International Journal of Advanced Science and Technology
Vol. 28, No. 20, (2019), pp. 764 – 776

A Cost Effective Scalable Scheme for Dynamic Data Service in Heterogeneous Cloud Environment

C.Venish Raja*, Dr.L. Jayasimman
*Research Scholar, Department of Computer Science, Bishop Heber College (Autonomous), Affiliated to Bharathidasan University, Trichy, Tamilnadu, India.
Assistant Professor, Department of Computer Science, Bishop Heber College College (Autonomous), Affiliated to Bharathidasan University, Trichy, Tamilnadu, India.

Abstract

Storing data in the past required a lot of resources and management. Because of tremendous utilization of web in distributed environment has created an urgent requirement for new techniques and frameworks that intelligently handled the information into valuable data and information. The past research methodologies and their frameworks focused only on static data which leads to wastage of resources and computational when the data is dynamic and soft. This paper proposed a new architectural framework to reduce the communication overhead, substantial switching cost and avoid lock-in dependency for the customers who uses the cloud services. The proposed framework uses K-means clustering analysis to filter the data in dynamic data services such as weather report, share market and streaming data like video or audio files while retrieving from the cloud. The unique feature of this framework is to provide the services using different service providers through a single interface. This research aims in enhancing the scalable resources based on the request made by the customers. The framework uses request analyzer to analyze the content which is based on media or numeric, resources are scheduled and allocated from the providers and deliver without communication delay to the customers. The experimental results indicates the proposed framework is better option in delivering the services without communication delay.

Keywords: *scalability, cost, heterogeneous cloud scalable, dynamic data service*

1. Introduction

For continuously growing businesses with increasing data, cloud-based services are one of the most accommodating solutions. Cloud service providers are focusing on 'scalability' because it's one of the primary requirements in IT environment. Scalability is the term often used, yet to be still understood. It is the compatibility of the process, network, software or appliance to grow and manage increased demands. This is one of the most valuable and predominant features in cloud computing. When business demands are increasing user can easily add nodes to increase the storage space, or user can increase the servers currently used. When the increased demand is used user can move back to their original configuration. Scalability enables to accommodate larger workloads without disruption or complete transformation of existing infrastructure. To effectively leverage scalability, one need to understand the complexity and the types of scalability. There are three types of scalability- Vertical, horizontal and diagonal.

Scale vertically- scale up is an easy way of scaling it can be done by moving the application to bigger virtual machines deployed in the cloud or cloud consumer can scale up by adding expansion units as well as with their current infrastructure. The ability to add resources to accommodate increasing workload volumes is vertical scaling.

Scale horizontally- scale out is the addition of nodes to the existing infrastructure to accommodate additional workload volumes. Contrary to vertical scaling, horizontal scaling also delivers performance along with storage capacity. The total workload volume is aggregated over the total number of nodes and latency is effectively reduced. The scaling is ideal for workloads that require reduced latency.

Scale diagonally or diagonal scaling helps to combine scaling up and scaling down. This type of scaling introduces enhanced budgeting and cost effectiveness for environment and business dealing with variable workload volumes.

Small data processes are a development concept that describes everything complex the amount of interrupted, semi-destructive and unstructured data that it has possible information retrieval. Big Data is different; it grows rapidly and requires intelligent management. In this paper, first examination of the revision of the big data document and then the second iteration related technologies like scalable cloud framework set has been discussed using iterative method [2]. A sophisticated visualization mechanism may involve the collection of data from different sources of different data formats and semantics. The identification of anomalies in the creation of this context requires a detailed distributed infrastructure with high efficiency and low latency [3].

The remainder of the paper is formed as follows. Section 2 describes the overview of related work in the field. Section-3 describes the proposed design and framework components. Section-4 describes the overview of architecture and its entities. Section-5 explains the implementation and experimental setup and results.Section-6 concludes the future work and report.

2. Related Works

Scalability of a cloud service is the broad area of research. Numerous studies have been conducted to identify various dimensions and issues in scalability of cloud. Most of the existing works focused on single layer scalability of cloud architecture whereas cloud is inherently heterogeneous and multi-layer [4]. To reduce the average waiting time and make-span dynamic task scheduling algorithms is proposed but it does not focus on the various metrics such as cost and number of users and task[5]. The cloud-based privacy protection model has been suggested to increase the security and scalability features [6].

This model does not consider the two important metrics total cost and response time which is must needed for scalable cloud. In order to achieve the efficient task scheduling adaptive task based scheduling algorithm is proposed [7] and evaluated in terms of cost and minimum resources used. The only drawback is it fails to concentrate on load balancing problems to achieve the efficient task based scheduling. .

