Efficient Task Scheduling Based on Dynamic Pre Allocation with Self Adaptive Ant Colony Optimization Method for Cloud Computing

[1]Rahul Upadhyay, [2]Umesh Lilhore

[1]M.Tech Research Scholar, [2]Associate Professor and PG Coordinator
NIIST Bhopal(M.P) , India

***ABSTRACT*:-Cloud computing is a new and innovative technology which serves computing resources on pay and use concepts. Cloud computing technology supports homogeneous as well as heterogeneous environments which attract cloud users to work and utilizes its various services such as PaaS, IaaS and SaaS. Cloud computing provides optimum utilization of computing resources in affordable cost, which attracts cloud users. Day by day sizes of cloud users and cloud services are getting increase rapidly. To serve computing resources and cloud services in efficient manner to cloud user is a challenging task for cloud service providers. Load balancing and task scheduling methods plays a vital role in load distribution for cloud service provider. Various types of load balancing methods are suggested by different cloud researchers. In this research paper we are presenting a dynamic pre allocation with self adaptive ant colony optimization method for cloud computing. Proposed DPSAACO uses efficient task scheduling based on hybrid concept of load balancing. Proposed DPSAACO method and existing round robin and ant colony optimization methods are implemented on cloud sim simulator and various performance comparison are calculated such as makespan, total response time and waiting time. A simulation result clearly shows that our proposed method performs outstanding over existing method.**

***Keywords*- Cloud computing, task scheduling, load balancing, Ant colony, DPSAACO, Round robin**

I. INTRODUCTION

Cloud computing technique is a further development computing model after the technology of distributed computing, grid computing, network storage, cluster technology and parallel computing. It provides dynamic and scalable virtualization resources through network service and form a virtual computing resource pool by integrating the large-scale computing storage resources. The resources need to be provided to the users in the form of service in this way [2]. Due to the diversity of applications in cloud computing platform and the heterogeneity of server node resources, some computers are overloaded and some computers are very light when the rapid growth of network traffic and data traffic. The present studies about adjusting the server road are mostly based on the migration of a single virtual machine.
However, with the emergence and development of cloud computing and big data, when dealing with the tasks of big data computing by virtual machines in cloud computing, due to the relevance of data, the migration of some virtual machines dealing with association data will bring more communication overhead between server in the process of migration and calculation, and reduce the utilization rate of system resource [4]. A load balancing strategy based on data correlation in cloud computing is proposed in this paper in view of the deficiency of existing research. This strategy defines and calculates the correlation factor based on the internal correlation of data and the correlation of virtual machines dealing the same data, and judges the relationship between them and overall migration by constructing load intensive data group. The communication overhead between servers is reduced and the utilization of the resources is improved based on the strategy [1,5].

II. RELATED WORK

More and more researchers focus on the load balancing strategy because it is very important. A resource load balancing scheduling algorithm based on ant colony algorithm in cloud computing was proposed in literate 1, the algorithm mainly implemented to guide the excess load to the low load node. A good task scheduler should adapt its scheduling strategy to the changing environment and the types of tasks [1, 2]. Therefore, a dynamic task scheduling algorithm, such as Ant Colony Optimization (ACO), is appropriate for clouds. A genetic algorithm for load balancing strategy in cloud computing was proposed in literate 2, the algorithm realized load balance by the mathematics of task load capacity and computing power. An efficient virtual machine migration strategy in cloud computing was proposed in literature 3; this strategy reduced the unnecessary page migration by combining the compression algorithm CBC to optimize the pre replication.

The method of improved dynamic migration of virtual machine based on the pre replication reuse distance was proposed in literature 4, this method determined the transferred page for each iteration migration stage by calculating the reuse distance. A dynamic migration framework of virtual machine named vagrant was created in literate 5 based on Xen and KVM, it realized the dynamic migration of heterogeneous virtual machines. A method about exploring and memory coding was proposed in literature 6 to achieve dynamic migration strategy for virtual machines efficiently, this strategy compressed the useful pages using long coding algorithm to reduce the total time of migration. In the literature 7, a hierarchical replication algorithm was proposed to improve the dynamic migration of virtual machines, this strategy calculated the number of update pages, the threshold value and the total write interrupt [3,5,6].

