ISSN 2321-807X
VOLUME 12 NUMBER 23
JOURNAL OF ADVANCES IN CHEMISTRY

ERAM2 - ENERGY BASED RESOURCE ALLOCATION WITH MINIMUM RECKON AND MAXIMUM RECKON

A.M.SenthilKumar[1], Dr. M.Venkatesan[2], Dr. A.RajivKannan[3]
[1]Research Scholar, [2,3]Professor
senthilam1185@gmail.com
[1]Dept. of CSE, Tejaa Shakthi Inst. of Tech. for Women, Coimbatore, Tamil Nadu, India
[2]KSR Institute for Engg. and Tech, Tiruchengode, Tamil Nadu, India
[3]Dept. of CSE, KSR College of Engineering, Tiruchengode, Tamil Nadu, India

ABSTRACT

The emerging field of cloud computing has flexibility and dominant computational architecture that offers ubiquitous services to users. It is different from traditional architecture because it accommodates resources in a unified way. Due to rapid growth in demands for providing the resources and computation in cloud environments, Resource allocation is considered as primary issues in performance, efficiency, and cost. For the provisioning of resource, Virtual Machine (VMs) is employed to reduce the response time and executing the tasks according to the available resources. The users utilize the VMs based on the characteristics of the tasks for effective usage of resources. This helps in load balancing and avoids VMs being in an idle state. Several resource allocation techniques are proposed to maximize the utility of physical resource and minimize the consuming cost of Virtual Machines (VMs). This paper proposes an Energy-Based Resource Allocation with Minimum Reckon and Maximum Reckon (ERAM2); which achieves an efficient scheduling by matching the user tasks on Resource parameters like Accessibility, Availability, Cost, Reliability, Reputation, Response time, Scalability and Throughput in the terms of Maximum Reckon and Minimum Reckon. This paper proposes an Ant Colony - Maximum Reckon and Minimum Reckon (AC-MRMR) method to consolidate all the available resource based on the pheromone value; the score is calculated for each pheromone value. When the score value exceeds Threshold limit λ then task migration process is carried out for optimized resource allocation of tasks.

Keywords: Cloud Computing, Virtual Machine, Resource Allocation, Minimum Reckon and Maximum Reckon, Ant Colony - Maximum Reckon and Minimum Reckon (AC-MRMR), Task Migration.

1. INTRODUCTION

In this emerging field, cloud computing [1] plays an important role by deploying several services in a disturbed system and these are accessed through networks. Cloud refers to collections of servers and devices in the networks. In cloud computing, the users are allowed to attain resources to achieve a powerful standard of networks by emerging QoS [1] and custom-need that are reliable for the end users. The resource allocation is based on the Service Level Agreement (SLA), established via service providers and customers [2]. "Resource allocation is used to allocate resources on the cloud based on user demand in an efficient and economic way. This is a way of scheduling the available resources and tasks to the required end users" [3]. Cloud has various resources and requires multiple policies for managing efficiently. Some of the factors that are affecting the resource management are performance, efficient allocation, and cost. In cloud computing, resource management is assigned with fluctuating tasks which pretense a major challenge in a maximum allocation of available resources [4].

The policies in the cloud computing for resource management are different from policies followed in traditional resource allocation systems. The general policies that are considered in a cloud environment are Admission control, Resource allocation, Resources parameter, Tasks Balancing, and Energy management. In a cloud, the architecture is framed for resource allocation [5] and to deal with the problems for efficient provisioning of VMs. In that architecture, the resource management satisfies three constraints such as CPU, Network I/O and RAM. The research focuses on CPU and RAM memory; meanwhile, a system allocates the jobs to the matched VMs. The proposed methodology does not overcome the scheduling issues faced in the cloud computing.

To overcome the scheduling issues new scheduling methodology is proposed, named as Haizea [6] that performs the scheduling policies like best effort reservation, immediate reservation and advanced reservation on Open Nebula. Major issue found in cloud computing is resource allocations by establishing Service Level Agreement with the end user. For provisioning of resource allocation the end user establish the contract with SLA [7]. The performance metrics termed in SLA is Latency in services, Consistency, Throughput, Security, Accessibility, Availability, Cost, Reliability, Reputation, Response time and Scalability.

