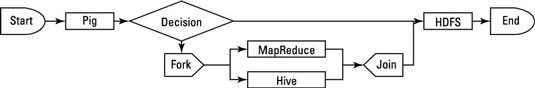
Explain in brief:

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

**Structure:**



Oozie workflows are, at their core, directed graphs, where you can define actions (Hadoop applications) and data flow, but with no looping — meaning you can’t define a structure where you’d run a specific operation over and over until some condition is met (a for loop, for example).

Oozie workflows are quite flexible in that you can define condition-based decisions and forked paths for parallel execution. You can also execute a wide range of actions.

In this figure, you see a workflow showing the basic capabilities of Oozie workflows. First, a Pig script is run, and is immediately followed by a decision tree. Depending on the state of the output, the control flow can either go directly to an HDFS (Hadoop Distributed File System) file operation (for example, a copyToLocal operation) or to a fork action.

If the control flow passes to the fork action, two jobs are run concurrently: a MapReduce job, and a Hive query. The control flow then goes to the HDFS operation once both the MapReduce job and Hive query are finished running. After the HDFS operation, the workflow is complete.

**Working:**

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](http://www.guru99.com/hive-tutorials.html)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.