Efficient Qos Based Resource Scheduling Using PAPRIKA Method for Cloud Computing

Hilda Lawrance*
Post Graduate Scholar
Department of Information Technology, Karunya University
Coimbatore, Tamilnadu, India 641114
Ph no: +917667882689, hildalawrance@gmail.com

Dr. Salaja Silas
Assistant Professor (SG),
Department of Information Technology, Karunya University
Coimbatore, Tamilnadu, India 641114
salaja_cse@karunya.edu

Abstract :

Cloud computing is increasingly been used in enterprises and business markets for serving demanding jobs. The performance of resource scheduling in cloud computing is important due to the increase in number of users, services and type of services. Resource scheduling is influenced by many factors such as CPU speed, memory, bandwidth etc. Therefore resource scheduling can be modeled as a multi criteria decision making problem. This study proposes an efficient QoS based resource scheduling algorithm using potentially all pair-wise rankings of all possible alternatives (PAPRIKA). The tasks are arranged based on the QoS parameters and the resources are allocated to the appropriate tasks based on PAPRIKA method and user satisfaction. The scheduling algorithm was simulated with cloudsim tool package. The experiment shows that, the algorithm reduces task completion time and improves resource utility rate.

Keywords: cloud computing; QoS; resource scheduling; multi criteria decision.

1. Introduction

Cloud computing is the rising technology that delivers many forms of resources as services, mainly over the internet. It permits customers to use applications without deployment and access the required files at any computer using internet [3]. Cloud Computing allows on-demand resource provisioning. It is the convergence of several concepts such as virtualization, distributed application design, grid, and enterprise IT management. It enables a more flexible approach for deploying and scaling applications [2].

In cloud, consumers can evade capital payments on hardware, software, and services. The users will pay the provider only for what they use, which reduces the complexity and cost for trading, configuring, and managing the hardware and software needed for applications. The applications are delivered as a service over the Internet. Cloud Computing considers both the applications provided as services over the Internet and the hardware in the data centers that provide those services such as CPU, memory, data storage and networking equipments.

In cloud computing, different consumers need different types of resources. The demand for the resources changes with the time and user. Scheduling of resources is one of the key areas in resource management. Resource scheduling is the allocation of resources that will be available to serve load in a given area or of a given utility. Because of different QoS parameters such as CPU speed, stability, memory etc., resource scheduling in cloud computing is different from conventional distributed computing environment. The demand for resources fluctuates dynamically so scheduling of resources is a difficult task. QoS based resource scheduling is necessary for efficient resource utilization, reducing cost, and for satisfying user requirement.

Most of the algorithms deal only about cost and a single QoS (Quality of Service) parameter. This paper proposes an algorithm that considers user satisfaction and Multi QoS parameters simultaneously for the allocation of resources to task. For the proper arrangement of tasks, potentially all pair-wise rankings of all possible alternatives (PAPRIKA), a multi criteria decision making method had been adopted. This increases the efficiency of allocation of resources in terms of maximize resource utility and minimize allocation time.

Literature review is discussed in section 2. Section 3 tells the factors that affect scheduling and Section 4 presents the design model for resource scheduling algorithm. Finally, Section 5 concludes this paper.

2. Literature Review

Linlin Wu et al., 2011, [6] had proposed ProfmaxVmMinAvaiSpace algorithm for the allocation of cloud resources for SaaS providers. The algorithm chose VM type which matched the request parameters using mapping technique. If there was an initiated VM which satisfied the request, then the algorithm checked whether it had enough space and allocated it to a VM in best fit manner. The algorithm maximized provider profit and reduced SLA violation.

Qi Zhang et al., had proposed a method for allocating resources dynamically to data centers based on fluctuating market demands [8]. It was based on periodic analysis of market demands for resources and allocated resources according to the market demand. This would increase cloud providers total revenue. Xindong YOU et al., 2009 [11] had proposed a resource allocation strategy based on market mechanism. It consisted of three modules: Consumer agent (CA), Resource agent (RA) and Market economy mechanism. CA assigned the consumer that had to participate in the market system and aimed to obtain maximal benefit for the consumer. RA assigned one type of resource to publish the resource's price and adjusted the price according to the relationship of supply and demand in the market system. Market Economy Mechanism was responsible for the balance of resource supply and the demand. This would balance supply and demand of resources and would utilize resource efficiently.

