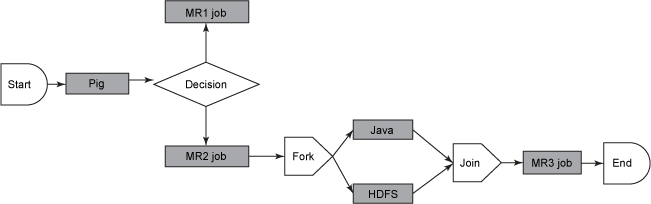
**Assignment 10.1**

**The Workflow of Oozie and its Benefits:**

* 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:**

* Multiple jobs with dependency.
* Job starts based on time or data availability.
* Monitoring and operational support for jobs.
* Horizontal Scalability: If load increases, add machines into Hadoop cluster
* Stability: Isolation of user code and system process
* Scalable, Reliable and multi-tenant
* Provides security by verifying its user before passing the job to hadoop
* Deployed in production and growing fast

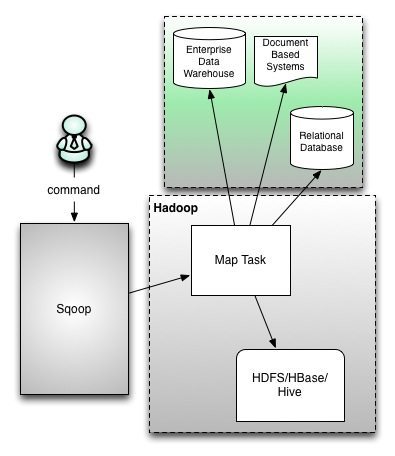
**The workflow of sqoop and its benefits:**

Sqoop = SQL – to- Hadoop

To use Hadoop for analytics requires loading data into Hadoop clusters and processing it in conjunction with data that resides on enterprise application servers and databases. Loading GBs and TBs of data into HDFS from production databases or accessing it from map reduce applications is a challenging task. While doing so, we have to consider things like data consistency, overhead of running these jobs on production systems and at the end if this process would be efficient or not. Using batch scripts to load data is an inefficient way to go with. Sqoop is one solution to this.

### How Sqoop Works

Sqoop provides a pluggable mechanism for optimal connectivity to external systems. The Sqoop extension API provides a convenient framework for building new connectors which can be dropped into Sqoop installations to provide connectivity to various systems. Sqoop itself comes bundled with various connectors that can be used for popular database and data warehousing systems.



* The sqoop action runs a Sqoop job.
* The workflow job will wait until the Sqoop job completes before continuing to the next action.
* To run the Sqoop job, you have to configure the sqoop action with the =job-tracker=, name-node and Sqoop command or arg elements as well as configuration.
* A sqoop action can be configured to create or delete HDFS directories before starting the Sqoop job.
* Sqoop configuration can be specified with a file, using the job-xml element, and inline, using the configuration elements.
* Oozie EL expressions can be used in the inline configuration. Property values specified in the configuration element override values specified in the job-xml file.
* Note that Hadoop mapred.job.tracker and fs.default.name properties must not be present in the inline configuration.
* As with Hadoop map-reduce jobs, it is possible to add files and archives in order to make them available to the Sqoop job.

**Benefits:**

* Apache Sqoop efficiently transfers bulk data between Apache Hadoop and structured datastores such as relational databases.
* Sqoop helps offload certain tasks (such as ETL processing) from the EDW to Hadoop for efficient execution at a much lower cost.
* Sqoop can also be used to extract data from Hadoop and export it into external structured datastores.
* Sqoop works with relational databases such as Teradata, Netezza, Oracle, MySQL, Postgres, and HSQLDB