Journal of Theoretical and Applied Information Technology
31st March 2025. Vol.103. No.6

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HOT TOPICS DETECTION AND PERFORMANCE OPTIMIZATION FROM MICROBLOGS USING HYBRID HADOOP FRAMEWORK

Dr. D. DAVID NEELS PONKUMAR[1*], Dr. S. RAMESH[2], Dr. R. KALPANA[3], K. MANIKANDAN[4], N. RAGHAVENDRAN[5], Y. HAROLD ROBINSON[6], Dr. G. UMA MAHESWARI[7], Dr. SANDRA JOHNSON[8], Dr. B. PRATHUSHA LAXMI[9], R. SARAVANAKUMAR[10]

[1*]Professor, Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India
[2]Assistant Professor, Department of Computing Technologies, School of Computing, College of Engineering and Technology, SRMIST, Kattankulathur Campus, Chengalpattu, Tamil Nadu, India
[3]Assistant Professor, Department of Electronics and Communication Engineering, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Avadi, Chennai, Tamil Nadu, India
[4]Department of Computer science and Engineering, P.S.R. Engineering College, Sivakasi, Tamil Nadu, India
[5]Assistant Professor, Department of Artificial Intelligence and Data Science, RMK College of Engineering and Technology, Chennai, Tamil Nadu, India
[6]Professor, Department of Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli, Tamil Nadu, India
[7]Professor, Department of Computer Science and Engineering, RMK College of Engineering and Technology, Chennai, Tamil Nadu, India
[8]Professor, Department of Artificial Intelligence and Data Science, R.M.K. Engineering College, Chennai, Tamil Nadu, India
[9]Professor, Department of Artificial Intelligence and Data Science, R.M.K. College of Engineering and Technology, Chennai, Tamil Nadu, India
[10]Software Analyst, Iconix Software Solution, Tirunelveli, Tamil Nadu, India
E-mail: [1*]david26571@gmail.com, [2]ramesh.isaiah@gmail.com, [3]kalpanar.ece25@gmail.com, [4]manikandan.k@psr.edu.in, [5]ragavendrannv2001@gmail.com, [6]yhrobinphd@gmail.com, [7]ramki_uma21@yahoo.com, [8]sjn.ad@rmkec.ac.in, [9]hod_ads@rmkcet.ac.in, [10]iconixsaro@gmail.com

ABSTRACT

The Hadoop framework's adoption is on the rise. We try to improve Hadoop's performance by incorporating a more sophisticated framework into the MapReduce paradigm, all while keeping the native Hadoop Framework's characteristics intact. The improved Hadoop framework sorts a large collection of microblogs according to the amount of attention they received from the social media site during a certain period 't'. There is a 3% decrease in the execution time of microblogs that have attained the attention level compared to those that have not. By distributing the load evenly, the EHF speeds up the processing of microblogs that have garnered a lot of interest. Global social media users may dynamically develop insoluble information. Social media networks employ big data to manage their massive data. Hadoop-based cloud platform provides large data fault tolerance and dependability. The foundation of big data analytics is Hadoop. The main drawback of Hadoop is processing massive configuration metrics. This paper proposes the Hybrid Hadoop Framework to improve big data processing by balancing workload, response time, network bandwidth, and hot topic detection for microblogs using cloud-based Apache Spark. To accurately find hot topics in large datasets, we purposely build MapReduce tasks. Experimental findings show that the suggested system is more accurate than comparable systems.

Keywords: *Hadoop Framework, Social Media, Mapreduce, Big Data Analytics, Apache Spark, Cloud, Workload, Response Time, Network Bandwidth, Hot Topic Detection*

1. INTRODUCTION

A comparative analysis of large data processing efficiency across various software and hardware platforms is crucial for high-performance computing, aimed at minimizing the load on current machines and informing future platform acquisition strategies. Fourteen prominent marketing SAS campaigns from a Kazakhstani bank were chosen as study samples. This decision facilitates a more accurate evaluation of computer systems' capabilities. Furthermore, using specialist tools, we examined the attributes of the investigated systems of massively parallel architectures, including Greenplum, Netezza, Exadata, and Oracle systems. Big data can address almost all critical functions of banks, including client acquisition, enhancing service quality, evaluating loans, and preventing fraud, among others. Big data technologies assist banks in fulfilling financial regulators' obligations by enhancing the speed and quality of reporting and fostering in-depth research. The data gathered inside the companies is inherently unstructured. Unstructured information is the most rapidly expanding kind of data produced today. Experts believe that 80-90% of organizational data is unstructured. The big data paradigm facilitates the resolution of unstructured data processing challenges. Valuable information is obtained by superimposing the raw data with a framework. This approach enables the interpretation of unstructured data. Hadoop and other systems are used to provide structure using key-value pairs in the absence of inherent organization. HDFS (Hadoop Distributed File System) is a self-repairing, distributed file system that offers dependable, scalable, and fault-tolerant data storage on hardware. It operates in conjunction with MapReduce by allocating storage and processing over extensive clusters to amalgamate storage resources that may expand according to demands and queries. HDFS accommodates data in many formats, including text, photos, and videos. HDFS utilizes a Master/Slave design, with the name node and secondary name node operating on the master node, while Data nodes function on each slave node. The NameNode retains and oversees metadata about the file system. This information is stored in main memory to provide expedited access for clients during read/write operations. The NameNode oversees the segmentation of files into blocks and designates the slave node responsible for storing these blocks. Data nodes are the principal storage components of HDFS that retain data blocks and fulfill read/write requests for files stored inside HDFS. The Secondary NameNode regularly accesses the file system and records the modifications. MapReduce is a programming methodology and software framework used in Hadoop for developing applications that process and analyze large information concurrently. The JobTracker Service operates on the master node and supervises the TaskTracker service on the slave nodes. It obtains input from the user and thereafter inquires the NameNode for the precise placement of the data inside the HDFS. The JobTracker identifies the Task Tracker on slave nodes and submits the jobs. The TaskTracker transmits a heartbeat message to the JobTracker as a form of acknowledgment. The TaskTracker receives tasks from the JobTracker and performs the MapReduce processes. Each TaskTracker has a limited quantity of task slots. The JobTracker is responsible for assigning the correct amount of jobs to the TaskTracker.

