

# Synchronized transform-aggregate model for big data analytics towards in distributed cloud ecosystem

Rajeshwari Dembala<sup>1</sup>, Kavya Ananthapadmanabha<sup>2</sup>, Shashank Dhananjaya<sup>1</sup>

<sup>1</sup>Department of Information Science and Engineering, The National Institute of Engineering, Mysuru, India

<sup>2</sup>Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysuru, India

## Article Info

### Article history:

Received Oct 2, 2024

Revised Mar 29, 2025

Accepted May 24, 2025

### Keywords:

Big data

Cloud

Internet-of-things

Optimality

Task

## ABSTRACT

The massively generated data from various technologically advanced applications hosted in the cloud and internet of things (IoT) in present times calls for effective management towards balancing the demands of both service providers and users. The conventional usage of distributed frameworks for such big data management is witnessed with various ongoing challenges. Hence, this manuscript presents a novel analytical framework for big data that can offer reduced cost and reduced time demanded to evaluate the distributed big data from multiple data points in the cloud in an optimal way. The core ideology of this framework is to gain a synchronized optimality for cost and time for executing a task demanded for big data analytics complying with the constraints associated with task deadline. The proposed framework is capable of fine-tuning the positioning of task operation using transform and aggregate strategy to exhibit 37% reduced delay, 41% efficient task completion performance, and 28% reduced execution time in contrast to existing frameworks.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Rajeshwari Dembala

Department of Information Science and Engineering, The National Institute of Engineering

Manandavadi Road, Vidyaranya, Mysuru, Karnataka 570008, India

Email: drrajeshwari@nie.ac.in

## 1. INTRODUCTION

The vast creation of data from internet of things (IoT) and cloud applications has resulted in large data volumes, necessitating the use of edge computing to process some data before sending it to the cloud for analysis [1]. Streaming analytics are used to manage real-time data on the cloud. This data contributes to the development of smart cities, healthcare, and manufacturing, as well as the big data idea, which involves the ingestion of data via stream and batch processing [2]–[4]. Data lakes and relational databases contain massive amounts of data for advanced processing and analysis. Big data analytics uses a variety of methodologies, including descriptive, diagnostic, predictive, and prescriptive analysis, as well as text, graph, and spatial analysis [5]–[10]. Big data is commonly processed using distributed frameworks like Hadoop and MapReduce. Hadoop is cost-effective, interacts well with ecosystems, and provides fault tolerance, however it suffers from latency and resource dependency. MapReduce provides efficient parallel computation, scalability, and fault tolerance; however, it suffers from latency caused by batch processing and difficulties in multitenant cloud systems. Despite these issues, both frameworks are frequently used for big data analytics, with efforts underway to increase scalability and fault tolerance. Distributed frameworks increase data processing speed through parallel processing, but also entail difficulties in configuration, maintenance, and monitoring, potentially leading to higher resource utilization and delay [11]–[15].

The related work has been studied to extract more insights into exiting frameworks used for managing big data analytical applications in cloud. The work presented by Hussain *et al.* [16] have used Apache Hadoop towards improving the better form of analysis on medical dataset. Adoption of Hadoop as well as MapReduce was noted in work presented by Kumar *et al.* [17] emphasizing on addressing issues related to resource allocation to accomplish reduced processing time. MapReduce has also been considered towards finetuning dynamic task as noted in work of Huang *et al.* [18] for addressing the degraded performance of existing cloud platforms. Azeroual and Nikiforova [19] have used Apache Spark along with machine learning libraries to secure the data. Deployment of Apache Flink is witnessed in model developed by Rank *et al.* [20] in order to improve the efficiency of task and processor while processing streams of incoming data. Liu *et al.* [21] have used Apache Beam in order to develop a parallel computing model for supporting analysis of big data. The model has also used particle swarm optimization for improving upon the operation of MapReduce for controlling the dimensional complexity. Investigation towards Apache Tez, which is an alternative MapReduce platform operating on fabric of Hadoop, has been carried out by Park *et al.* [22] considering a use-case of monitoring data lake. Bartolini and Patella [23] have investigated usage of Apache Samza leading to development of a middleware module for analyzing heavier files like multimedia. Muchenje and Seppänen [24] have presented discussion about the suitability of big data analytics toward varied business scores using a matrix-based modelling towards explaining the interaction of task. Jing and Dan [25] have presented a centralized scheduler design that is claimed to perform high-end data transmission along with placement of task.

After reviewing the existing related work, following *research issues* have been identified viz. i) it is noted that MapReduce is one of the frequently adopted approach while it has still an open-ended issues related to uniform performance of different types of hardware in extensive distributed cloud environment and IoT, ii) the mechanism of scheduling the task during analytical operation is also controlled by variable resource cost, which has not been analyzed in existing system, iii) existing studies with open-source distributed framework is found not to consider influence of differences in resources on multiple position of data points, and iv) Existing scheduler design is witnessed to evaluate time towards task execution in order to cater up the incoming request to the optimal server without any consideration of constraints associated with this scheduling. Hence, it is necessary to consider variability of regions of data points in order to improve the relaying of big data analytical services and its availability in a much cost-effective manner.

