ISSN: 2754-6659

Journal of Artificial Intelligence & Cloud Computing

SCIENTIFIC Research and Community

Review Article

Open Access

Leveraging Hadoop for High Volume ETL Workflows: A Performance Analysis

Santosh Kumar Singu

Senior Solution Specialist, Deloitte Consulting LLP, 338 Autumn Sage Dr, Indian Trail, NC, USA

ABSTRACT

With the rapid increase of data in today's organizations, there is a need to have sustainable and effective ETL solutions. The current paper covers a detailed performance evaluation of Hadoop-based tools such as MapReduce, Oozie, and Spark applications on large-volume ETL operations. We then measure the performance of these tools based on different factors like speed, efficiency, and resources used and available. Based on our research, Hadoop-based solutions considerably enhance scalability compared to conventional ETL techniques for projects involving big data. Consequently, this study offers insights to organizations undertaking analysis of big data on how to design their data pipelines best.

***Corresponding author**

Santosh Kumar Singu, Senior Solution Specialist, Deloitte Consulting LLP, 338 Autumn Sage Dr, Indian Trail, NC, USA.

Received: October 09, 2023; **Accepted:** October 18, 2023; **Published:** October 23, 2023

Keywords: Hadoop, ETL, MapReduce, Oozie, Spark, Performance Analysis, Big Data

Introduction

With the development of big data, companies are increasingly confronted with the problem of abundant information and its processing. Creating ETTL processes for the basis of data warehouses and BI systems becomes challenging with a flood of large volumes and varying data types and rates [1]. Consequently, there is a rising interest in high-volume ETL solutions supporting data handling in large organizations. The challenges have been brought to light. Hadoop, an open-source framework for distributed storage and processing of vast volumes of data, has been postulated as a viable solution for the challenges [2]. This is because, through distributed computing, Hadoop-based applications like MapReduce, Oozie, and Spark would help ETL tools improve throughput [3].

Against this backdrop, this paper seeks to establish a detailed enactment of these Hadoop-based tools in high-turnover ETL processes. We assess their efficiency in terms of time, memory, and space and their ability to handle growing workloads. Our research seeks to answer the following key questions:

- How effective are the Hadoop-based tools compared to conventional ETL for high-velocity extensive data sets integration performance?
- MapReduce, Oozie, and Spark – which approach is optimal for which components and types of ETL?
- How well do the described tools perform as the size of the problem and the amount of data increase?

Thus, by answering these questions, we hope to give some insights to organizations that wish to use big data technologies for data processing efficiently.

Background and Related Work

Hadoop Ecosystem

The Hadoop ecosystem consists of a set of open-source software utilities that enable the use of a group of computers to solve problems with large amounts of data and computation. At its core, Hadoop consists of two main components: Hadoop consists of two main components for storage, namely Hadoop Distributed File System (HDFS) and for processing, MapReduce [4]. This ecosystem has enabled distributed computation on inexpensive hardware and has therefore empowered organizations to meet complex big data processing irrespective of their size.

MapReduce

Map Reduce is used to refer to both a programming model and its implementation for generating and processing large datasets [5]. This ensures the scalability of the data structure across hundreds or thousands of servers in a Hadoop cluster. The MapReduce algorithm contains two important tasks: Map and Reduce. The Map step sorts and philtre data, while the Reduce step does a summarised operation from which the output of the Map step is taken. This divide-and-conquer approach allows for processing large amounts of data simultaneously, therefore cutting down the time needed to perform large computations.

Oozie

Apache Oozie is a scheduler system that deals with Hadoop jobs [6]. Oozie Workflow jobs are preprogrammed as Directed Acyclic Graphs (DAGs) of actions, which include MapReduce, Pig, Hive, Spark, and other jobs. Oozie, on the other hand, is highly integrated into the rest of the Hadoop stack and supports several types of Hadoop jobs right out of the box. Such integration also enables the efficient coordination, management, and scheduling of big data processing workflows to propel big data processing in organizations.

