International Journal of Engineering & Technology, 7 (4.36) (2018) 200-203

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paperv

A Framework For Effective Processing Of Jobs In Hadoop

Amarendra Mohanty[1], Dr.P.Ranjana [2]

[1]Research Scholar, Dept. of MCA,Hindustan Institute of Technology & Science,Chennai, India
mohanty.amarendra@gmail.com
[2]Associate Professor, School of Computing Sciences, Hindustan Institute of Technology & Science,Chennai, India
**Corresponding author E-mail: pranjana@hindustanuniv.ac.in*

Abstract

The main challenges in oozie based scheduling is high computing, high CPU usage and resource intensive. This leads to resource contention in production because it was not load balanced optimally.
The objective of the proposed New Sql Server built Java based (NSSJ) Scheduling is to overcome some of the current challenges in the existing oozie based scheduling. It stores all the inventory information on SQL Server environment. SQL Server is preferred over Hbase, because at any given point of time, there were multiple threads hitting same inventory table to ensure transaction level processing. One can run or kill or put on hold any number of deamons or jobs at any point of time. This gives complete flexibility to the end user to load balance based on the number of jobs. It has auto restart feature when a task or job fails. It will try to attempt for one re-run, if it fails second time, it will put the job in abandoned state.
Thus the proposed NSSJ scheduling load balances the resource optimally during production.

Keywords— *Hadoop, Daemons, Oozie, CPU, Cluster, Capacity*

1.Introduction

Data ingestion is one of the critical Hadoop workflows. Massive amounts of data are moved from various sources into Hadoop for performing analysis. Data ingestion is the process of obtaining, importing and processing data for later use or storage in a database.

There are few studies about performance of the Hadoop framework and cluster management in the literature. Multiple clusters can be used for the processing of Big Data and each cluster can have multiple nodes. It is difficult to manage with the increasing number of nodes in a cluster, Virtualization solutions have been developed for the installation, scheduling, deployment and efficient resource management.

In this study, Data ingestion framework defines the processes and guidelines for landing data from a variety of sources "as-is" into Hadoop environment. The framework encapsulates all the components, processes and "how-to" guidelines, that project teams can utilize to land their data into Hadoop environment from varieties of data sources e.g. databases, rest APIs, FTP/SFTP servers with ease.

2.Background

The diagrams below depict the data ingestion framework, with the typical flow of data from source systems into Hadoop environment.

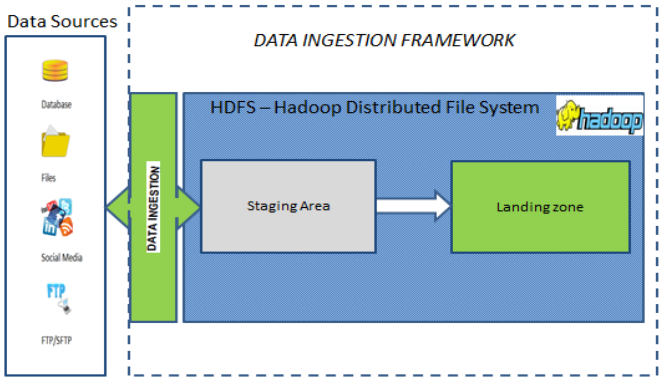

Figure 1: Data Ingestion Framework

As data is brought in from different types of sources, the ingestion framework provides common routines for each type of source that shall connect to and extract data from the source system.

2.1. Data Ingestion has four major operations (irrespective of sources system)

1. **Pre Data Ingest** – Preparations to occupy the source system and adding metadata entries to the reputed systems
2. **Data Ingest** – Process of ingesting data from external or internal sourced to Hadoop Staging Area without applying any transformations – as is data from source system.
3. **Data Movement** – Moves the data from Staging area to landing zone for analytical users access.

4. Post Data Ingest – Mails are triggers to team on success/failure of ingestion and movement. Audit and logging will be done using HBASE and Unix box.

