iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 Issue: VI Month of publication: June 2018

DOI: http://doi.org/10.22214/ijraset.2018.6298

An Efficient Cloud Server Resource Scheduling using Round Robin and Ant Colony Optimization Algorithm

Snehil Agrawal[1], Prof. Sumit Nigam[2]
[1, 2]M Tech Scholar, Computer Science and Engineering, Indore 452001, India

Abstract: Cloud computing is a computational and storage infrastructure. A number of applications and growing technologies are getting resources from the cloud infrastructure. Resources in cloud computing are not statically distributed that are shared between all the applications therefore according to the demand of applications the cloud schedulers provides the resources to the jobs or processes. The schedulers are works to evaluate the available resources and manage the resource allocation to jobs for efficiently execution. But if the resource management is not performed in proper scenario then the job execution cost and process waiting time is increases in significant amount. In this context the proposed work is dedicated to find an efficient method for scheduling jobs according to the available resources. Therefore a hybrid technique that incorporates the properties of round robin technique and ACO (ant colony optimization) technique is developed. the proposed technique first categorize the entire work load in three main classes low, average, and high work load. Additionally then after the ACO optimization technique is employed for finding best match of resource sequence and jobs to maximize the job execution. The implementation of the proposed technique is performed using cloudSim simulator and their performance is computed in terms of waiting time, CPU utilization and execution time. The obtained performance of the proposed system shows efficient management of resources and low resource consumption.

Keywords: Ant Colony Optimization, Space Shared, Virtual Machine Scheduling, Cloud Computing, Data Center, Round Robin

I. INTRODUCTION

Cloud computing is not a just software or hardware platform it is a computational infrastructure. The cloud infrastructure involves computational units, storage units and the efficient network. Cloud computing is basically popular for efficient and scalable computing and storage resources. These resources are help to execute the end client submitted task. On the other hand the due to large number of requests the availability of resources are fluctuating. The availability of resources is impact on the jobs execution time and cost of job processing. Therefore a suitable technique is required that evaluate the available resources and the reached jobs for processing to efficiently allocate the resources to the jobs. This phenomenon is known as resource scheduling. In this presented work the resource scheduling technique is key area of study. In addition of that the effort is made to optimize the performance of resource scheduling. In cloud computing the resources are much expensive and their availability is depends on the traffic or workloads on the cloud server. The misuses of resources impact the performance and the cost of exaction of the jobs. Therefore resource management and better utilization of available resources are a key challenge of efficient cloud computing. In this presented work the different cloud resource scheduling or job allocation techniques are studied and a new soft computing based technique is proposed for the efficient job scheduling. In this context the proposed work leads to design and development of a new resource scheduling technique that combines the goodness of round robin and Ant colony optimization technique to deploy and efficient resource scheduling technique.

II. PROPOSED WORK

This chapter provides the detailed understanding about the proposed cloud resource scheduling algorithm for improving the cloud server performance. In this context the chapter includes the system overview, detailed methodology of system design and proposed algorithm.

A. System Overview

Cloud computing infrastructure is offers the shared resources for computing and storage. These resources are used when jobs are submitted to cloud for processing. Basically the cloud server includes a resource manager that manager evaluate the jobs appeared for execution. Additionally according to the requirements of the jobs the resources are allocated to the jobs. If the resource allocation is not much effective then the problem of resources are occurred. Due to this the execution time of jobs are increases and the cost of execution of jobs are also affected. Therefore the scheduling of jobs is a primary technique that helps to improve the performance of cloud servers. During the study there are different cloud resource scheduling techniques and load balancing techniques are studied. Most of the techniques are broadly categorized in two categories. In first kinds of technique the scheduling system only considers the list of appeared jobs and for each job a resource is randomly allocated. In second technique the scheduling algorithm considers both the resource list and appeared jobs and using both the lists the decision is made for allocation. Such kind of techniques are effectively manages the resources and appeared jobs. Therefore the proposed technique is conceptually similar to the second technique where both the lists are evaluated for finding the best suitable combination of jobs and resources for efficiently processes the work load appeared on the cloud servers. This section provides the basic of the proposed concept in next section the proposed methodology is explained.

B. Methodology

The proposed technique of resource scheduling is demonstrated in figure 2.1. Additionally their components are explained with their functional aspects.