A load balancing algorithm based on genetic algorithm was proposed in literature 8, the algorithm has a certain improvement in the response time and load balance degree. In literature 9, a strategy was proposed using resorting and compressing the dynamic pages to realize high performance and real time migration. These methods stabilized the load balancing strategy and improved the resource utilization in a certain degree, but they haven't taken the correlation between the data and the relationship among the virtual machines which are handing same data into account. Dorigo M. introduced the ant algorithm based on the behavior of real ants in 1996[8], it is a new heuristic algorithm for the solution of combinatorial optimization problems. Investigations show that: Ant has the ability of finding an optimal path from nest to food[4].On the way of ants moving, they lay some pheromone on the ground; while an isolated ant encounter a previously laid trail, this ant can detect it and decide with high probability to follow it. Hence, the trail is reinforced with its own pheromone. The probability of ant chooses a way is proportion to the concentration of a way's pheromone. To a way, the more ants choose, the way has denser pheromone, and the denser pheromone attracts more ants. Through this positive feedback mechanism, ant can find an optimal way finally [5].

III. LOAD BALANCING AND CHALLENGES

Based on literature survey of load balancing methods for cloud computing following challenges are identified [1,7,11] -

- **Slower Response Time-**Slower response time shows poor performance for the system.
- **Higher Execution Time-**Higher execution time shows poor performance.
- **Selection of Load balancing-**Dynamic Load balancing shows better performance.
- **Selections of partitioning method-**Many load balancing methods are based on static partitioning, which are less efficient in large environment.
- **Prediction of Task arrival patterns-**Jobs are arrive from various nodes in cloud atmosphere, so it is quite difficult to identified exact arrival pattern.
- **Priority of Task-** During load balancing it is also challenging to execute jobs priority wise.

IV. PROPOSED METHOD FOR TASK SCHEDULING

We utilize the characteristics of ant algorithms mentioned above to schedule task [1]. We can carry out new task scheduling depending on the result in the past task scheduling. It is very helpful in the cloud environment. In contrast to other ACO algorithm, the DPSAACO algorithm inherits the basic ideas from ACO algorithm to decrease the computation time of tasks executing, it also considers the loading of each VM. We can carry out new task scheduling depending on the result in the past task scheduling. It is very helpful in the cloud environment. The main contributions of this paper are as follow:

- Set dynamic parameters such as queue size, number of VMs, processors and memories.
- Given tasks and processors information, compute the pre-allocation template size for each processor
- Use roulette wheel selection algorithm to improve the resource selection mechanism.
- Apply ACO at sender as well as receiver both level.
- Improve the way getting the initial pheromone of resource and the pheromone the tasks carry. At time same time improve the way updating the pheromone.
- Schedule the combined tasks sequentially to the appropriate processor

// Proposed DPSAACO algorithm for task scheduling in cloud

Input: List of Cloudlet (Tasks) and List of VMs, k= ant
Output: the best solution for tsaks scgeduling on VMs
Step 1- Initialization of variable
Set value of Current_iteration_t=1
Set value of Current_optimal_solution=null
Set an initial value $\tau_{ij}(t)$=c for each path between tasks and VMs.
Step 2- Placethe m ants on the starting VMs randomly.
Step 3- For k =1 to m do
Place the starting VM of the k-th ant in $tabu_k$
Do ants_trip while all ants don't end their trips
The k-ant chooses VM j for next task i with a probability that is computed by Equation

$$P_{ij}^{k} = [\ \{[T_{ij}(t)]^{\alpha}\ [\acute{\eta}_{ij}]^{\beta} / \sum_{s \,€\, allowed\ k} [T_{is}(t)^{\alpha}]\ [\acute{\eta}_{is}]^{\beta}\ \} \quad \text{if } j \,€\, allowed_k \text{ ------- eq (1)}$$

Where-
$T_{ij}(t)$ = shows the pheromone concentration at the t time on the path between task i and VM j
$allowed_k=\{0,1,\ldots,n-1\}-tabu_k$ express the allowed VMs for ant k in next step and $tabu_k$ records the traversed VM by ant k
Insert the selected VM to $tabu_k$

End Do

Step 4- For k=1 to m do

Compute the length L_k of the tour described by the k-th ant by using-

$L^k(t)=\arg\max_{j\in J}\{ sum_{i\in IJ}(d_{IJ}) \}$

Update the current_optimal_solution with the best founded solution.

Step 5- For every edge (i,j), apply the local pheromone by using equation-

$T_{ij}(t)=(1-p)\ T_{ij}(t)+\ \Delta T_{ij}(t)$

Step 6- Apply global pheromone update by using –

$\Delta T_{ij}(t)=\sum^{m}_{k=1}=\ \Delta T^{k}_{ij}$

Step 7- Increment Current_iteration_t by one.

