Explain in brief:

● The complete structure and the working of “Oozie Workflow scheduler “

Workflow nodes are classified in control flow nodes and action nodes:

* **Control flow nodes:** nodes that control the start and end of the workflow and workflow job execution path.
* **Action nodes:** nodes that trigger the execution of a computation/processing task.

Node names and transitions must be conform to the following pattern =[a-zA-Z][\-\_a-zA-Z0-0]\*=, of up to 20 characters long.

## **What is Oozie?**

Apache Oozie is an open source project based on Java™ technology that simplifies the process of creating workflows and managing coordination among jobs. In principle, Oozie offers the ability to combine multiple jobs sequentially into one logical unit of work. One advantage of the Oozie framework is that it is fully integrated with the Apache Hadoop stack and supports Hadoop jobs for Apache MapReduce, Pig, Hive, and Sqoop. In addition, it can be used to schedule jobs specific to a system, such as Java programs. Therefore, using Oozie, Hadoop administrators are able to build complex data transformations that can combine the processing of different individual tasks and even sub-workflows. This ability allows for greater control over complex jobs and makes it easier to repeat those jobs at predetermined periods.

In practice, there are different types of Oozie jobs:

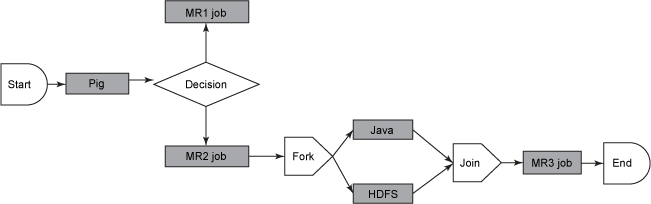
* Oozie Workflow jobs — Represented as directed acyclical graphs to specify a sequence of actions to be executed.
* Oozie Coordinator jobs — Represent Oozie workflow jobs triggered by time and data availability.
* Oozie Bundle— Facilitates packaging multiple coordinator and workflow jobs, and makes it easier to manage the life cycle of those jobs.

## **How does Oozie work?**

An Oozie workflow is a collection of actions arranged in a directed acyclic graph (DAG). This graph can contain two types of nodes: control nodes and action nodes. Control nodes, which are used to define job chronology, provide the rules for beginning and ending a workflow and control the workflow execution path with possible decision points known as fork and join nodes. Action nodes are used to trigger the execution of tasks. In particular, an action node can be a MapReduce job, a Pig application, a file system task, or a Java application. (The shell and ssh actions have been deprecated).

Oozie is a native Hadoop stack integration that supports all types of Hadoop jobs and is integrated with the Hadoop stack. In particular, Oozie is responsible for triggering the workflow actions, while the actual execution of the tasks is done using Hadoop MapReduce. Therefore, Oozie becomes able to leverage 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. Figure 1 illustrates a sample Oozie workflow that combines six action nodes (Pig scrip, MapReduce jobs, Java code, and HDFS task) and five control nodes (Start, Decision control, Fork, Join, and End). Oozie workflows can be also parameterized. When submitting a workflow job, values for the parameters must be provided. If the appropriate parameters are used, several identical workflow jobs can occur concurrently.

##### Figure 1. Sample Oozie workflow



In practice, it is sometimes necessary to run Oozie workflows on regular time intervals, but in coordination with other conditions, such as the availability of specific data or the completion of any other events or tasks. In these situations, Oozie Coordinator jobs allow the user to model workflow execution triggers in the form of the data, time, or event predicates where the workflow job is started after those predicates get satisfied. The Oozie Coordinator can also manage multiple workflows that are dependent on the outcome of subsequent workflows. The outputs of subsequent workflows become the input to the next workflow. This chain is called a data application pipeline.

Oozie workflow definition language is XML-based and it is called the Hadoop Process Definition Language.Oozie comes with a command-line program for submitting jobs. This command-line program interacts with the Oozie server using REST. To submit or run a job using the Oozie client, give Oozie the full path to your workflow.xml file in HDFS as a parameter to the client. Oozie does not have a notion of global properties. All properties, including the jobtracker and the namenode, must be submitted as part of every job run. Oozie uses an RDBMS for storing state.

It is necessary to keep workflow.xml (a workflow definition file) in the top level directory (parent directory with workflow name). Lib directory contains Jar files containing MapReduce classes. Workflow application conforming to this layout can be built with any build tool e.g., Ant or Maven.

Such a build need to be copied to HDFS using command, for example -

% hadoop fs -put hadoop-examples/target/ name of workflow **Steps for Running an Oozie workflow job**

In this section we will see how to run a workflow job. To run this, we will use Oozie command-line tool (a client program which communicates with the Oozie server).

**1. Export OOZIE\_URL environment variable which tells the oozie command which Oozie server to use (here we’re using one running locally):**

% export OOZIE\_URL="http://localhost:11000/oozie"

**2. Run workflow job using-**

% oozie job -config ch05/src/main/resources/max-temp-workflow.properties -run

The -config option refers to a local[Java](http://www.guru99.com/java-tutorial.html)properties file containing definitions for the parameters in the workflow XML file, as well as oozie.wf.application.path, which tells Oozie the location of the workflow application in HDFS.

Example contents of the properties file:

nameNode=hdfs://localhost:8020 jobTracker=localhost:8021 oozie.wf.application.path=${nameNode}/user/${user.name}/

**3. Get the status of workflow job-**

Status of workflow job can be seen using subcommand 'job' with '-info' option and specifying job id after '-info'.

e.g., % oozie job -info

Output shows status which is one of: RUNNING, KILLED, or SUCCEEDED.

**4. Results of successful workflow execution can be seen using Hadoop command like-**

% hadoop fs -cat

### **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.

### **FEATURES OF OOZIE**

* Oozie has client API and command line interface which can be used to launch, control and monitor job from Java application.
* Using its Web Service APIs one can control jobs from anywhere.
* Oozie has provision to execute jobs which are scheduled to run periodically.
* Oozie has provision to send email notifications upon completion of jobs.