Enhanced Network Traffic Classification through Priority Based User Scheduling Algorithm (PUSA)

[1]S. Somasundaram, [2]Dr. M. Ravichandran,
[1]Research scholar, [2]Associate Professor
[1, 2,] Dept. of Computer Science,
[1,2,] Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore, India

Abstract

The Most colleges and universities are embracing new technologies that can deliver powerful resources with ease of use. Cloud computing is one of the smart technologies they are using. It is a new paradigm that can provide on-demand services such as servers, storage disks, platforms and applications to any cloud service customers over a network. To address both issues, efficient task scheduling optimization methods must be provided.

Task scheduling algorithms are one of the major theoretical challenges in the field of network classification and scheduling. Some intensive researches have been done in the field of task scheduling of cloud figures. In this paper, we proposed a new algorithm for priority classification and user scheduling in cloud computing. The proposed algorithm is based on a decision making model with multiple criteria to use the supervised machine learning classification algorithms to classify the priority tasks into priorities User Scheduling (PUS) different tasks priority queue to improve the task scheduling response time A users scheduling algorithm task in a multi-class network is also to categorize the users into one of the predefined classes. Since this approach has more users and the parameters considered for classification are more independent, simulation results suggest that the classification technique clusters the task more effectively and provides consistent use of available resources.

Keywords: classification, simulation, embracing, independent, priority

Introduction

In the field of distributed computing, several job scheduling algorithms have been proposed [10-11, 12, 13]. The main goal of job scheduling is to achieve high computational performance and the best system throughput. Traditional job scheduling algorithms are not able to provide scheduling in cloud environments. After a simple classification [8], jobs are queued and combined into a set when they arrive in the system. The scheduling algorithm starts after a fixed period of time. [9]

Task scheduling is the process of assigning tasks to available CSP resources based on various characteristics and requirements [3-7]. Therefore, for efficient task scheduling, we need to use intelligent task scheduling algorithms. In the practice of task scheduling, we can use two types of algorithms: priority-based and non-priority-based [8]. Generally, when an organization needs a service or resources from CSP, a contract called SLA is signed between them. SLA is a commitment paper based on the goals and desires between the CSPs and the CSCs. The goal of this contract is to define the rules of interaction between the CSPs and the CSCs. Thus, if the CSC is satisfied with the SLA, then the cloud services are requested [1].

Scheduling algorithms are used to solve the competition for shared resources in a network. Therefore, it specifies and distributes bandwidth among users and determines their transmission order to enable improved QoS in WiMAX networks [3, 11 and 12].Task scheduling, one of the most well-known combinatorial optimization problems, is one of the most important factors that plays a key role in improving the performance of flexible and reliable systems. The main objective is to allocate tasks to adaptive resources in accordance with adaptive time, which involves finding an appropriate order in which tasks can be executed under the constraints of transaction logic [5]. The key here is to allocate resources to tasks in the queue in order to make the best use of the available computing system. In order to achieve this, the tasks must be divided into different priority queues. The main goal is to allocate the tasks to the adaptive resources in accordance with the adaptive time, which involves finding an appropriate order in which the tasks can be executed under transactional logic constraints [5]. The key here is to allocate resources to tasks in the request queue in order to make the best use of the available computing system. To achieve this, tasks are grouped and resources are allocated later [6]. The motivation is to determine, through experimental work, which method is more efficient for task classification in the cloud computing user environment [7].

Related Work

Wang et al. proposed a “multidimensional algorithm for task scheduling according to the availability of CPU memory, but it does not consider the network bandwidth, and the main flaw of this type of scheduling is that it does not consider the dynamic change of resource demand." [8]

Fang et al A two level task scheduling scheme which is based on load balancing algorithm”. In this first level, we assign user applications to the virtual machines and in the second level, we provide appropriate host resources to the virtual machines. However, this algorithm does not consider the network bandwidth and resource usage for each task. “A nonlinear programming model has been proposed to reduce the data transfer cost and task execution time, but it does not consider network bandwidth." Therefore, although this algorithm takes less time to execute each task, the total time to execute all tasks is higher due to the transmission time [9].

The priority task scheduling strategy proposed by Naoufal et al. is based on three parameters: task deadline, task age, and task length. Moreover, this strategy considers that each CSP has a multi-cloud data centre, each data centre has multiple clusters, each cluster has multiple physical servers, and each physical server has one or more virtual machines, which can give a global view of task scheduling in cloud computing.