To reduce the cost and waste of computing resources [8], a new heterogeneous resource allocation called multi-resource skewness avoidance allocation is proposed. However, it completely eliminates the total cost of the resources used and focuses only on resource allocation and time delay.

Although it is proposed to achieve better performance scalability in the dynamic output and utilization algorithm, it does not take into account response time and overall cost during the completion of jobs [9].It has been suggested that the resource allocation model based on feed forward neural network and back propagation neural network reduces the cost estimate in the cloud [10].Cost-based job scheduling is proposed to reduce response time and in job allocation processes use data transfer, processing power and network features. The main drawback is it does not calculate the total cost based on the number of users[11].

For task scheduling with minimum cost, an improved cost-based algorithm is proposed [12] to achieve reduced response time and task grouping to increase scalability in heterogeneous cloud. But the cost and resource committed to the expected level is not increasing. To minimize the end users' cost and ensure fairness among the service providers a dynamic pricing scheme is proposed [13]. This scheme changes the price accordingly to promote the usage of resources with low utilization rate and discourage the usage of resources with high utilization rate.

A priority based min-min algorithm using two phase technique for task scheduling is proposed to reduce execution time and enhance its performance using the available resources based on the priority[14]. This model completely ignores the number of users and total cost required to generate those jobs or task.

In this study [15], Enhanced Max-Min and Min-Min algorithms are studied and their performance is evaluated on the basis of makespan by increasing the number of cloudlets in time shared and in space shared mode. The problem with this approach it does not consider the cost and number of users to generate the task.

Optimization based algorithm is proposed [16] for efficient task scheduling using particle swarm optimization (PSO) in cloud computing environment. This approaches only consider the response time, execution time and energy consumption .The main drawback is the lack of priority determination which results in failure of deadline tasks. Resource Aware Min-Min algorithm is proposed [17] to produce the better response time and resource efficiency. The traditional Max-Min and Min-Min algorithms are not capable of producing better response time and efficient use of resources.

Different task scheduling algorithms using Evolutionary Multi Objective and Self Adaptive Learning Particle Swarm Optimization have been proposed [18], [19] to reduce cost, make-span and maximizing the profit of IAAS providers and Quality of services. These methods do not take into account the maximum number of users and the required allocation of resources necessary to complete a particular number of tasks. Various pricing schemes are discussed in [20]. The proposed model therefore uses the real-time pricing scheme[21],[22][23] needed to minimize costs and resources with an effective response time in terms of number of tasks, number of users, allocation of bandwidths for a specific task, etc.

3. Proposed Design

The over view of Cost Effective Scalable Framework is shown in figure-1 and the details are provided in Section 4. In the proposed methodology the resource allocation provisions are based on the request data it guarantees the efficient communication. The framework examines the Data preprocessor is used to study the numeric data and applies K-means clustering to filter the fake data and sends the accurate data to the user.
In this approach it is based on the assumption that all public service providers have a common interest in securing the infrastructure and data against any attacks and adversaries. This work acknowledges those challenges and focus on the scalability of cloud. It should be noted that framework is tested with soft or insensitive dynamic data.

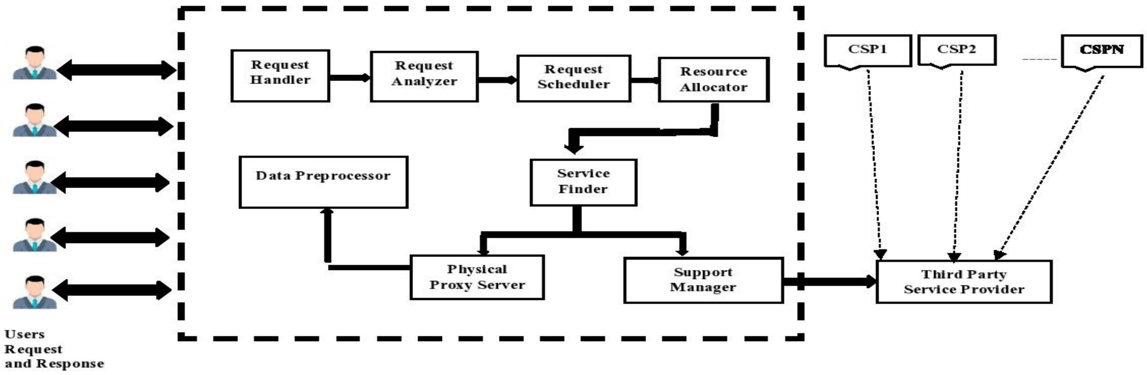

Figure-1 Architecture of CESFDD

A.CESFDD Framework

The framework is a web application and it has been comprised with various components. Request Handler, Request Analyzer, Request Coordinator, Resource Allocator, Service Finder, Data Preprocessor are the components used to perform the request process submitted by the user.

