International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-1, November 2019

PE2 - A Service Oriented Meta Task Scheduling Framework in Cloud Environment

H. Krishnaveni, V. Sinthu Janita

Abstract: ***Cloud computing is an emerging computing environment which facilitates on demand services. As it contributes a large pool of computing resources, scheduling of tasks in an efficient manner is one of the main problems. Poor allocation of tasks affects the performance of the whole system. Hence it is very important to schedule the tasks for better utilization of resources by allocating them properly to particular resources in particular time. Efficient scheduling algorithms fulfill the user requirements and also satisfy the needs of the cloud service providers without affecting the performance of the environment. Execution Time based Sufferage Algorithm (ETSA), Cost and Completion Time based Sufferage Algorithm (CCTSA) and Modified Artificial Fish Swarm(MAFSA) Algorithm are efficient task scheduling approaches developed in cloud environment. These algorithms considered the parameters such as makespan, cost and resource utilization while scheduling the tasks and produced better performance. This paper presents a scheduling framework which converts the above said algorithms in to services and deployed in the cloud. Depends on the user's requirements, the services will be delivered.***

Keywords: ***cloud computing, task scheduling, cost, resource utilization, makespan***

I. INTRODUCTION

Cloud computing is a distributed system which has centralized server resources to provide on demand network access. It offers numerous application services over the Internet and also provides computing as a service to its users. It allows its users to lease resources and services in a pay as you use manner and release them when they are no longer useful. Cloud computing provides different services to different users. These services are called as cloud delivery models; they are Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) [1].

IaaS: It provides the computing architecture such as processing power, storage, network bandwidth, CPU and power. Once the user acquires the infrastructure, he/she controls the operating system, data, applications, services etc.

PaaS: The user is provided with the hardware infrastructure, network and operating system to form a hosting environment. So that he is able to install his applications and activate the services from the hosting environment.

Revised Manuscript Received on November 30, 2019.

***** Correspondence Author

Mrs H Krishnaveni*, Associate Professor in Department of Computer Science, Cauvery College for Women(Autonomous), (affiliated to Bharathidasan University), Tiruchirapalli

Dr Sinthu Janita, Professor and Head in the PG &Research Department of Computer Science, Cauvery College for Women(Autonomous),(affiliated to Bharathidasan University), Tiruchirapalli.

SaaS: This service allows the users to access an application. He does not control the hardware, network, security or operating system. This is the biggest public category of cloud services.

Resource and data management, network infrastructure management, and virtualization are the some of the problems related to IaaS. Scheduling algorithms have a direct impact on the adeptness of users' tasks and also in efficient consumption of resources in IaaS computing environment.

Scheduling can be categorized into job scheduling and task scheduling. Job scheduling means mapping of jobs (processes) to a set of resources. Jobs can be divided into a number of independent tasks, which have no dependency between each other. Task scheduling means the assignment of independent tasks to a group of machines. It directly affects the efficiency of the entire cloud system. The major parameters to be considered while scheduling the tasks are the overall completion time, overall execution cost, throughput, load balance and consumption of resource [2].

Efficient task scheduling results in better utilization of the machines with minimized overall execution time and cost. Also it improves the throughput of the system with a balanced load [3].

The energy efficiency, scalability, resource computation and network transmission cost are parameters, which are used in lot of scheduling problems. But only few work focus dynamic load balancing of resources. This work focuses the following parameters: makespan, cost and resource consumption.

This research paper presents a framework which delivers the services to the cloud users based on their requirements. The remaining part of the work is prepared as follows. Section 2 explains the related works of the research; Section 3 describes the presented scheduling framework for efficient task scheduling; Section 4 discusses the experimental results and the last section 5 concludes the paper.

II. RELATED WORKS

As cloud is a heterogeneous environment consists of a large pool of resources, scheduling plays an essential role. Task scheduling is used to schedule the tasks impartially to suitable resources in such a way to enhance the resource consumption, improve the performance and sustaining with reduction of execution time. There are a variety of scheduling models based on the scheduling criteria. This section explains some of the related works pertaining to task scheduling in cloud environment based on time, cost and optimality.

Saeed Parsa and Reza Entezari -Maleki [4] proposed an innovative scheduling algorithm known as Resource-Aware-Scheduling algorithm.