The above-mentioned identified research problem is addressed in proposed system by a novel analytical model that *aims* for leveraging big data analytical services and availability to user in cost-effective manner on distributed cloud ecosystem. The value-added contribution of the study are as follows: i) the proposed system introduces a new analytical framework which works on basis of transform and aggregate considering variability attributes of both cost and resources, ii) a novel minimization algorithm has been presented to control the cost and time associated to comply with the defined deadline over distributed cloud platforms, and iii) to mechanize a proper synchronization between time and cost in order to offer consistent performance delivery while performing data analytical operation.

## 2. METHOD

The prime purpose of the proposed study is to present a simplified and yet novel form of analytical framework that can leverage the big data analytical task to be hosted on multiple cloud regions. The prime idea is to offer faster task execution and reduced cost of data transfer toward streams of big data. The architecture of proposed method is shown in Figure 1. According to Figure 1, the incoming stream of big data is subjected to a software framework performing transform and aggregate task. During transform stage, the input stream is processed resulting in an intermediate information where the incoming big data stream is divided into smaller packets and are subjected to parallel processing. Obtained data are the rearranged considering intermediate data and input data. During aggregate stage, all these processed rearranged data generated final outcome. The first part of the architecture is related to computation of cost associated with forwarding the data during data analytical operation followed by computation of completion time considering their respective involved duration. The proposed scheme also introduces algorithm towards minimizing cost, time, and optimizes the effective cost synchronization between cost and time attributes. This architecture contributes to ensure higher degree of availability of an enriched quality analyzed data which can be accessed with reduced delay, higher throughput, minimal resource consumption, and faster accomplishment of analytical task computation considering multi-stages of transform and aggregate operation introduced in the framework. Apart from this, architecture also offers a fault tolerance performance where none of the ongoing analytical processing will be ever affected in case of any abnormal circumstance in cloud environment.

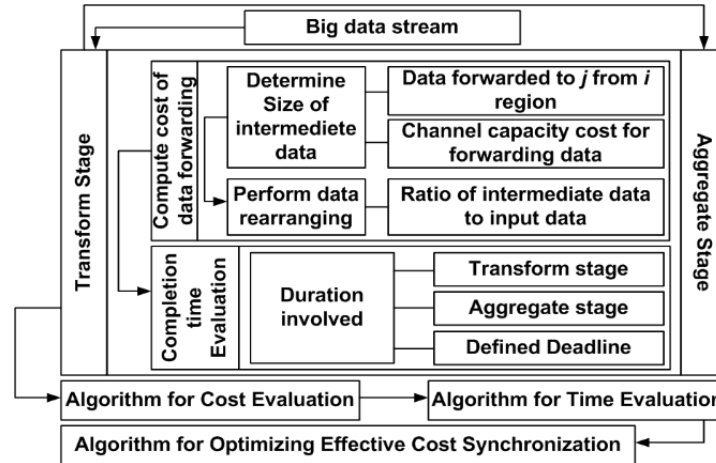


Figure 1. Adopted research method

## 2.1. System implementation

The proposed scheme presents an unconventional data analytical process in cloud where a distinct form of positioning of requested task is carried out. The prime ideology is to determine the number of transformed task and aggregate task required to be positioned at each location of cloud environment thereby assisting towards reading the data by the assigned task in effective way. The proposed scheme implements a cost-based computation that targets to optimize cost and duration of completion of task. The discussion of system implementation is as follows:

### 2.1.1. Implemented optimization principle

The proposed system implements two discrete algorithms which are responsible for computing cost and reduce the overall cost associated with completion of ongoing task for a defined set of deadlines. The first algorithm targets towards evaluation of cost which is also extended towards time evaluation along with optimal performance. The second algorithm further optimizes the overall performance by addressing the possible tradeoff between them on dynamic environment. The algorithms are discussed as follows:

#### Algorithm for cost evaluation

Input:  $G_l, idp, d_{tr}, d_{agg}, \alpha_{tr}, \alpha_{agg}$

Output:  $\beta$

Start

1.  $\gamma = f_1(G_l, d_{tr}, \alpha_{tr})$

2.  $\psi = f_2(\gamma, idp)$

3.  $\pi = f_3(\psi, d_{agg}, \alpha_{agg})$

4.  $(\beta_{1c}, \beta_{2c}) = f_4(\gamma, \pi)$

End

The discussion of algorithmic step are as follows: The algorithm takes the input of  $G_l$  (generated data at  $l$  location),  $idp$  (instantaneous data proportion),  $d_{tr}$  (duration of transform),  $d_{agg}$  (duration of aggregate),  $\alpha_{tr}$  (data for transform), and  $\alpha_{agg}$  (data for aggregate) that upon processing yields  $\beta$  (cost metric). The algorithm initially implements a function  $f_1(x)$  considering the input argument of  $G_l$ ,  $d_{tr}$ , and  $\alpha_{tr}$  for reducing the cost of transform operation while the transformed data is stored in  $\gamma$  (Line-1). Further, another explicit function  $f_2(x)$  is developed that can obtain all the processing intermediate information with respect to  $\gamma$  and  $idp$  (Line-2), while the outcome is restored within  $\psi$  buffer in cloud. It should be noted that variable  $idp$  represents proportion of all processed intermediate data divided by all input data. Further, another function  $f_3(x)$  is implemented in order to minimize the cost of aggregate operation with respect to newly obtained  $\psi$  and  $d_{agg}$ ,  $\alpha_{agg}$  while the data is stored in  $\pi$  matrix (Line-3). It will mean that during the transform operation, the algorithm obtains all segment of data pertaining to aggregate task  $\pi$  that are assigned to each region for accessibility. Finally, another function  $f_4(x)$  is implemented which is meant to obtain information of time and cost associated with  $\gamma$  and  $\pi$  which is retained in  $\beta_{1c}$  matrix for time and  $\beta_{2c}$  matrix for cost respectively (Line-4). The variable  $\gamma$  and  $\pi$  represents data at specific region with respect to cost and time respectively. It should be noted that exactly same algorithmic structure can be also used for minimizing the time attribute too.

**Algorithm for time evaluation**Input:  $G_l, idp, d_{tr}, d_{agg}, \alpha_{tr}, \alpha_{agg}$ Output:  $\beta$ 

Start

1.  $\gamma_1 = f_{1t}(G_l, d_{tr}, \alpha_{tr})$
2.  $\psi = f_2(\gamma_1, idp)$
3.  $\pi_2 = f_{3t}(\psi, d_{agg}, \alpha_{agg})$
4.  $(\beta_{3c}, \beta_{4c}) = f_4(\gamma_1, \pi_2)$

End

It can be seen from above algorithmic steps that all the lines of execution are exactly similar to that of prior one only with few differences in usage of variable: function  $f_{1t}(x)$  is used for obtaining reduced transformed time which is then stored in new matrix  $\gamma_1$  representing data (Line-1). Further, function  $f_{3t}(x)$  is meant for reducing aggregate time while the outcome is stored in matrix  $\pi_2$  (Line-3). Hence first algorithm is meant for cost computation while second algorithm is meant for time calculation. The objective function developed for further optimization is represented as (1), (2):

$$\psi_1 = \arg_{\min}(\chi_{tf} + \chi_{tr}) \quad (1)$$

$$\psi_1 = \arg_{\min}(\chi_{ra} + \chi_{agg}) \quad (2)$$

According (1), the proposed system computes minimal cost  $\psi_1$  with respect to incoming traffic load ( $\chi_{tf}$ ) and transform operation ( $\chi_{tr}$ ) in such a way that generated data at  $l$  location  $G_l$  is equivalent to summation of  $\gamma_f$  quantity of traffic data that are forwarded to destination location from specific data location in cloud *i.e.*,  $G_l = \Sigma \gamma_f$ . The equation (2) states that system targets to minimize the cumulative cost  $\psi_2$  computed by summation of cost towards rearranging the data ( $\chi_{ra}$ ) and cost towards aggregate task ( $\chi_{agg}$ ) such that summation of  $\pi$  *i.e.*, segment of aggregate task carried out at destination location is equivalent to unity *i.e.*,  $\Sigma \pi = 1$ . It can be noted that above two algorithms can successfully optimize both cost as well as time using its transform-aggregate framework to carry out data analytical operation with a sole motive of controlling the cost of task completion associated to highly distributed cloud analytical network with a defined constraint.

**2.1.2. Optimization towards effective cost synchronization**

The first and second algorithm is not meant to perform synchronized task with each other. Therefore, proposed system implements third algorithm which is meant towards accomplishing a better form of synchronization between task cumulative cost and completion time. According to this algorithm, the cumulative cost is optimized by provoking the prior algorithms towards computing reduced cost operation for selecting a cost-effective wireless channel in order to forward the data as well as in order to perform cheaper cost completion with respect to each slot. The system attempts to optimize the time for completion of task by forwarding data to this wireless channel with increased channel capacity in order to ensure reduced time involvement while sufficient resources are utilized to perform computation of task on these regions. The algorithmic operation carried out of this purpose is as shown below:

**Algorithm for optimizing effective cost synchronization**Input:  $G_l, idp, d_{tr}, d_{agg}, \alpha_{agg}$ Output:  $\gamma_f, \pi_f$ 