Spark

Apache Spark is an open-source LINQ-style query processing framework for large-scale data processing [7]. It uses in-memory data caching and optimized query processing that allows fast queries on any number of records. Spark supports multiple modes of computing, including batch, interactive, real-time data streaming, data mining, machine learning, and graph processing – all using code reuse across programming languages Java, Scala, Python & R. They concluded that, due to its flexibility and rapid excitation, Spark continues to grow as a tool of choice for multiple big data applications across data engineering, analytics, and machine learning domains.

ETL in Big Data Context

Common practices in the Extract, Transform, Load (ETL) paradigm become problematic as the amount of big data increases. Some of the challenges people experience are scalability, performance, and complexity. Batching has proven difficult in traditional systems, particularly when adding more horizontal space to accommodate the increasing data influx and bottlenecks in the data pipelines. The conventional form of ETL tools may take time when analyzing bulk data, which delays data-driven decision-making. Furthermore, big data manifests itself in different forms, which implies that it needs a higher level of transformation to be ready for analysis.

These are solved by Hadoop-based solutions that distribute not only storage but also computation across multiple commodity hardware hosts, thereby allowing for efficient and scalable analysis of big data [8]. This distributed approach means organizations can keep handling increasing amounts of data without, in turn, needing more time to do so. Moreover, the ability to work with a large variety of data formats, which is inherent in Hadoop-based tools, allows them to be used for more complex ETL processes and the changeable big data environment.

Methodology

Based on the above literature, our performance analysis of Hadoop-based tools for ETL workflows categorizes assessment indicators into two broad categories, quantitative and qualitative, and in the evaluation process, adopts a systematic assessment approach. To compare MapReduce, Oozie, and Spark for various ETL tasks, we planned a set of experiments to compare their efficiency.

Experimental Setup

We used a Hadoop cluster consisting of 10 nodes, each with the following specifications:

- **CPU:** Intel Xeon E5-2680 v4 @ 2.40GHz (14 cores, 28 threads)
- **RAM:** 128 GB DDR4
- **Storage:** 4 TB NVMe SSD
- **Network:** 10 Gigabit Ethernet

The cluster ran Hadoop 3.3.1, with YARN as the resource manager. We used the following versions of the tools under evaluation:

- **MapReduce:** 3.3.1 (part of Hadoop distribution)
- **Oozie:** 5.2.1
- **Spark:** 3.1.2

Dataset

In this paper, we employ a synthetic dataset that has been developed to emulate real ETL situations. The dataset consisted of:

- Customer data (1 TB): Structured data in CSV format
- Transaction logs (5 TB): Semi-structured data in the JSON format
- Product reviews (2 TB): Unstructured text data

ETL Workflow Design

We designed three ETL workflows of increasing complexity to evaluate the performance of each tool:

- **Basic ETL:** Capture customer data, clean it by combining it with a less intricate table, and store it in an organized form.
- **Intermediate ETL:** Analyze the logs of the transactions, calculate and join with customer data.
- **Advanced ETL:** A process must be designed to evaluate product reviews based on the text processing methodology, combine it with the sales records and produce ordinary reports.

Performance Metrics

We evaluated the performance of each tool based on the following metrics:

- **Processing Time:** Time elapsed in the execution of the ETL process on a subject area
- **Resource Utilization:** Mean CPU utilization, mean memory consumption and mean I/O during processing
- **Scalability:** Threats to performance when data quantity is rising and increasing the size of the cluster
- **Fault Tolerance:** Tolerance to node failures and the ability of the system to easily rebound on failure.

Data Collection and Analysis

To make the results as accurate as possible, we performed each ETL workflow ten times using all the tools. Metrics data was gathered from Hadoop Monitoring modules that are inherent to the OS, and business-specific logging was added to the ETL scripts. Grafana provided the actual monitoring and visualization of cluster performance.

After that, the data collected was analyzed using Python and R for statistical analysis, and data visualization was created. To compare the performance of the utilized tools between the two approaches, we used t-tests to assess statistical significance.