It is the combination of two conventional scheduling algorithms such as Max-Min and Min-Min. It makes use of the goodness of these algorithms and wrap up their shortcomings. This algorithm does not consider the, arrival time of the tasks, expenditure of the task execution on every machine and transmission cost.

Chawda et al. [5] proposed, Improved MinMin task scheduling algorithm to balance the load and enhances the utilization of the machines, to reduce the total completion time. It is a two stage algorithm. MinMin algorithm is executed is the first stage and the subsequent stage it selects the task with highest completion time and assigns it to suitable resource.

An MaxMin algorithm is improved by J.K Konjaang et al. [6] for Resource Allocation and Task Scheduling. It effectively optimize the allocation of resources in the cloud by taking the advantage of MaxMin scheduling algorithm and make it better to enhance the effectiveness in terms of makespan i.e. completion time. The conventional MaxMin algorithm gave importance to choose the tasks with highest execution time on a quicker available machine i.e. proficient of giving lowest completion time. So maximum execution time of the tasks may enhance the makespan that directs to a holdup of minimum execution time of the tasks.

Su et al. [7] designed a cost-effective task scheduling algorithm based on two heuristic techniques. The initial technique assigns tasks to the suitable cost-efficient VMs dynamically using the concept of Pareto dominance. The next technique follows the reverse of the first strategy that reduces the economic costs of non-critical tasks. This approach is not performed well as it did not take into account the completion time and cost for scheduling.

In [8], a hybrid algorithm Performance and Cost Algorithm (PCA) was proposed, to minimize the services cost compensated by the customer and/or to increase the revenue grow by the service provider. It also aims to optimize the performance by minimizing the overall completion time and maximizing the employment of the resources, so as to allow the service provider to bring the greatest and most competent services with extremely competitive prices.

Mosleh et al., [9] proposed an Adaptive Cost-based Task Scheduling to get to the virtual machines within the time limit exclusive of increasing the cost. The Data Access Completion Time (DACT) is considered to select the cost efficient path to access the data. For this, the DACT is calculated via taking into account the mean and variance of the network service time and the appearance rate of network input/output requirements. Based on the task precedence, the cost of information paths are examined and allocated. For low priority tasks minimum cost path is allocated and rapid access path are assigned to high priority tasks so as to meet the time constraint.

Load Balancing Mutation Particle Swarm Optimization (PSO) based on mathematical model was proposed in [10] which consider reliability, make span, transmission and round trip time and transmission cost. LBMPSO improves the reliability of the cloud computing environment by reschedule the failed tasks after considering the resources available. The Improved Particle Swarm Optimization algorithm was proposed in [11] aims to give the finest allotment of enormous amount of errands to the virtual machines. Dordaie and Jafari Navimipour [12] have introduced a PSO with hill climbing algorithm (local search) for scheduling the task in cloud environment.

Xu et al. [13] proposed, task scheduling on heterogeneous computing system. It used numerous priority queues genetic algorithm for task scheduling. Keshanchi et al. [14] combined genetic algorithm and priority queues for task scheduling in the cloud environment

Scheduling a task in an efficient way is a noteworthy study issue in the part of cloud computing. For different types of user, it is very vital element for organizing and sharing resources of cloud. Some of the task scheduling algorithms is explained in the literature. Every algorithm has a few confinements, for example, highest time taken for scheduling, overloaded, time and computation complexity. Also, nearly all of the research works utilize a less number of tasks and single objective function. These works cannot provide better results. To conquer the difficulties present in recent scheduling algorithms, there is an huge requirement for creating novel approaches. This work presents a framework for task scheduling in cloud environment that reduce time, cost and maximize the resource utilization.

III. METHODOLOGY

This research paper exhibits a PE2 (Priority Service, Economic Service and Exemplary Service) framework for meta task scheduling based on the requirement of users in cloud environment. The framework uses three different task scheduling algorithms: ETSA,CCTSA and MAFSA. Fig. 1 shows the flow of proposed framework. This framework provides the services to the users of the cloud based on their requirements. It comprises three algorithms that are delivered as services to the users. They are Execution Time based Scheduling Algorithm (ETSA), Cost and Completion Time based Algorithm(CCTSA) and Modified Artificial Fish Swarm Algorithm (MAFSA).Each algorithm has its own feature to adopt as a service. Service cost is fixed for each algorithm based on their functionality.

