**ASSIGNMENT – 34.1**

**Problem Statement:**

1. **Explain in brief:**

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

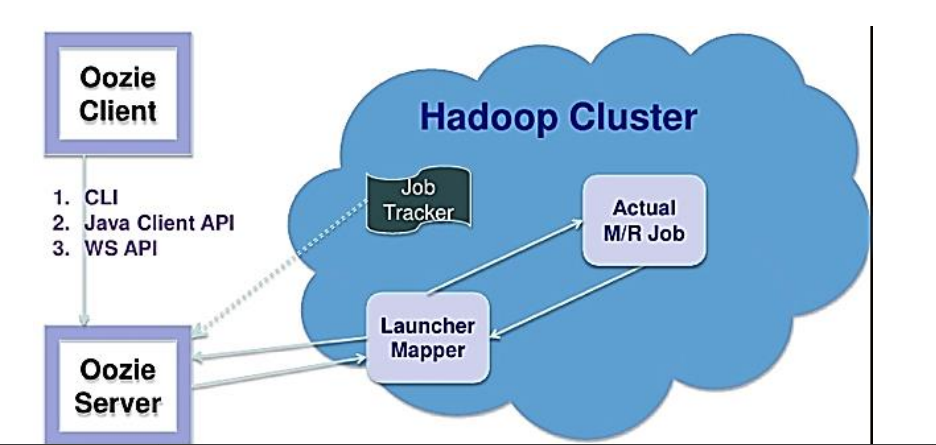
Apache Oozie is a real time scheduler and workflow engine that blends well with large production environments

• Oozie can run workflow jobs with MapReduce and Pig action nodes

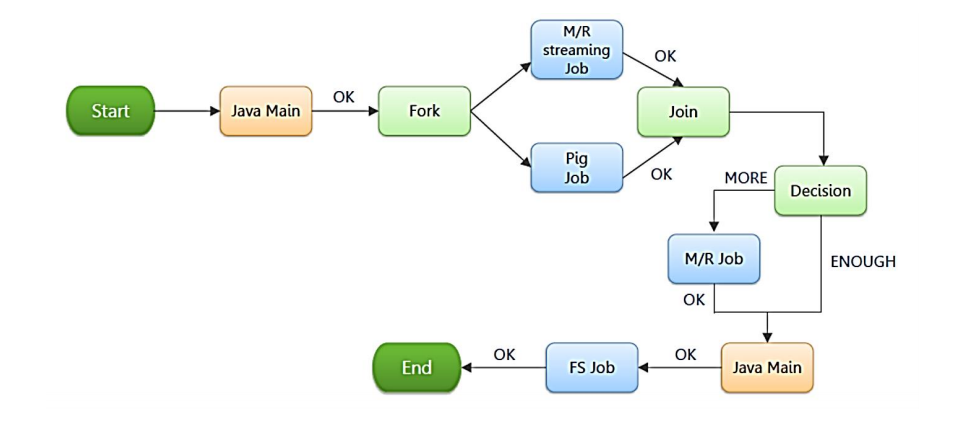
• Oozie Workflow jobs are Directed Acyclical Graphs (DAGs) of actions.

• It is a server based workflow engine

**OOZIE ARCHITECTURE :**



**OOZIE WORKFLOW :**



**Oozie Workflow Nodes :**

**Control Flow –**

• Start/end/kill

• Decision

• Fork/join Actions

• Map-reduce

• Pig

• Hdfs

• Sub-workflow

• Java-run custom java code

• To run oozie workflows, two files are needed.

1. workflow.xml (stored in HDFS)

• It contains the structure of workflow.

1. job.properties (stored in local)

• It contains the configuration properties.

• The Oozie server is designed to work with either MR V1 or YARN.

Please note that it cannot work with both simultaneously

• It can be configured with CATALINA\_BASE variable in /etc/oozie/conf/oozie-env.sh Hadoop 1

• CATALINA\_BASE = /usr/lib/oozie/oozie-server-0.20 Hadoop 2

• CATALINA\_BASE=/usr/lib/oozie/oozie-server

**OOZIE COORDINATOR :**

Oozie Coordinator is a collection of predicates (conditional statements based on timefrequency and data availability) and actions (i.e. Hadoop Map/Reduce jobs, Hadoop file system, Hadoop Streaming, Pig, Java and Oozie sub-workflow). • Actions are recurrent workflow jobs invoked each time predicate returns true.

Coordinator applications allow users to schedule complex workflows, including workflows that are scheduled regularly.

Oozie Coordinator models the workflow execution triggers in the form of time, data or event predicates. The workflow job mentioned inside the Coordinator is started only after the given conditions are 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**.**

**Schedule a Job Based on Time**

<coordinator-app name=*"tf-idf" frequency="1440"*

start=*"2013-01-01T00:00Z" end="2013-12-31T00:00Z" timezone="UTC"*

xmlns=*"uri:oozie:coordinator:0.1">*

<action>

<workflow>

<app-path> hdfs://node:8020/home/train/tfidf/workflow

</app-path>

</workflow>

</action>

</coordinator-app>

**Schedule a Job Based on Data Availability**

<coordinator-app name=*"file\_check" frequency="1440"*

start=*"2012-01-01T00:00Z" end="2015-12-31T00:00Z" timezone="UTC"*

xmlns=*"uri:oozie:coordinator:0.1">*

<datasets>

<dataset name=*"input1">*

<uri-template>

hdfs://node:8020/job/result/

</uri-template>

</dataset>

</datasets>

<action>

<workflow>

<app-path>hdfs://node:8020/myapp/</app-path>

</workflow>

</action>

</coordinator-app>