3.Key Features

- Generic, works in batch based/adhoc based mode
- Capable of performing Ingestion from Databases/File based systems
- Reusable code libraries that can be adapted by any new source system with the minimal changes to cetain files(Properties and Catalog updates)
- Contains Hive based services.
- Common Hadoop file handling utilities in Java.
- Properties which drives the framework to act on a particular source.
- Workflows & shell scripts that run the java code.
- Falcons which govern the scheduling of the whole ingestion process.
- Metadata, Auditing and Logging is handled within the code framework.
- Exception Handling and necessary mail notification on success/failures are incorporated.
- Ease to perform build and deploy to different environments.

4.Hadoop File System Hierarchy

The Hadoop File System Hierarchy(HDFS) covers the folder structure for both HDFS and local Unix file systems in Development, Quality Assurance and Production environments. This file system stores the ingested data, and code required for data ingestion and supporting Hadoop related files.
Key elements stored in HDFS:

- Source Data Files
- Hive external tables
- Pig and Hive scripts
- MapReduce code and user defined functions
- Intermediate processing files
- Scheduling configuration files
- File watchers
- Workflows
- Java Binaries

Staging Area: This acts as a temporary storage area which holds the raw un-altered data from the source systems.
Landing Area: Raw un-altered data stored in staging area is copied into this zone which acts as a permanent storage area for the data ingested.