ETSA service is executed when the user wants to execute their tasks quickly i.e. in short time (low makespan). CCTSA service is executed, when the user wants to minimize the cost of execution. MAFSA is an optimization algorithm; it gives optimal results for all the parameters i.e minimize makespan and cost, maximize resource utilization. If the user wants to reduce the overall completion time and cost of the tasks execution, MAFSA service is called.

To formulate the task scheduling problem, consider the set of tasks, defined as $T=\{t_1, t_2,.., t_n\}$, where n is the number of tasks. Set of resources defined as $R=\{r_1, r_2, \dots r_m\}$, where m is the number of resources. Two matrices such as Expected Time to Compute (ETC) and Expected Cost to Compute (ECC) are formed. ETC (T_i, R_j) is the estimated execution time of task (T_i) on resource (R_j). ECC (T_i, R_j) is the estimated execution cost of task (T_i) on resource (R_j).

The completion time of task t_i on resource r_j can be computed as follows,

$$CT_{ij} = ET_{ij} + r_j \quad (1)$$

where, CT_{ij} completion time

ET_{ij} execution time of task (T_i) on resource (R_j)

r_j availability time of resource (R_j) after completing the previously assigned tasks.

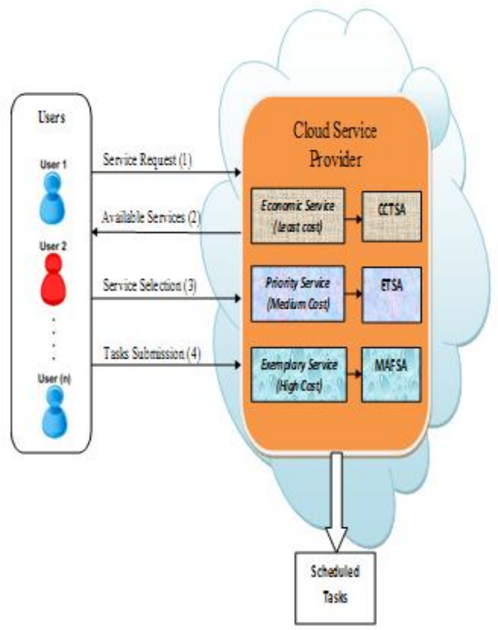

Fig 1. Flow of Scheduling framework

The Completion Cost CC_{ij} of task t_i on resource r_j can be calculated by using (2).

$$CC_{ij} = Cost_{ij} + rc_j \tag{2}$$

where, $Cost_{ij}$ -execution cost of task i on resource j

rc_j - ready cost of resource j

A. ETSA

This algorithm schedule the tasks to various resources based on the execution time. Fig. 2. shows the block diagram of ETSA.