The existing scheduling methodology does not utilize the Resources parameters for maximizing the resources for end users. The proposed ERAM2 framework uses resource parameters like Accessibility, Availability, Cost, Reliability, Reputation, Response time, Scalability and Throughput. The resources are allocated based on the User parameter list. At first, the user loads are scheduled by resource weight and resource capacity, this is done using First to Suit Scheduling (FSS) algorithm. FSS algorithm allocates the cloudlets by selecting the VMs based on requirements that are closest to the requested amount of resource requirements. Finally, it sorts the VMs so that the cloudlets are scheduled, this results in least usage of VMs. This algorithm sorts the items so that the user loads are scheduled based on first and best VM in the list at each iteration. FSS fails during idleness of VM's after completion of cloudlets and arrival of dynamic cloudlets. To overcome this dynamic situation, resource consolidation is done using Ant Colony - Maximum Reckon and Minimum

Reckon (AC-MRMR). The resource consolidation is based on the Maximum and Minimum reckons attributes. Both attributes are used to improve the quality of resource allocation among VMs.

Finally, validation and comparison of the proposed methods, First to Suit Scheduling (FSS) and Ant Colony - Maximum Reckon and Minimum Reckon (AC-MRMR) method are done in a simulated environment by using Cloud Sim toolkit. In this paper, rest of the contents organized as follows. Section 2 discusses related works and Section 3 discusses proposed methodology. Section 4 shows Experimental results and Section 5 concludes this paper.

2. RELATED WORK

Sanjaya K. Panda et al. [8] have focused a multi-objective task scheduling algorithm which reduces makespan and cost in heterogeneous multi-cloud systems. This work also focuses on performance metric average cloud utilization. However, it does not consider other performance metrics. Wang et al [9] have proposed Ant Colony Optimization based task scheduling policy to minimize the makespan of the submitted tasks in the cloud system. However, it does not evaluate various metrics to improve task scheduling. GuPing et al [10] have addressed adaptive Ant Colony Optimization algorithm to solve the convergence problem in the conventional ant colony algorithm. However, this technique does not apply to task scheduling problems.

Yangyang Dai et al. [11] have discussed task scheduling algorithm by combining Ant Colony Optimization algorithm (ACO) with Genetic Algorithm (GA). It focuses on time-consumption, expenditure, security and reliability in the scheduling method. However, it does not include more Resources parameters in task scheduling.Most of VMs consolidation problems are formulated from greedy heuristics models such as Best Fit [12], First Fit Decreasing (FFD) [13]. However, greedy methods do not provide near optimal solutions. Also, many models fail to focus multi-dimensional view of resource utilization. Xiaoying Wang et al. [14] proposed decentralized virtual machine design that focuses on minimizing the energy costs and load balancing by making idle nodes into a sleep state. Two thresholds levels are used to decide virtual machine migration.

Liu et al [15] have implemented a model for live VM migration. They have analyzed the energy cost and performance for migration. Choi et al. [16] have proposed a model based on resource utilization history to find the VMs migration thresholds at run time. Park et al. [17] implemented a design of virtual machine migration in an autonomy virtualized environment .They proposed an optimization model based on linear programming. Yongqiang Gao et al. [18] focused a multi-objective ACO algorithm for optimal VM placement to minimize resource wastage and power consumption in data centers. However, it does not consider other performance metrics.