Step 8- If(Current_iteration_t < tmax)

Empty all tabu lists.

Go to step 2

Else

Print current_optimal_solution

End If

Return.

V. IMPLEMENTATION & RESULTS ANALYSIS

5.1 Cloud Sim parameters- The experiment is implemented with 20 Datacenters with 100 VMs and 50-1000 tasks under the simulation platform of cloud sim. The length of the task is from 1000 MI (Million Instructions) to 20000 MI. The parameters setting of cloud simulator are shown in Table 5.1.

S NO	Entity	Parameter Name	Value
1	CloudLets(Tasks)	No of tasks	50-1000
		Length of task	1000 to 20000 MI
2	VMs (Virtual Machines)	No of VMs	100
		MIPS	250-2500
		RAM (VM Memory)	250-3000
		Bandwidth	250-1500
		Cloud Let Scheduling method	Time Shared Method & Space Shared
		No of PEs requirements	1 to 4
3	Data Center	No of Data Center	20
		VMS Scheduler	Time Shared Method & Space Shared
		No of Hosts	1 to 10

Table 5.1 Cloud Sim parameters

5.2 ACO and DPSAACO Parameters - We implemented the existing ACO algorithm, Round robin algorithm and proposed DPSAACO algorithm and investigated their relative strengths and weaknesses by experimentation. The parameters (α, β, P, T_{max} , k shows number of aunts). The default value of the parameters was α=1, β=1, ρ=0.5, Q=100, T_{max}=150 and k=8.

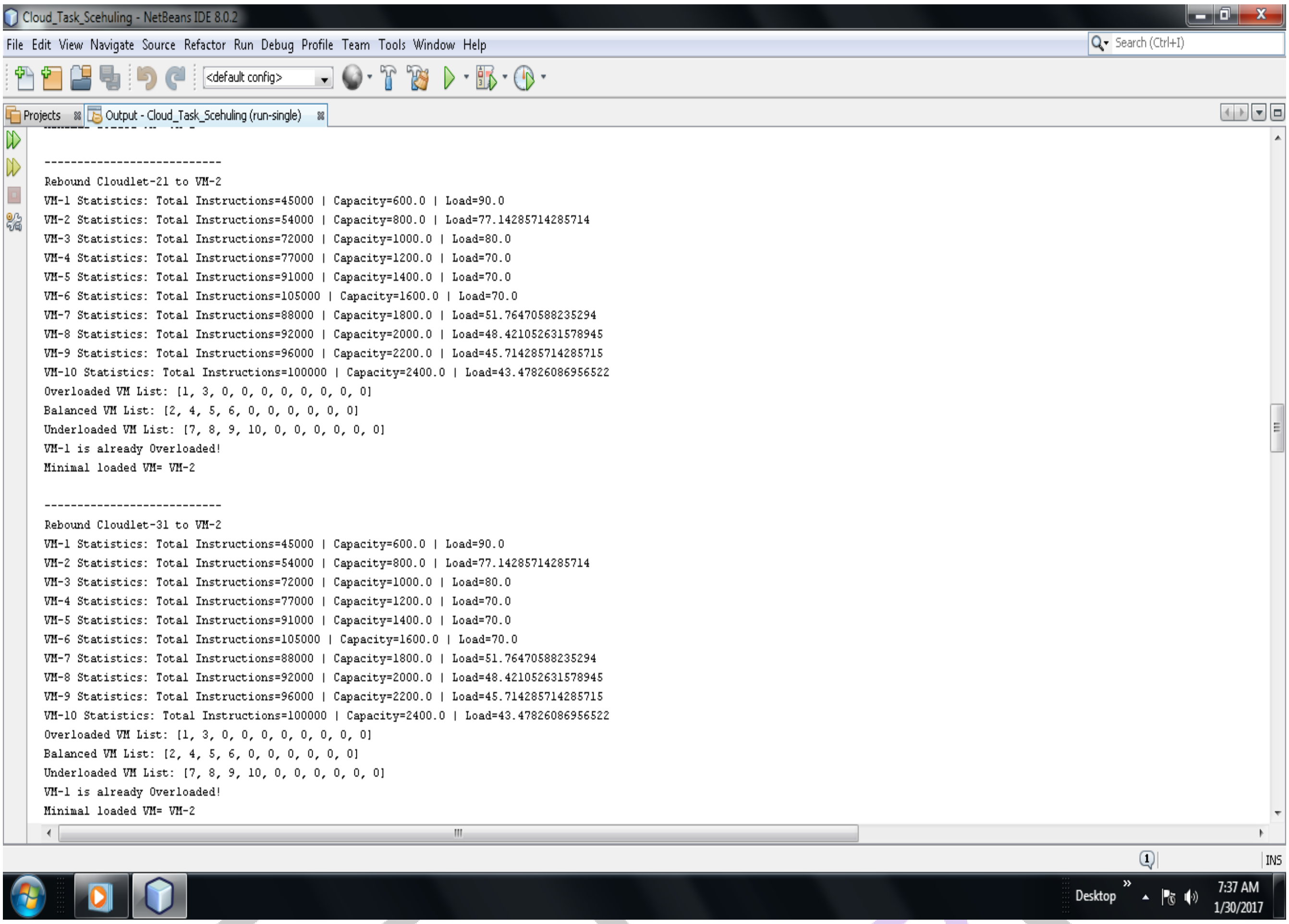
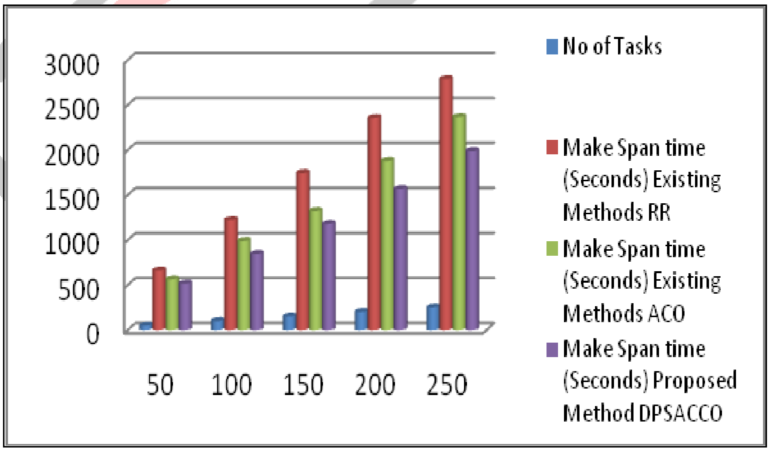