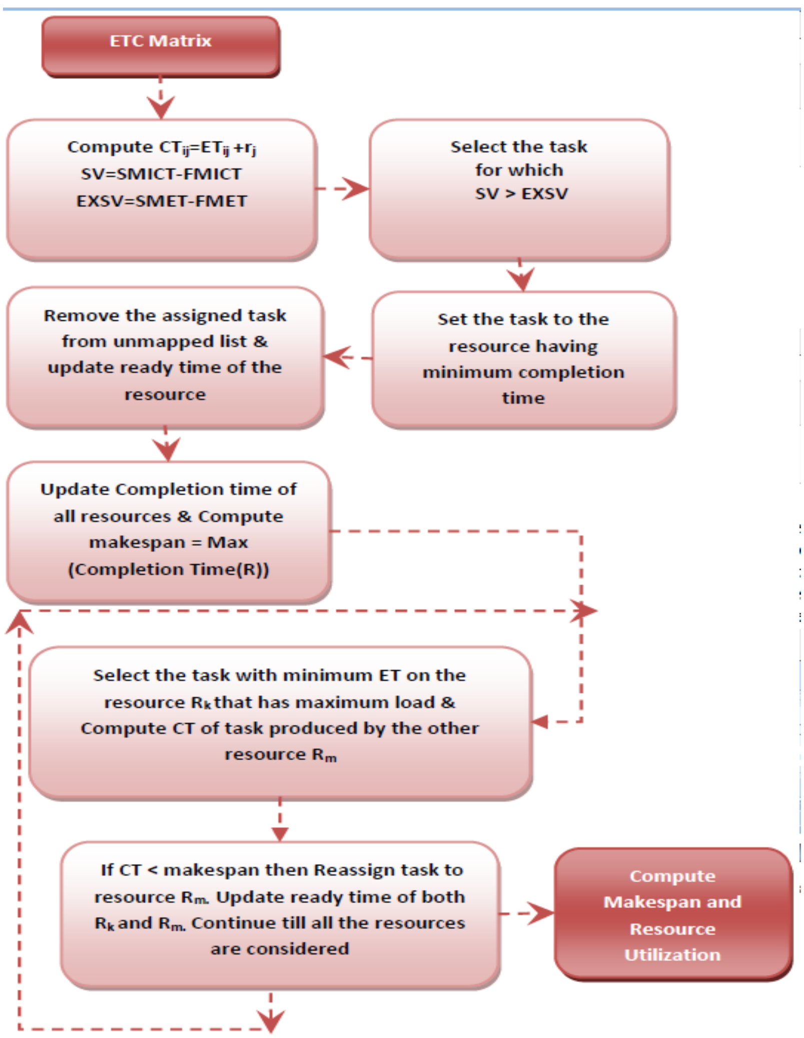

Fig 2. ETSA Block Diagram

For each task T_i, First Minimum Completion Time ($FMCT_i$) and Second Minimum Completion Time ($SMCT_i$) are calculated. Also First Minimum Execution Time ($FMET_i$) and Second Minimum Execution Time ($SMET_i$) are computed. Sufferage Value for completion time using equation (3) is computed.

$$SV_i = SMCT_i - FMCT_i \quad (3)$$

Sufferage value for execution time $EXSV_i$ is calculated using equation (4).

$$EXSV_i = SMET_i - FMET_i \quad (4)$$

Then, the tasks are sorted based on the sufferage value SV_i. The $EXSV_i$ is arranged based on the value of SV_i. For each task (T_i) from the task list (N), if $SV_i > EXSV_i$, then the task (T_i) is allotted to the resource that provides the lowest completion time of task. The chosen task (T_i) is eliminated from task list and update ready time of resource $R_j = C_{ij}$, otherwise, T_n will be selected and assigned. Next the algorithm attempts to decrease the load of the resource containing more number of tasks comparing with different resources. This will modify the task that has smallest amount of execution time on the greatest loaded resource to the resource which is under utilized by holding the condition that its completion time in other resources is less than the makespan acquired. The time complexity of this algorithm is $O(n^2m)$ where m = resource count and n = task count.

B. CCTSA

The cost and completion time based sufferage algorithm is used to schedule the tasks to the machines in an effective manner. Fig. 3. shows the block diagram of CCTSA.

For each task in the unassigned task list, the first and second minimum completion time is selected and sufferage value (SVCT) is calculated. In the same way, from the completion cost matrix, first and second minimum completion cost is selected. Then cost sufferage (SVC) which is the difference between Second Minimum Cost and First Minimum Cost is computed.

Both the sufferage values SVCT and SVC are compared with first minimum completion time and first minimum completion cost respectively to select a particular task for scheduling. If the condition is satisfied, the corresponding task is allotted to the resource that provides lowest completion time for that task. Eliminate the task from the un-assigned task list. Ready time, completion time of the resource and completion cost are updated. Then makespan, total execution cost, and resource utilization are calculated. The time complexity of this algorithm is $O(n^2m)$ where m = resource count and n = task count.