3. PROPOSED METHODOLOGY

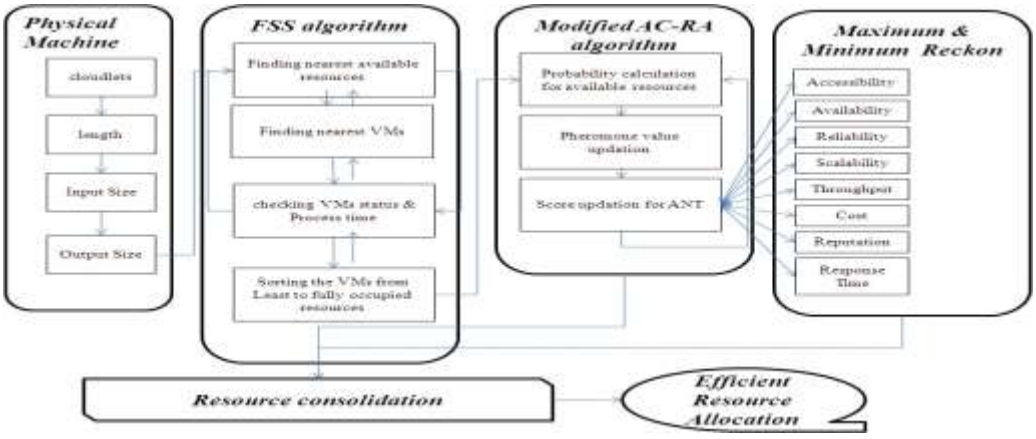

Fig. 1 : ERAM2 Framework

The proposed scheduling technique is shown in figure 1. It is an automated model which is divided into two steps. First, FSS scheduler updates the accessible resources list by requested resources and capacity. The resource list contains the accessible resources which are closest to the required amount of resources. Based on task requirement, suitable closest resource i.e. VM's are selected from the accessible resources list. Then the scheduler uses AC-MRMR to consolidate the least used and idle resources into a single resource for efficient resource allocation. Consolidation of resources is based on maximum and minimum reckon values on Resource parameters. During consolidation, tasks are migrated to the other VM's which can occupy the tasks as per threshold policy. Before migration destination VM's capacity is checked and based on an availability of memory capacity, tasks are migrated to new VM's. The AC-MRMR is worked with the combination of pheromone value, Maximum reckons and Minimum reckons. Finally, consolidation of the resource will occupy less memory storage, consume less energy and resource scheduling will be efficient.

3.1 First to Suit Scheduling Algorithm

In this framework, the status of a virtual machine is calculated by monitoring the cloudlets and computation of resource utilization in the VM's and the average of theirs monitoring values. The status of the VM's in the cloud is calculated, and the following function is used to calculate the cloudlets

$$r(n)=\begin{Bmatrix} Machinecount_{\max}\exists_j R_j < d_j \\ Taskcount_{\max}\sum_{j=1}^{n}\mu_j\left(d_j - R_j\right) > \lambda \\ Machinecount_{\min}\exists_i R_i < d_i \\ Taskcount_{\min}\sum_{i=1}^{n}\left(d_i - R_i\right) > \lambda \end{Bmatrix} \quad \textbf{(1)}$$

In above function, R_j and $R_i$represents the max and min resource utilization, μ_j $\mu_i$represents the weight of the resource max and min, d_j represents the ideal resource utilization and λ defines threshold limit. Since fully occupied resource will lead to poor performance of the system, the under loaded cloudlets are determined to consume resources that are available. This algorithm sorts the items so the user loads are scheduled to the closest, in order to use least VMs as possible. To calculate the least resources the following function is used,

$$r_{ij}=\frac{C_{ij}\times R_{ij}}{d_{ij}-\varepsilon} \quad \textbf{(2)}$$

Hence, C_{ij} represents the capacity of original resource $iandj$ allocation, R_{ij} represents the in progress allocated resource $iand\ j$ and d_{ij} represents the supreme utilization of the resource. In this ε denotes the distance factor between utilized resource and ideal resource and it is used to avoid the unhinged state after completion of the process.

Algorithm: First to Suit Scheduling

Input: client Machine $M=\{E_1,E_2\ldots\ldots\}$ $E=\{E\ id, status, p, i, pt\}$

Output: Efficient resource allocation at low cost.

Tasks demand Resources

While (true)

Server sends request to clients E.Request=listen()

Server check status of each client E(request).status=idle

E(request).p++

Scanning process stop when all client sends acknowledgement E(request.t.stop())=M.ack()

Task (workload)

For each E∈ client Machines M

Checks status and resources

E=(E.status==idle)::E(min(e,p))

E.p++

Assigning workload to Resources

E.pt=workload

E⟵workload

Repeating the process till all tasks allocated by resources

S.t.continue ()

End

3.2 Ant Colony Resource Allocation using AC-MRMR

In dynamic workloads and tasks, the FSS resource allocation takes more time. During completion of tasks and dynamic workloads, FSS scans the nearest resources that are available in a cloud to handle this dynamic situation. Hence it consumes more resource, energy, and cost for executing the process. Therefore resource consolidation is required to

overcome those issues. In resource consolidation, it identifies the cloudlets count which runs on Virtual Machine that is less than the threshold value (λ). Then the identified cloudlets are marked for migration to other VMs which is capable of executing the cloudlets. This resource consolidation is done by proposing an algorithm named as Ant Colony - Maximum Reckon and Minimum Reckon (AC-MRMR).

The consolidation of resources is done at a frequent interval of time and also consolidation is done when the cloudlets get executed from the resources. After a consolidation of resources, hundred joules of energy are minimized in this work. The proposed ERAM2 notations as follows,

τ_{jl}	Pheromone value	η_{jl}	Heuristic Information
p_{jl}	Probability of Pheromone Value	J	Total number of Tasks
vm	Number of Virtual Machines	C_{jl}	Cost of Tasks J
E_{jl}	Reputation of Tasks J	T_{jl}	Response Time of Tasks J
e_{jl}	Accessibility of Tasks J	A_{jl}	Availability of Tasks J
L_{jl}	Reliability of Tasks J	S_{jl}	Scalability of Tasks J