a) Request Handler: It handles the service request submitted by the user and check whether the request can be processed or illegal request.
b) Request Analyzer: It analyze the service request processed by the request handler and identifies the requirement of request is for numeric data or media.
c) Resource Allocator: It identifies type of the request and allocate the number of instances, processor, etc. required to complete the request.
d) Request Coordinator: The scheduler allocates the time submits the request to the service finder and also to the third party request repository.
e) Service Finder: The service finder first search API service in the physical proxy server and if the service is found it immediately submits the response to the user.
f) Support Manager: The support manager submits the request to the third party service provider. It also calculates the local resources used and third party resources used which helps in generating a billing report.
g) Data Preprocessor: Data Preprocessor uses k-means clustering to filter the numeric data obtained from third party service provider. It is also used for unsupervised learning process and it is also to be noted that whenever data is fetched from physical proxy server it is the filtered data there is no need for further preprocessing.

4. Architecture Overview

Users: The registered users will submit their request by using various devices which will be handled by the request analyzer in the proposed framework. They interact with the various service providers using the proposed framework.

Physical Proxy Server: It is a local cloud server (private cloud server) which is used to store large number of API calls or web services and also separates the storage for media files in a separate repository. When the client makes a call or request for the service, the proposed framework first communicates with physical proxy server and at the same time it also communicates with Third Party service provider so it searches among the public cloud service providers in parallel for the required service.

Third Party Service Provider: It consists of trusted cloud coordinator which is a programming interface used deploying the features such as resource allocation, scheduling etc. in the federation clouds. Third party service provider has the ability to provision the resource allocation based on the service resource details sent by the request coordinator. Based on the details sent by the request coordinator server resource gets allocated after the analysis of required bandwidth, latency etc. and also for the same cost.

Algorithm-1 CESFDD

The algorithm which has been proposed is shown below:

Input: Any request to dynamic data or streaming data
Output: Response time, type of streaming video files, file size resource allocated and cost based on the request
Step1:
User submits the request
Step2:
Request Handler analyze the type of the request and send it to the resource allocator
Step 3:
The default resources (latency, bandwidth, memory etc) allocated for the type of request.
Step 4:
Request Coordinator schedules the time to allocated request.
Step 5:
Service availability is checked by the service finder
If service found in PPS(physical proxy server)
If service type=="dynamic data"
Store the information in allocated resources and sent response.
Else
Get the service from TPSP and store the information in the allocated resources and sent it to the physical server
Call the Data Preprocessor()
Store the information in allocated resources and sent response
End if
If service type="streaming Data"
Store the information in allocated resources and sent response.
Else
Get the service from TPSP and store the information in the allocated resources and sent it to the physical server
End if
End if
Step 6:
Obtain the details of the resource allocated from support manager and calculate the cost.
Step 7:
Request Handler catches the request submitted time and finishing time is noted from the client machine once the response is received.

Algorithm-1 explains the user request is interpreted by the request analyzer and it identifies type of service request. Resource allocation provisions are based on the type of data so that for numeric type of data limited resources can be allocated with minimum cost. For streaming data maximum resources can be allocated. The time is scheduled for each service request by the coordinator. Availability of service is founded by the service finder. The service is found in physical proxy server immediate response is submitted to the user otherwise the user obtain the service from the third party service provider since the allocation and scheduling details are also submitted to the third party service provider in parallel once the resource allocation and scheduling is completed by the resource coordinator. This process makes the response time to be minimized and yields a better support for the users. The numeric data is fetched from trusted third party service provider's data preprocessor is used to filter the data. The request time and response time can be obtained by the request handler and service finder.

5. Implementation

The Cost Effective Scalable Framework is proposed to provide following benefits:

- Allocate resources based on the demand of the users and type of data requested.
- Increase interdependency among the cloud service providers.
- Dynamic resource or service allocation. Resources or services can be added or removed at any time.
- Optimized resource utilization based on dynamic and streaming data.
- Reduced cost
- Highly elastic in nature
- Better Quality of services in terms of response time.

