**The workflow of Oozie and its Benefits**.

Apache Oozie is a workflow scheduler for Hadoop. It is a system which runs workflow of dependent jobs. Here, users are permitted to create “Directed Acyclic Graphs” of workflows, which can be run in parallel and sequentially in Hadoop.

It consists of two parts:

* **Workflow engine** : Responsibility of a workflow engine is to store and run workflows composed of Hadoop jobs e.g., MapReduce, Pig, Hive.
* **Coordinator engine**: It runs workflow jobs based on predefined schedules and availability of data.

Oozie is scalable and can manage timely execution of thousands of workflows in a Hadoop cluster.

Oozie is very much flexible, as well. One can easily start, stop, suspend and rerun jobs. Oozie makes it very easy to rerun failed workflows. One can easily understand how difficult it can be to catch up missed or failed jobs due to downtime or failure. It is even possible to skip a specific failed node.

**How Oozie Works**

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 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.



**Benefits of Oozie**

1. Oozie is designed to scale in a Hadoop cluster. Each job will be launched from a different data node. 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 job tracker jobs with the workflow actions.
5. Oozie is proven to scale in some of the world’s largest clusters. The [**white paper**](https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxzd2VldHdvcmtzaG9wMjAxMnxneDo1NzRhYjZlNzdmNTM1Yjgw#!)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.