$$\rho_{jl} = \begin{cases} 1, R\_resources \\ 0, otherwise \end{cases}$$

$\max \sum_{j=0}^{J} \sum_{l=0}^{vm} e_{jl} A_{jl} L_{jl} S_{jl}$ Maximum Reckon

$\min \sum_{j=0}^{J} \sum_{l=0}^{vm} c_{jl} E_{jl} T_{jl}$ Minimum Reckon

$\sum_{j=0}^{J} \sum_{l=0}^{vm} J\rho, J \le vm$ Tasks J assigned to VMs

$\sum_{l=0}^{vm} \rho, J = 1$ Assigning Tasks J to resources R

3.3 Dynamic Updating of Pheromone Value

To choose the next VMs, strengthening of pheromone value plays an important role. The pheromone value decides the ability of the resources to process the cloudlets. Each ant iterates the pheromone value for migration plan by limiting within maximum reckons and minimum reckons. In this, maximum reckon are always maximum τ_{max} and minimum reckon are always minimum τ_{min}. After completing the pheromone update, if the pheromone value is high, then maximum reckon is selected. In turn, if the value is low, then minimum reckon is selected. For efficient resource allocation, maximum reckons should be maximum and minimum reckons should be minimum with specified λ threshold limit. The modification in pheromone helps to consolidate the resources at a frequent interval of time. The probability values p_{jl} of each ant are as follows,

$$p_{jl} = \frac{\alpha + \tau_{\max} + (1-\alpha)\eta_{jl}}{\sum_{i \notin FSS} \left(\alpha + \tau_{\max} + (1-\alpha)\eta_{jl}\right)} \forall i \notin FSS \quad \textbf{(3)}$$

$$p_{jl} = \frac{\alpha + \tau_{\min} + (1-\alpha)\eta_{jl}}{\sum_{i \notin FSS}\left(\alpha + \tau_{\min} + (1-\alpha)\eta_{jl}\right)} \forall i \notin FSS \quad \textbf{(4)}$$

Where α is AC-MRMR parameter that controls a ratio between $\tau_{\max}$, $\tau_{\min}$ and η_{jl} is heuristic information and that is used to compute all moves of each ant. At each iteration, the ant moves from current VMs to another VMs by applying resource transition rule until every VMs encounters resources for consolidation. The heuristic information is calculated for both maximum Reckon $\boldsymbol{\tau_{max}}$ and Minimum Reckon $\boldsymbol{\tau_{min}}$ as follows,

$$\eta_{\max(jl)} = \sum_{j=1}^{J-1}\sum_{l=0}^{vm} e_{jl} A_{jl} L_{jl} S_{jl} \quad \textbf{(5)}$$

$$\eta_{\min(jl)} = \frac{1}{Z + \sum_{j=1}^{J-1}\sum_{l=0}^{vm} c_{jl} E_{jl} T_{jl}} \quad \textbf{(6)}$$

Where Z is a small integer, which is used to avoid zero dividends.

Algorithm Modified Ant Colony –Resource allocation using AC-MRMR

Input: set of resources $R_m \leftarrow r_1, r_2, \ldots \ldots r_n$,

Set of resources requests $RR_j \leftarrow rr_1, rr_2 .. rr_n$,

A is number of ants

Output: consolidation of Resources

Set pheromone value as τ_{jkl} ←0;

Calculate threshold limit $\lambda \leftarrow \sum R_m$;//

Searching resources ∀Ant

While $(J - E) > 0$

Reckon p_{jl}

then

$\tau \in$ While $\in (R_m) \neq 0$

Selecting the phase for ant $vm \leftarrow R_m$

Schedule j task at $\in (R_m)$

Label1: Update Pheromone value← τ_{jkl}

$E \leftarrow E \cup \{J\}$//consolidating the VMs

Reckon $\sum dR$ // calculating score for resources

If score excites the λ limit then

selectResources.migration ()

Else

Continue to L1

End

3.4 Workload Migration

After pheromone is updated, the score of the ant is increased when it visit the VM s in the cloud environment. The ant score is compared with the threshold limit, when a score is higher than the threshold limit then workload migration is preceded to improve the system performance. Figure 2 shows workload migration when the current resource is needed by another task. The server machine holds all client machine and represents them using vector $\mathrm{E} = \{E.Id, Status, p, i, pt\}$. When a new task is requesting a server, a server chooses the client machine with required resources and dispatches the task to available resources. The workload migration process is done when the score of the ant is greater than the threshold limit. The timer c is set to check client machine periodically. If any client executing the workload continuously for a long time then the score of ant gets increased. Once the ant score exceeds the threshold limit, then server send migration request to a client. After receiving an acknowledgment from a client, a server performs rescheduling task to some other resources which have the capacity to execute the tasks. Workload migration function is as follows,

$$R_{j,f} = \frac{R_j \times S_j + R_{j,r} \times S_{j,r}}{S_{j,r}} \qquad (7)$$

Above function, $R_{j,f}$ represents resource j utilization of the virtual migration process, $S_{j,r}$represents the capacity of resources j. $R_{j,r}$ represents utilization of resources in a physical machine. From this function, it can estimate the utilization of the resources after task migration.

