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?

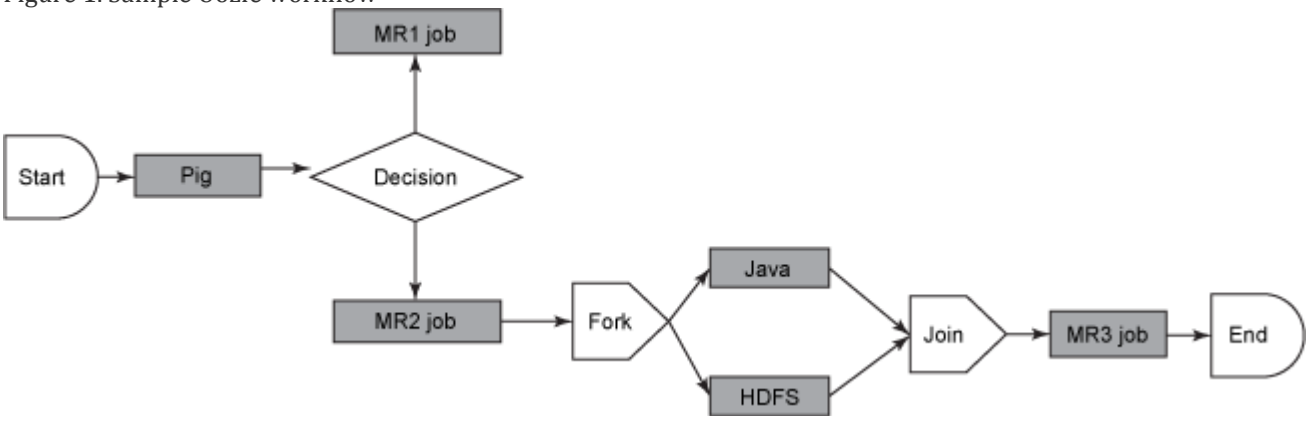
An Oozie workflow is a collection of actions arranged in a directed acyclic graph (DAG). This graph can contain two types of nodes: control nodes and action nodes. Control nodes, which are used to define job chronology, provide the rules for beginning and ending a workflow and control the workflow execution 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.

Benefits of Oozie:

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.

Conclusion:

Helping Hadoop users in chaining and automating the execution of big data processing tasks into a defined workflow is quite useful feature in real-world practices. In this article, you were introduced to Oozie, an Apache open source project that simplifies the process of creating workflow and coordination between Hadoop-based jobs.

However, Oozie is not the only project that can help you to achieve this goal. Other projects include Azkaban (written and open-sourced by LinkedIn), Luigi(Python-based workflow engine) and Cascading (supports any JVM-based language such as Java, JRuby, and Clojure).

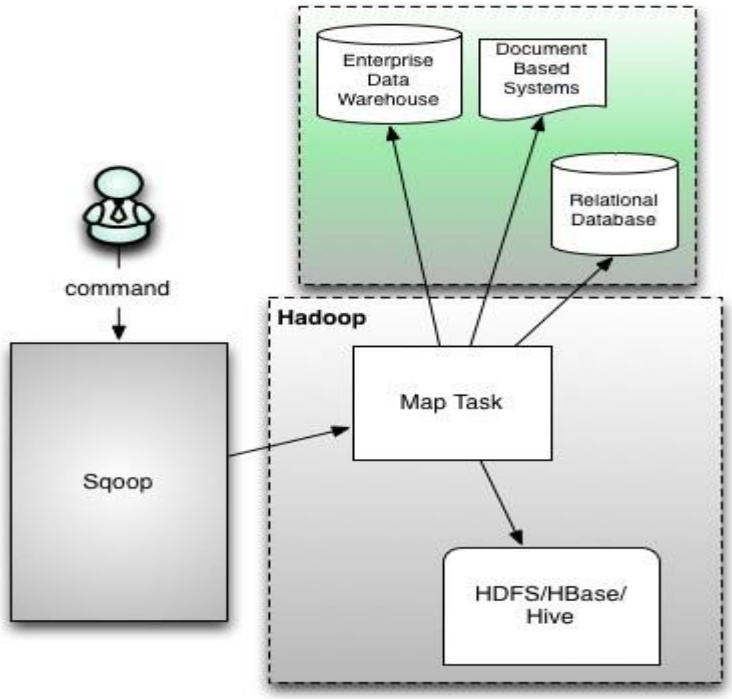
What is Sqoop?

Apache Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and external datastores such as relational databases, enterprise data warehouses.

Sqoop is used to import data from external datastores into Hadoop Distributed File System or related Hadoop eco-systems like Hive and HBase. Similarly, Sqoop can also be used to extract data from Hadoop or its eco-systems and export it to external datastores

such as relational databases, enterprise data warehouses. Sqoop works with relational databases such as Teradata, Netezza, Oracle, MySQL, Postgres etc.

How does Sqoop works?



Sqoop Architecture

Sqoop provides command line interface to the end users. Sqoop can also be accessed using Java APIs. Sqoop command submitted by the end user is parsed by Sqoop and launches Hadoop Map only job to import or export data because Reduce phase is required only when aggregations are needed. Sqoop just imports and exports the data; it does not do any aggregations.

Sqoop parses the arguments provided in the command line and prepares the Map job. Map job launch multiple mappers depends on the number defined by user in the command line. For Sqoop import, each mapper task will be assigned with part of data to be imported based on key defined in the command line. Sqoop distributes the input data among the mappers equally to get high performance. Then each mapper creates connection with the database using JDBC and fetches the part of data assigned by Sqoop and writes it into HDFS or Hive or HBase based on the option provided in the command line.

Where is Sqoop used?

Relational database systems are widely used to interact with the traditional business applications. So, relational database systems has become one of the sources that generate Big Data.

As we are dealing with Big Data, Hadoop stores and processes the Big Data using different processing frameworks like MapReduce, Hive, HBase, Cassandra, Pig etc and storage frameworks like HDFS to achieve benefit of distributed computing and distributed storage. In order to store and analyze the Big Data from relational databases, Data need to be transferred between database systems and Hadoop Distributed File System (HDFS). Here, Sqoop comes into picture. Sqoop acts like a intermediate layer between Hadoop and relational database systems. You can import data and export data between relational database systems and Hadoop and its eco-systems directly using sqoop.

Benefits of Sqoop:

For Hadoop developers, the interesting work starts after data is loaded into HDFS. Developers play around the data in order to find the magical insights concealed in that Big Data. For this, the data residing in the relational database management systems need to be transferred to HDFS, play around the data and might need to transfer back to relational database management systems. In reality of Big Data world, Developers feel the transferring of data between relational database systems and HDFS is not that interesting, tedious but too seldom required. Developers can always write custom scripts to transfer data in and out of Hadoop, but Apache Sqoop provides an alternative.

Sqoop automates most of the process, depends on the database to describe the schema of the data to be imported. Sqoop uses MapReduce framework to import and export the data, which provides parallel mechanism as well as fault tolerance. Sqoop makes developers life easy by providing command line interface. Developers just need to provide basic information like source, destination and database authentication details in the sqoop command. Sqoop takes care of remaining part.

<u>Sqoop provides many salient features like:</u>

1. Full Load
2. Incremental Load
3. Parallel import/export
4. Import results of SQL query
5. Compression

6. Connectors for all major RDBMS Databases

7. Kerberos Security Integration

8. Load data directly into Hive/Hbase

9. Support for Accumulo

Sqoop is Robust, has great community support and contributions. Sqoop is widely used in most of the Big Data companies to transfer data between relational databases and Hadoop.