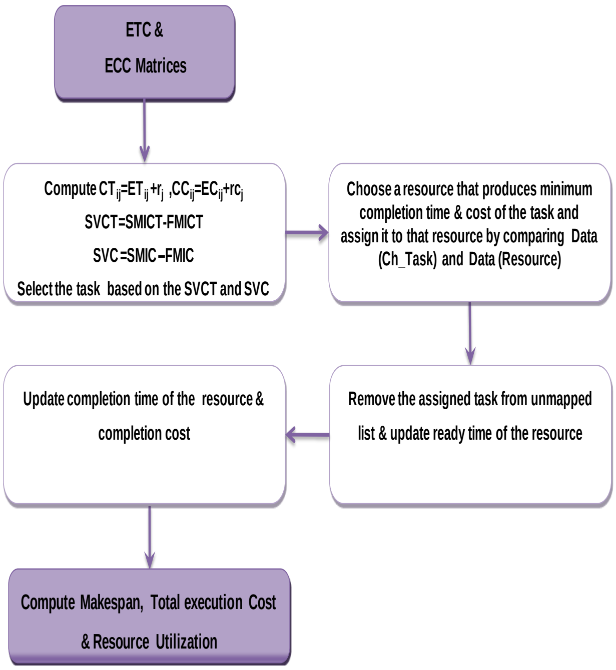

Fig. 3. CCTSA Block Diagram

C. MAFSA

MAFSA is proposed for efficient task scheduling. The output of CCTSA is given as one of the population in MAFSA tol increase the efficiency of task scheduling. Every artificial fish searches its natural conditions and its associates to pick a proper behavior to move at the speediest towards the optimal path. At last, the artificial fish assembles around a few neighborhoods extreme.

Assume X is artificial fish swarm vector. Let $X=(x_1, x_2,\ldots,x_n)$ where $x_1,x_2,\ldots,x_n$ is current position of fishes. The activity of fishes happens just in the span of a loop with vision. X_v is the visual position at some moment; $X_v = \left(x_1^v, x_2^v, \ldots, x_n^v\right)$. X_{next} is the latest location of the fish. The parameter Step (moving step length), Visual (visual distance), and crowd factor ($0 < \delta < 1$) are used in scheduling. The practices of artificial fish incorporate prey behavior, swarm behavior along with follow behavior explained in [17]. Fig. 4. shows the block diagram of MAFSA. The execution of the algorithm stops simply if the upper limit of iterations is reached. The result which is holding the optimal objective value is chosen based on the ranking strategy and displayed in the bulletin board.

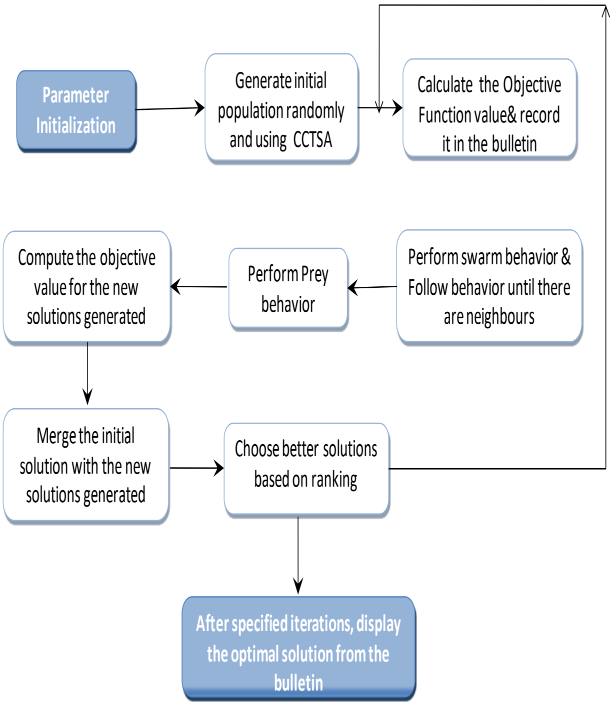

Fig.4. MAFSA Block Diagram

The time complexity of this algorithm is $O(2{*}N^2+2{*}N)$, where N is the number of population.

IV. RESULTS AND DISCUSSION

This section explains the performance evaluation of proposed frame work. The proposed scheduling framework is implemented using Cloud Sim 3.0. The evaluation metrics of this work is makespan, cost and resource utilization.

For experiments, sample of 10 tasks and 3 resources are taken. Table-I shows the resource information and Table-II shows the task information.