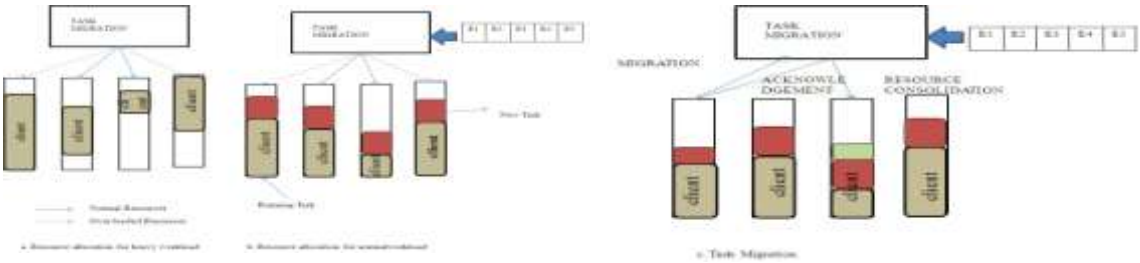

Fig. 2 : Workload Migration Process

4. EXPERIMENTAL RESULTS

The proposed algorithm is carried out using Cloud-Sim. The simulation parameters in a cloud environment are shown in Table 1 and Table 2 shows the consolidated recourse parameters details. The proposed Energy-Based Resource Allocation with Minimum Reckon and Maximum Reckon has been executed and analyzed using different criteria in the consolidating Virtual Machine. Figure 3 shows System accessibility of various user tasks. Figure 4 explains the throughput of the system for different task range. Figures 5 - 10 shows scalability, response time, availability, reliability, cost and reputation of the system for different user tasks.

Table 1 : Simulation Parameter for Proposed Algorithm

Type	Parameters	Value
Task	No.of.cloudlets	10
	Length	10000-40000
	Input size	200-600
	Output size	300-900
Resource	No.of.VM's	5
	MIPS	250-1000
	Image size	10000 MB
	RAM	256-1024 MB
	Bandwidth	1000-2000 B/S
	Pes number	1-5

Table 2 : Consolidated Resource Parameters

Resources parameters	First-to-suit scheduling					Modified Ant colony Resource Allocation with Workload migration				
	No of Tasks					No of Task				
	10	20	30	40	50	10	20	30	40	50
Time	2.1	3.5	5.6	7.0	8.5	1.4	2.8	4.2	5.6	7.0
Cost	3.9	5.9	7.9	12.4	15.7	2.8	5.2	7.0	10.4	13.0
Availability	2	4	5	8	10	3	5	9	12	15
Accessibility	1.7	3.2	4.7	6.3	8.0	2.2	3.6	4.3	7.7	9.6
Reliability	1.9	3.8	5.2	7.1	9.0	2.8	4.0	4.9	8.7	10.8
Throughput	1.1	2.2	3.5	4.7	6.0	1.8	2.6	3.3	5.7	7.1
Reputation	1.4	2.8	4.1	5.7	7.0	2.1	3.1	3.9	6.8	8.2
Scalability	2.1	4.3	6.7	8.9	11.0	3.2	5.0	6.0	10.5	13.2

Accessibility

It is the ratio of the actual amount of resource accessed by tasks to the total time taken to complete all tasks. Figure 3 details that the accessibility of resources using Ant Colony - Maximum Reckon and Minimum Reckon method is greater than 29.41% than FSS algorithm when cloudlets value is 10.

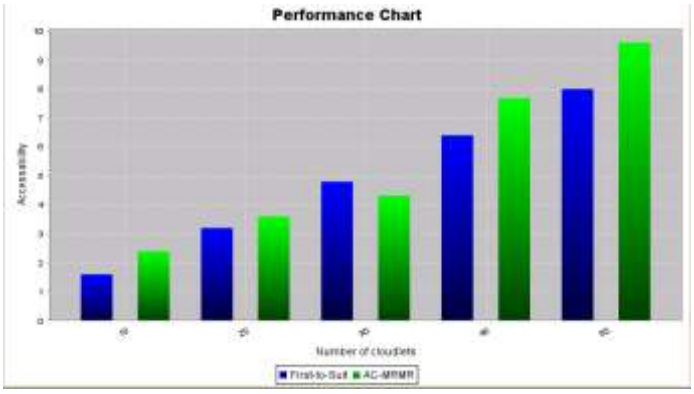

Fig. 3 : Accessibility

Throughput

Throughput is defined as a ratio of the successful tasks performed in resources to the total time taken to complete all tasks in a resource. Figure 4 shows that throughput of resources using Ant Colony - Maximum Reckon and Minimum Reckon algorithm method is 63.63% higher than FSS when cloudlets value is 10.