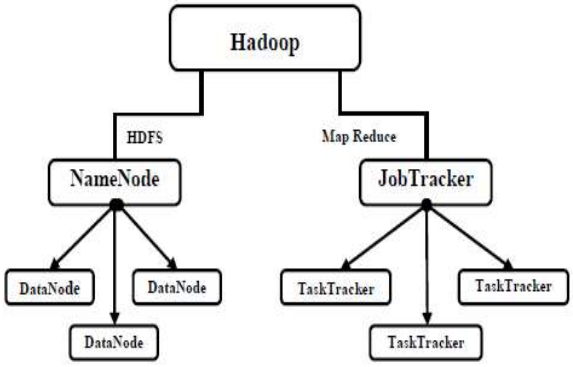

Figure 1: Hadoop Daemons

The HDFS is tailored on the MapReduce paradigm to accommodate big files shown in Figure 1. HDFS divides the large files over numerous partitions and distributes them among hundreds of nodes. To control the partitions, the HDFS central stores the metadata—index information. The fundamental HDFS data piece is these partitions, whose size is 64MB [2]. The overall processing time is dominated by the data movement time in HDFS. Furthermore, included in the MapReduce Model is the need for a TaskTracker to wait for a new task before starting one. The master node should let the slave node know about the necessary chores and data placement. Integration of technology is driving the need for BD to get bigger and more expanding [1,2,3,4, 5]. This is mostly resulting from the development of industry and more intelligent operating systems. It is used in all the communication variations of learning techniques [6], intelligent approaches [7], cyber-physical infrastructures [8-14], new V2G technology systems, photovoltaic interactions, renewable energy

integration, etc. Moreover, BD is rather important for all the information-driven companies and sectors to have that technological competitive advantage. For instance, Cisco Systems has presented a projection about internet data traffic reaching 4.8 zeta-bytes yearly [15]. For current IT security firms, however, the volume of created data as well as its great speed and adaptability in variety have presented some difficulties. Furthermore, a roadblock towards technological interactions has been the shift from traditional data collecting to BD, VMs, and cloud infrastructure [16]. Even open-source developers like Hadoop are struggling to cover BD's security flaws, which are ultimately prone and susceptible to bad actors, hackers, and cybercriminals [17].

The main contributions of the paper

- The Twitter dataset is used to categorize buzz and non-buzz using big data analytics.
- The MapReduce approach is used to optimize workload distribution, and task allocation, and enhance data transmission rates.
- A Hybrid Hadoop Algorithm has been created to differentiate between buzz and non-buzz components inside the Hadoop cluster in the HDFS system.
- The execution time, total process execution duration, and Hadoop cluster creation are assessed in the performance assessment, demonstrating that the suggested technique performs well across all parameters.

2. RELATED WORK

The authors of [18] address cyber forensics and provide a Hadoop analytical framework to prevent polyn time complexity and increase an accuracy and detection ratio. Introduced as a distributed file system based on Hadoop. The restriction of the effort was to evaluate the model in a more dynamic test situation. For cyber security management, the authors of [19] suggested a cloud computing architecture with a data storage and job scheduling module. Additionally included in the suggested architecture are end-user devices and a monitoring center. The restriction of the work is its use towards scalability and different purposes. Towards the incident response process in BD systems, the authors of [20] suggested a blockchain technology-driven solution. The restriction of this work is its confirmation of the optimum parametrization of the method and the suggested solution for many attack situations. The cybersecurity possibilities in smart grids, smart cities, and possibly related solutions are covered by the writers of [21]. They also visit IoT technologies and the blockchain and their participation in these systems. The approach does not concentrate on a specific solution that would handle BD systems' associated cybersecurity issues. The study in [22] suggests a fresh approach based on the attack likelihood score to identify BD system cyber-attacks. Data-flow sacks help the probability score to be executed more quickly. The restriction of the work is an architecture based on this scoring system as the suggested scheme is built on a virtual software-based cluster. The weaknesses in the Apache Hadoop architecture are covered by the writers of [23]. There are several instances of work tackling BD and security concerns of computers seen in the literature. Work addressing the combination of BDC security was lacking, nevertheless. This was a novel path unexplored and may provide directions of answers if gaps were identified along with tests as researchers were looking for methods of addressing security concerns utilizing BD technology. [24-26]

3. PROPOSED WORK

Hadoop provides computing applications to become highly scalable distributed environments. The developer focuses on the dataset and its logic only and no need to worry about processing. The HDFS stores large no of files in many machines this helps in achieving high consistency by information duplication across many hosts and avoiding RAID devices for hosts. The HDFS generated the data nodes within the cluster and data over the network using a chunk framework. Data over HTTP will help the user to allow access to a client that all data nodes are connected to gather to rebalance, copy/move the data, and ensure the replication of data is achieved. If any single node fails it will become a dead node and any new need added will become a live node.

The data volume of social media to be processed by cloud applications is growing much faster than computing power. Optimizing such big data volume is always a challenge for Hadoop's performance. Enhancing the data processing speed has to be concerned more than reducing the latency of data. The workload has to be balanced among the map slots and reduced slots to reduce the network bandwidth. Hadoop performance tuning parameters should be identified for each issue which is a time-consuming task. A Map Reduce program for predicting buzz has to be done in analyzing the collected Twitter dataset. The Hadoop cluster processes large datasets with an efficient throughput that leads to a promising conclusion that Hadoop is the most beneficial framework to analyze and

manage huge social media big data like Twitter, Facebook, LinkedIn, etc. Fig. 1 demonstrates the concept of big data analytics using tweet data. The data acquisition layer is utilized for storing the Twitter data in the related database management system. The data storage and processing layer is utilized to format the unstructured data into structured data using the concept of MapReduce methodology. The HDFS is used to transform the data into a client-oriented format. The web browser and the analysis tool are used for reporting the data in a report format for big data analytics.

Fig. 2 and 3 demonstrates the concept of MapReduce using Task tracer. The client program can submit the job in the job tracker, it is responsible for assigning the work to the map and reducing the phase for finishing the scheduled job. The Task tracker separates the big data into individual regions. Each region is connected with the partitioning and combining of the big data into the allotted Task tracer and produces the output file for the concerned task tracer. The reducer phase again modifies the content of the output file and produces the result.