Monika et.al have proposed a scheduling algorithm. Here the incoming tasks are grouped based on the task requirements like minimum execution time and minimum cost. The research paper only says that the tasks are grouped or clustered, but it does not discuss the grouping or clustering technique [3]. [8]

Problem Statements

Monitoring: in task scheduling, tasks are assigned to the virtual machine based on the computational power and available user resources. If the virtual machine is capable of executing the task and the user resources required for the additional task are available, then we assign the additional task to the user's virtual machine. While a task is running, we assign the additional task to the same virtual machine without checking whether the available bandwidth is sufficient for the task or not, and the task is in the waiting position without being able to run the next task.

Therefore, we check the available bandwidth of the two virtual machines and the required task and assign the task based on the computational power of the bandwidth and resources of the virtual machine to reduce the wastage of resources and execution time." [10]

Techniques and technologies: Both technical and physical controls are used here to help set and enforce policies, diagnose errors and accurately troubleshoot the network. Organisational access and management policies: Task scheduling assigns the task to the virtual machine based on the user's processing power and available user resources. [11] If the virtual machine is capable of performing the task and the user resources required for the additional task are available, the additional task is assigned to the user's virtual machine. While a task is running, we assign the additional task to the same virtual machine without checking whether the available bandwidth is sufficient for the task or not.

Techniques and technologies: several tools and techniques that support the network. Through the task scheduling algorithm, the classification of priority task scheduling algorithm [13].

As a result of the problems described above, in this paper, task scheduling for users and user network usage task classification algorithm.

Proposed Methodology

Our planning algorithm consists of three priority levels: Planning level (target level), resource level (attribute level), and order level (alternative level), see Fig. 1 and Fig. 2.

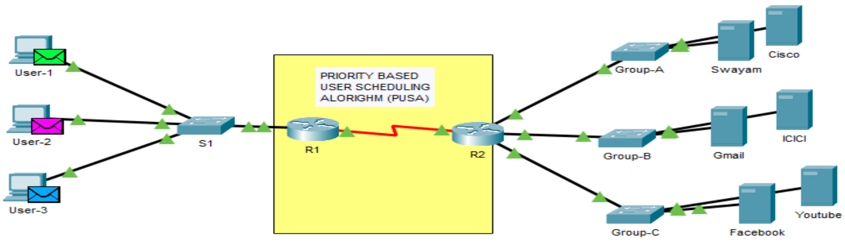


Fig 1 : Request Transfer to PUSA

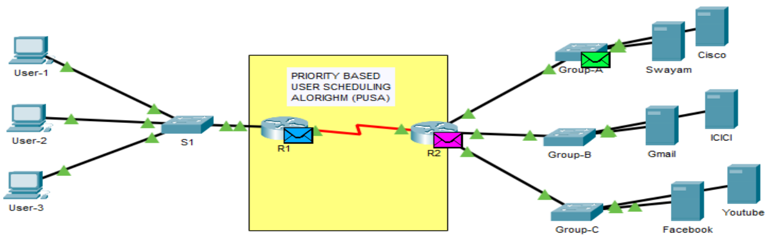


Fig 2 : PUSA assign the priority

Assume that ψ={U1,U2,...,Um} is a set of jobs requesting resources in a user cloud environment, and that φ={S1,S2,…,Sd} is a set of resources available in a server cloud environment ($d << m$). Each user requests a resource with a certain priority. Each user requests a resource with a specific priority. The priority of each user is compared separately with other users [14].

for example, suppose that the ratio of the priorities of Ji and Jj for obtaining a particular resource such as Cg is 7, in which case we write pgU=1/7. It is clear that. In general, this can be represented by equation (1)

$$P_g^{ij} = \begin{cases} 1 \\ pgij \quad i \neq j\,,\, i = j \\ 1 \end{cases} \qquad \text{Eq--------(1)}$$

In Eq. (6), Pg denotes a matrix with m rows and m columns. This matrix is also a comparison matrix. Suppose that Q1, Q2…..... Qn are n comparison matrices of orders created according to the priority of resource additions. For each of the comparison matrices, we should compute a priority user [15].

The vector of priority users can be obtained by solving the following equation

$wA = \lambda_{\text{MAX}} x^{w}$ --------------Eq(2)

. There are several methods for computing the priority vector [1-5]. An iterative method for solving Eq. (3) can be found in [6], which solves Eq. (3) using numerical methods. Using iterative methods, the priority vector (vector of weights) can be computed without worrying about consistency problems.