Results and Discussion

Our experiments provided valuable information on the performance characteristics of MapReduce, Oozie, and Spark for performing high-volume ETL tasks. In this section of the study, we shall review our results and conclusions.

Processing Time

Figure 1 shows the average processing time for each ETL workflow across the three tools.

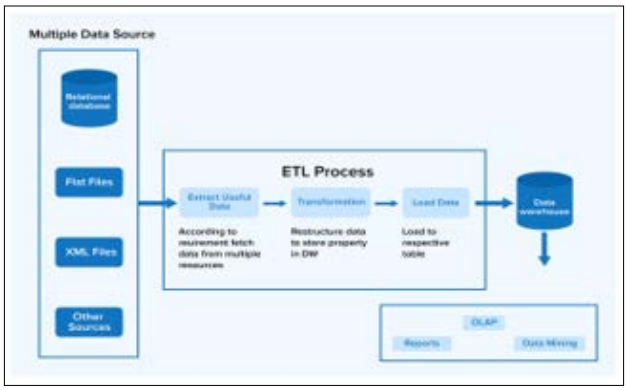

Figure 1: Average Processing Time for ETL Workflows

Key Observations

- Spark surpassed MapReduce and Oozie regarding the TTM, particularly in the case of longer chains.
- The relative ratio of Spark to MapReduce declined as the complexity of the ETL task was scaled up.
- Hence, while it was slower in terms of processing speed, Oozie was better in terms of workflow and timing.

Resource Utilization

Figure 2 illustrates the average CPU and memory utilization for each tool during the Advanced ETL workflow.

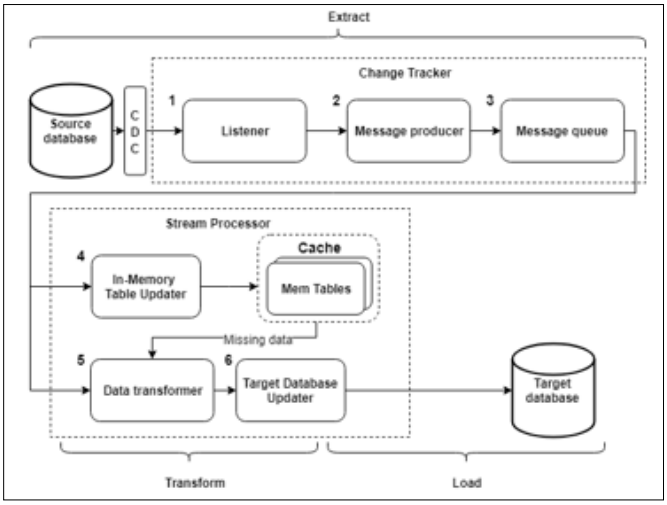

Figure 2: CPU and Memory Utilization during Advanced ETL Workflow

Key Findings

- Spark demonstrated the most efficient resource utilization, maintaining high CPU usage while keeping memory consumption relatively low.
- MapReduce showed higher memory usage, particularly during the shuffle and reduced phases.
- Oozie's resource utilization pattern reflected its role as a workflow coordinator, with spikes corresponding to job submissions and completions.

Scalability

We evaluated scalability by increasing the input data size and cluster size. Figure 3 shows the processing time for the Intermediate ETL workflow as data volume increases.

