Oozie Workflow

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

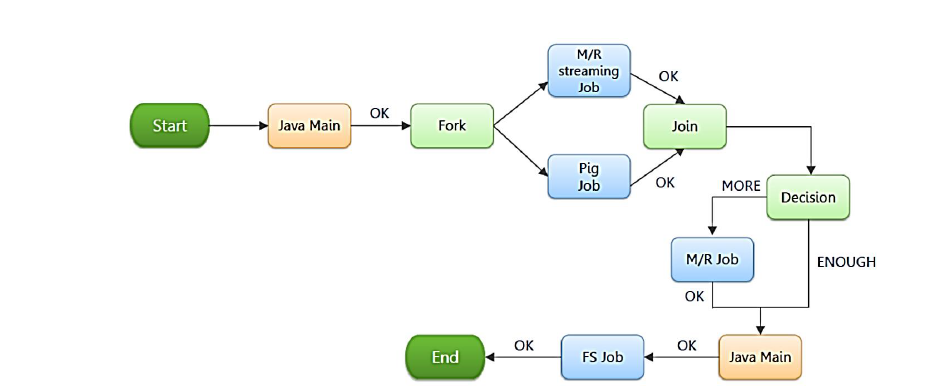
For the purposes of Oozie, a workflow is a collection of actions (i.e. Hadoop Map/Reduce jobs, Pig jobs) arranged in a control dependency DAG (Direct Acyclic Graph). "control dependency" from one action to another means that the second action can't run until the first action has completed.

Oozie 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.

**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.

There can be **decision trees** to decide how and on which condition a job should run. A **fork** is used to run multiple jobs in parallel.



# Advantages

Oozie does have some benefits that are worth considering:

1.Oozie is designed to scale in a Hadoop cluster. Each job will be launched from a different datanode. This means that the workflow load will be balanced and no single machine will become overburdened by launching workflows. This also means that the capacity to launch workflows will grow as the cluster grows.

2.Oozie is well integrated with Hadoop security. This is especially important in a kerberized cluster. Oozie knows which user submitted the job and will launch all actions as that user, with the proper privileges. It will handle all the authentication details for the user as well.

3.Oozie is the only workflow manager with built-in Hadoop actions, making workflow development, maintenance and troubleshooting easier.

4.Oozie UI makes it easier to drill down to specific errors in the data nodes. Other systems would require significantly more work to correlate jobtracker jobs with the workflow actions.

5.Oozie is proven to scale in some of the world’s largest clusters. The white paper discusses a deployment at Yahoo! that can handle 1250 job submissions a minute.

6.Oozie gets callbacks from MapReduce jobs so it knows when they finish and whether they hang without expensive polling. No other workflow manager can do this.

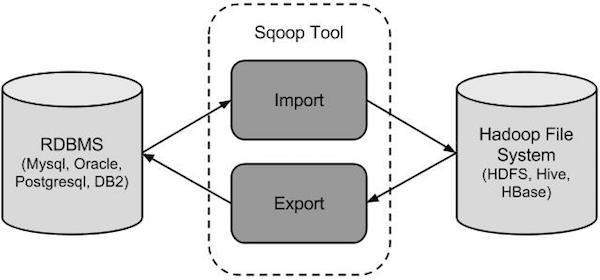
7.Oozie Coordinator allows triggering actions when files arrive at HDFS. This will be challenging to implement anywhere else.

8.Oozie is supported by Hadoop vendors. If there is ever an issue with how the workflow manager integrates with Hadoop – you can turn to the people who wrote the code for answers.

Source: apache.org

Sqoop

Apache SqoopTM is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases. 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.



# 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.

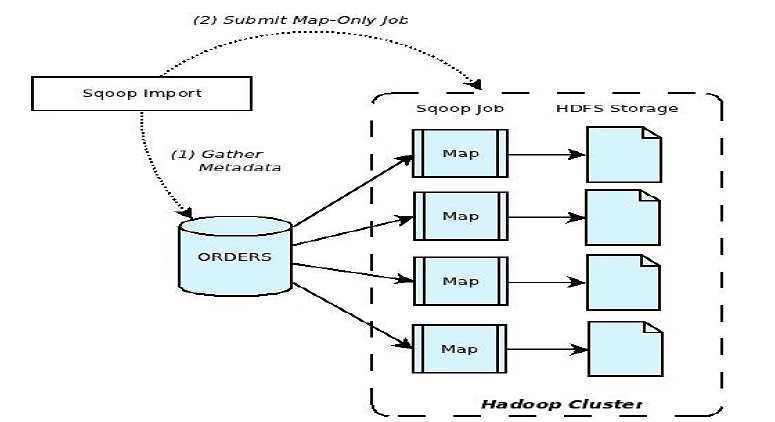
The import is done in two steps:

•In the first Step Sqoop introspects the database to gather the necessary metadata for the data being imported.

•The second step is a map-only Hadoop job that Sqoopsubmits to the cluster.

# 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.



Source: [www.tutorialspoint.com](http://www.tutorialspoint.com)

# Advantages of Sqoop

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.

Below are the advantages of Apache Sqoop, which is also the reason for choosing this technology in this layer.

* 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. This helps not only to limit the usage of the mainframe, but also reduces the high cost in executing certain jobs using mainframe hardware.

Source: [www.safaribooksonline.com](http://www.safaribooksonline.com)