Table – I: Resource Details

Resource	Processing Speed (MIPS)	Related Bandwidth (Mbps)	Cost in $/Hr
R1	50	100	0.03
R2	100	200	0.12
R3	200	250	0.24

Table – II: Task Details

Tasks	No. of Instructions (MI)	Data (Mb)
T1	206	44
T2	50	95
T3	128	64
T4	69	30
T5	118	59
T6	112	47
T7	21	39
T8	200	61
T9	90	23
T10	45	23

The execution time for each task on each resource can be computed as

Execution time = (MI/MIPS + Mb/Mbps)

The execution cost of each task on each resource is calculated by

Execution cost=Mb * Cost

Where,
MI - Million Instructions
MIPS – Million Instructions per Second
Mb – Mega bits
Mbps – Mega bits per Second
Initially Expected Time to Compute (ETC) and Expected Cost to Compute (ECC) matrices are generated.
Table-III presents the expected execution time of tasks in milliseconds for 10 * 3 matrix

Table – III: ETC Matrix (msec.)

Task	R1	R2	R3
T1	4.56	2.28	1.206
T2	1.95	0.975	0.63
T3	3.2	1.6	0.896
T4	1.68	0.84	0.465
T5	2.95	1.475	0.826
T6	2.71	1.355	0.748
T7	0.81	0.405	0.261
T8	4.61	2.305	1.244
T9	2.03	1.015	0.542
T10	1.13	0.565	0.317

Table-IV presents the expected cost to compute for tasks.

Table –IV: ECC Matrix ($)

Task	R1	R2	R3
T1	1.32	5.28	10.56
T2	2.85	11.4	22.8
T3	1.92	7.68	15.36
T4	0.9	3.6	7.2
T5	1.77	7.08	14.16
T6	1.41	5.64	11.28
T7	1.17	4.68	9.36
T8	1.83	7.32	14.64
T9	0.69	2.76	5.52
T10	0.69	2.76	5.52

Table-V shows the makespan comparison of ETSA, CCTSA and MAFSA for different number of tasks.

Table – V: Makespan Comparison

No. of Tasks	No. of Resources	Makespan (msec.)		
		ETSA	CCTSA	MAFSA
25	5	9.30	11.41	8.87
50	7	14.16	15.47	12.89
100	10	20.29	21.99	15.22
200	14	25.17	28.95	20.45
300	17	34.02	37.14	31.25
500	22	41.24	45.88	34.16

Fig.5. shows the graphical representation of makespan analysis. From that result, the MAFSA algorithm produce less makespan compared to other algorithms

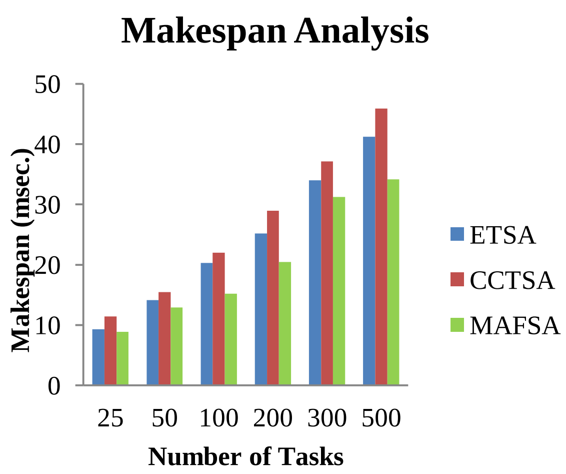

Fig.5. Makespan Analysis

Table-VI shows the resource utilization obtained by ETSA, CCTSA and MAFSA for different combination of inputs.

Retrieval Number: L34211081219/2019©BEIESP
DOI: 10.35940/ijitee.L3421.119119
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table –VI: Resource Utilization Comparison

No. of Tasks	No. of Resources	Resource Utilization (%)		
		ETSA	CCTSA	MAFSA
25	5	85.23	90.08	91.66
50	7	86.11	89.18	90.54
100	10	87.46	91.30	93.90
200	14	94.95	95.81	95.55
300	17	95.57	94.96	94.08
500	22	96.67	96.78	96.80

Fig.6. shows the analysis of resource utilization. From that result, the MAFSA algorithm produce high percentage compared to other algorithms. When the number of tasks is increased, the resource utilization is increased.