Figure 5.1 Simulation results for proposed method

5.3 Results Analysis- Following results are calculated for existing RR, ACO and proposed DPSAACO method.

5.3.1 MakeSpan Time- Make span can be defined as the overall task completion time.

No of Tasks	Make Span time (Seconds)		
	Existing Methods		Proposed Method
	RR	ACO	DPSACCO
50	660	562	515
100	1222	989	845
150	1744	1322	1178
200	2355	1878	1566
250	2788	2365	1988



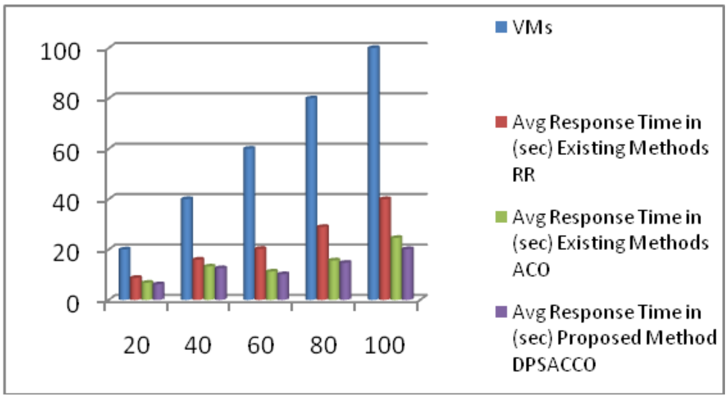
MakeSpanTimeTable **Makespan Time Compairison between Existing Methods and Proposed**

Influences- The above results and graph clearly shows that proposed method have better make span time for task 50,100,150,200 and 250 over existing round robin and ACO methods.

5.3.2 Average Response Time- It is the amount of time taken from when a process is submitted until the first response is produced. Average response times for each algorithm have decreased by increasing the number of CPUs.

VMs	Avg Response Time in (sec)		
	Existing Methods		Proposed Method
	RR	ACO	DPSACCO
20	8.75	6.8	6.2
40	15.99	13.25	12.6
60	20.25	11.25	10.2
80	28.96	15.66	14.75
100	39.99	24.6	20.15

Average Rsponse Time Table



Avg Response Time Comparison between Existing Methods and Proposed

Influences- The above results and graph clearly shows that proposed method have Avg response time time for VMs 20, 40, 60, 80, and 100 over existing round robin and ACO methods.

