Oozie: Run & Manage jobs in HADOOP

Manish Gupta[1*], Dr. Anish Gupta[2], Prateek Chaturvedi[3]
[1,3]Assistant Professor, Amity University Greater Noida Campus
[2]Assistant Dean Academics, Amity University Greater Noida Campus

ABSTRACT:

The size of data has been growing day by day in rapid way. Using traditional approach, it make expensive to process large set of data. Hadoop is a popular framework written in java, being used by company like Yahoo, facebook, Youtube etc. to store and process large set of data on commodity hardware. Processing of data is a tough task when we do not know format of data (Structured, Semi Structured and Unstructured). To process structured types of data we used HIVE and for semi structured and unstructured types of data we use PIG. Hive was introduced by FACEBOOK and PIG was introduced by YAHOO. Scheduling and managing of jobs in HADOOP is done by Oozie. Scheduling techniquesin HADOOP are different from traditional schedulingtechniques(SJF, FCFS, RR etc.). Oozieallow to combine multiple complex jobs to be run in a sequential order to achieve a bigger task. One of the main advantages of Oozie is that it is tightly integrated with HADOOP stack supporting various HADOOP jobs like Hive, Pig, Sqoop as well as system-specific jobs like Java and Shell.

Keywords: HADOOP, Data, Hive, Pig, Sqoop, Structured, Semi Structured, Unstructured.

INTRODUCTION:

Oozie is a Java web application workflow manager and coordinator, are used to manage and coordinate jobs in hadoop ecosystem. Its work flow job is just like as a Direct Acyclic Graph (DAG)[1].Oozie is scalable in nature and can manage thousands jobs in hadoop cluster.

There are 3 common jobs in Oozie.

1. **Oozie Workflow Jobs:** It specify the sequence of action to be executed.
2. **Oozie Coordinator Jobs:** Coordinates the job and triggered by time and data availability.
3. **Oozie Bundle:** Package of multiple Workflow and Coordinator.

Oozieworkflow contains**action node** and **control flow node.**

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. In simple language, how to execute a job is decided by action node.

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.

There is only one control node but number of action node depend on number of jobs. Action node always will be equal to the number of job.

OOZIE ARCHITECTURE:

Oozie server is deployed as a java web application and all the necessary information are stored in a database. Database can be any type either Derby, MySQL, Oracle etc[2].

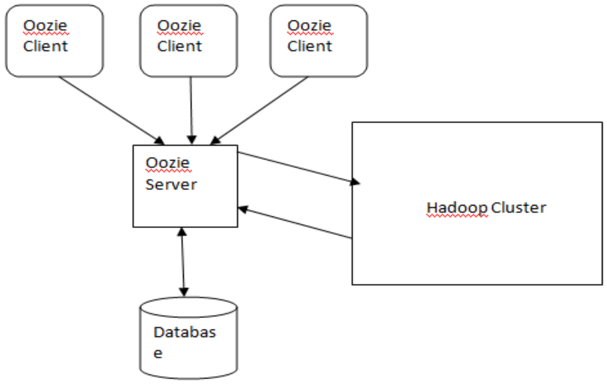

Fig1: *Oozie Architecture*

All jobs are stored in Hadoop cluster. Oozie client contact to Oozie server, for managing and processing the jobs. After processing useful information are store in database.

OOZIE WORKFLOW:

Workflow is a collection of action arranged in a DAG[3]. Oozie workflow definition written in hPDL. Oozie workflow contain a collections of node ie. Start control node, end control node, kill control node, decision node, fork node and join node.

a. **Start control node:**Every Oozie workflow must contain a start control node, and it always start the execution from start control node.
b. **End control node:**After successfully completion, it goes to end control node. Reaching to end control node means there is no error.
c. **Kill control node:**If we want to kill the execution of a workflow, then we use kill control node. There may be more than one kill control node.