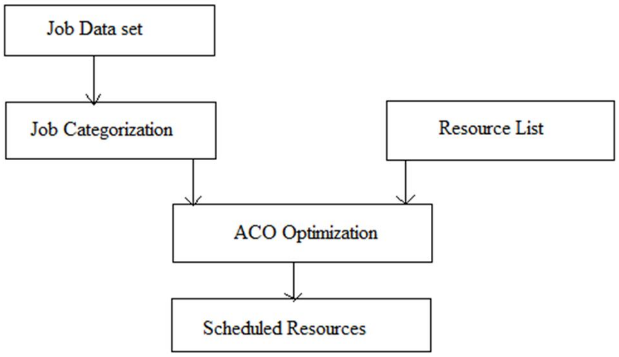

Figure 2.1 proposed system

The job dataset consist of a list of jobs with some additional information such as the job processing length and the required resource. In addition of that the job list has a random organization of job appearance. Therefore first the job dataset is process using the proposed system for categorizing the jobs in three major categories:

1) low load
2) average load
3) high load

In order to categorize the data in three categories threshold is developed first the given technique of dataset categorization is demonstrated below:

```
Input: dataset D
Output: categorized list $L_{low}, L_{avg}, L_{high}$

Process:
    1. $max = findMaxjob(D)$
    2. $min = findMinjob(D)$
    3. $th = \frac{max-min}{2}$
    4. $for(i = 1; i \leq D.length; i++)$
        a. $if\left(D_i > \min \&\& D_i \leq \frac{min+th}{2}\right)$
            i. $L_{low}.Add(D_i)$
        b. $else\ If\left(D_i \geq \frac{min+th}{2}\ \&\&\ D_i < \frac{max-th}{2}\right)$
              i. $L_{avg}.Add(D_i)$
        c. $else$
              i. $L_{high}.Add(D_i)$
        d. $end\ if$
    5. $end\ for$
```

Table 2.1 load categorization

After categorizing the load in three categories the available resources are need to be considered. Let the available resources are$R = \{r_1, r_2, \dots r_n\}$. Then we need to find the set of n jobs which are effectively satisfied by the combination of the resources R. therefore using the three developed list random sequences are developed with the length of n and total of m as the population. This list of solutions is processes using ACO algorithm to find the best combination of resources and jobs for execution.

```
Input: load categories $L_{low}, L_{avg}, L_{high}$, list of resources $r_1, r_2, \dots r_n$, total population M
Output : optimal schedule O

Process:
    1. $for(i = 1; i \leq m; i++)$
        a. $J_n = selectJobs(L_{low}, L_{avg}, L_{high})$
        b. $POP.add(J_n)$
    2. $end\ for$
    3. $O = ACO.optimize(POP, R)$
    4. Return O
```

Table 2.2 optimization process

III. RESULT ANALYSIS

The implementation of the proposed hybrid load balancing approach is described in previous section. This section includes the detailed understanding about experimental evaluation and performance of the proposed system. The essential parameters are used for evaluation are listed with observed values.

A. CPU Utilization Time

In order to execute an individual process for that an amount of time is consumed. This time of execution is termed as CPU Utilization. The CPU Utilization time can be calculated by using following formula:

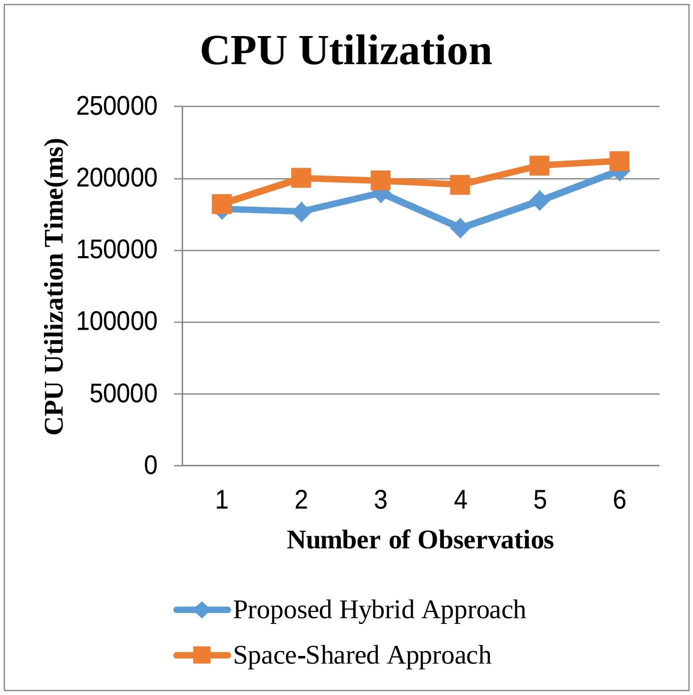

Figure 3.1: Comparative CPU utilization

$$\text{CPU}_{\text{ut}} = 100\% - (\%\text{ of time spent in idle task})$$

$$\%\text{ time in idle task} = \frac{(\text{Average Period of background task with no load}) * 100}{\text{Average Period of background task with some load}}$$

Where, CPU_{ut} = CPU Utilization Time

The figure 3.1 and table 3.1 shows the CPU utilization time both implemented algorithm i.e. proposed and traditional approaches of load balancing. In given figure X axis contains number of observation of scheduling and the Y axis contains the amount of CPU Time consumed. In each of the load balancing approach the results analysis for the CPU utilization time is given in terms of milliseconds. According to the comparative performance analysis the proposed technique needs less CPU cycles to execute the task as compared to the traditional space shared. Therefore CPU utilization is comparable by their result demonstration. Additionally we show the tabular values of for both approaches in table 3.1.

Table 3.1: Tabular values of CPU Utilization

Number of Observation	Proposed Hybrid Approach(ms)	Space-Shared Approach(ms)
1	178643	182301
2	176891	200250
3	189986	198561
4	165562	195561
5	184801	208941
6	205561	212136

B. Execution Time

The execution time sometimes also called the runtime can be defined as the amount of time required during a program is executing, in context to other program lifecycle such as compilation time, and/or load time. Additionally the comparative performance of both the approaches i.e. ACO based hybrid load balancing approach and traditional space shared is demonstrated using the figure 3.2 In the given performance graphs the X axis shows the number of observation during the code execution and the Y axis contains the execution time in terms of milli-seconds.

Table 3.2: Tabular values of Execution Time

Number of Observation	Proposed Hybrid Approach(ms)	Space-Shared Approach(ms)
1	186525	193670
2	206526	210265
3	216598	225489
4	208561	234781
5	220215	245898
6	235501	256548

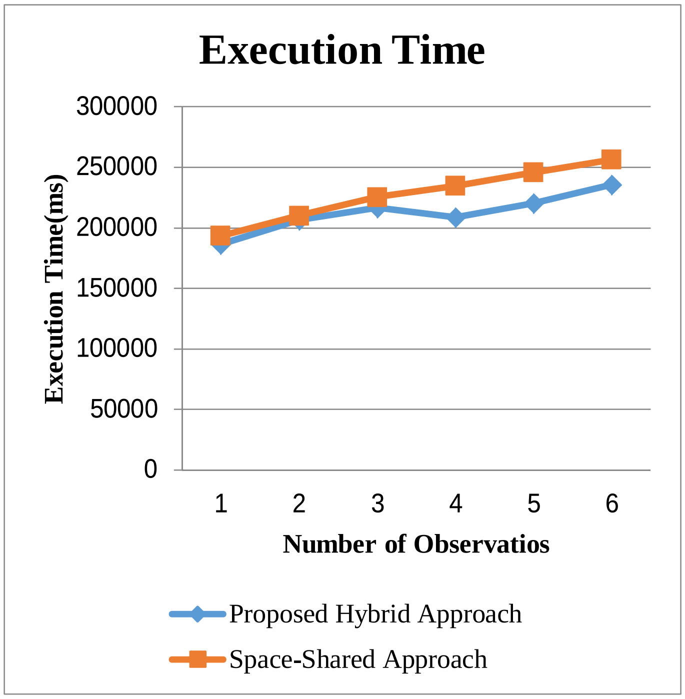

Figure 3.2: Comparative Execution Time

In the given graph figure 3.2, comparative performance is represented in terms of execution time, in this diagram the blue line shows the performance of the proposed technique and the orange line shows the performance of the traditional space shared based approach. According to the experimental results the proposed technique needs less time for load or execute a process with respect to the traditional technique. Therefore the proposed technique is comparatively more efficient and adoptable as compared to the traditional technique.