Abirami S.P. et al., 2012, [1] had proposed a scheduling algorithm named as Linear Scheduling for Tasks and Resources (LSTR), which performed tasks and resources scheduling respectively. The method collected requests in every pre-determined interval of time and found the initial threshold which was the sum of all resources. Two matrices were created and according to the request type, the requests would add to either upper matrix or lower matrix. The requests allotted to resources according to the order and threshold value. The method had high system throughput but the waiting time for processing was high. Nagendram S. et al., [7] had proposed Multi-dimensional Resource Integrated Scheduling algorithm for efficient resource scheduling in data centers. The author discussed scheduling of resources based on multi-dimensional resource requests. The algorithm calculated the relationship between multi-dimensional resource supply and demand and based on that assigned priority to jobs which required less scarce resources. The method had high system efficiency and application performance but required high processing time.

Hai Zhong et al., [5] had proposed an approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems. The algorithm allocated VM for the maximum usage of physical resources. Scheduling model was divided into three steps. First, scheduler updated the available resource list when allocation or de-allocation happened and updated the VM request list when each time new VM requests came. And then, scheduler used Improved Genetic Algorithm to find out fitness and economical allocation. Finally, cloud launched the corresponding VMs at the physical resource and suspended the VMs when the leasing time's up. If there was no enough resource for the VM request, the scheduler would reject the request automatically. Here, the speed of scheduling was high comparatively.

Gunho Lee et al., had proposed Heterogeneity-Aware Resource Allocation algorithm for the allocation of resources in a heterogeneous environment [4]. The nodes were divided into two pools. First was long-living core nodes to host both data and computation and another was accelerator nodes that were added to the cluster temporarily when extra computing power was required. The utilization of core nodes were minimal but had high computing rate.

Zhongni Zheng et al., [12] had proposed Cloud Resource Scheduling Based on Parallel Genetic Algorithm. There were three main steps in the scheduling. First, the system set an idle resource list and VM request list, update them at the initiate time and each time new VM requests came or VMs were shutdown or physical resource change were being detected. The parallel genetic algorithm, found the optimal allocation sequence. The specified physical machines were launched by the VM requests. This would take less time for finding best allocation.

Wuqi Gao et al., 2012, had proposed Cloud Simulation Resource Scheduling Algorithm Based on Multi-Dimension Quality of Service that discussed task based VM scheduling by analyzing multiple quality of service parameters [10]. The approach took values of all quality of service (QoS) parameters for every task and considered as task matrix. Same way VM QoS parameters were taken and resource matrix was obtained. Analytic Hierarchy Process (AHP), was used to arrange tasks and VMs according to order of QoS weight. Next step was finding user satisfaction. For that, it took first task in the task line, computed the satisfaction of task at every VM and found out the resource which gave highest satisfaction to this particular task. If the particular VM was free, allotted task to that VM or searched for next VM from the list. Tasks were deleted from the task line, as soon as it had completed processing and repeated the same procedure until the task list was blank. The method considered multi QoS parameters such as CPU speed, bandwidth, stability and length as the scheduling factors and considered user satisfaction for allocation of the resources which improved the resource utility rate.

3. Factors Affecting Resource Scheduling

The factors that affect resource scheduling are as follows [10]

CPU Speed (C) - It is the measure of a computer's processor speed. It measures the number of operations that could be done per second.

Bandwidth (B) - It is the data transfer rate, the amount of data that can transfer from one host to another.

Stability (S) - It represent the current task status code. It returns a string value which indicates whether the task status is created, ready, in execution, success, queued, failed and so on.

Length (L) - It is the length of the task. It measures in bytes.

4. Resource Scheduling Using PAPRIKA (RS-PAPRIKA algorithm)

The proposed system performs task based scheduling of resources by using a multi-criteria decision making method called potentially all pair-wise rankings of all possible alternatives (PAPRIKA). This system takes multiple QoS parameter values for both tasks and resources. Using PAPRIKA method, finds priority for resources. Here, both resource and task QoS parameter values are taking and creating resource matrix and task matrix respectively [10]. For calculating priority for the resources, multi-criteria decision making method is using. PAPRIKA method particularly applies to additive multi-attribute value models with performance categories such as points. It is based on the fundamental principle that an overall ranking of all possible alternatives represent able by a given value model. Creating a threshold value for all the QoS parameters and doing pair-wise comparison of all QoS parameters of resources and finding priority for each resource.