A. Experimental setup:

The proposed methodology involves the creation of physical proxy server for cloud storage services. There is no federated system to evaluate the performance the technique. The proposed framework is implemented in Visual Studio 2010 Asp.Net with C#.K-means clustering is implemented using c# .net libraries. Various request data formats such as numeric, dynamic and streamed data are used in the experiments. The experiments are carried out using the following data. For streaming data the proposed framework is integrated with the following free API'S such as Daily Motion, Wurl Video API, Vimeo API and physical server API. Private clouds were operated on Windows 10 64 bit machine. The machine uses Intel Core (TM) 2 Duo CPU T6500 that runs at 2.10 GHz with 4 GB of DDR3 RAM and 100MbPs internet connection. The proposed work allocates the default resources provided and obtained the result for single tenant environment. The obtained values are stimulated in cloud sim to evaluate the experiment in multi-tenant environment by specifying the number of instances, processors, and datacenters and so on.

The first set of experiments uses the following characteristics:
RAM: 4GB, Bandwidth: 100Mbps, Data Transfer Limit 2000GB and average communication cost per hour is estimated from internet service provider like cherrinet and airtel. Computation or Vm used cost is taken from azure price web portal [21].

The second set of experiments is carried out by minimizing the vm characteristics as follows:
RAM: 1 GB, No. of VM/user: 4, bandwidth speed is limited to 0.5Mbps for dynamic data
RAM: 2GB, No. of VM/user: 4, bandwidth speed is limited to 2.0Mbps for streaming data.

Bandwidth speeds are fixed by using [22] [23] .The Soft Perfect Bandwidth manager tool is used to limit the bandwidth speed for each user to achieve the minimum resource required. The datacenter characteristics remain the same as before. Communication cost is calculated based on the bandwidth allocated. Computation cost is obtained from [22]. Communication cost and bandwidth speeds are obtained from Telecom Regulatory Authority of India (TRAI) recommendation which is used with minimum resources in the proposed model.

Acronyms used in Table 1 and 2

NOU -Number of Users
NOT -Number Tasks
NOVM/USR -Number of Virtual Machine per User
COVM/HR -Cost of Virtual Machine per Hour
RT -Request Type
RAM SIZE/USR -RAM SIZE per User
CCFRHR/USR -Communication Cost for an Hour per User
AVG.RT -Average Response Time
B/W SPD -Bandwidth Speed

Table-1 Scalability features with full usage of available resources

Service Used	NOU	NOT	VM/ USR	COVM/HR	RT	RSZ/ USR	CCFRHR/USR	AVG. RT	B/W SPD
Weather	5	5	1	Rs.0.25	Numeric	4GB	Rs.1.636	1 sec	100Mbps
Traffic	5	5	1	Rs 0.25	Numeric	4GB	Rs.1.636	1 sec	100Mbps
Video	5	5	1	Rs 0.25	Stream	4 GB	Rs 1.636	1 sec	100Mbps
Share Price	5	5	1	Rs 0.25	Numeric	4GB	Rs1.636	1 sec	100Mbps

Table-2 Scalability features with minimum cost and limited usage of available resources

Service Used	NOU	NOT	VM/ USR	COVM/ HR	RT	RSZ/ USR	CCFRHR/USR	AVG. RT	B/W SPD
Weather	5	5	4	Rs 0.15	Numeric	1GB	Rs 1.25	1.5 sec	0.5Mbps
Traffic	5	5	4	Rs 0.15	Numeric	1GB	Rs. 1.25	1 sec	0.5Mbps
Video	5	5	4	Rs 0.20	Stream	2GB	Rs 1.25	2 sec	2Mbps
Share Price	5	5	4	Rs 0.15	Numeric	1 GB	Rs 1.25	1.5 sec	0.5 Mbps

From the table-1 and 2 it is shown that table-2 average response time is obtained by using the minimum resources and also with minimum cost. Default allocation for dynamic data request is obtained by using 4 virtual machines each with 256 MB RAM and bandwidth is 0.5Mbps.Similarly for streamed data 4vms with 2 GB ram and bandwidth speed is 2Mbps is used. Table-2 shows that the proposed model consumes only 25% of RAM and 0.5% of bandwidth to achieve the minimum response time.