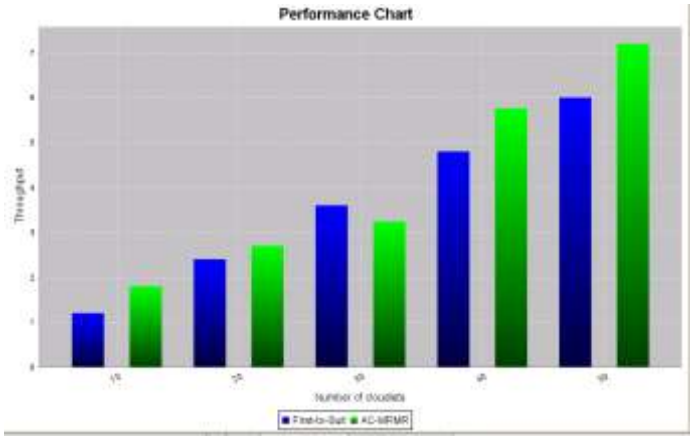

Fig. 4 : Throughput

Scalability

It is the ratio of actual number of VMs created in a resource to the total number of tasks executed in a resource. Figure 5 depicts that resource scalability by Ant Colony - Maximum Reckon and Minimum Reckon algorithm is 52.38 % higher than FSS algorithm when the cloudlet is 10.

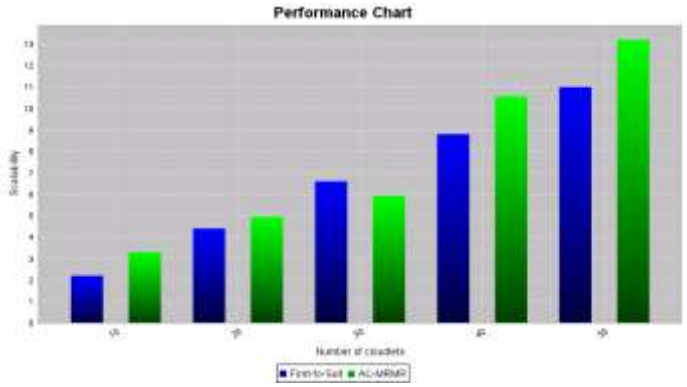

Fig. 5 : Scalability

Response Time

It is defined as a ratio of actual execution time taken by tasks in resource and the total number of tasks executed in a resource. Figure 6 presents that response time using Ant Colony - Maximum Reckon and Minimum Reckon algorithm is 33.33 % lower than FSS algorithm method when the number of cloudlets is 10

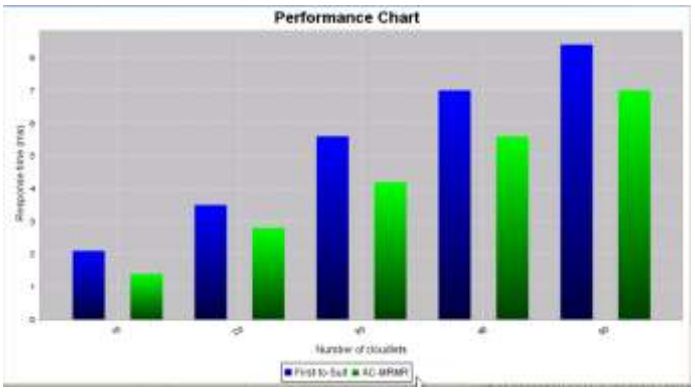

Fig. 6 : Response Time

Availability

Availability is the ratio of a difference between task completion time and resource downtime and the total time taken to complete all tasks in a resource. Figure 7 depicts that availability of resources by Ant Colony - Maximum Reckon and Minimum Reckon algorithm which is higher than 50% compare to FSS algorithm when cloudlets value is 10.