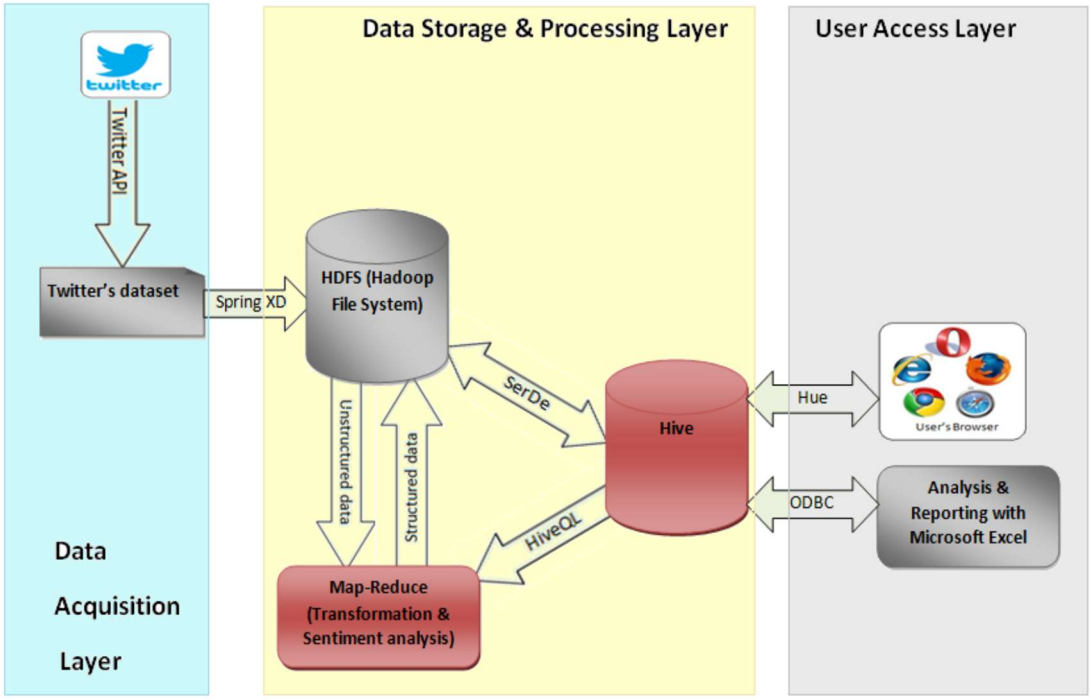

Figure 2: Big data analytics using tweet

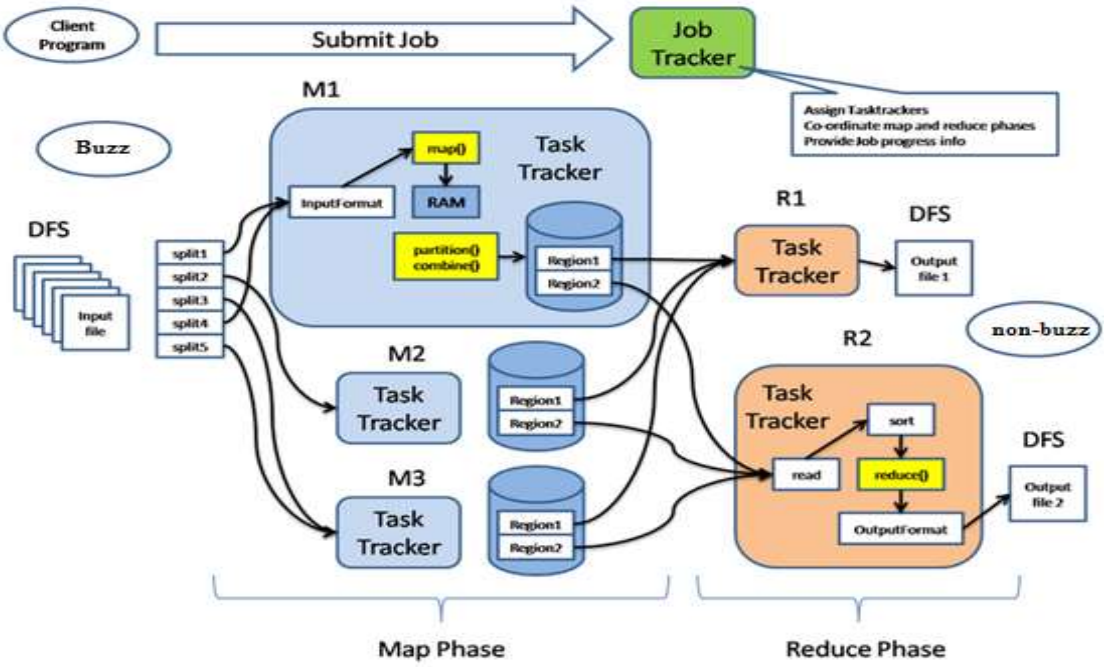

Figure 3: MapReduce concept using Task Tracer

3.1 Formal representation of the multi-dimensional schema

Currently, analysts are extremely relevant in all areas of business. This study looks at how to get current information from the accumulated large volumes of raw information. The use of advanced analytical technologies, the ability to extract the necessary knowledge from big data, integrate them into operational processes, and insert, and convert all this into operational management decisions. The article uses big data, the business analysis of innovations as a competitive advantage, and the construction of a mathematical model for decision-making.

In addition, the problems associated with data management such as collection, storage, structuring, and classification, using Hadoop, and MapReduce methods in the study, it was decided to develop a mathematical model for determining the preferences of customers for the specific choice of a particular company. It should be noted that regardless of their volume and quality, the data is not very useful if they are not retained in such an environment and format which gives them access and the most important thing is to analyze them. Big data does not provide success and progress. To benefit from the data, it is necessary to analyze them and perform some action based on the results of the analysis. Hadoop and MapReduce systems do not automatically interpret data collected from various data sources, so, we attempted to create mathematical models for further analysis and decision-making based on test experimental data. The use of scoring models to provide accelerated and extended analysis makes it possible to quickly make decisions on the choice of the desired product and gives good results for processing large raw data. For the analysis and forecast of the statistical data, it is necessary to construct a mathematical model that reflects the relationship between the four solutions to increase the performance of big data processing.

