Q. The workflow of Oozie and its Benefits

Ans. Apache Oozie is a system for running workflows of dependent jobs. It is composed of two main parts:

1. A workflow engine that stores and runs workflows composed of different types of Hadoop jobs (MapReduce, Pig, Hive, and so on).
2. A coordinator engine that runs workflow jobs based on predefined schedules and data availability.

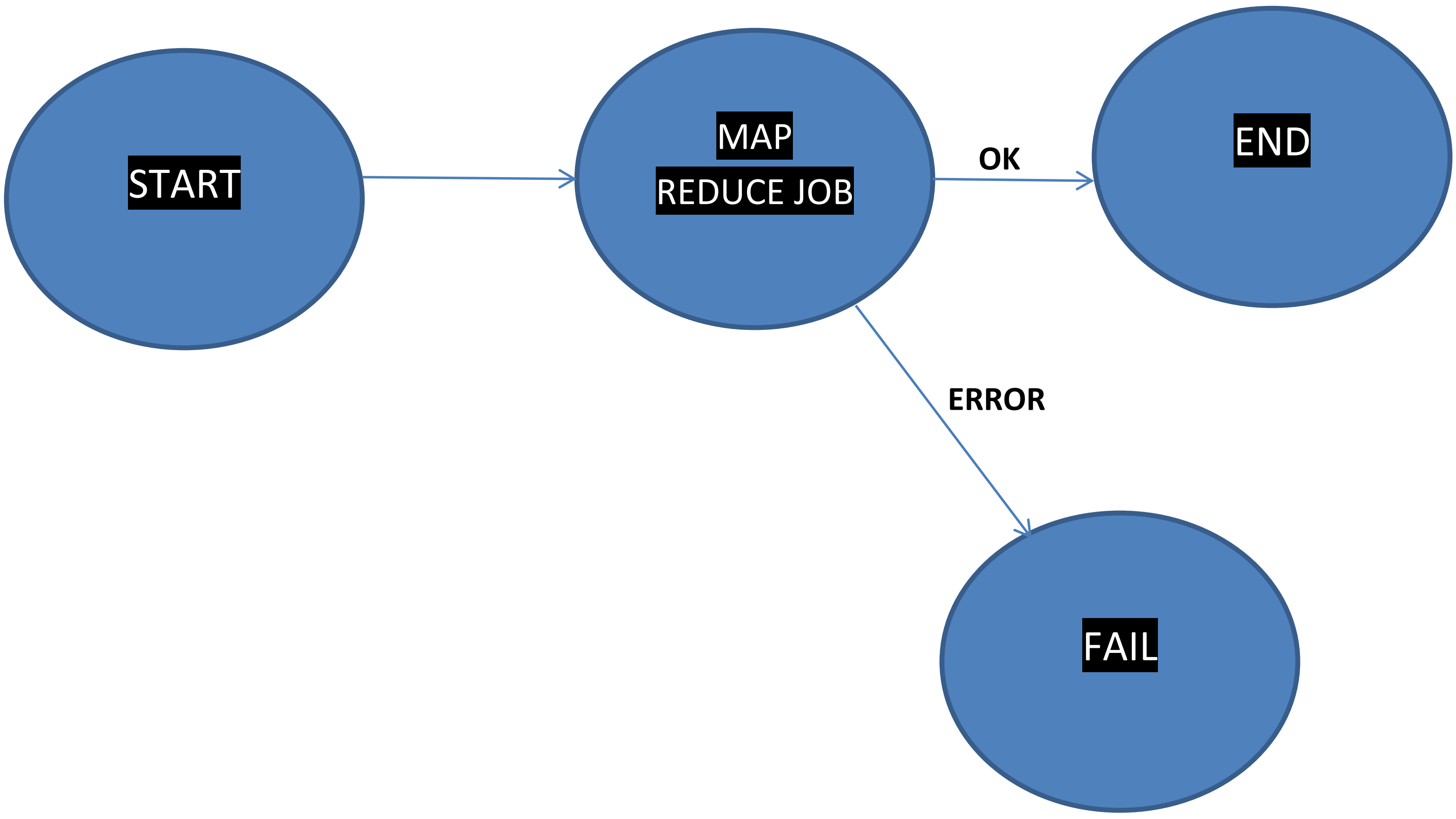
Oozie has been designed to scale, and it can manage the timely execution of thousands of workflows in a Hadoop cluster, each composed of possibly dozens of constituent jobs. Oozie makes rerunning failed workflows more tractable, since no time is wasted running successful parts of a workflow. Unlike JobControl, which runs on the client machine submitting the jobs, Oozie runs as a service in the cluster, and clients submit workflow definitions for immediate or later execution. In Oozie parlance, a workflow is a DAG of action nodes and control-flow nodes.
An action node performs a workflow task, such as moving files in HDFS; running a MapReduce, Streaming, Pig, or Hive job; performing a Sqoop import; or running an arbitrary shell script or Java program. A control-flow node governs the workflow execution between actions by allowing such constructs as conditional logic (so different execution branches may be followed depending on the result of an earlier action node) or parallel execution. When the workflow completes, Oozie can make an HTTP callback to the client to inform it of the workflow status. It is also possible to receive callbacks every time the workflow enters or exits an action node.