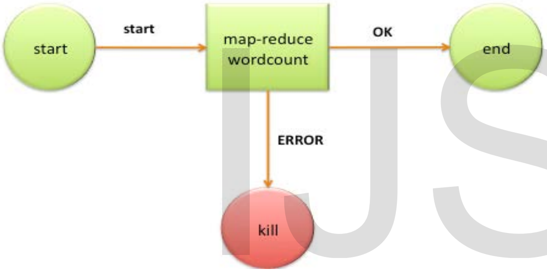

Fig2: Oozie workflow

REFERENCES:

[1] http://oozie.apache.org/
[2] https://devstacks.wordpress.com/2017/02/16/oozie-architecture-and-job-scheduling/
[3] https://oozie.apache.org/docs/3.1.3-incubating/DG_Overview.html
[4] https://www.oreilly.com/library/view/apache-oozie/9781449369910/ch04.html
[5] Dan Gunter, EwaDeelman, TaghridSamak, ChristopherBrooks, Monte Goode, Gideon Juve, Gaurang Mehta,Priscilla Moraes, Fabio Silva, Martin Swany, Karan Vahi.Online Workflow Management and Performance Analysiswith Stampede, 7th International Conference on Network andService Management (CNSM-2011), Paris, France, October2011
[6] Weiwei Chen, EwaDeelman, Workflow Overhead Analysisand Optimizations, 6th Workshop on Workflows in Supportof Large-Scale Science (WORKS 11), Seattle, Washington,November 14th, 2011.
[7] The Azkaban Project. (n.d.). Retrieved February 24, 2012from http://github.com/azkaban/azkaban
[8] Apache Pig. (n.d.). Retrieved February 24, 2012 fromhttp://pig.apache.org
[9] Gates, A. F., Natkovich, O., Chopra, S., Kamath, P.,Narayanamurthy, S. M., Olston, C., Reed, B., Srinivasan, S.& Srivastava, U. 2009. Building a high-level dataflowsystem on top of map-reduce: The Pig experience. In Proc.VLDB.
[10] Thusoo, A., Sarma, J., Jain, N., Shao, Z., Chakka, P.,Anthony, S., Liu, H., Wyckoff, P. & Murthy, R. 2009. HiveAWarehousing Solution Over a Map-Reduce Framework,IN VLDB '09: PROCEEDINGS OF THE VLDBENDOWMENT
[11] The Hive Project. (n.d.). Retrieved February 24, 2012 fromhttp://hadoop.apache.org/hive/
[12] Hunt, P., Konar, M., Junqueira, F. P., & Reed, B. 2010.ZooKeeper: Wait-free coordination for internet-scalesystems. In Proc. USENIX Annual Technical Conference.
[13] Google App Engine Multitenancy. (n.d.). RetrievedFebruary 10, 2012 fromhttp://code.google.com/appengine/docs/java/multitenancy/ov erview.html
[14] Chong, F., Carraro, G., &Wolter, R. 2006. Multi-TenantData Architecture. Retrieved February 10, 2012 fromhttp://msdn.microsoft.com/en-us/library/aa479086.aspx
[15] Wang, C. (2009, November 18). Cloud Security Front andCenter [Web log comment]. Forrester Research. Retrievedfrom http://blogs.forrester.com/srm/2009/11/cloud-securityfront-and-center.html
[16] Brodkin, J. 2008, July 02. Gartner: Seven cloud-computingsecurity risks. InfoWorld. Retrieved fromhttp://www.infoworld.com/d/security-central/gartner-sevencloud-computing-security-risks-853
[17] Security Guidance for Critical Areas of Focus in CloudComputing. (n.d.). Cloud Security Alliance. RetrievedFebruary 24, 2012 fromhttps://cloudsecurityalliance.org/research/projects/securityguidance-for-critical-areas-of-focus-in-cloud-computing/
[18] Zhang, K. 2009. Adding user and service-to-serviceauthentication to Hadoop. Retrieved February 24, 2012 fromhttps://issues.apache.org/jira/bro
[19] wse/HADOOP