The linear equation form is generated using a matrix computed in Eq. (1) and Eq. (2).

$$\begin{pmatrix} 1_D & 1_D \\ X_{ND} & X_{ND} \end{pmatrix}\begin{pmatrix} 1_D & X_D \\ 1_{ND} & X_{ND} \end{pmatrix}\begin{pmatrix} w_0 \\ w^T \end{pmatrix} = \begin{pmatrix} 1_D & 1_D \\ X_{ND} & X_{ND} \end{pmatrix}\begin{pmatrix} I_D \\ 0 \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} n & n_D\mu_D + n_{ND}\mu_{ND} \\ n_D\mu_D^T + n_{ND}\mu_{ND}^T & X_D^T X_D + X_{ND}^T X_{ND} \end{pmatrix}\begin{pmatrix} w_0 \\ w^T \end{pmatrix} = \begin{pmatrix} n_D \\ n_D\mu_D^T \end{pmatrix} \quad (2)$$

Where μ_D is the mean variable vector of completed task and μ_{ND} is the mean variable vector of non-completed task. The sample for learning methodology is equivalence to the linear regression and the assumption is computed using Eq. (3).

$$X_D^T X_D + X_{ND}^T X_{ND} = nE[X_i X_j] = Cov(X_i X_j) + n_D\mu_D\mu_D^T + n_{ND}\mu_{ND}\mu_{ND}^T \quad (3)$$

Let the covariance function can be computed using Eq. (4).

$$X_D^T X_D + X_{ND}^T X_{ND} = nC + n_D \mu_D \mu_D^T + n_{ND} \mu_{ND} \mu_{ND}^T \quad (4)$$

To obtain the new Eq. (5) and Eq. (6).

$$nw_0 + (n_D \mu_D + n_{ND} \mu_{ND}) w^T = n_D \quad (5)$$

$$\begin{pmatrix} n_D \mu_D^T \\ n_{ND} \mu_{ND}^T \end{pmatrix} w_0 + \begin{pmatrix} nC + n_D \mu_D \mu_D^T \\ n_{ND} \mu_{ND} \mu_{ND}^T \end{pmatrix} w^T = n_D \mu_D^T \quad (6)$$

The weight factor is computed using Eq. (7).

$$Cw^T = a(\mu_D - \mu_{ND})^T \quad (7)$$

The optimal weight vector is assigned the values using Eq. (8).

$$w = (w_0, w_1, \dots, w_p) \quad (8)$$

The probability value is computed using Eq. (9) and Eq. (10).

$$P(x_i) = G(x_i, w) \quad (9)$$

$$G(x_i, w) = \frac{e^{x_i, w}}{1 + e^{x_i, w}} \quad (10)$$

The probability distribution is computed using Eq. (11).

$$\frac{P(x_i)}{1 - P(x_i)} = w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots . + w_p x_{ip} + e_i \quad (11)$$

The Length of the optimal weight vector is computed using Eq. (12).

$$L(w) = \prod_{i=1}^{n} P(x_i)^{yi} (1 - P(x_i))^{1-yi} \quad (12)$$

The discrete value is computed using Eq. (13).

$$\frac{dl(w)}{dw_j} = \sum_{i=1}^{n} (y_i - p(x_i)) = 0 \quad (13)$$

The back-propagation methodology is used to implement the layers for analyzing the connection-based neural networks is computed using Eq. (14).

$$E(w) = \frac{1}{2} \sum_{j=1}^{p} (y_j - d_j)^2 \quad (14)$$

Where p is the total amount of neurons in the output layer, d_j is the target value of j^{th} output, y_j and is the j^{th} neural network output.

The varying weight coefficient in the iteration is measured using Eq. (15) and Eq. (16).

$$\Delta w_{ij} = -\mu \frac{\partial E}{\partial w_{ij}} \quad (15)$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \text{ x } \frac{\partial y_j}{\partial S_i} \text{ x } \frac{\partial S_i}{\partial w_{ij}} \quad (16)$$

Where μ is the learning speed parameter, S_i is the weighted sum of the input signals and it is computed using Eq. (17).

$$S_j = \sum_{i=1}^{n} w_i x_{ij} \quad (17)$$

The output value is computed using Eq. (18), Eq. (19) and Eq. (20).

$$\frac{\partial S_i}{\partial w_{ij}} = x_i \quad (18)$$

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_j} \text{ x } \frac{\partial y_j}{\partial S_i} \text{ x } \frac{\partial S_i}{\partial y_j} \quad (19)$$

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_j} \text{ x } \frac{\partial y_j}{\partial S_i} \text{ x } w_{jk}^{n+1} \quad (20)$$

The middle layer function is established using Eq. (21).

$$d_j^n = \frac{\partial E}{\partial y_j} \text{ x } \frac{\partial y_j}{\partial S_i} \quad (21)$$

The final layer value is computed using Eq. (22).

$$d_j^n = \left[\sum_k \delta_k^{n+1} x \ w_{jk}^{n+1} \right] \text{ x } \frac{\partial y_j}{\partial S_i} \quad (22)$$

Table 1 demonstrates the notations used in this formation and the meaning of the notations.

Table 1: Notations used

Notation	Meaning
w	Weights
$\in$	least squares value
x_{ij}	matrix value from the input layer to the output layer
x_{iD}	matrix value for derivation
μ_D	Mean variable vector of completed task
μ_{ND}	Mean variable vector of non-completed task
X_D	input value in the derivation
X_{ND}	input value in non-derivation

y	output parameter
X	decision value
p	the total amount of neurons
n	the total amount of vector values
$Cov(X_iX_j)$	covariance function
C	weight factor
$P(x_i)$	probability value
$G(x_i, w)$	optimal weight vector
$L(w)$	Length of the optimal weight vector
$\frac{dl(w)}{dw_j}$	discrete value
$E(w)$	back-propagation methodology
Δw_{ij}	weight coefficient in iteration

d_j	target value of jth output
y_j	jth neural network output
μ	learning speed parameter
S_i	the weighted sum of the input signals

4. PERFORMANCE EVALUATION

The perfecting mechanism will help MapReduce prepare the required data before the task is launched. The settings of the systems of each product are illustrated in Table 2.