Workflow definitions are written in XML using the Hadoop Process Definition Language.
A Workflow application is DAG that coordinates the following types of actions: Hadoop, Pig, and sub-workflows. Flow control operations within the workflow applications can be done using decision, fork and join nodes. Cycles in workflows are not supported. Actions and decisions can be parameterized with job properties, actions output (i.e. Hadoop counters) and file information (file exists, file size, etc). A Workflow application is a ZIP file that contains the workflow definition (an XML file), all the necessary files to run all the actions: JAR files for Map/Reduce jobs, shells for streaming Map/Reduce jobs, native libraries, Pig scripts, and other resource files.
Before running a workflow job, the corresponding workflow application must be deployed in Oozie.
When submitting a workflow job, a set of properties resolving all the formal parameters in the workflow definitions must be provided. This set of properties is a Hadoop configuration.
Possible states for a workflow jobs are: PREP , RUNNING , SUSPENDED , SUCCEEDED , KILLED and FAILED .
In the case of a action start failure in a workflow job, depending on the type of failure, Oozie will attempt automatic retries, it will request a manual retry or it will fail the workflow job.
Oozie can make HTTP callback notifications on action start/end/failure events and workflow end/failure events.
In the case of workflow job failure, the workflow job can be resubmitted skipping previously completed actions. Before doing a resubmission the workflow application could be updated with a patch to fix a problem in the workflow application code.
A workflow definition is a DAG with control flow nodes (start, end, decision, fork, join, kill) or action nodes (map-reduce, pig, etc.), nodes are connected by transitions arrows.
The workflow definition language is XML based and it is called hPDL (Hadoop Process Definition Language).
Oozie does not support cycles in workflow definitions, workflow definitions must be a strict DAG.
At workflow application deployment time, if Oozie detects a cycle in the workflow definition it must fail the deployment.

Workflow nodes are classified in control flow nodes and action nodes:

- **Control flow nodes:** nodes that control the start and end of the workflow and workflow job execution path.
- **Action nodes:** nodes that trigger the execution of a computation/processing task.

Below diagram shows the transition workflow of an Oozie job-



The start node is the entry point for a workflow job, it indicates the first workflow node the workflow job must transition to.When a workflow is started, it automatically transitions to the node specified in the start .A workflow definition must have one start node.

The end node is the end for a workflow job, it indicates that the workflow job has completed successfully. When a workflow job reaches the end it finishes successfully (SUCCEEDED).
If one or more actions started by the workflow job are executing when the end node is reached, the actions will be killed. In this scenario the workflow job is still considered as successfully run.
A workflow definition must have one end node.

The kill node allows a workflow job to kill itself. When a workflow job reaches the kill it finishes in error (KILLED). If one or more actions started by the workflow job are executing when the kill node is reached, the actions will be killed. A workflow definition may have zero or more kill nodes.

A decision node enables a workflow to make a selection on the execution path to follow. The behavior of a decision node can be seen as a switch-case statement. A decision node consists of a list of predicates-transition pairs plus a default transition. Predicates are evaluated in order or appearance until one of them evaluates to true and the corresponding transition is taken. If none of the predicates evaluates to true the default transition is taken.

The map-reduce action starts a Hadoop map/reduce job from a workflow. Hadoop jobs can be Java Map/Reduce jobs or streaming jobs. A map-reduce action can be configured to perform file system cleanup and directory creation before starting the map reduce job. This capability enables Oozie to retry a Hadoop job in the situation of a transient failure (Hadoop checks the non-existence of the job output directory and then creates it when the Hadoop job is starting, thus a retry without cleanup of the job output directory would fail). The workflow job will wait until the Hadoop map/reduce job completes before continuing to the next action in the workflow execution path. The counters of the Hadoop job and job exit status (=FAILED=, KILLED or SUCCEEDED) must be available to the workflow job after the Hadoop jobs ends. This information can be used from within decision nodes and other actions configurations.
The map-reduce action has to be configured with all the necessary Hadoop JobConf properties to run the Hadoop map/reduce job. Hadoop JobConf properties can be specified in a JobConf XML file bundled with the workflow application or they can be indicated inline in the map-reduce action configuration.
The configuration properties are loaded in the following order, streaming , job-xml and configuration , and later values override earlier values. Streaming and inline property values can be parameterized (templatized) using EL expressions.

Below are some benefits of using Oozie-

- 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.
- 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.
- Oozie is the only workflow manager with built-in Hadoop actions, making workflow development, maintenance and troubleshooting easier.
- 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.
- Oozie is proven to scale in some of the world’s largest clusters.
- 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.
- Oozie Coordinator allows triggering actions when files arrive at HDFS. This will be challenging to implement anywhere else.
- Oozie is supported by Hadoop vendors.