The third set of experiments is carried out by using the cloud analyst simulation software. The efficiency of the proposed approach is compared in terms computation cost,

communication cost, average response time, number of tasks, task type and number of users. The cloud analyst is configured by adding more number of users and tasks. Physical proxy server is created by using datacenter region code. If the region code and users code belong to the same code then the user request are handled by the physical proxy server in this simulation. Numeric and streamed data are simulated by varying the task length or file length parameter. The cost is calculated as per the configured cost estimated in cloud analyst and the number of virtual machines with memory used. The following configuration is made constant throughout our experimental setup as follows:
Data Center used-2, size of dynamic or streamed data site is 3 MB, 4 virtual machines of 256 MB RAM, bandwidth is 0.3Mbps for dynamic data and 2 Mbps for streaming data, 1 CPU and default cost as specified.
Table-3 shows the Comparison of our proposed approach with other approaches such as Adaptive cost based task scheduling (ACTS), improved cost based task scheduling algorithm (IABC) for 10 tasks.

Table-3 Comparison of scalability features with other existing schemes using 10 tasks.

Features	**ACTS[7]**	**IABC[12]**	**CESFDD (Proposed)**
No.of Tasks/hour	10	10	12
Task Type	Numeric/stream	Numeric/stream	Numeric/Stream
RAM per VM	1GB	1 GB	256 MB
No.of users/hr	10	10	30
Bandwidth	900 bps	1000 bps	300bps
Avg.Response Time	0.2 sec	0.1sec	0.3 sec
Total Cost per Hour(in Rs)	40	28.93	29.00
VMs/user	5	8	4

Table-4 shows the Comparison of our proposed approach with other approaches such as Adaptive cost based task scheduling (ACTS), improved cost based task scheduling algorithm (IABC) for 20 tasks
.

Table-4 Comparison of scalability features with other existing schemes using 40 tasks.

Features	**ACTS[7]**	**IABC[12]**	**CESFDD (Proposed)**
No.of Tasks/hour	20	25	20
Task Type	Numeric/stream	Numeric/stream	Numeric/stream
RAM per VM	1GB	1 GB	256 MB
No.of users/hr	20	20	40
Bandwidth	900 bps	1000 bps	300bps
Avg.Response Time	0.2 sec	0.5sec	0.3 sec
Total Cost per Hour(in Rs)	100.00	72.34	29.00
VMs/user	5	8	4

Table- 5 shows the Comparison of our proposed approach with other approaches such as Adaptive cost based task scheduling (ACTS), improved cost based task scheduling algorithm (IABC) for 80 tasks with 75 users per hour.

Features	ACTS[7]	IABC[12]	CESFDD (Proposed)
No.of Tasks/hour	30	50	80
Task Type	Numeric/stream	Numeric/stream	Numeric/stream
RAM per VM	1GB	1 GB	256 MB
No.of users/hr	35	60	75
Bandwidth	900 bps	1000 bps	300bps
Avg.Response Time	0.4 sec	0.37sec	0.39 sec
Total Cost per Hour(in Rs)	700	370	30.00
VMs/user	5	8	4

Table-5 Comparison of scalability features with other existing schemes using 80 tasks.

Table- 6 shows the Comparison of our proposed approach with other approaches such as Adaptive cost based task scheduling (ACTS), improved cost based task scheduling algorithm (IABC) for 1000 tasks with 100 users per hour.

Table-6 Comparison of scalability features with other existing schemes using 1000 tasks.

Features	ACTS[7]	IABC[12]	CESFDD(Proposed)
No.of Tasks/hour	50	100	1000
Task Type	Numeric/stream	Numeric/stream	Numeric/stream
RAM per VM	1GB	1 GB	256 MB
No.of users/hr	35	60	100
Bandwidth	900 bps	1000 bps	300bps
Avg.Response Time	0.45 sec	0.48sec	0.40 sec
Total Cost per Hour(in Rs)	1000	543	40.00
VMs/user	5	8	4

Acronyms used in Table 1 and 2

NOU -Number of Users
NOT -Number Tasks
RT -Request Type
RAM SIZE/USR -RAM SIZE per User
TCO/HR -Total Cost for all users per Hour
AVG.RT -Average Response Time
B/W SPD -Bandwidth Speed
ACTS - Adaptive Cost Based Task Scheduling
IABC -Improved Cost Based Task Scheduling
CESFDD -Cost Effective Scalable Framework for Dynamic Data

ISSN: 2005-4238 IJAST

Table-7 shows the proposed algorithm is evaluated using the following parameters such as number of tasks per hour, number of users per hour, average response time and total cost per hour includes computation, storage and communication cost and it is not for individual user.
None of the existing schemes consider the users and the response time for the performance comparison while the proposed scheme considers those two metrics along with the cost is presented in the table below.