Vol 12 Issue 04 2023
ISSN NO: 2230-5807

Suppose that w1, w2, ...wd are the corresponding priority vectors of Q1, Q2, ... Qd are. In this case, we can define a normal matrix of the workspace plane as in Eq. (3).

Δ=[w1,w2…..wd] --------------------Eq(3)

It is clear that this is a matrix with m [number of users] rows and d [number of resources] columns. The next step of the proposed algorithm is to create a comparison matrix for resources by priority. This matrix determines which resource has a higher priority than others based on the decision maker(s). In this case, we obtain a matrix with d rows and d columns [20]. Assume that S is the comparison matrix for the resource level, then γ is defined as the priority vector of S. The next step of the algorithm is to calculate PVS, which is called the priority vector of scheduling users [16]. PVS can be calculated using equation (4). Finally, we choose the maximum element of PVS and then the corresponding element of ψ to allocate an appropriate resource. A general algorithm for PJSC can be outlined in Table 2.

A general algorithm for PUSA

Give U= {u^1... U^m} a set of users

Enter S= {s^1,...,s^d} a set of resources

Create a consistent comparison matrix for all users by priority of resource accessibilities – [d-matrix with m rows and m columns]

Calculate priority vector

For all d matrices with m rows and m columns, based on

Δ= [w^1, w^2…......w^d]

Create a matrix with priority vectors based on

PVS=Δ.γ and name it.

Compute a consistent comparison matrix for C corresponding to the decision maker(s).

Compute the priority vector for the matrix in the comparison matrix C based on Compute priority vector

Δ=[w^1,w^2…......w^d] and name it.

Compute PVS, which is a vector that represents the priority values of users from

$\upsilon = d^{2.81} + d \times m^{2.81}$.

Select the user with the highest priority value based on PVS and assign him an appropriate resource.

Update the list of users.

End.

Finite-time speed of the server (FTSS)

The proposed algorithm mainly focuses on user priority. However, we do not expect this algorithm to have an optimal final time (FTSS). This means that the algorithm must consider the priority of the users instead of considering the end time (FTSS). To further explain, we assume that A={x2},B={x3,x4}and x={y2,y5,y6} are three users. We also assume that x1 and x2 are on server 1 and x3 is on server 2, and that two processors p1 and p2 are available [18].

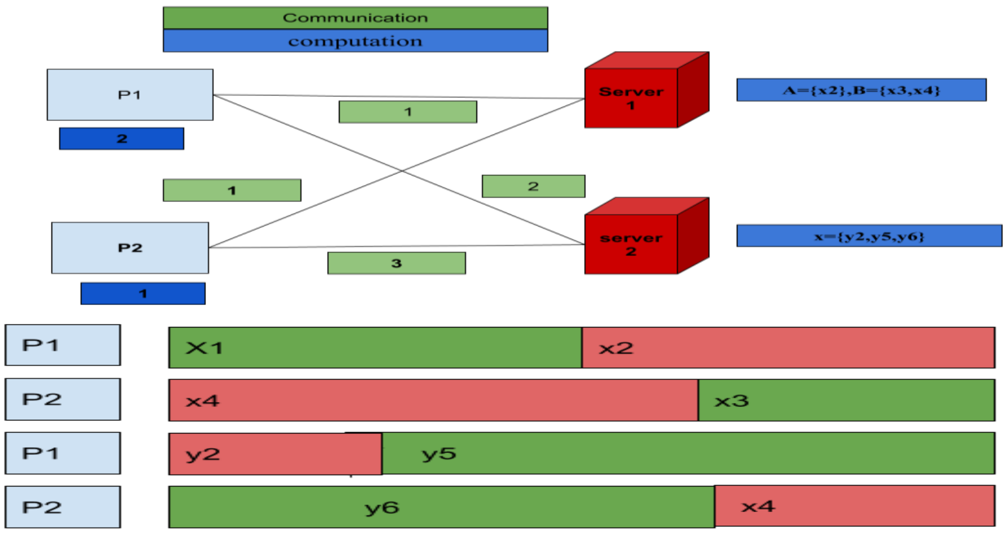
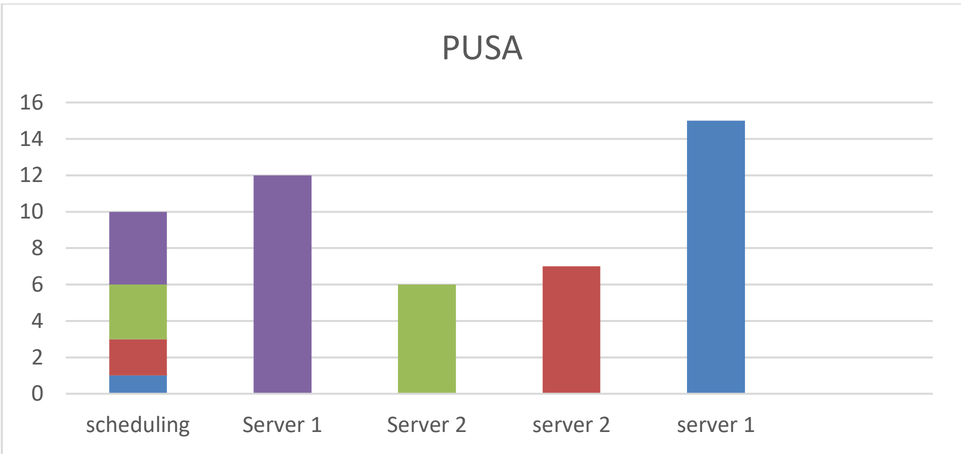