C. Waiting Time

The waiting time is the amount of time required to schedule the job for execution after placing the job execution request in the waiting queue. The waiting time of the process execution is reported in this section, figure 3.3 shows the comparative waiting time in terms of milli-seconds.

Table 3.3: Tabular values of Waiting Time

Number of Observation	Proposed Hybrid Approach(ms)	Space-Shared Approach(ms)
1	226512	235569
2	215648	239584
3	198951	248851
4	203654	253621
5	206546	245610
6	210215	228791

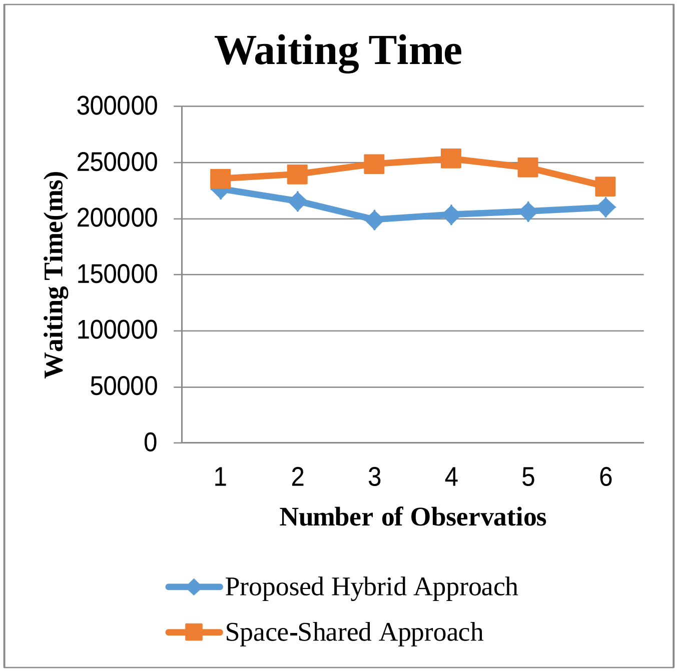

Figure 3.3 Comparative Waiting Time

To demonstrate the performance of implemented approach, X axis of diagrams depicts the different program execution and the Y axis shows the waiting time of the processes in terms of milliseconds. According to the comparative performance analysis as given in figure 3.3 the proposed technique needs less amount of time for waiting in wait queue. Therefore the proposed technique is adaptable as compared to the traditional approach.

IV. CONCLUSION AND FUTURE WORK

This chapter provides the summary of the proposed study conducted. Therefore based on the performance and implementation the conclusion of work is reported additionally the feasible extension of the work is also given in this chapter.

A. Conclusion

Cloud infrastructure is a computational and storage platform for the different applications. A significant amount of applications and processes are taking advantages of this infrastructure. But due to mismanagement of resource on cloud various serious issues are occurred for example if the server is running and no jobs are appeared for execution the running cost is required to maintain. If large amount of jobs are appeared and less resources are available then need to borrow additional resources from other infrastructure. All these situations increase the cost of cloud computing. Therefore need an optimal manner by which the maximum jobs are get satisfied and low execution cost required to pay.

In this context the proposed work investigates about the different cloud resource management and scheduling techniques. The scheduling techniques are the intelligent techniques by which the decisions about the job allocation to a resource are conducted. In order to perform such task the proposed technique implements a hybrid resource scheduling technique for computational cost reduction. The proposed technique combines the goodness of round robin technique and ACO algorithm for developing a new technique. Therefore first the entire appeared jobs are categorize in three classes and then using the categorized task the ACO scheduling is performed. Finally the best combination of jobs and resources are used for allocation of resources to the appeared jobs. the proposed method scale the performance of cloud resources and improves the jobs waiting time. The implementation of the proposed approach is performed using JAVA technology and by using the CloudSim simulator. After the implementation of required simulation the performance of the proposed technique is evaluated and compared with the traditional space shared technique. The comparative performance study is demonstrated using table 6.1.

S. No.	Parameters	Proposed technique	Traditional technique
1	CPU Utilization	High	Low
2	Processing time	Low	High
3	Waiting time	Low	High

Table 4.1 performance summary

According to the obtained performance as listed in table 4.1 the proposed technique enhances the performance of computational cloud by optimizing the resource scheduling. Therefore the proposed technique is acceptable for real world use.