Table 2: Specification.

Variables	Oracle current	Netezza	Greenplum	Exadata
Configuration database	24 core Power 7 (3.4 GHz) 160 GB RAM 3000 MB/sec SAN	NZ1000-E (4 SPU-only 24 CPU + FFGA)	4server segment to 16 CPU cores each	XZ-2 Half Rack 4 database nodes to CPU cores storage node 7-12 CPU cores
SAS compute server configuration	Wirth. (VMWare) 12 vCPU-300 MB/sec SAN Storage SAS 9.2	16 physical cores CPU 1 GB/sec DAS storage SAS 9.3	16 physical cores CPU 1 GB/sec DAS storage SAS 9.3	16 physical cores CPU 1 GB/sec DAS storage SAS 9.3
Configuring SAS	Wirth. (VMWare)	Wirth. (VMWare)	Wirth. (VMWare)	Wirth. (VMWare)
Mid-Tier Server	4 vCPU 12 GB RAM	4 vCPU 12 GB RAM	4 vCPU 12 GB RAM	4 vCPU 12 GB RAM
Network Interface SAS-DB	Unknown	10 GB	10 GB	10 GB
Cost (in Million $)	0.5 – 1	2.5 – 3.5	1 – 2.5	4 – 5

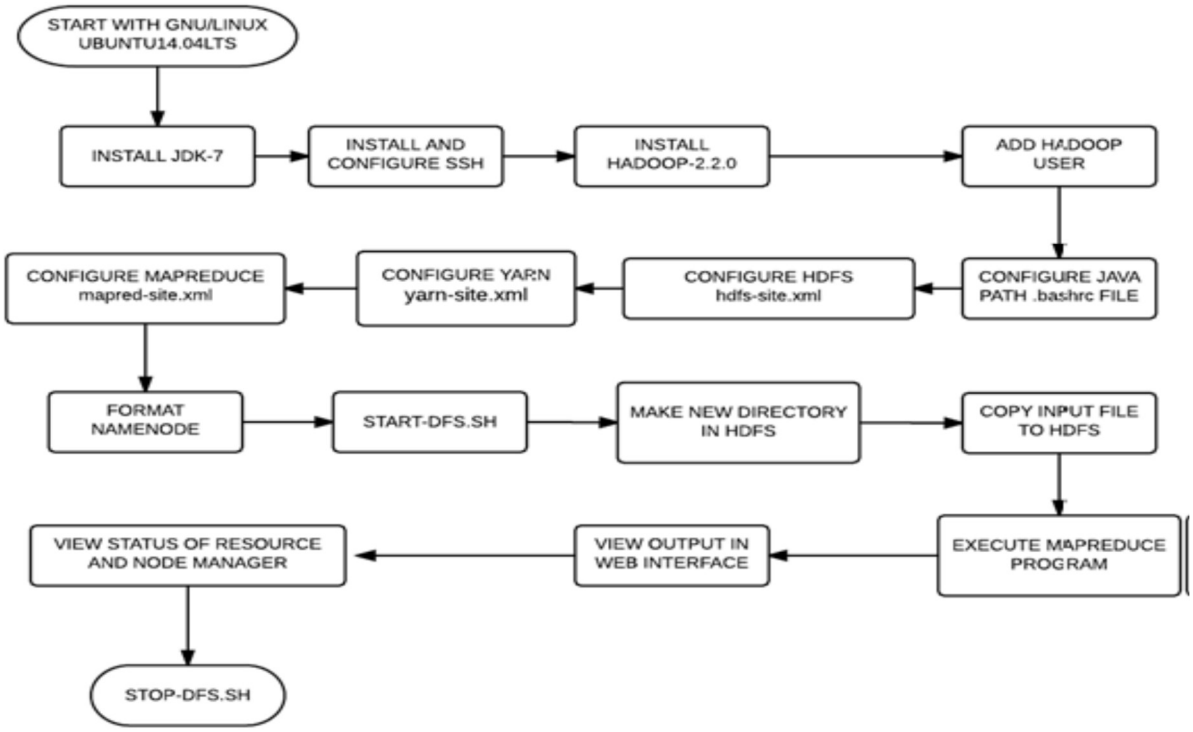

Figure 4: Steps to Execute the Hadoop Job

Figure 4 sets the environment for the Hadoop cluster after the Hadoop environment-ready data is loaded with HDFS and data is split into different blocks. Figure 6 shows the various steps before executing the Hadoop job. It explains the step-by-step process from the installation of Java JDK to configuring the Hadoop node.

```
File System Counters
        FILE: Number of bytes read=1257416
        FILE: Number of bytes written=3067201
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=68187557
        HDFS: Number of bytes written=694558
        HDFS: Number of read operations=21
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=10
Job Counters
        Launched map tasks=2
        Launched reduce tasks=5
        Data-local map tasks=2
        Total time spent by all maps in occupied slots (ms)=37244
        Total time spent by all reduces in occupied slots (ms)=90717
Map-Reduce Framework
        Map input records=140707
        Map output records=140707
        Map output bytes=975972
        Map output materialized bytes=1257446
        Input split bytes=180
        Combine input records=140707
        Combine output records=140707
        Reduce input groups=3026
        Reduce shuffle bytes=1257446
        Reduce input records=140707
        Reduce output records=140707
        Spilled Records=281414
        Shuffled Maps =10
        Failed Shuffles=0
        Merged Map outputs=10
        GC time elapsed (ms)=2359
        CPU time spent (ms)=16760
        Physical memory (bytes) snapshot=1009299456
        Virtual memory (bytes) snapshot=3423612928
        Total committed heap usage (bytes)=754974720
```

Figure 5: The procedure for MapReduce

Figure 5 shows that the Hadoop cluster is ready to perform the tasks and sample I/O processed and the Hadoop cluster status becomes live and the map reduces done successfully.