Table-7 Comparison of scalability features of the proposed model in terms of number of tasks, users, total cost and response time with existing models

S.NO	Existing Cost Based Scheduling Schemes								Proposed			
	ACTS[7]				IABC[12]				CESFDD			
	NOT	NOU	AVG.RT(ms)	TCO/HR (rs)	NOT	NOU	AVG. RT (ms)	TCO/HR (rs)	NOT	NOU	AVG. RT (ms)	TCO/HR (rs)
1	10	10	200	40	10	10	100	28	10	10	300	29
2	20	20	200	100	25	20	300	72	40	40	300	29
3	30	35	400	700	50	60	370	370	80	75	390	30
4.	50	45	450	1000	100	80	480	573	1000	100	499	40

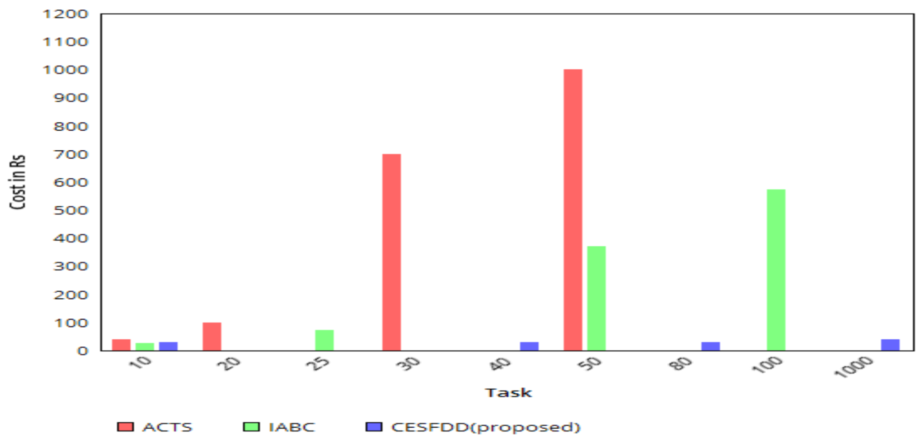

Figure-2 comparison of task vs cost

Total Cost: It includes the cost of used resources, storage cost and data transfer from datacenter to receiver's machine or device.
Figure-2 shows the comparison of existing schemes with the proposed Cost Effective Scalable Framework in terms of cost. The tasks are taken in the x-axis while the total cost (price) per hour is taken along the y-axis. If the number of tasks is 50 the existing adaptive cost-effective task scheduling costs ₹1000, in improved cost based algorithm when the task is 100 costs is ₹573 but the proposed cost-based task scheduling has ₹40. This indicates that the proposed CESFDD requires less expense than the existing system.
Average Response Time: It is the time taken to complete all the request and corresponding events to get to the server and then back to the user's device or machine.

Figure-3 shows the comparison of existing schemes with the proposed Cost Effective Scalable Framework in terms of average response time. In the x-axis, the tasks are taken whereas on the y-axis the execution time in milliseconds (ms) is taken. If the number of tasks is 50 the existing adaptive cost-effective task scheduling average response time is 450(ms), in improved cost based algorithm when the task is 100 average response time is 480 ms but the proposed cost-based task scheduling is 499(ms) when the task is 1000. This indicates that the proposed CESFDD will reduce the time taken for the overall process.

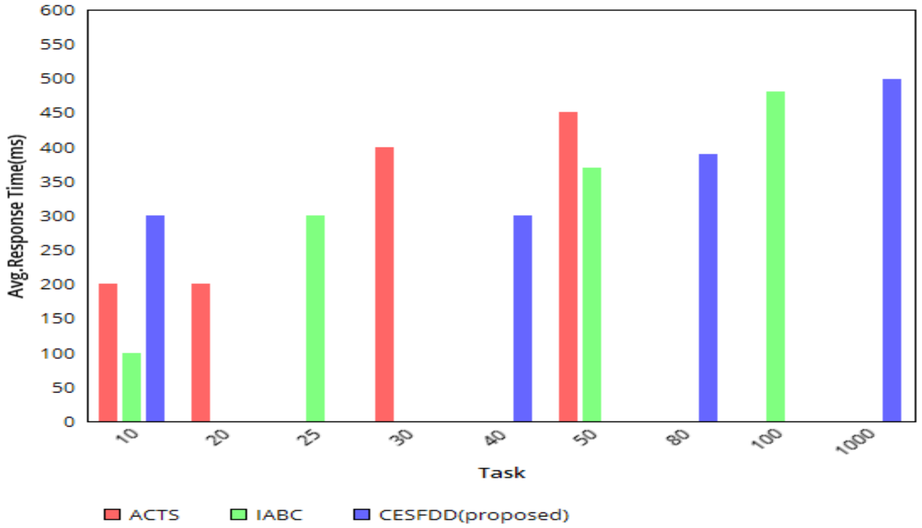

Figure-3 comparison of task vs average response time

Figure-4 shows the total number of users per hour submitted their task in each scheme. It is observed that the proposed model uses the large number of users and tasks to indicate that cost and response time can be better with the minimum resources used.

