclical term refers to the graph having no loops i.e. the action graph has a separate starting point as well as an end point. The DAG process is built using both the action nodes and the various defined dependencies, each one having a starting point and pointing towards the end without ever going back to starting point.

OOZIE is quite flexible in manner of the different type of tasks it can handle, as the action node in the program can be a job to ReduceMap, a Java app, a file system job, or even a Pig application. The representation of the flow control in the develop DAG graph is done by the use of node elements, which function on the logic taken from the input generated by the preceding job in the same DAG graph. Flow control nodes in the OoZIE programs can be join nodes, forks nodes, and decisions nodes. The figure shown below is an example of workflow in the OOZIE application.

The workflow process in OOZIE is a collection of different action types (including Hadoop map jobs, pig jobs), which are arranged based on a DAG (Direct Acyclic Graph), which defines the sequence for the different actions to execute. The GAD graph is defined in a specific XML language for process definition known as hDPL. hDPL is a very compact language and makes use of minimal amount of action nodes and flow control nodes. In this language, the control nodes are used to specify the execution flow for different actions (marking start, end and fail nodes) and also allows the instruction set to define the execution path by help of nodes like- decision node, fork node, and join node.

Action nodes are the instruction that function as necessary triggers that initiate the execution of an action when the necessary conditions are met. Using OOZIE application, the following types of actions can be controlled and mapped- Hadoop file system actions, Hadoop map/reduce actions, Pig map action, and Java sub workflow actions.

****

**Oozie is a server based Workflow Engine specialized in running workflow jobs with actions that run Hadoop Map/Reduce and Pig jobs.**

**Oozie is a Java Web-Application that runs in a Java servlet-container.**

**For the purposes of Oozie, a workflow is a collection of actions (i.e. Hadoop Map/Reduce jobs, Pig jobs) arranged in a control dependency DAG (Direct Acyclic Graph). "control dependency" from one action to another means that the second action can't run until the first action has completed.**

**Oozie workflows definitions are written in hPDL (a XML Process Definition Language similar to JBOSS JBPM jPDL).**

**Oozie workflow actions start jobs in remote systems (i.e. Hadoop, Pig). Upon action completion, the remote systems callback Oozie to notify the action completion, at this point Oozie proceeds to the next action in the workflow.**

**Oozie workflows contain control flow nodes and action nodes.**

**Control flow nodes define the beginning and the end of a workflow ( start , end and fail nodes) and provide a mechanism to control the workflow execution path ( decision , fork and join nodes).**

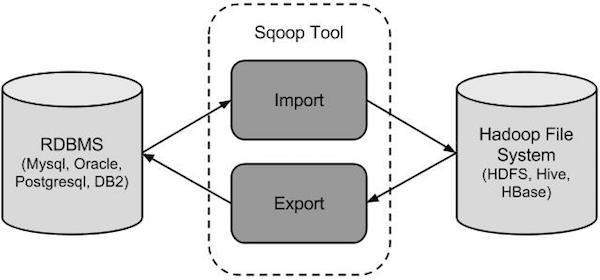
**Action nodes are the mechanism by which a workflow triggers the execution of a computation/processing task. Oozie provides support for different types of actions: Hadoop map-reduce, Hadoop file system, Pig, SSH, HTTP, eMail and Oozie sub-workflow. Oozie can be extended to support additional type of actions.**

**Oozie workflows can be parameterized (using variables like ${inputDir} within the workflow definition). When submitting a workflow job values for the parameters must be provided. If properly parameterized (i.e. using different output directories) several identical workflow jobs can concurrently.**

● The​ ​ workflow​ ​ of​ ​ Sqoop​ ​ and​ ​ its​ ​ Benefits

Sqoop is a tool designed to transfer data between Hadoop and relational database servers. It is used to import data from relational databases such as MySQL, Oracle to Hadoop HDFS, and export from Hadoop file system to relational databases. This is a brief tutorial that explains how to make use of Sqoop in Hadoop ecosystem.

The following image describes the workflow of Sqoop.



Sqoop Import

The import tool imports individual tables from RDBMS to HDFS. Each row in a table is treated as a record in HDFS. All records are stored as text data in text files or as binary data in Avro and Sequence files.

Sqoop Export

The export tool exports a set of files from HDFS back to an RDBMS. The files given as input to Sqoop contain records, which are called as rows in table. Those are read and parsed into a set of records and delimited with user-specified delimiter.

Benefits**:**

Allows the transfer of data with a variety of structured data stores like Postgres, Oracle, Teradata, and so on.

Since the data is transferred and stored in Hadoop, Sqoop allows us to offload certain processing done in the ETL (Extract, Load and Transform) process into low-cost, fast, and effective Hadoop processes.

Sqoop can execute the data transfer in parallel, so execution can be quick and more cost effective.

Helps to integrate with sequential data from the mainframe.