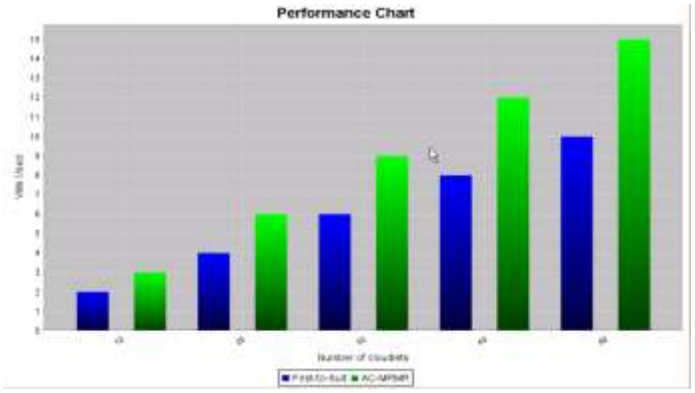

Fig. 7 : Availability

Reliability

It is defined as a ratio of a difference between task completion time and resource downtime to expected resource failure when performing tasks. Figure 8 depicts that resource reliability by Ant Colony - Maximum Reckon and Minimum Reckon algorithm that is 47.36% higher than FSS algorithm when cloudlets value is 10.

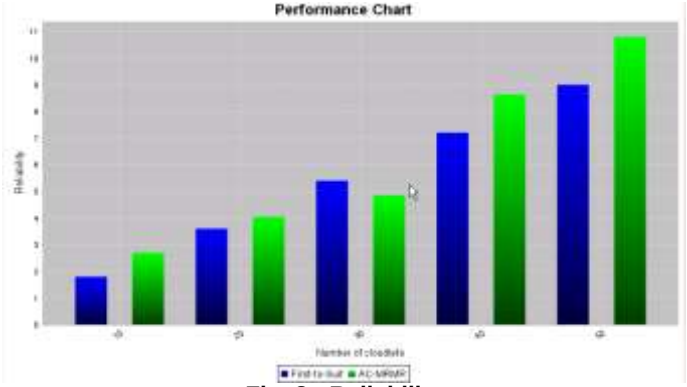

Fig. 8 : Reliability

Cost

Cost is the product of task instance cost per second and total time taken to complete all tasks in a resource. Figure 9 shows that cost of resources using Ant Colony - Maximum Reckon and Minimum Reckon algorithm is 28.2 % lesser than FSS algorithm method when cloudlets value is 10.

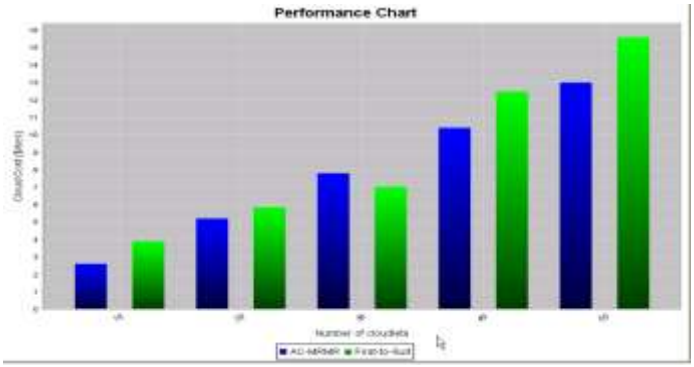

Fig. 9 : Cost

Reputation

It is defined as a ratio of actual reputation that a task calculated in resource and the expected reputation by a task in a resource. Figure10 concluded that reputation rate of resources using Ant Colony - Maximum Reckon and Minimum Reckon algorithm is 50 % better than FSS algorithm when the number of cloudlets is 10.

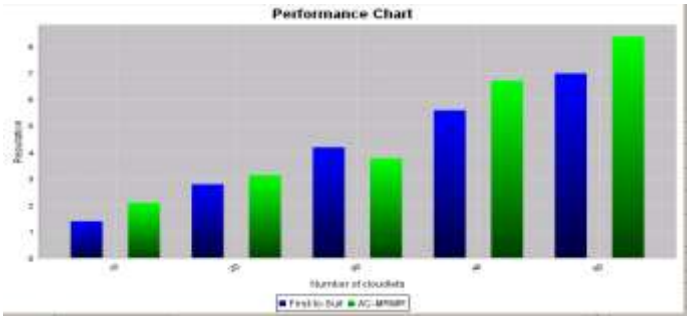

Fig. 10 : Reputation

5. CONCLUSION

Resource scheduling and Virtual machine consolidation issues were addressed in this paper. The Experimental result shows that the proposed Ant Colony - Maximum Reckon and Minimum Reckon based resource scheduling has been compared with the First-to-suit resource scheduling in all the cases. In this work, it is recognized that the proposed method has optimal resource scheduling, occupy less memory storage, and consume less energy. Simulation result shows that

the proposed technique minimize response time, cost and maximize the accessibility, scalability, availability, reliability, and reputation of the system. Security related issues have not been considered in this research. In future, such issues will be implemented in the real world cloud environment.