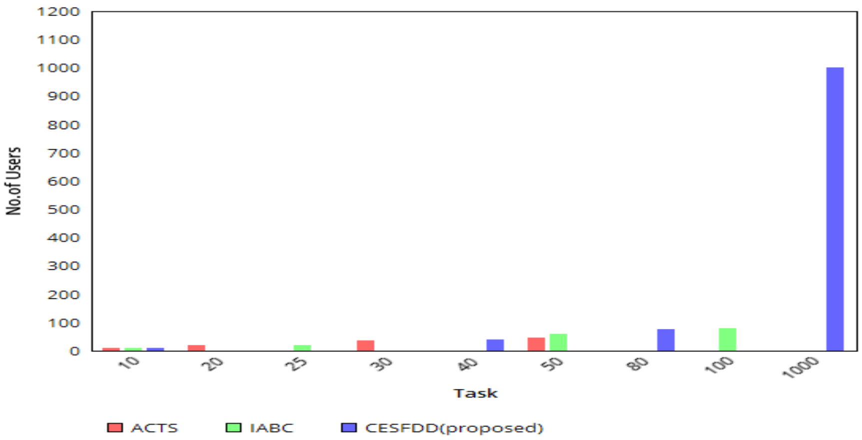

Figure-4 comparison of users in each scheduling scheme

6. Future Enhancement

The redundant streamed data present in the private server has not been reduced to obtained more efficiency. Mobility challenges have not been observed in this work. The amount of energy consumed in this work is not examined in this work. Since there is no real time heterogeneous cloud environment the proposed architectural framework should be tested by submitting request through various devices and domains such as big data, artificial intelligence and internet of things.

Conclusion

In this paper, heterogeneous cloud-based scheduling model has been proposed. It uses minimum resources to complete the task with minimum cost and better response time. The proposed model also Consider the number of users along with various metrics such as RAM per virtual machine, Number of virtual machines per user, bandwidth and so on. The experimental results indicates that the proposed cost effective scalable framework model has provides better performance in terms of number of users and tasks, total cost , response time and other virtual resources.

References

[1] Lijin P, Dr.S.Sivasathiya, K.S Guru Prakash, " A Survey on Resource Allocation Techniques in Cloud Computing", International Journal of Engineering and Technology vol. 7, no.3 ,2018,pp. 823-828

[2] Imen Chebbi, W. B. "Big Data: Concepts, Challenges and Applications", Computational Collective Intelligence, Springer, Lecture Notes in Computer Science, vol. 9330, 2013, pp. 638-647.

[3] Mohiuddin Solaimani, M. I. "Statistical technique for online anomaly detection using Spark over heterogeneous data from multi-source VMware performance data" Proceedings of the IEEE International Conference on Big Data, Washington, (2014), October 27-30.

[4] C. A. Ardagna, E. Damiani, F. Frati, G. Montalbano, D. Rebeccani and M. Ughetti, "A Competitive Scalability Approach for Cloud Architectures," Proceedings of the IEEE 7th International Conference on Cloud Computing, Anchorage, AK, 2014, June 27- July 2.

[5] A. K. M. Mashuqur Rahman Mazumder, K. M. Aslam Uddin, N. Arbe, L. Jahan and M. Whaiduzzaman, "Dynamic task scheduling algorithms in cloud computing," Proceedings of the 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, June 12-14.

[6] P. Santhosh Kumar, L. Parthiban and V. Jegatheeswari, "Scalable Anomaly Detection for Large-Scale Heterogeneous Data in Cloud Using Optimal Elliptic Curve Cryptography and Gaussian Kernel Fuzzy C-Means Clustering", Journal of Circuits, Systems and Computers, 2019, pp. 2050074-1 -2050074-38.

[7] Mohammed A. S. Mosleh, G. Radhamani, Mohamed A. G. Hazber, and Syed Hamid Hasan, "Adaptive Cost-Based Task Scheduling in Cloud Environment," Scientific Programming, vol. 2016, Article ID 8239239, 2016, pp.1-9.