Cluster Summary

Security is *OFF*
1 files and directories, 0 blocks = 1 total.
Heap Memory used 52.26 MB is 40% of Commited Heap Memory 130.50 MB. Max Heap Memory is 889 MB.
Non Heap Memory used 18.03 MB is 67% of Commited Non Heap Memory 26.75 MB. Max Non Heap Memory is 176 MB.

Configured Capacity	:	9.04 GB
DFS Used	:	24 KB
Non DFS Used	:	5.24 GB
DFS Remaining	:	3.80 GB
DFS Used%	:	0.00%
DFS Remaining%	:	42.00%
Block Pool Used	:	24 KB
Block Pool Used%	:	0.00%

DataNodes usages	:	Min %	Median %	Max %	stdev %
		0.00%	0.00%	0.00%	0.00%

Figure 6: Summary of HDFS cluster

Figure 6 shows the cluster summary and configuration of the node and DFS and Non-DFS used.

Figure 7 shows the summary of file system counts, number of job counters, and map-reduce framework before optimizing.

Figure 8 shows the summary of file system counters, the number of job counters, and a map-reduced framework after optimization.

Figure 9 shows the final output of the Twitter dataset number of buzz and nonbuzz detected by the Hadoop cluster. The number of buzzes is 112932 and non-buzz is 27775, the detected as per the Twitter dataset in an optimal way using a Hadoop cluster environment.

```
File System Counters
        FILE: Number of bytes read=46
        FILE: Number of bytes written=236660
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=68187557
        HDFS: Number of bytes written=21
        HDFS: Number of read operations=9
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
Job Counters
        Launched map tasks=2
        Launched reduce tasks=1
        Data-local map tasks=2
        Total time spent by all maps in occupied slots (ms)=14934
        Total time spent by all reduces in occupied slots (ms)=11217
Map-Reduce Framework
        Map input records=140707
        Map output records=140707
        Map output bytes=1125656
        Map output materialized bytes=52
        Input split bytes=180
        Combine input records=140707
        Combine output records=4
        Reduce input groups=2
        Reduce shuffle bytes=52
        Reduce input records=4
        Reduce output records=2
        Spilled Records=8
        Shuffled Maps =2
        Failed Shuffles=0
        Merged Map outputs=2
        GC time elapsed (ms)=799
        CPU time spent (ms)=5960
        Physical memory (bytes) snapshot=461471744
        Virtual memory (bytes) snapshot=1470328832
        Total committed heap usage (bytes)=315883520
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
File Input Format Counters
        Bytes Read=68187377
File Output Format Counters
        Bytes Written=21
dcser@saraaya-Invalid-entry-length-0-DMI-table-is-broken-Stop:/$ hadoop dfs
EFRECATED: Use of this script to execute hdfs command is deprecatec.
nstead use the hdfs command for it.
```

Figure 7: MapReduce function before applying tuned configuration

```
        File System Counters
                FILE: Number of bytes read=46
                FILE: Number of bytes written=236678
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=68187557
                HDFS: Number of bytes written=21
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=32824
                Total time spent by all reduces in occupied slots (ms)=113
        Map-Reduce Framework
                Map input records=140707
                Map output records=140707
                Map output bytes=1125656
                Map output materialized bytes=52
                Input split bytes=180
                Combine input records=140707
                Combine output records=4
                Reduce input groups=2
                Reduce shuffle bytes=52
                Reduce input records=4
                Reduce output records=2
                Spilled Records=8
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=1459
                CPU time spent (ms)=6940
                Physical memory (bytes) snapshot=536293376
                Virtual memory (bytes) snapshot=1473015808
                Total committed heap usage (bytes)=404488192
        Shuffle Errors
                Total committed heap usage (bytes)=470024192
         Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
         File Input Format Counters
                Bytes Read=68187377
         File Output Format Counters
                Bytes Written=21
hduser@saranya-Invalid-entry-length-0-DMI-table-is-broken-Stop:
DEPRECATED: Use of this script to execute hdfs command is depre
Instead use the hdfs command for it.
```