After finding the priority by using pair wise comparison, arranging the resources in the order. Taking the first task from the task line and finding the user satisfaction of that task with all resources [10]. The following is the equation for finding user satisfaction.

$$Satisfaction = \begin{cases} \frac{rQ_{i,j}}{tQ_{i,j}}, rQ_{i,j} < tQ_{i,j} \\ 1, \ rQ_{i,j} \geq tQ_{i,j} \end{cases} \quad (1)$$

In eq. (1), $rQ_{i,j}$ represents service ability provisioned by QoS parameter at dimension J in resource Ri. $tQ_{i,j}$ stands for need volume by QoS in dimension J for task I. After calculating user satisfaction of all the resources, finds the resource which gives highest satisfaction for that particular task. If the particular resource is free, then allocates that particular resource to the task. Repeating this same procedure until all the tasks in task line finishes. When allocation finishes for the particular task, deletes that task from task line.

The proposed method has several advantages. It optimizes the allocation of task to appropriate resources in minimum time. Another is the improved resource utility rate.

Algorithm is as follows:

Step 1: Set up task matrix $T_{n,k}$ with required QoS parameter values (C, B, S, L) for the tasks that to be scheduled.

$$T_{n,k} = \begin{bmatrix} t_{1,1} & \cdots & t_{1,k} \\ \vdots & \ddots & \vdots \\ t_{n,k} & \cdots & t_{n,k} \end{bmatrix}$$

Step 2: Set up resource matrix $R_{m,k}$ with QoS parameter values (C, B, S, L) of all virtual machines in the available resource set.

$$R_{m,k} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,k} \\ \vdots & \ddots & \vdots \\ r_{m,k} & \cdots & r_{m,k} \end{bmatrix}$$

Step 3: Set up threshold value for resource QoS parameters.

Step 4: Finding priority for the resources by using PAPRIKA method

- Step 4.1: Take each resource QoS parameters.
- Step 4.2: Compare each QoS parameter values with threshold value.
- Step 4.3: Assigning point value to each parameter based on the comparison.
- Step 4.4: Calculating the sum of each parameter points and finding the point value for each resource.
- Step 4.5: Arranging resources according to the priority values.

Step 5: Arranging tasks based on the QoS need volume of each tasks.

Step 6: Taking first task in the task line and computing user satisfaction of this task at every resource, then find out the resource which gives highest satisfaction to this task.

Step 7: If the particular resource is free, allocate it to task. Search for the next resource in the case if particular resource is not available.

Step 8: Delete task from task line, after the completion of allocation.

Step 9: Check whether task line is blank. If not repeat from step 5.

5. Simulation Results

The cloud environment was simulated by using CloudSim toolkit in Java platform. CloudSim [9] provide a generalized and extensible simulation framework that enables modeling, simulation, and experimentation of emerging Cloud computing infrastructures and application services, allowing its users to focus on specific system design issues that they want to investigate, without getting concerned about the low level details related to Cloud-based infrastructures and services. Ten virtual machines and cloudlets and two data centers were created in CloudSim to perform scheduling. Data center broker binds every single task to single fixed VM to be operated.

The users who require cloud resources were submitting the tasks as cloudlets. Each cloudlet had certain parameters such as CPU speed, stability of cloudlet, bandwidth and length. These parameters called as Quality of Service (QoS) parameters. In each virtual machine (VM), the QoS parameters values were added and both the cloudlet and VM were sent to data center broker for binding. The simulation section mainly concentrates on task finishing time and resource utility rate.

Comparison of task completion time: While comparing the analysis result of both PAPRIKA method and S-CSRSA algorithm, it shows, scheduling of resources based on PAPRIKA method takes fewer tasks allocation time. Figure 1 and 2 shows the analysis result of both the methods. While the scheduling based on PAPRIKA method takes time only 241.86ms, the second method takes 450.2ms. The RS-PAPRIKA method reduces the task allocation time.

```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID      Start Time    Time    Finish Time
    8         SUCCESS         2             5          15         0.2          15.2
    2         SUCCESS         3             9          37.5       0.2          37.7
    3         SUCCESS         3             8          66.67      0.2          66.87
    5         SUCCESS         3             8          75         0.2          75.2
    7         SUCCESS         2             0          107.14     0.2          107.34
    9         SUCCESS         2             3          110        0.2          110.2
    6         SUCCESS         2             4          150        0.2          150.2
    4         SUCCESS         2             1          350        0.2          350.2
    0         SUCCESS         2             6          400        0.2          400.2
    1         SUCCESS         2             6          450        0.2          450.2
*****Datacenter: Datacenter_0*****
User id        Debt
4              7179.2
**********************************
*****Datacenter: Datacenter_1*****
User id        Debt
4              3076.8
**********************************
Simulation finished!
```