Start

1.  $Mat_1 = \arg_{\min} dur()$
2.  $Mat_2 = \arg_{\min} cos()$
3. If  $\beta_{3t} > \beta$
4. break();
5. If  $\beta_{1c} < \beta$
6. compute ( $\gamma, \pi$ )
7. While  $\beta_n > \beta$
8.  $\gamma_n, \pi_{2n} = f_5(\lambda) \gamma_n, \pi_{2n} = f_5(\lambda)$
9.  $\beta_n = f_6(\gamma_n, \pi_{2n})$
10.  $\gamma_f = \gamma_n, \pi_f = \pi_{2n}$

End

According to the above-mentioned operational steps, the algorithm takes the input argument of  $G_l$  (generated data at  $l$  location),  $idp$  (instantaneous data proportion),  $d_{tr}$  (duration of transform),  $d_{agg}$  (duration of aggregate), and  $\alpha_{agg}$  (data for aggregate) in order to generate an outcome of  $\gamma_f$  (quantity of information forwarded to destination region in transform stage from specific location),  $\pi_f$  (segment of aggregate task

operated at destination region). The algorithm initially executes a method  $\arg_{\min}dur()$  in order to minimize the time followed by constructing a matrix  $Mat_1$  which retains  $\gamma_1$  (quantity of information forwarded to destination region in transform stage from specific location with respect to computed time),  $\pi_2$  (segment of aggregate task operated at destination region with respect to computed time),  $\beta_{3t}$  (task deadline with respect to computed time), and  $\beta_{4t}$  (cost of slot with respect to time) (Line-1). Similar computation is carried out by introducing method  $\arg_{\min}cos()$  towards reducing cost where cost-based attributes (similar to that of prior step is considered) are stored in matrix  $Mat_2$  (Line-2). If the computed deadline of task  $\beta_{3t}$  is found to be less than cut-off deadline  $\beta$  (Line-5), the operation break as it fails to identify possible solution (Line-4). The algorithm further checks if deadline of task with respect to cost  $\beta_{1c}$  is found to be more than cut-off deadline  $\beta$  (Line-5), the algorithm returns the computed value of  $\gamma$  (quantity of information forwarded to destination region in transform stage from specific location with respect to computed cost) and  $\pi$  (segment of aggregate task operated at destination region with respect to computed cost) (Line-6). In the consecutive step, the algorithm constructs three buffers  $\gamma_n$ ,  $\pi_{2n}$ , and  $\beta_n$  which are assigned with newly computed values of  $\gamma$ ,  $\pi$ , and  $\beta_{1c}$  respectively. The algorithm consecutive checks if value of new deadline  $\beta_n$  is found more than cut-off deadline  $\beta$  (Line-7), then a method  $f_5(x)$  is executed that performs finetuning of  $\lambda$ , a finetuning coefficient configured with specific value ( $\lambda=0.001$ ) while the outcome is positioned within  $\gamma_n$  and  $\pi_{2n}$  (Line-8). Similarly, another method  $f_6(x)$  is used for obtaining time information with respect to  $\gamma_n$  and  $\pi_{2n}$  while the outcome is substituted in new deadline attribute  $\beta_n$  (Line-9). The final score of  $\gamma_f$  and  $\pi_f$  is finally obtained from extracting the updated values of  $\gamma_n$  and  $\pi_{2n}$  respectively (Line-10). Further, the implementation of the finetuning coefficient (FTC) using function  $f_5(x)$  is empirically carried out as (3):

$$FTC = \gamma_n - A1.A_2 \quad (3)$$

In the expression (3), it can be noted that proposed system used FTC computation based on three attributes i.e.,  $\gamma_n$ ,  $A_1$ , and  $A_2$ . The attribute  $A_1$  is equivalent to difference of  $\gamma$  and  $\gamma_1$  i.e.,  $A_1=(\gamma-\gamma_1)$  while the attribute  $A_2$  is computed as a product of finetuning coefficient  $\lambda$  with simulation round  $h$  i.e.,  $A_2=(\lambda.h)$ . However, this expression (3) is only valid if  $\gamma > \gamma_1$ . However, for vice-versa condition (i.e.,  $\gamma < \gamma_1$ ), following is the formulation of FTC as shown in expression (4),

$$FTC = \gamma_n + A1.A_2 \quad (4)$$

A closer look into the third algorithm showcases its capability to optimize the cumulative cost associated with meeting the deadline for all distributed analytical task on cloud considering time and cost. However, a real-time analytical task of big data conventionally consists of different stages of dependencies. Furthermore, there is a need to perform aggregation of all the ultimate outcomes of such task from all locations of cloud as the data is stored disperse form in multiple cloud storage units. This calls for more extensive cost as well as more consumption of duration in order to meet the deadline of task completion. This research challenge is addressed by effectively planning for functioning the jobs in multistage form. A careful observation of proposed algorithm also exhibits that there is a reduction in information size as the completion of task progresses. Hence, the proposed algorithm is further more finetuned so that it can accomplishes overall optimized performance towards the completion of data analytical task of big data over cloud environment in distributed manner.