B. Future Work

The main aim of the proposed work is to design and develop a soft computing based technique for cloud resource scheduling is completed successfully. The proposed technique can also extended for the following areas:

1) The proposed technique utilizes the concept of round robin and ACO (ant colony optimization) in near future more techniques are explored and for improving the technique more efforts are placed

2) The proposed work is currently dedicated for improving resource scheduling in near future more parameters are collected to improve the scheduling strategy.

REFERENCES

[1] Mesbahi, Mohammad Reza, Mahnaz Hashemi, and Amir Masoud Rahmani, "Performance evaluation and analysis of load balancing algorithms in cloud computing environments", 2016 Second International Conference on Web Research (ICWR). IEEE, 2016, pp. 145-151

[2] Karandeep Kaur. “A Review of Cloud Computing Service Models”, International Journal of Computer Applications (IJCA), Volume 140 – No.7, April (2016)

[3] Kelvin, “Basic Overview on Cloud Computing”, Online: https://www.hostdepartment.com/blog/2014/08/05/cloud-computing/ [Accessed on 23 June 2017].

[4] “What is cloud computing? A beginner's guide”, online available at: https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/

[5] “The Benefits of Cloud Computing”, Dynamic Infrastructure July (2009), available online at: https://www.ibm.com/ibm/files/H300444G23392G14/13Benefits_of_Cloud_Computing_634KB.pdf

[6] Buyya, Rajkumar, and Karthik Sukumar. "Platforms for building and deploying applications for cloud computing." arXiv preprint arXiv: 1104.4379 (2011).

[7] BOX, B. "Cloud computing in telecommunications." Available online at: http://www.ramonmillan.com/documentos/bibliografia/CloudComputingInTelecomunications_Ericsson.pdf

[8] Ajay Gulati, and Irfan Ahmad, “Cloud-scale resource management: challenges and techniques, In Proceedings of the 3rd USENIX conference on Hot topics in cloud computing, Hot Cloud’11, pages 3–3, Berkeley, CA, USA, 2011, USENIX Association.

[9] Lombardi, Michele, and Michela Milano, "Optimal methods for resource allocation and scheduling: a cross-disciplinary survey", Constraints 17.1 (2012): 51-85.

[10] B. Sotomayor, R.S. Montero, I.M. Llorente, and I. Foster, “Virtual infrastructure management in private and hybrid clouds”, Internet Computing, IEEE, 13(5):14 –22, sept.-oct. 2009.

[11] http://www.amazon.com/b?ie=UTF8&node=201590011. Amazon Elastic Compute Cloud (Amazon EC2).

[12] Li, Jiayin, Meikang Qiu, Jian-Wei Niu, Yu Chen, and Zhong Ming. "Adaptive resource allocation for preemptable jobs in cloud systems." In Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on, pp. 31-36. IEEE, 2010.

[13] P.Ruth, J.Rhee, D.Xu, R.Kennell and S.Goasguen, “Autonomic Adaptation of virtual computational environments in a multi-domain infrastructure”, IEEE International conference on Autonomic Computing, 2006, pp.5-14.

[14] RerngvitYanggratoke, FetahiWuhib and Rolf Stadler: Gossip-based resource allocation for green computing in Large Clouds: 7th International conference on network and service management, Paris, France, 24-28 October, 2011.

[15] HadiGoudaezi and MassoudPedram: Multidimensional SLA-based Resource Allocation for Multi-tier Cloud Computing Systems, IEEE 4th International conference on Cloud computing 2011, pp.324-331.

[16] V. Vinothina and Dr. R. Sridaran, “A Survey on Resource Allocation Strategies in Cloud Computing”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No.6, 2012.

[17] Ali M. Alakeel, “A Guide to Dynamic Load Balancing in Distributed Computer Systems”, IJCSNS International Journal of Computer Science and Network Security, Volume 10 No.6, June 2010.

[18] Sharma, Eep, Sarabjit Singh, and Meenakshi Sharma, "Performance Analysis of Load Balancing Algorithms", in proceeding of 38th World Academy of Science, Engineering and Technology, 2008.

[19] Abhijit A. Rajguru, S.S. Apt” A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters”, International Journal of Recent Technology and Engineering (IJRTE), Volume-1, Issue-3, August 2012.

[20] Mohsen and HosseinDelda, “Balancing Load in a Computational Grid Applying Adaptive, Intelligent Colonies of Ants”, Informatica 32, PP. 327–335, 2008.