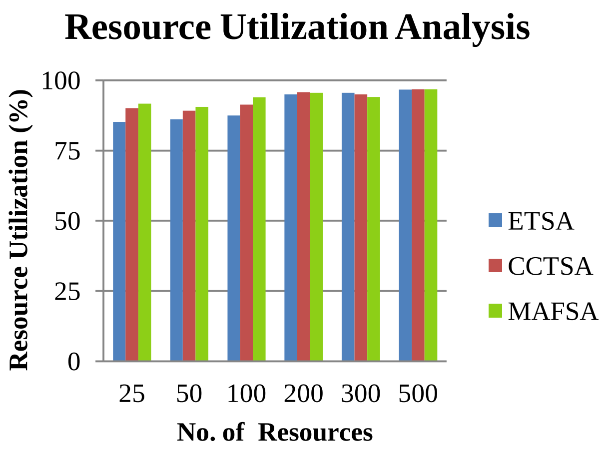

Fig.6. Resource Utilization Analysis

Table -VII shows the cost comparison of ETSA, CCTSA and MAFSA for different number of tasks and resources.

Table- VII: Cost Comparison

No. of Tasks	No. of Resources	Total Execution Cost ($)		
		ETSA	CCTSA	MAFSA
25	5	82.12	76.61	70.15
50	7	156.33	150.80	143.95
100	10	192.80	185.95	181.70
200	14	257.37	231.67	220.60
300	17	335.43	327.73	316.71
500	22	399.39	377.80	360.79

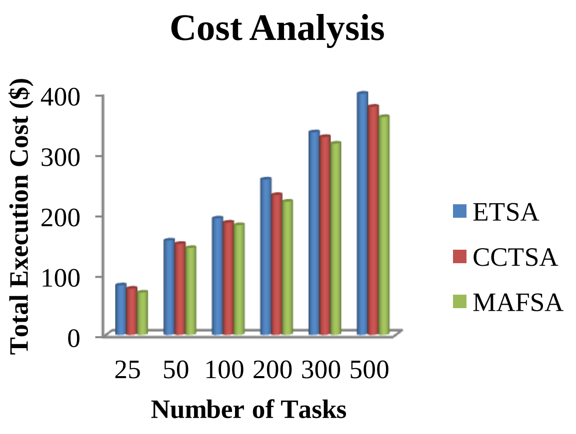

Fig.7. Cost Analysis

Fig.7. shows graphical representation of cost analysis. From that result, it is observed that, MAFSA obtained low cost compared to other algorithms. The overall result shows that the MAFSA algorithm performs well in terms of makespan, resource utilization and cost.

V. CONCLUSION

Efficient scheduling algorithm can give up more enviable services to consumers and enhance the performance offer by cloud environment. The primary intent of task scheduling in cloud environment is to lessen the execution time of errands and to maximize the resource consumption. This work presents a scheduling framework for efficient task scheduling in cloud environment based on execution time (ETSA), cost and completion time (CCTSA) and optimization (MAFSA). ETSA is applied to schedule the amount of errands to the different resources dependent on the execution time efficiently. It provides improved outcomes in terms of makespan and resource utilization with a balanced load. CCTSA schedule the tasks to the machines in an effective manner. The outcome of the algorithm proves that the cost is minimized with the increased utilization of the resources and makespan. MAFSA technique utilizes the principles of evolutionary to reduce some redundant computations. This algorithm decreases the execution cost, makespan and increases the resource utilization while scheduling the tasks in the cloud. The user can go with any type of service based on his requirement.

REFERENCES

1. Vijayalakshmi A. Lepakshi, Prashanth C. "A Study on Task Scheduling Algorithms in Cloud Computing", International Journal of Engineering and Innovative Technology (IJEIT), Vol. 2, Iss.11, 2013
2. Gawali M., Shinde S. "Task scheduling and resource allocation in cloud computing using a heuristic approach", Journal of Cloud Computing, Springer, Vol.7, Pp. 1-16, 2018.
3. Armbrust M, Fox A, Griffith R, Joseph AD. "A view of cloud computing", Communication of ACM, Vol. 53, Iss.4, Pp.50–58, 2010.
4. Parsa S., Entezari-Maleki R. " RASA: A New Task Scheduling Algorithm in Grid Environment", World Applied Sciences Journal, Vol. 7, Pp.152-160, 2009.