For this purpose, the proposed scheme initially evaluates the plan towards positioning the task related to transform and aggregate on all the region of analytical data points in cloud. In case of execution of analytical task in stage wise, the input arguments are generated by the completed jobs being executed in prior stage while the size of generated data keeps on minimizing with each progressive execution. It will eventually mean the processed and analyzed data size is very much minimized in size in contrast to actual source data. Hence, a single region of data point is opted by proposed algorithm where all the generated data is received followed by planned execution (of all algorithms) in every upcoming operational stages. This significantly curtails extensive cost and time towards analytical data forwarding. Hence, the success factor of this algorithm completely depends upon undertaking an evaluation to determine the cardinality of operational stages involved in task execution on multiple regions of data points in cloud. The proposed scheme computes the reduced cost associated with task execution at multiple data point regions while it ends up choosing the task which is characterized by minimal cost. Hence following are the steps of execution towards finetuning the algorithm:

- The algorithm initiates by obtaining number of stages involved in computation. Initially, the complete stages of the ongoing task are computed followed by computation of strategy towards positioning the task as well as evaluating the associated cost for all the stages.

- Once the complete number of stages are found, the algorithm selects specific  $h$  number of simulation rounds for computing the task featured with minimal cost. It will mean that out of all the tasks, the algorithm only selects tasks with reduced cost sampled for  $h^{\text{th}}$  number of iterations towards multiple data points in cloud.
- After the minimal cost is determined in  $h^{\text{th}}$  rounds, the algorithm identifies the best strategy to position its task.
- The proposed algorithm determines the number of stages associated with  $(h-1)$  distributed data points in cloud. It is possible that the number of data recorded in each stage keeps on reducing in contrast to size of source data.
- The first and second algorithm towards minimal cost and time consumption is further computed with the updated values of data while similar conditional logic associated to comparing variable ( $\beta^{\text{st}}$  and  $\beta_n$ ) with cut-off ( $\beta$ ) is carried out to obtain complied outcomes.

The execution of the algorithm is resumed till it meets the finally reduced cost in order to represent its accomplished state of converging. Hence, the proposed scheme offers a very simplified and yet quite sophisticated operation towards leveraging analytical operation of big data associated with distributed environment of cloud. The next section presents discussion of the outcomes accomplished from implementation of proposed study model.

### 3. RESULT

The proposed system presents a novel framework towards leveraging big data analytical operation and hence it demands to be assessed over a planned environment mapping with near-real world scenario. Apart from this, owing to the novelty of newly introduced features associated with analytical data positioning over multiple distributed data points, the proposed model is required to be subjected to comparative analysis with similar form of conventional framework. Further, a standard set of performance metric with universal adoption is chosen to evaluate the effectiveness of present model. The following are more elaborate highlights of the adopted strategy of assessment and accomplished study outcomes.

#### 3.1. Assessment strategy

The proposed system is scripted in Python environment considering Kaggle dataset [26]. The dataset consists of traces of big data varied form of google cluster's traces with exclusive fields associated with utilized resources, decision of scheduling, and submitted jobs. It also has inclusion of information related to task along with reservation of shared resources and usage of CPU. The channel capacity considered is 2 gigabytes per second with 100 computing slots and 0.001 finetuning coefficient. The overall size of this dataset is around 2.4 terabytes. The outcome of the study has been evaluated with respect to four types of performance parameters *i.e.*, i) delay, ii) cost associated with task completion, iii) time associated with task completion, and iv) overall execution time. Further, the proposed scheme has been compared with existing software framework that is a current adopted in both research and enterprises viz. Apache Hadoop, Apache MapReduce, Apache Spark, Apache Flink, Apache Beam, Apache Tez, and Apache Samza. Conventional software framework has been suitably finetuned to fit in uniform testbed where data forwarding is carried out from analytical units to cloud distributed data storage units.

#### 3.2. Accomplished results

As the core implementation of proposed study model is mainly associated with meeting the deadlines of task in cost effective manner. Hence, the first performance-based investigation is carried out towards cost associated with task completion as shown in Figure 3. For retaining generality, the cost is represented in probability score while the outcome eventually shows proposed system to incur much reduced cost in contrast to existing frameworks.