Figure 8: MapReduce function after applying tuned configuration

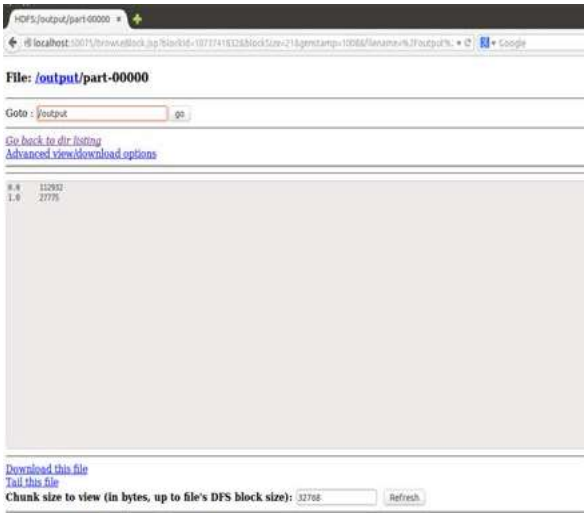

Figure 9: Hadoop output with buzz and non-buzz

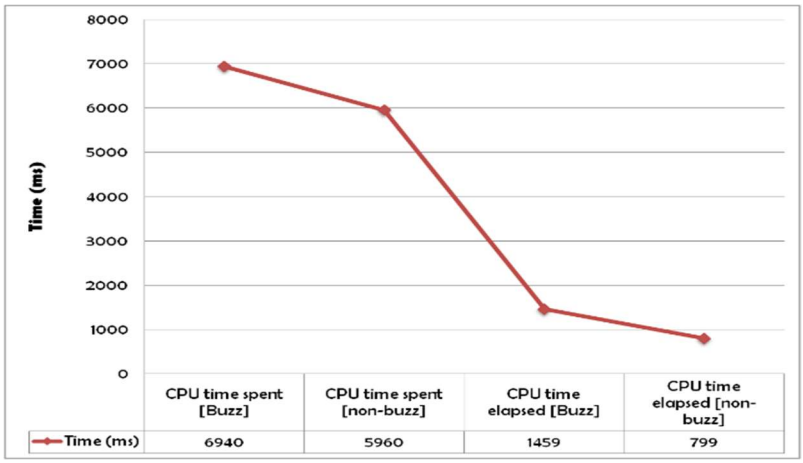

Figure 10: Overall process execution time for MapReduce

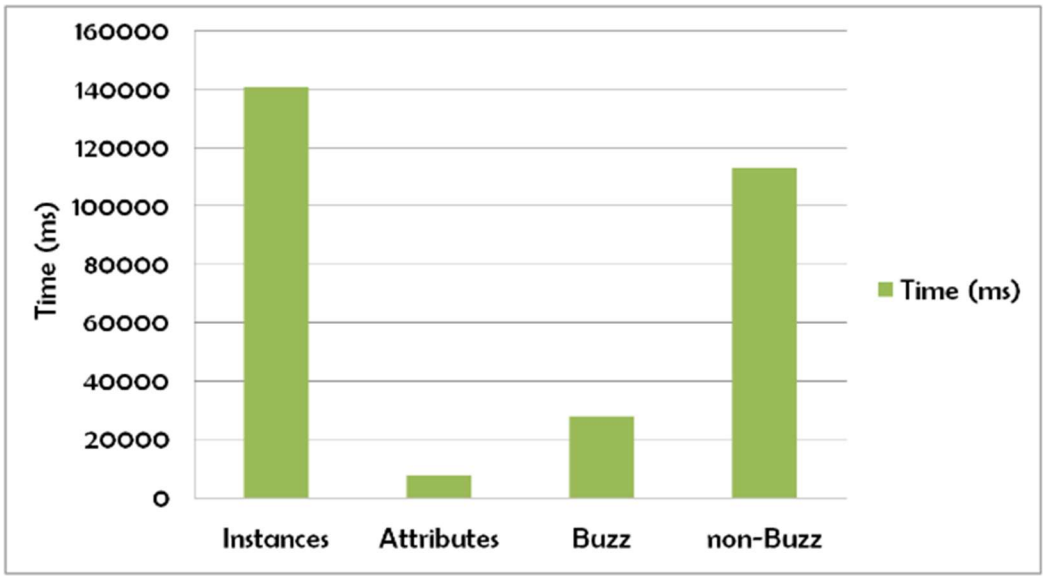

Figure 11: Hadoop cluster result

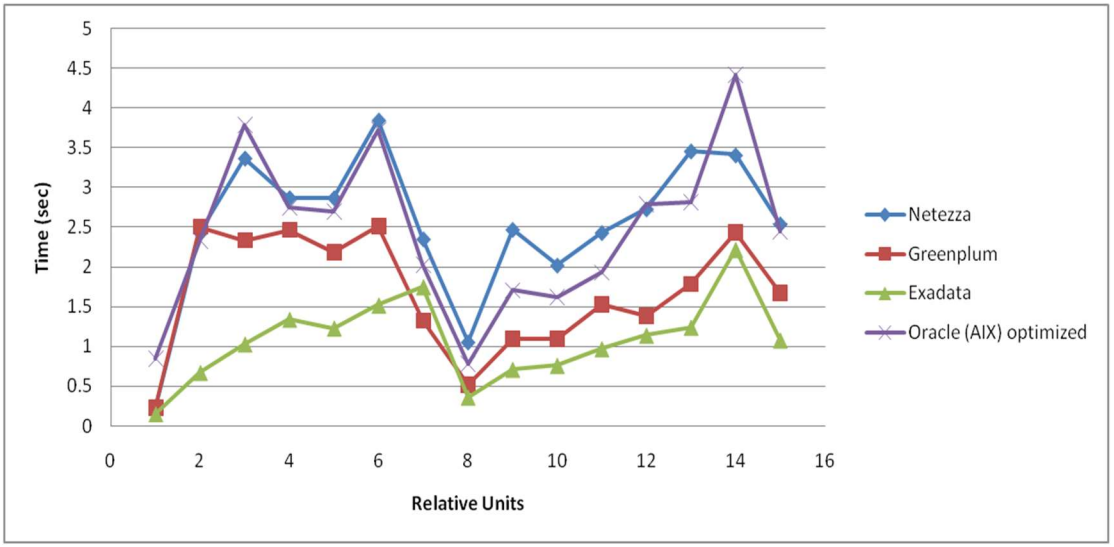

Figure 12: Relative Units

Figure 10 shows the CPU time spent and CPU time elapsed for processing the dataset which has 6940 and 1459 ms, respectively in dark blue color bars represent. The light color bars represent the CPU time spent and CPU time elapsed for processing the dataset which has 5960 and 799 ms, respectively.

Figure 11 shows the total number of instances, attributes, and buzz predicted and non-predicted from the Twitter dataset. The MapReduce output, this classification has been made Positives instances (Buzz): 27775 (19 %) negative instances (Non Buzz): 112932 (81 %).

The performance test was designed to measure the maximum performance with a given set as the campaign of circumstances and settings. Measurement of the duration of SAS campaigns was conducted before the experiment and is illustrated in Fig. 12.

5. CONCLUSION

While big data is engaged to preserve the vast information and keep fault tolerance and dependability, the Social Media Network might develop the semantic data in a dynamic method. With the main KPIs including the cloud, response time, and bandwidth for hot subject identification, the hybrid Hadoop Framework has been used to

improve data processing. While the MapReduce with the Hadoop setup, the performance enhancement of the Hadoop framework might be useful in Hadoop-related clusters from the Twitter dataset. The testing findings showed that the suggested method improves the whole performance measures employing multi-node clusters.

REFERENCES:

[1] F. Han and J. Ren, "Analyzing Big Data Professionals: Cultivating holistic skills through university education and market demands," *IEEE Access*, Vol. 12, 2024, pp. 23568–23577.

[2] S. Ahmadi, "A Comprehensive Study on Integration of Big Data and AI in Financial Industry and its Effect on Present and Future Opportunities," *International Journal of Current Science Research and Review*, Vol. 07, No. 01, 2024.