5. Chawda, P., Chakraborty, P.S. "An improved min-min task scheduling algorithm for load balancing in cloud computing", Internal Journal on Recent Innovation and Trends in Computing and Communication, Vol.4, Pp.60–64, 2016.
6. J.K Konjaang, J.Y Maipanuku and K.K Kubuga. "An Efficient Max-Min Resource Allocator and Task Scheduling Algorithm in Cloud Computing Environment", International Journal of Computer Applications, Vol. 142, Iss.8, Pp. 25 –30, 2016.
7. Su S, Li J, Huang Q, Huang X, Shuang K, and Wang J. "Cost efficient task scheduling for executing large programs in the cloud", Parallel Computing, Vol. 39, Iss. 4-5, Pp. 177–188, 2013.
8. Naseem A.AL-Sammarraie, Mohammed .F. Al-Rahmawy and Magdi Z. Rashad. "A scheduling algorithm to enhance the performance and the cost of Cloud Services", Computer Engineering and Intelligent Systems, Vol. 6, Iss.8, 2015.
9. Mosleh M. A. S., Radhamani G, Mohamed A. G. Hazber, and Syed Hamid Hasan. "Adaptive Cost-Based Task Scheduling in Cloud Environment", Scientific Programming, vol. 2, Article ID 8239239, 2016
10. Awad A.I., El-Hefnawy N.A., Abdel_kader H.M. "Enhanced Particle Swarm Optimization For Task Scheduling In Cloud Computing Environments", International Conference on Communication, Management and Information Technology (ICCMIT2015), Elsevier, Science Direct, Procedia Computer Science, Vol. 65, Pp. 920 – 929, 2015.
11. Saleh H, Nashaat H, Saber W, and Harb H.M. "IPSO task scheduling algorithm for large scale data in cloud computing environment". IEEE Access, Vol. 7, Pp.5412–5420, 2019.
12. Dordaie N., and Navimipour N. J. "A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments", ICT Express, Vol. 4, Issue. 4, Pp. 199–202, 2017.
13. Xu Y, Li K, Hu J, Li K. "A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues", Information Sciences Vol.270, Pp. 255–287, 2014.
14. B.Keshanchi, A.Souri, N.Jafari Navimipour. "An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: Formal verification, Simulation,and statistical testing", Journal of Systems and Software, Vol.124, Pp.1–21, 2017.
15. Krishnaveni H., Sinthu Janita V. "Execution Time based Suffrage Algorithm for Static Task Scheduling in Cloud", Advances in Big Data and Cloud Computing,Springer,Vol.750, Pp. 61-70, 2018.
16. Krishnaveni H, George Amalarethinam D.I., Sinthu Janita V. "Cost and Completion Time based Sufferage Algorithm for Task Scheduling in Cloud Environment", International Journal of Research in Electronics and Computer Engineering (IJRECE), Vol. 7, Iss.2, Pp.2807-2812, 2019.
17. Krishnaveni H., Sinthu Janita V. "Modified Artificial Fish Swarm Algorithm for Efficient Task Scheduling in Cloud Environment", International Journal of Computer Sciences and Engineering (IJCSE), Vol.7, Iss.5, Pp.1363-1371, 2019.

AUTHORS PROFILE

Mrs H Krishnaveni pursed Bachelor of Science from Bharathidasan University, Trichy in 1997 and Master of Computer Applications in Bharathidasan University in year 2000. She is currently pursuing Ph.D. and currently working as Associate Professor in Department of Computer Science, Cauvery College for Women(Autonomous), (affiliated to Bharathidasan University), Tiruchirapalli since 2001. She is a member of ISSE. She has published more than 5 research papers in reputed international journals and conferences including Springer and it's also available online. Her main research work focuses onCloud Scheduling, Cloud Security and Privacy, Big Data Analytics and IoT. She has 18 years of teaching experience.

Dr Sinthu Janita pursed Master of Computer Applications and Ph.D from Bharathidasan University. She is currently working as a Professor and Head in the PG &Research Department of Computer Science, Cauvery College for Women(Autonomous),(affiliated to Bharathidasan University), Tiruchirapalli. She is a member of IEEE, Computer Society of India and a life member of ISSE.She has published more than 35 research papers in reputed international journals including Thomson Reuters (SCI & Web of Science) and conferences including IEEE, Elsevier, ACM, Springer and it's also available online. Her main research work focuses on Communication Networks, Network Security, Big Data Analytics, Data Mining and Mobile Communications. She has 23 years of teaching experience and 14 years of Research Experience achievements, with photo that will be maximum 200-400 words.