REFERENCES

1. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I, 2009. Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Future Generation Computer Systems, Vol.25, No.6, 599-616.
2. Choi, Yeongho, and Yujin Lim, 2016. Optimization Approach for Resource Allocation on Cloud Computing for IoT, International Journal of Distributed Sensor Networks, Vol.10, No.11, 1-6.
3. Neeraj Mangla,, and Jaspreet kaur, 2014. Resource Allocation in Cloud Computing, International Journal of Science and Research (IJSR), Vol.3, No.8, 124-128.
4. Ergu, Daji, Gang Kou, Yi Peng, Yong Shi, and Yu Shi, 2013. The Analytic Hierarchy Process: Task Scheduling and Resource Allocation in Cloud Computing Environment, The Journal of Supercomputing, Vol. 64, No. 3, 835-848.
5. S.Thamarai Selvi, C. Valliyammai, and V. Neelaya Dhatchayani, 2014. Resource Allocation Issues and Challenges in Cloud Computing, International Conference on Recent Trends in Information Technology (ICRTIT),1-6.
6. Sotomayor, Borja, Rubén S. Montero, Ignacio M. Llorente, and Ian Foster, 2009. Virtual Infrastructure Management in Private and Hybrid Clouds, IEEE Internet Computing, Vol.13, No.5, 14-22.
7. Syed Hamid Hussain MadniI., Muhammad Shafie Abd LatiffI., Yahaya CoulibalyI., and Shafi'i Muhammad AbdulhamidI, 2016. Resource Scheduling for Infrastructure as a Service (IaaS) in Cloud Computing: Challenges and Opportunities, Journal of Network and Computer Applications, Vol.68, 173-200.
8. Sanjaya K. Panda and Prasanta K. Jana, 2015. A Multi-Objective Task Scheduling Algorithm for Heterogeneous Multi-Cloud Environment, International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV), 82-87.
9. Lin Wang and Lihua Ai, 2012. Task Scheduling Policy based on Ant Colony Optimization in Cloud Computing Environment, International Conference on Logistics, Informatics and Service Science (LISS2012), 953-957.
10. Gu Ping, Xiu Chunbo, Cheng Yi, Luo Jing, and Li Yanqing, 2014. Adaptive Ant Colony Optimization Algorithm, International Conference on Mechatronics and Control (ICMC), 95-98.
11. Yangyang Dai., Yuansheng Lou., and Xin Lu, 2015. A Task Scheduling Algorithm based on Genetic Algorithm and Ant Colony Optimization Algorithm with Multi-QoS Constraints in Cloud Computing, 7th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 428-431.
12. Mishra, M., and Sahoo, 2011. A.: On Theory of VM placement: Anomalies in Existing Methodologies and their Mitigation using a Novel Vector based Approach. In: International Conference on Cloud Computing (CLOUD), 275-282.
13. Wood, T., Shenoy, P., Venkataramani, A., Yousif, M, 2009. Sandpiper: Black-box and Gray-box Resource Management for Virtual Machines, Computer Networks, Vol.53, 2923-2938.
14. X.Wang, X.Liu,L.Fan and X.Jia, 2013. A Decentralized Virtual Machine Migration Approach of Data Centers for Cloud Computing, 1-10.
15. H.Liu,C-Z. Xu, H.Jin, J: Gong and X.Liao, 2011. Performance and Energy Modeling for Live Migration of Virtual Machines, In Proceedings of the 20th ACM International Symposium on High Performance Parallel and Distributed Computing (HPDC),171-181.
16. H.W.Choi,H.Kwak,A.Sohn and K.Chung, 2008. Autonomous Learning for Efficient Resource Utilization of Dynamic VM Migration, In Proceedings of the 22nd ACM International Conference on Supercomputing (ICS), 185-194.
17. J.-G.Park,J.-M. Kim, H. Choi and Y.-C. Woo, 2009. Virtual Machine Migration in Self-Managing Virtualized Server Environments, In Proceedings of the 11th International Conference on Advanced Technology (ICACT), 2077-2083.
18. Yongqiang Gao,Haibing Guan,Zhengwei Qi,Yang Hou and Liang Liu., 2013. A Multi-objective Ant Colony System Algorithm for Virtual Machine Placement in Cloud Computing, Journal of Computer and System Sciences, Vol.79, 1230-1242.
19. Zhan,Z.H.,Liu,X.F.,Gong,Y.J.,&Zhang, J, 2015. Cloud Computing Resource Scheduling and of its Evolutionary Approaches. ACM Computing Surveys, Vol.15, No.63, 1-33.
20. Singh, S., & Chana, I, 2015. QRSF: QoS-aware Resource Scheduling Framework in Cloud Computing, JOURNAL of Supercomputing, Vol.71, 241-292.