[3] H. Kamyab, T. Khademi, S. Chelliapan, M. SaberiKamarposhti, S. Rezania, M. Yusuf, M. Farajnezhad, M. Abbas, B. H. Jeon, and Y. Ahn, "The latest innovative avenues for the utilization of artificial Intelligence and big data analytics in water resource management," *Results in Engineering*, Vol. 20, 2023, p. 101566.

[4] C. Acciarini, F. Cappa, P. Boccardelli, and R. Oriani, "How can organizations leverage big data to innovate their business models? A systematic literature review," *Technovation*, Vol. 123, 2023, p. 102713.

[5] Q. Gao, C. Cheng, and G. Sun, "Big data application, factor allocation, and green innovation in Chinese manufacturing enterprises," *Technological Forecasting and Social Change*, Vol. 192, 2023, p. 122567.

[6] U. Inayat, M. F. Zia, S. Mahmood, H. M. Khalid, and M. Benbouzid, "Learning-Based Methods for Cyber Attacks Detection in IoT Systems: A survey on methods, analysis, and future prospects," *Electronics*, Vol. 11, No. 9, 2022, p. 1502.

[7] Z. Said, P. Sharma, Q. T. B. Nhuong, B. J. Bora, E. Lichtfouse, H. M. Khalid, R. Luque, X. P. Nguyen, and A. T. Hoang, "Intelligent approaches for sustainable management and valorisation of food waste," *Bioresource Technology*, Vol. 377, 2023, p. 128952.

[8] Z. Said, P. Sharma, Q. T. B. Nhuong, B. J. Bora, E. Lichtfouse, H. M. Khalid, R. Luque, X. P. Nguyen, and A. T. Hoang, "Intelligent approaches for sustainable management and valorisation of food waste," *Bioresource Technology*, Vol. 377, 2023, p. 128952.

[9] H. M. Khalid et al., "WAMS Operations in Power Grids: A Track Fusion-Based Mixture Density Estimation-Driven Grid Resilient Approach Toward Cyberattacks," *in IEEE Systems Journal*, Vol. 17, No. 3, 2023, pp. 3950-3961.

[10] H. M. Khalid, F. Flitti, M. S. Mahmoud, M. M. Hamdan, S. M. Muyeen, and Z. Y. Dong, "Wide area monitoring system operations in modern power grids: A median regression function-based state estimation approach towards cyber attacks," *Sustainable Energy Grids and Networks*, Vol. 34, 2023, p. 101009.

[11] A. Yazdinejad, A. Dehghantanha, H. Karimipour, G. Srivastava, and R. M. Parizi, "A robust Privacy-Preserving federated learning model against model poisoning attacks," *IEEE Transactions on Information Forensics and Security*, Vol. 19, 2024, pp. 6693–6708.

[12] J. Sakhnini, H. Karimipour, A. Dehghantanha, A. Yazdinejad, T. R. Gadekallu, N. Victor, and A. Islam, "A generalizable deep neural network method for detecting attacks in industrial Cyber-Physical systems," *IEEE Systems Journal*, 2023, pp. 1–9.

[13] A. Yazdinejad, A. Dehghantanha, G. Srivastava, H. Karimipour, and R. M. Parizi, "Hybrid privacy Preserving federated learning against irregular users in Next-Generation Internet of Things," *Journal of Systems Architecture*, Vol. 148, 2024, p. 103088.

[14] A. Yazdinejad, A. Dehghantanha, and G. Srivastava, "AP2FL: Auditable Privacy-Preserving Federated Learning Framework for Electronics in Healthcare," *IEEE Transactions on Consumer Electronics*, Vol. 70, No. 1, 2023, pp. 2527–2535.

[15] K. Compton, "Cisco's Global Cloud Index Study: Acceleration of the Multicloud Era," *Cisco Blogs*, 16-Feb-2018. [Online]. Available: https://blogs.cisco.com/news/acceleration-of-multicloud-era.

[16] "Top 10 big data security and privacy challenges report released," 2013-06-18, *Security Magazine*, 18-Jun-2013. [Online]. Available: https://www.securitymagazine.com/articles/84461-top-10-big-data-security-and-privacy-challenges-report-released

[17] "Hadoop Wiki." [Online]. Available: https://www.projectpro.io/hadoop-wiki.

[18] G. S. Chhabra, V. Singh, and M. Singh, "Hadoop-based analytic framework for cyber

forensics," *International Journal of Communication Systems*, Vol. 31, No. 15, 2018.

[19] G. Xu, W. Yu, Z. Chen, H. Zhang, P. Moulema, X. Fu, and C. Lu, "A cloud computing based system for cyber security management," *International Journal of Parallel Emergent and Distributed Systems*, Vol. 30, No. 1, 2014, pp. 29–45.

[20] J. Moreno, M. A. Serrano, E. B. Fernandez, and E. Fernández-Medina, "Improving incident response in big data ecosystems by using blockchain technologies," *Applied Sciences*, Vol. 10, No. 2, 2020, p. 724.

[21] S. Sadik, M. Ahmed, L. F. Sikos, and A. K. M. N. Islam, "Toward a sustainable cybersecurity ecosystem," *Computers*, Vol. 9, No. 3, 2020, p. 74.

[22] S. Aditham and N. Ranganathan, "A novel framework for mitigating insider attacks in big data systems," *2021 IEEE International Conference on Big Data (Big Data)*, 2015, pp. 1876–1885.

[23] A. Kaushik, and V. K. Srivastava, "Threat to big data: Common weakness enumerations and vulnerabilities for Hadoop framework," *International Journal of Research and Analytical Reviews*, Vol. 7, 2020, pp. 280–286.

[24] A. A. Cardenas, P. K. Manadhata, and S. P. Rajan, "Big data Analytics for security," *IEEE Security & Privacy*, Vol. 11, No. 6, 2013, pp. 74–76.

[25] Y. Fernando, R. R. M. Chidambaram, and I. S. Wahyuni-Td, "The impact of Big Data analytics and data security practices on service supply chain performance," *Benchmarking an International Journal*, Vol. 25, No. 9, 2018, pp. 4009–4034.

[26] Abhijit, "What is Apache Ambari?," Intellipaat, 27-Nov-2024. [Online]. Available: https://intellipaat.com/blog/what-is-apache-ambari/?US.