5.Nssj Framework

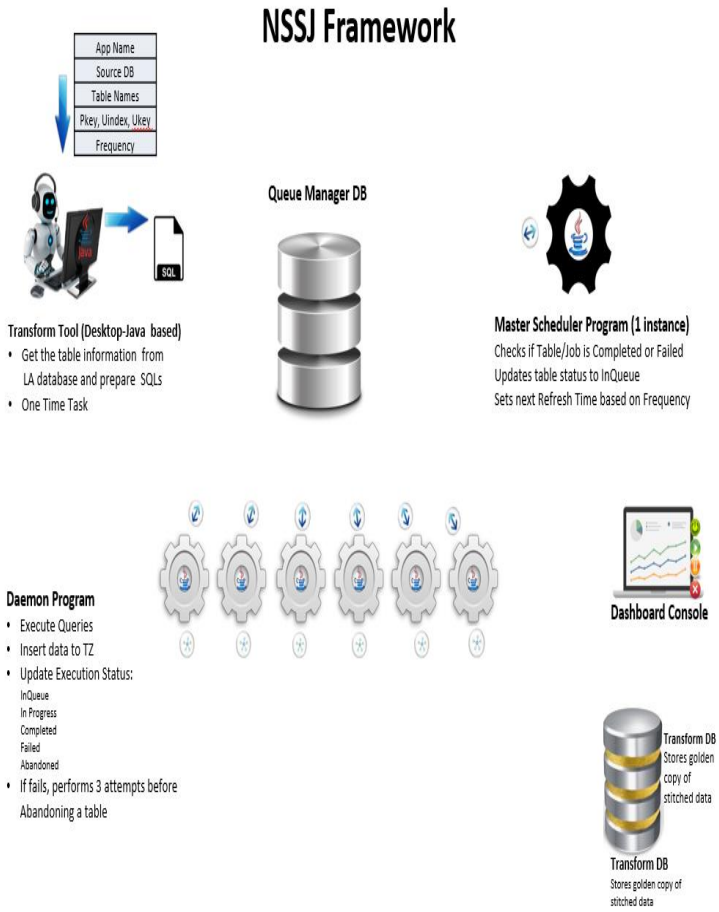

Figure 2: NSSJ Framework

5.1. High Level Architecture:

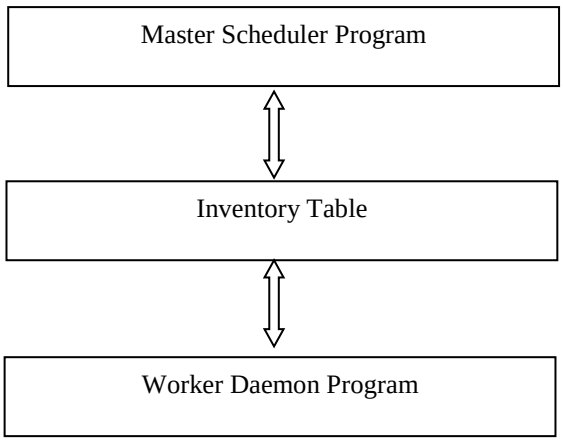

Figure 3: High Level Architecture of NSSJ Framework

Master scheduler program will be running always in the background and it will keep hitting polling/queue table, fetch the next set of the tables , that needs to processed based on priority, next refresh time, Status.
Inventory table will have following information:
Table Name, Application name, frequency of the refresh (Daily, Weekly, Every 4 hours, every 30 minutes). Last refreshed date time status: (In queue, In progress, Ready, Pending for submission, completed, Failed), Next Refresh date time, Priority

Order, Manual Override Flag (to override the priority and give higher precedence

Worker Daemon Program will process those set of tables sequentially. Based on the status, it needs to write back the status of that process back to the Inventory table. Once it completes all the tables, the daemon process return the control back to Master Scheduler program with number of success/failures and spawn out from the queue.

Master Scheduler Flow Diagram:

Step 1: with next Refresh Date time & it's Status as Completed through the SMF Framework tool

Step 2: if the current date time is greater than Next Refresh Date time, please set all the tables to In Queue State

Worker Daemon Flow Diagram:

Step 1: Worker Daemon will look for tasks/jobs that are in "InQueue" Status and start changing the status to "InProcess" based on the priority order

Step 2: Each and every task/job is associated to hql /some scripts, which is stored in hdfs and framework will read hql and run the same

Step 3: Based on the status of the output it will set to either completed or abandoned status.

6.Comparision Between Oozie Vs Nssj

Here are the number of jobs consuming the clusters and no.of job failures in a typical peak time scenario

OOZIE		
Cluster Usage	# No. of Jobs Running	#No. of failure
552.10	608	0
552.30	616	0
551.60	741	0
550.80	771	0
551.40	771	0
551.90	539	0
552.50	815	1
551.30	781	0
551.50	962	7
551.00	976	79
552.00	864	0
551.80	1000	130
376.90	630	0
502.70	648	0
551.40	602	0
551.60	999	132
548.60	997	245
551.20	1000	252
550.50	1000	154
550.50	1000	357
550.40	1000	250
550.80	999	48
551.70	977	18

Figure 4: Cluster Usage Metrics for OOZIE

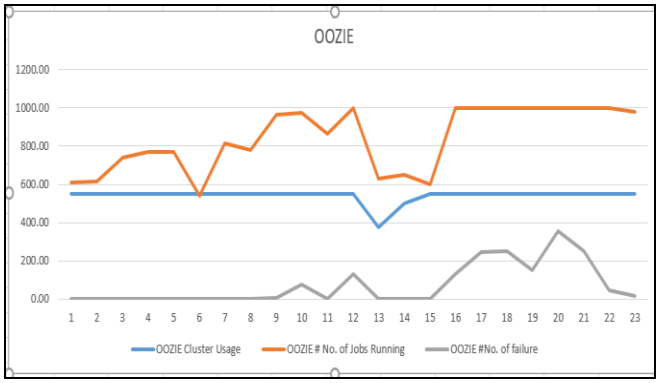

Figure 5: Cluster Usage Graph for OOZIE

Here are the number of jobs consuming the clusters and no.of job failures in a typical peak time scenario.