Fig 1: Simulation result of S-CSRSA algorithm

```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID      Start Time    Time    Finish Time
    5         SUCCESS         2             2          31.25      0.2          31.45
    4         SUCCESS         3             7          70         0.2          70.2
    3         SUCCESS         2             0          85.71      0.2          85.91
    2         SUCCESS         2             4          100        0.2          100.2
    8         SUCCESS         2             1          149.99     0.2          150.19
    6         SUCCESS         2             0          157.14     0.2          157.34
    0         SUCCESS         3             8          200        0.2          200.2
    7         SUCCESS         2             0          200        0.2          200.2
    1         SUCCESS         3             8          233.33     0.2          233.53
    9         SUCCESS         3             8          241.66     0.2          241.86
*****Datacenter: Datacenter_0*****
User id        Debt
4              7179.2
**********************************
*****Datacenter: Datacenter_1*****
User id        Debt
4              3076.8
**********************************
Simulation finished!
```

Fig 2: Simulation result of Scheduling based on PAPRIKA method

Figure 3 shows the comparison result of task completion time based on both the methods. The number of tasks will be varied from 0 to 100 and its corresponding task allocation time is noted. Task allocation time of S-CSRSA algorithm is compared with Scheduling based on PAPRIKA method and graph is plotted corresponding to the values.

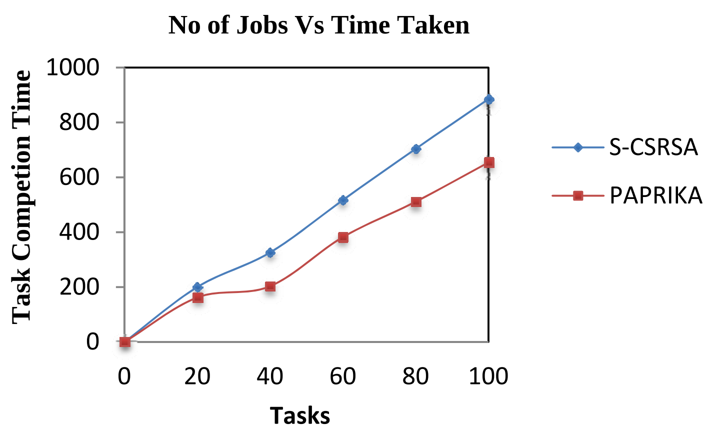

Fig 3: Comparison of task finishing time between S-CSRSA and PAPRIKA

From the graph it is evident that scheduling based PAPRIKA method takes less task allocation time while comparing with S-CSRSA algorithm.

Comparison of resource utility rate: Resource utility rate of S-CSRSA algorithm is low, while comparing to the scheduling algorithm which is based on PAPRIKA method. The proposed method shows 10% increase in the resource utility rate. To get high resource utility rate, the resources are allotted to the tasks based on the priority value. The resources with high priority will allotted to the suitable task, and in this way efficient utilization of resources achieves.

Figure 4 reveals the difference in resource utility rate of the both methods. A range of 0 to 100 tasks are created and the corresponding resource utility rate is obtained.

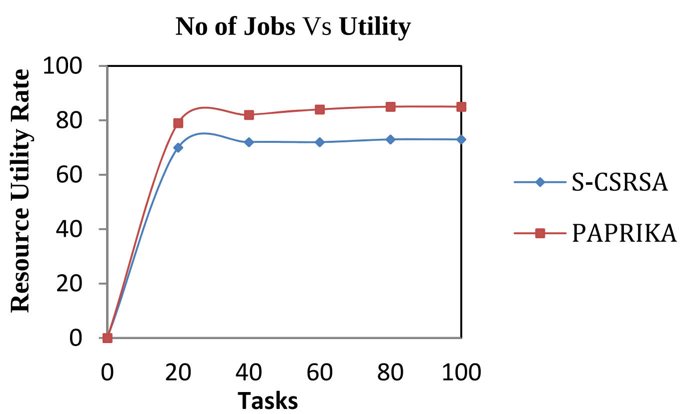

Fig 4: Comparison of resource utility rate between S-CSRSA and PAPRIKA

The graph compares the utility rate of both algorithms and it proves that scheduling based PAPRIKA method having increased resource utility rate than S-CSRSA algorithm.