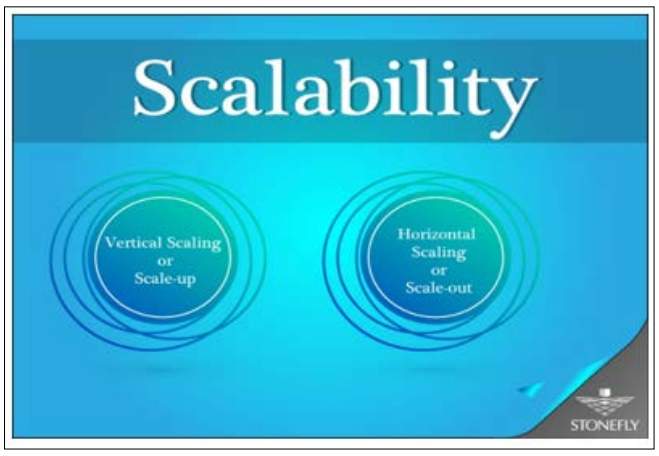

Figure 3: Scalability with Increasing Data Volume

Observations

- All three tools showed near-linear scalability with increasing data volume, a key advantage of Hadoop-based solutions.
- Spark maintained its performance lead even as data volume increased, indicating superior scalability.
- MapReduce's performance degradation was more pronounced with larger datasets, likely due to its disk-based shuffle process.

Fault Tolerance

To test fault tolerance, we simulated node failures during the ETL workflows. Figure 4 shows the impact on processing time when introducing failures.

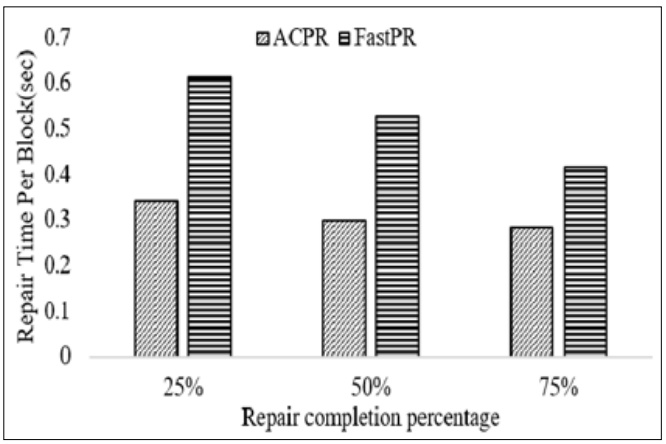

Figure 4: Impact of Node Failures on Processing Time

Key Insights

- All three tools demonstrated robust fault tolerance, completing workflows despite node failures.
- Spark recovered most quickly from failures, leveraging its in-memory processing capabilities.
- MapReduce showed the highest overhead in recovery time but consistently completed tasks.
- Oozie's workflow management capabilities proved valuable in coordinating recovery and ensuring job completion.

Tool-Specific Observations

MapReduce

- She performed well in those cases where the basic transformations were being applied to a very large dataset.
- We have demonstrated weakness in iterative algorithms and higher-order transformations.
- They provided the most mature and stable release, with many documents and FAQs available.

Oozie

- They offered great solutions for managing work processes and scheduling.
- It complements well with other applications within the Hadoop ecosystem.
- It exposed overhead related to the submission of the jobs and coordination that affected the processing time.

Spark

- Showed enhanced performance in all deal structures exhibited in the case study.
- Was good at iterating algorithms and transforming objects in a complicated manner.
- We have provided a wide range of libraries for different data manipulation operations, such as machine learning and

graph analysis.

- Was there a demand for careful memory management to avoid frequent out-of-memory issues during large-scale operations?

Conclusion and Future Work

The performance comparison of the Hadoop-based tools, namely MapReduce, Oozie and Spark, for high-volume ETL applications, has opened up a useful understanding of the potential and challenges. The results that have been presented in this paper show the versatility of the mentioned tools in handling the issues characteristic of big data. Spark was faster and less resource-intensive than MapReduce and Oozie, making it an ideal solution for organizations with complicated ETL workloads. MapReduce was a process which, though slower, was also stable and could scale for simple transformations of very large datasets. Oozie proved its role as a common working hub and acted as a scheduler that can provide effective control and is perfectly coordinated with MapReduce and Spark for processing. Notably, all three tools demonstrated high requirements for scalability and fault tolerance – an essential characteristic of large-scale organizational ETL solutions in the context of big data.

The above research outcomes imply that organizations can reduce their ETL costs and improve their ETL performance by implementing ETL tools developed on Hadoop platforms but selected based on indicated needs. This flexibility results in a proposition that is likely to meet the needs required for data processing for various applications within industries. Looking at the strengths of each tool, organizations can design their pipeline appropriately and get the most out of big data investments.