Figure 2(a) showcase Apache Spark to possess reduced cost while Apache Beam to offer increased cost in perspective of existing frameworks. Apache Spark offering better batch processing of big data stream while it is still suffering from extensive memory consumption. Execution with Apache Beam has nearly similar batch and streaming capability but it has an additional layer of abstraction which extensively increases the cost while attempting to complete the task prior to defined deadline. Other conventional frameworks too are found to work in sub-optimal manner that doesn't contribute much towards massive big data processing. Figure 2(b) showcases Apache Hadoop to offer reduced time while Apache Flink to offer more time consumed towards task completion. It can be justified by adoption of disk-based storage by Hadoop leading to slower processing although it is capable of scaling thousands of nodes. Similarly, Apache Flink demands extensive dependability of deployed resources although it is well known for its fault tolerance. In all this regards, MapReduce do offer reduced time performance; however, it offers limited interactivity

leading to keep the user waiting for job completion that adversely affect the interactive data analysis. Figure 2(c) showcase proposed system to excel in highly reduced delay whereas majority of existing framework of distributed database management has similar trend of increased delay. This can be attributed by inclusion of less iterative operation and inclusion of more logical operations. This process speeds up the process of task completion and offers modeling a solution complying with defined deadline by proposed system. Figure 2(d) showcase that highly reduced algorithm execution time for proposed system. The outcome exhibited by MapReduce follows the next reduced execution time which is mainly due to its data locality feature where it moves the computation in proximity of data minimizing network congestion; however, it is not found suitable for low-latency application due to its batch-oriented properties. It eventually shows that proposed system offers approximately 37% reduced delay, 38% of reduced cost, 45% of minimized time, and 28% of reduced execution time in contrast to mean value of conventional software frameworks. The outcome eventually showcases the proposed scheme excels optimally cost-effective solution towards leveraging distributed and parallel data analytical operation in cloud.

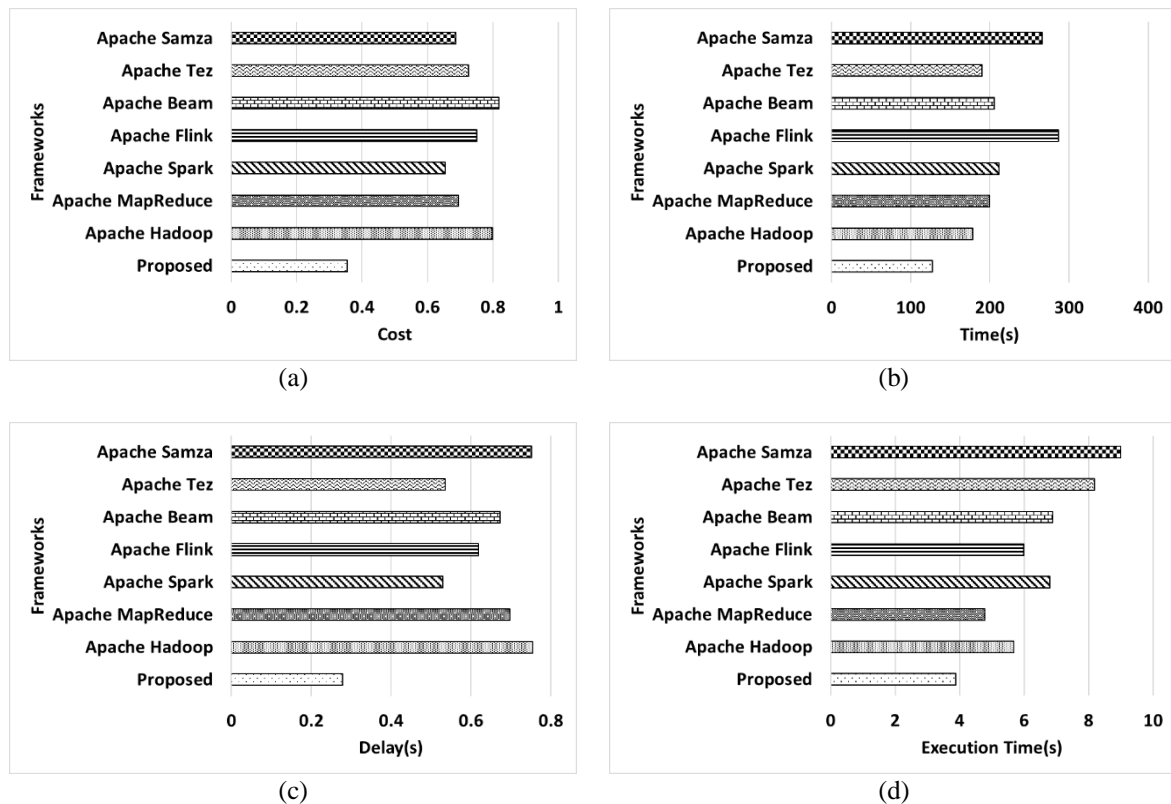


Figure 2. Comparative outcome: (a) cost, (b) time, (c) delay, and (d) execution time

#### 4. CONCLUSION

The proposed study presents a unique computational model which is capable of leveraging the data analytical operation in distributed environment. The contribution of the proposed study are as follows: i) the presented framework is capable of obtaining the big data of distributed application of IoT from multiple regions of data points in cloud environment, ii) the model considers heterogeneous prices on distributed data points to offer reduced task completion associated with job related to analytical operation for a defined deadline, iii) a distributed set of algorithms has been present capable of minimizing the cost and time attribute related to task associated with data analytical operation in multi-stages of transform and aggregate-based parallel framework, and iv) the study outcomes exhibited proposed scheme to excel cost-effective performance on near-real world dataset in contrast to conventional distributed frameworks. The future work will be further oriented towards considering more network-specific and traffic-oriented attributes to realize the impact of more challenging traffic states on task completion.