NSSJ		
Cluster Usage	# No. of Jobs ru	#No. of failure
351.46	850	0
255.66	837	0
400.00	841	0
293.75	849	0
350.76	820	0
115.65	823	0
320.98	850	0
298.76	850	0
365.00	850	0
312.24	850	0
351.36	850	0
351.16	850	0
276.26	832	0
302.06	791	0
400.76	850	0
290.65	850	0
188.97	850	0
332.34	850	0
334.00	850	0
333.43	801	0
349.76	820	0
350.16	832	0
398.32	821	0

Figure 6: Cluster Usage Metrics for NSSJ

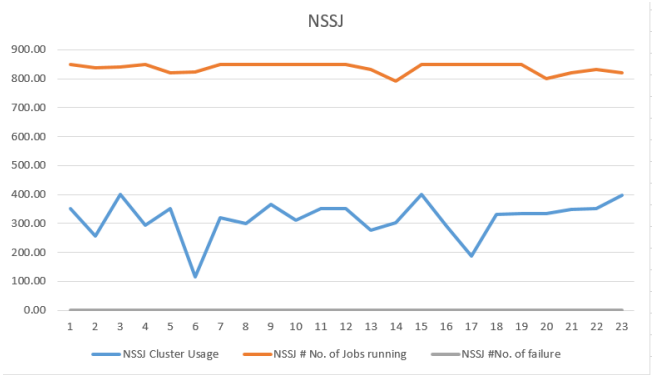

Figure 7: Cluster Usage Graph for NSSJ

7.Conclusion & Next Steps

The usage of cluster is optimized through the NSSJ framework. It would enhance the performance of the system. It results in reducing the no. of job failures and improves the processing time due to optimization of resource utilization. It is controlled job execution framework. No. of jobs can be decided based on the availability of resources and cluster capacity. This framework ensures the efficient use of resources (CPU memory). The scope of this framework can be extended to optimize the job processing time.

References

[1] R. Gu, X. Yang, J. Yan, Y. Sun, B. Wang, C. Yuan. (2014) "Hadoop: Improving MapReduce performance by optimizing job execution mechanism in Hadoop clusters", Journal of Parallel and Distributed Computing, vol. 7, n. 03, pp. 2166-2179.

[2] Yuansong Qiao, Xueyuan Wang, Guiming Fang, Brian Lee. (2016) "Doopnet: An Emulator for Network Performance Analysis of Hadoop Clusters Using Docker and Mininet", IEEE Symposium on

[3] Computers and Communication. Pp 784-790.

[4] Rammohan, N., Baburaj, E. (2014) "Genetic Clustering with Workload Multi-task Scheduler in Cloud Environment", International Journal on Communications Antenna and Propagation, pp. 77-86.

[5] C. Vorapongkitipun and N. Nupairoj. (2014) "Improving performance of small-file accessing in Hadoop", 11th International Joint Conference on Computer Science and Software Engineering, pp. 200-205.

[6] M. Ishii, J. Han, and H. Makino. (2013) "Design and performance evaluation for hadoop clusters on virtualized environment", International Conference on Information Networking (ICOIN), pp. 244-249.

[7] J. B. Buck, N. Watkins, J. LeFevre, K. Ioannidou, C. Maltzahn, N. Poly-zotis. (2011) "Array-based query processing in Hadoop", International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1-11.

[8] J. Xie, S. Yin, X. Ruan, Z. Ding, Y. Tian, J. Majors. (2010) "Improving mapreduce performance through data placement in heterogeneous hadoop clusters", IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum, pp. 1-9.

[9] Zaharia, M. (2009) "Job scheduling for multi-user MapReduce clusters", EECS Department, University of California, Berkeley, Vol. 55, pp. 1-16.

[10] Guo S. (2013). "Hadoop Operations and Cluster Management", Packt Publishing.

[11] J. Dean and S. Ghemawat. (2008) "MapReduce: Simplified data processing on large clusters", Communication of the ACM, vol. 51, pp. 107-113.

[12] Tan YS, Tan J, Chng ES. (2011) "Hadoop framework: impact of data organization on performance". Wiley Online Library, 43: 1241-1260.

[13] Lee SW, Yu F. (2014) "Securing KVM-based cloud systems via virtualization introspection", 47th Hawaii International Conference on System Sciences, pp. 5028-5037.