6. Conclusion

Resource scheduling is one of the key factors in resource management. Efficient utilization of resources will reduce cost and time needed for processing. Literature review reveals the limitations of different scheduling and allocation methods. The major limitation is high re source utility rate and processing time. The proposed method improves resource utility rate and minimizing the time needed for allocation.

7. Acknowledgment

The authors wish to thank Karunya University for providing infrastructure for carrying out the simulation and financial support. The authors thank the senior professors and the technical experts for providing valuable suggestions to improve the quality of the research paper.

References

[1] Abirami S.P. and Shalini Ramanathan, 2012. "Linear Scheduling Strategy for Resource Allocation in Cloud Environment". International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol.2, No.1, February 2012. Available from: http://airccse.org/journal/ijccsa/papers/2112ijccsa02.pdf.

[2] An Oracle White Paper in Enterprise Architecture, 2009. "Architectural Strategies for Cloud Computing", August 2009. Available from: http://www.oracle.com/technology/architect/entarch/pdf/architectural_strategies_for_cloud_computing.pdf.

[3] Esha Bansal, Nisha Bansal, 2011. "An Analysis of Cloud Computing". International Journal on Computing and Corporate Research (IJCCR), Vol.1, Issue 3, Manuscript 7, November 2011. Available from: http://www.ijccr.com/November2011/7.pdf.

[4] Gunho Leey, Byung-Gon Chunz, Randy H. Katz, 2011. "Heterogeneity-Aware Resource Allocation and Scheduling in the Cloud". 3rd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud), 2011. Available from: http://static.usenix.org/event/hotcloud11/tech/final_files/Lee.pdf.

[5] Hai Zhong, Kun Tao1, Xuejie Zhang, 2010. "An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems" The Fifth Annual China Grid Conference, Publisher: Ieee, Pages: 124-129. DOI: 10.1109/ChinaGrid.2010.37;

[6] Linlin Wu, Saurabh Kumar Garg and Rajkumar Buyya, 2011. "SLA-based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments". 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Pages 195-204. Available from: http://gridbus.cs.mu.oz.au/papers/SLA-SaaS-CCGrid2011.pdf.

[7] Nagendram S., Vijaya Lakshmi J., Venkata Narasimha Rao D., Naga Jyothi Ch, 2011. "Efficient Resource Scheduling in Data Centers using MRIS" Indian Journal of Computer Science and Engineering (IJCSE), Vol.2, Issue 5, pages: 764-769. Available from: http://www.ijcse.com/docs/INDJCSE11-02-05-144.pdf.

[8] Qi Zhang, Eren G¨urses, Raouf Boutaba, Jin Xiao, 2011. "Dynamic Resource Allocation for Spot Markets in Clouds". Fourth IEEE International Conference on Utility and Cloud Computing (UCC), Pages:178-185. Available from: http://static.usenix.org/event/hotice11/tech/full_papers/Zhang.pdf.

[9] Rajkumar Buyya, Rajiv Ranjan, Rodrigo N. Calheiros, 2009. "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities". 7th High Performance Computing and Simulation Conference (HPCS 2009). Available from: http://www.cloudbus.org/papers/CloudSim-HPCS2009.pdf.

[10] Wuqi Gao and Fengju Kang, 2012. "Cloud Simulation Resource Scheduling Algorithm Based on Multi-Dimension Quality Of Service". Information Technology Journal, 11: 94-101. Available from: http://scialert.net/qredirect.php?doi=itj.2012.94.101&linkid=pdf.

[11] Xindong YOU, Xianghua XU, Jian Wan, Dongjin YU, 2009. "RAS-M: Resource Allocation Strategy based on Market Mechanism in Cloud Computing" Fourth China Grid Annual Conference. DOI: 10.1109/ChinaGrid.2009.41.

[12] Zhongni Zheng ,Rui Wang, Hai Zhong, Xuejie Zhang; "An Approach for Cloud Resource Scheduling Based on Parallel Genetic Algorithm"