## ACKNOWLEDGEMENTS

We wish to confirm that no known conflicts of interest are associated with this publication and all the authors have contributed equally.

## FUNDING INFORMATION

Authors state no funding involved.

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Rajeshwari Dembala	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Kavya Ananthapadmanabha		✓				✓		✓	✓	✓	✓	✓		
Shashank Dhananjaya	✓		✓	✓			✓			✓	✓		✓	✓

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

## CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## REFERENCES




- [1] N. G. Puttaswamy and A. N. Murthy, "An efficient reconfigurable workload balancing scheme for fog computing network using internet of things devices," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, pp. 6525–6537, Dec. 2023, doi: 10.11591/ijece.v13i6.pp6525-6537.
- [2] M. Aljarah, M. Shurman, and S. Alnabelsi, "Cooperative-hierarchical based edge-computing approach for resources allocation of distributed mobile and IoT applications," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 296–307, Feb. 2020, doi: 10.11591/ijece.v10i1.pp296-307.
- [3] A. Q. Khan, M. Matskin, R. Prodan, C. Bussler, D. Roman, and A. Soyly, "Cost modelling and optimisation for cloud: a graph-based approach," *Journal of Cloud Computing*, vol. 13, no. 1, p. 147, Sep. 2024, doi: 10.1186/s13677-024-00709-6.
- [4] K. Rahul, R. K. Banyal, and N. Arora, "A systematic review on big data applications and scope for industrial processing and healthcare sectors," *Journal of Big Data*, vol. 10, no. 1, p. 133, Aug. 2023, doi: 10.1186/s40537-023-00808-2.
- [5] B. Berisha, E. Mëziu, and I. Shabani, "Big data analytics in cloud computing: an overview," *Journal of Cloud Computing*, vol. 11, no. 1, p. 24, Aug. 2022, doi: 10.1186/s13677-022-00301-w.
- [6] A. Nambiar and D. Mundra, "An overview of data warehouse and data lake in modern enterprise data management," *Big Data and Cognitive Computing*, vol. 6, no. 4, p. 132, Nov. 2022, doi: 10.3390/bdcc6040132.
- [7] H. Jahani, R. Jain, and D. Ivanov, "Data science and big data analytics: a systematic review of methodologies used in the supply chain and logistics research," *Annals of Operations Research*, Jul. 2023, doi: 10.1007/s10479-023-05390-7.
- [8] M. Khalid and M. M. Yousaf, "A comparative analysis of big data frameworks: An adoption perspective," *Applied Sciences*, vol. 11, no. 22, p. 11033, Nov. 2021, doi: 10.3390/app112211033.
- [9] X. Xu, L. Sun, and F. Meng, "Distributed big data storage infrastructure for biomedical research featuring high-performance and rich-features," *Future Internet*, vol. 14, no. 10, p. 273, Sep. 2022, doi: 10.3390/fi14100273.
- [10] A. Gandomi, M. Reshadi, A. Movaghar, and A. Khademzadeh, "HybSMRP: a hybrid scheduling algorithm in Hadoop MapReduce framework," *Journal of Big Data*, vol. 6, no. 1, p. 106, Dec. 2019, doi: 10.1186/s40537-019-0253-9.
- [11] G. Di Modica and O. Tomarchio, "A hierarchical Hadoop framework to process geo-distributed big data," *Big Data and Cognitive Computing*, vol. 6, no. 1, p. 5, Jan. 2022, doi: 10.3390/bdcc6010005.
- [12] A. Băicoianu and I. V. Scheianu, "Managing and optimizing big data workloads for on-demand user centric reports," *Big Data and Cognitive Computing*, vol. 7, no. 2, p. 78, Apr. 2023, doi: 10.3390/bdcc7020078.
- [13] R. N. S. Widodo, H. Abe, and K. Kato, "Hadoop data reduction framework: Applying data reduction at the DFS layer," *IEEE Access*, vol. 9, pp. 152704–152717, 2021, doi: 10.1109/ACCESS.2021.3127499.
- [14] W. Li and M. Tang, "The performance optimization of big data processing by adaptive MapReduce workflow," *IEEE Access*, vol. 10, pp. 79004–79020, 2022, doi: 10.1109/ACCESS.2022.3193770.