VI. CONCLUSIONS & FUTURE WORKS

In cloud computing load balancing plays an important role. A Load balancing method transfer task from an over loaded machine to another under loaded machine, to achieve optimum solution. This research paper presents a dynamic pre allocation with self adaptive ant colony optimization method for efficient task scheduling in cloud computing. Proposed method use pre allocation of task by using dynamic distribution and its self adaptive nature makes it more efficient. Experimental results for proposed DPSACCO have better makespan time and response time over existing methods. According to the self adaptability of the ACO parameters and the new way to calculated and update the pheromone, DPSAACO has a good performance. In future work we will implement and compare our proposed method on a realistic environment instead of just only on a simulator.

REFERENCES

[1] Weifeng Sun, Zhenxing Ji, Jianli Sun, Ning Zhang, Yan Hu," SAACO: A Self Adaptive Ant Colony Optimization in Cloud Computing", 2015 IEEE Fifth International Conference on Big Data and Cloud Computing", PP 148-154.

[2] Aarti Singh, Dimple Juneja, Manisha Malhotra," Autonomous Agent Based Load Balancing Algorithm in Cloud Computing", International Conference on Advanced Computing Technologies and Applications (ICACTA-2015), Science Direct, PP 832-841

[3] Sidra Aslam, Munam Ali Shah," Load Balancing Algorithms in Cloud Computing: A Survey of Modern Techniques", 2015 National Software Engineering Conference (NSEC 2015) IEEE, PP 30-36

[4] Geethu Gopinath P P, Shriram K Vasudevan," An in-depth analysis and study of Load balancing techniques in the cloud computing environment", 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15), Science Direct, Procedia Computer Science 50 (2015)PP 427-432.

[5] Shahrzad Aslanzadeh, Zenon Chaczko," Load Balancing Optimization in Cloud Computing: Applying Endocrine-Particle Swarm Optimization" 2014, PP 165-170

[6] M. Katyal and A. Mishra, "A comparative study of load balancing algorithms in cloud computing environment." in International Journal of Distributed and Cloud Computing, Volume 1 Issue 2 December 2013.

[7] R. G. Rajan and V. Jeyakrishnan, "A survey on load balancing in cloud computing environments," in International Journal of Advanced Research in Computer and Communication Engineering,vol. 2, no. 12, pp. 4726–4728, 2013.

[8] X. Xu, H. Yu, and X. Cong, "A qos-constrained resource allocation game in federated cloud," Seventh Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Compute. pp. 268–275, Jul. 2013.

[9] Beaumont, Casanova H, Legrand A, etc. Scheduling Divisible Loads on Star and Tree Networks: Results and Open Problems. Parallel and Distributed Systems, IEEE Transactions on, 2005, 16(3): 207-218.

[10] Characranoon S, Robertazzi T G, Luryi S. Parallel processor configuration design with processing Transmission costs. IEEE Transactions on Computers, 2000, 49(9): 987-991.

[11] D. Puthal, B. P. S. Sahoo, S.Mishra and S.Swain, "Cloud Computing Features, Issues, and Challenges: A Big Picture," 2015 International Conference on Computational Intelligence and Networks (CINE), pp. 116-123, Bhubaneshwar, 2015.

[12] T. Liao, K. Socha, M. A. Montes de Oca, T. Stutzle, M. Dorigo, "Ant colony optimization for mixed-variable optimization problems," IEEE Transactions on Evolutionary Computation, vol 18, NO. 4, pp. 503-518, 2014.

[13] Z. H. Xu, J. Z. Sun . “A Review of Job Scheduling for Grid Computing. Computer Application,” vol .25, NO. 10, pp. 2236-2237, 2005.

[14] M. A. Tawfeek, A. El-Sisi, A. E. Keshlk F, A, Torkey, “Cloud Task Scheduling Based on Ant Colony Optimization,” 2013 8th International Conference on Computer Engineering & Systems (ICCES), pp. 64-69, Cairo, 2013.

[15] W. F. Sun, N. Zhang, H. T. Wang, W. J. Yin, T. Qiu, “PACO: A Period ACO Based Scheduling Algorithm in Cloud Computing,” 2013 International Conference on Cloud Computing and Big Data (CloudCom-Asia), pp. 482-486, Fuzhou, 2013.