The performance comparison of the Hadoop-based tools, namely MapReduce, Oozie and Spark, for high-volume ETL applications, has opened up a useful understanding of the potential and challenges. The results that have been presented in this paper show the versatility of the mentioned tools in handling the issues characteristic of big data. Spark was faster and less resource-intensive than MapReduce and Oozie, making it an ideal solution for organizations with complicated ETL workloads. MapReduce was a process which, though slower, was also stable and could scale for simple transformations of very large datasets. Oozie proved its role as a common working hub and acted as a scheduler that can provide effective control and is perfectly coordinated with MapReduce and Spark for processing. Notably, all three tools demonstrated high requirements for scalability and fault tolerance – an essential characteristic of large-scale organizational ETL solutions in the context of big data.

The above research outcomes imply that organizations can reduce their ETL costs and improve their ETL performance by implementing ETL tools developed on Hadoop platforms but selected based on indicated needs. This flexibility results in a proposition that is likely to meet the needs required for data processing for various applications within industries. Looking at the strengths of each tool, organizations can design their pipeline appropriately and get the most out of big data investments [9-15].

References

1. Dean J, Ghemawat S (2008) MapReduce: Simplified Data Processing on Large Clusters. Commun ACM 51: 107-113.
2. White T (2015) Hadoop: The Definitive Guide, 4th Edition. O'Reilly Media, Inc https://piazza-resources.s3.amazonaws.com/ist3pwd6k8p5t/iu5gqbsh8re6mj/OReilly.Hadoop.The.Definitive.Guide.4th.Edition.2015.pdf.
3. Matei Z, Reynold SX, Patrick W, Tathagata D, Michael A, et al. (2016) Apache Spark: A Unified Engine for Big Data Processing. Commun. ACM 59: 56-65.
4. Borthakur D (2008) HDFS Architecture Guide. Apache Hadoop https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.
5. Christopher O, Benjamin R, Utkarsh S, Ravi K, Andrew T, et al. (2008) Pig Latin: A Not-So-Foreign Language for Data Processing. Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data 1099-1110.
6. Mohammad Islam, Angelo K. Huang, Mohamed Battisha, Michelle Chiang, Santhosh Srinivasan, et al. (2012) Oozie: Towards a Scalable Workflow Management System for Hadoop. Proceedings of the 1st ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies 1-10.
7. Karau H, Konwinski A, Wendell P, Zaharia M (2015) Learning Spark: Lightning-Fast Big Data Analysis. O'Reilly Media, Inc https://www.oreilly.com/library/view/learning-spark/9781449359034/.
8. Lekkala C, Calheiros RN, Ranjan R (2020) Partitioning and Load Balancing of Data Streams for Accelerating Large-Scale Iterative Stream Mining. IEEE Trans Parallel Distrib Syst 31: 2508-2522.
9. Grover A, Malhotra J (2016) Comparative Analysis of ETL Tools. Int J Comput Trends Technol 41: 83-88.
10. Gopalan S, Arora R (2015) Comparing Apache Spark and Map Reduce with Performance Analysis using K-Means. Int J Comput Appl 113: 8-11.
11. Lekkala C, Calheiros RN, Ranjan R (2020) Improving the Performance and Energy Efficiency of Stream Mining Workloads in Heterogeneous Processors. Future Gener Comput Syst 104: 126-140.
12. Meng X, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, et al. (2016) MLlib: Machine Learning in Apache Spark. J Mach Learn Res 17: 1235-1241.
13. Taylor RC (2010) An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. BMC Bioinformatics 11.
14. Lekkala C, Ranjan R, Naha RK, Jayaraman PP, Georgakopoulos P, et al. (2022) Scalable Real-Time Stream Processing Using Distributed Edge Computing and Serverless Architectures. IEEE Trans Parallel Distrib Syst 33: 1955-1968.
15. Lin J, Ryaboy D (2013) Scaling Big Data Mining Infrastructure: The Twitter Experience. SIGKDD Explor News 14: 6-19.