- [15] M. Saadoon *et al.*, "Experimental analysis in Hadoop MapReduce: A closer look at fault detection and recovery techniques," *Sensors*, vol. 21, no. 11, p. 3799, May 2021, doi: 10.3390/s21113799.
- [16] F. Hussain, M. Nauman, A. Alghuried, A. Alhudhaif, and N. Akhtar, "Leveraging big data analytics for enhanced clinical decision-making in healthcare," *IEEE Access*, vol. 11, pp. 127817–127836, 2023, doi: 10.1109/ACCESS.2023.3332030.
- [17] A. Kumar, N. Varshney, S. Bhatiya, and K. U. Singh, "Replication-based query management for resource allocation using Hadoop and MapReduce over big data," *Big Data Mining and Analytics*, vol. 6, no. 4, pp. 465–477, Dec. 2023, doi: 10.26599/BDMA.2022.9020026.
- [18] T.-C. Huang, G.-H. Huang, and M.-F. Tsai, "Improving the performance of MapReduce for small-scale cloud processes using a dynamic task adjustment mechanism," *Mathematics*, vol. 10, no. 10, p. 1736, May 2022, doi: 10.3390/math10101736.
- [19] O. Azeroual and A. Nikiforova, "Apache Spark and MLlib-based intrusion detection system or how the big data technologies can secure the data," *Information*, vol. 13, no. 2, p. 58, Jan. 2022, doi: 10.3390/info13020058.
- [20] J. Rank, J. Herget, A. Hein, and H. Krcmar, "Evaluating task-level CPU efficiency for distributed stream processing systems," *Big Data and Cognitive Computing*, vol. 7, no. 1, p. 49, Mar. 2023, doi: 10.3390/bdcc7010049.
- [21] J. Liu, T. Zhu, Y. Zhang, and Z. Liu, "Parallel particle swarm optimization using Apache Beam," *Information*, vol. 13, no. 3, p. 119, Feb. 2022, doi: 10.3390/info13030119.
- [22] S. Park, C.-S. Yang, and J. Kim, "Design of vessel data lakehouse with big data and AI analysis technology for vessel monitoring system," *Electronics*, vol. 12, no. 8, p. 1943, Apr. 2023, doi: 10.3390/electronics12081943.
- [23] I. Bartolini and M. Patella, "The Metamorphosis (of RAM3S)," *Applied Sciences*, vol. 11, no. 24, p. 11584, Dec. 2021, doi: 10.3390/app112411584.
- [24] G. Muchenje and M. Seppänen, "Unpacking task-technology fit to explore the business value of big data analytics," *International Journal of Information Management*, vol. 69, p. 102619, Apr. 2023, doi: 10.1016/j.ijinfomgt.2022.102619.
- [25] C. Jing and P. Dan, "JHTD: An efficient joint scheduling framework based on hypergraph for task placement and data transfer across geographically distributed data centers," *IEEE Access*, vol. 10, pp. 116302–116316, 2022, doi: 10.1109/ACCESS.2022.3219873.
- [26] A. Hussain and M. Aleem, "GoCJ: Google Cloud Jobs dataset for distributed and cloud computing infrastructures," *Data*, vol. 3, no. 4, p. 38, Sep. 2018, doi: 10.3390/data3040038.

## BIOGRAPHIES OF AUTHORS






**Rajeshwari Dembala**    holds a B.E., M.Tech, and Ph.D. in computer science and engineering from Visvesvaraya Technological University, Belagavi, India. With over 20 years of teaching experience, she is currently an associate professor in the Department of Information Science and Engineering at The National Institute of Engineering, Mysuru. Professor Rajeshwari D has published 18 journal papers in reputed journals, in addition to more than 5 conference publications, book chapters, and patents. Her research interests include data mining, artificial intelligence, the internet of things, blockchain technology, and data analytics. She currently supervises two research scholars. She can be contacted at email: drrajeshwari@nie.ac.in.



**Kavya Ananthapadmanabha**    received the Ph.D. degree from Visvesvaraya Technological University, Belagavi. M.Tech. degree in digital electronics and communication systems and B.E degree in electronics and communication from Visvesvaraya Technological University, Belagavi, Karnataka. She is presently working as an assistant professor in the Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysuru. She has 10 years of teaching experience. Her research interests include wireless sensor networks and communication systems. She has published three papers in Scopus indexed journals and also in National and International Journals. She can be contacted at email: kavyaap@gmail.com.



**Shashank Dhananjaya**    has completed his B.E. in computer science and engineering, M.Tech. in software engineering and Ph.D. in security of cognitive radio networks from the Visveswaraya Technological University, Belagavi, India. He is working as associate professor in the Department of Information Science and Engineering, NIE Mysuru. He has published four papers in Scopus Indexed journals and presented papers in international conferences winning the Best Paper Award. He is also awarded with Interscience Young Investigator Award from Interscience Network India. He has completed EMCCIS cloud certification, also a Certified Cyber Crime Intervention Officer from ISAC Inda. His area of interest includes radio networks, cybersecurity and cloud computing. He can be contacted at email: shashank@nie.ac.in.