[8] L. Wei, C. H. Foh, B. He and J. Cai, "Towards Efficient Resource Allocation for Heterogeneous Workloads in IaaS Clouds," in IEEE Transactions on Cloud Computing, vol. 6, no. 1, 2018 pp. 264-275,.

[9] Deivendran P, E.R Naganathan "Scalability in Dynamic Performance and Utilization Techniques in Scalable Cloud Computing", Journal of Advanced Research in Dynamical and Control Systems Vol. 9. 2017 pp.2975-2996.

[10] Amritdeep Singh Mann, Vijay Kumar " An Efficient Method for Estimation of Cost in Cloud Computing using Neural Network", Indian Journal of Science and Technology Vol.9 no.44 2016 pp.1-10 .

[11] Mansouri, N. and Javidi, M.M. "Cost based job Scheduling strategy in cloud computing environments" Distributed Parallel Databases, 2019, pp.1-36.

[12] S. Selvarani and G. S. Sadhasivam, "Improved cost-based algorithm for task scheduling in cloud computing," Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, 2010, December 28-29.

[13] N. binti Muhamad Shaari, T. Ang, L. Por and C. Liew, "Dynamic Pricing Scheme for Resource Allocation in Multi-Cloud Environment", Malaysian Journal of Computer Science, vol. 30, no. 1, 2017, pp. 1-17,.

[14] Abdulrahman Mohammed Hussein Obaid, Santosh Kumar Pani, Prashant Kumar Pattnaik "A priority based on min-min algorithm for reducing make span task scheduling in cloud computing", International Journal of Engineering & Technology vol.7 no.4, 2018, pp. 2039-2045.

[15] Vasrha Soni, Dr. N.C.Barwar "Performance Analysis of Enhanced Max-Min and Min-Min algorithm in cloud computing environment", International Journal of Management Technology and Engineering, October 2018, pp. 1281-1286.

[16] Khairunnisa "Optimization Algorithms based Task Scheduling Method for cloud computing environment" International Journal of Recent Technology and Engineering Vol.8 no. 3, September 2019, pp. 3608-3613.

[17] Syed Arshad, Samiya Khan, Mansaf Alam, " Resource-Aware Min-Min Algorithm for Resource Allocation in Cloud Computing", International Journal of Recent Technology and Engineering Vol.8 no. 3, September 2019, pp. 1863-1870.

[18] Z. Zhu, G. Zhang, M. Li, and X. Liu, "Evolutionary multiobjective workflow scheduling in cloud," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 5, 2016 pp. 1344–1357.

[19] X. Zuo, G. Zhang, and W. Tan, "Self-adaptive learning psobased deadline constrained task scheduling for hybrid iaas cloud," IEEE Transactions on Automation Science and Engineering,vol. 11, no. 2, 2014, pp. 564–573.

[20] Artan Mazrekaj, Isak Shabani, Besmir Sejdiu "Pricing Schemes in Cloud Computing – an Overview International Journal of Advanced Computer Science and Applications vol.7 no .2,2016, pp. 80-86 .

[21] "Pricing - Windows Virtual Machines | Microsoft Azure", Azure.microsoft.com, 2019. [Online]. Available:https://azure.microsoft.com/en-in/pricing/details/virtual machines/windows/. [Accessed: 27- Oct- 2019].

[22].Manish Ranjan, Abdul Kayum, M.suri, "Broadband definition and standardization of Measurement method in the context of developing countries like India having large rural population" Telecom Engineering Center, Department of Telecommunications, Government of India. Tec.gov.in, 2019. [Online]. Available: http://tec.gov.in/pdf/Studypaper/Study%20paper%20on%20BB%20Definition.pdf.

[23] "What is Bandwidth? - Definition and Details", Paessler.com,2019.[Online].Available: https://www.paessler.com/it-explained/bandwidth.

Author's Profile

Author I

Mr.C.Venish Raja is currently working as an AssistantProfessor, in the Departmen tof Information Technology, St.Joseph's College (Autonomous), Trichy, India. He received his M.Phil Degree in Bharathidasan University, Trichy, India in 2012 and also He is pursuing Ph.D (Computer Science) Bharathidasan University, Trichy.

Author II

Dr. L. Jayasimman working as an Assistant Professor, in the Department of Computer Science, Bishop Heber College, Trichy, India.He received his M.Tech Degree in Bharathidasan University, Trichy, India in 2008 and He completed his PhD (Computer Science)Bharathidasan University, Trichy In 2014.