Figure 3: scheduling of these four users

Fig. 3 shows how these four users can be scheduled. Fig. 3 also shows the end time of scheduling these three jobs in different possible states.[19,21] According to Fig. 3, the priority scheduling time of x4, y2, and y5 can be changed depending on which user can be scheduled first.
We calculated the priority scheduling time of PUSA. Calculate a deterministic value for the end time of PUSA. It depends on the priority of users and can vary from the worst to the best value (see Fig. 3). One possible method to calculate the PUSA completion time is to find the average value of the completion time [22].



Let Ji be the set of users scheduled on device i.
Then $\ell i = \sum j \in Ji\ pi,j$ is the load of device i.
The maximum load
$\ell max = cmax = maxi \in M\ \ell i$
is called the time span of the schedule.

Example of an experiment

In this section, we have described an example of the proposed job scheduling algorithm. In this case, we define three resources and four jobs. Comparison matrices and priority vectors are computed in a cloud environment for all resources and jobs according to the resource allocation [17]. (See Table 1-2).

Table 1.Priority of resource according to allocation

Priority	server1	server2	server3	Priority Vector
Server 1	1	3	5	0.64
server2	½	1	3	0.66
server3	1/3	1/5	1	0.26

Table 2.Priority of jobs according to allocation accessibility to Users

Resource 1	user 1	user 2	user 3	Priority Vector
User1	1	1/2	4	0.13
User2	4	1	4	0.24
User3	1/4	1/4	1	0.07

Conclusion

Priorities are an important issue in job scheduling in cloud environments. In this paper, we proposed a priority-based user scheduling algorithm that can be used in cloud environments. We called it “PUSA”." We also discussed some issues related to the proposed algorithm, such as complexity, consistency, and end time. The results of this work show that the proposed algorithm has reasonable complexity. Moreover, improving the proposed algorithm with the aim of taking less time for planning priority time is considered as future work.

Reference

[1] Naoufal Er-Raji, Faouzia Benabbou, Ahmed Eddaoui, Task Scheduling Algorithms in the Cloud Computing Environment: Survey and Solutions , (2016) International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), 1 (6).

[2] Vigindra, Sudhir Shenai, Survey on Scheduling Issue in Cloud Computing, international conference on modeling optimization and computing – (ICMOC-2012).

[3] Nidhi Bansal, Dr. Maitreyee Dutta, Performance Evaluation of Task scheduling with priority and non-priority in cloud computing, International Conference on Computational Intelligence and Computing Research – (IEEE-2014).

[4] G. Justy Mirobi , Dr. L. Arockiam, Service Level Agreement In Cloud Computing. An Overview, International Conference on Control Instrumentation, Communication and Computational Technologies (LCCICCT) – (IEEE-2015).

[5] M. Vaidehi, T.R. Gopalakrishnan Nair, and V. Suma,” An Efficient Job Classification Technique to Enhance Scheduling in Cloud to Accelerate the Performance”, Chapter · January 2014, DOI: 10.1007/978-3-319-03107-1_65, Research Gate.

[6] Polo, J., Castillo, C., Carrera, D., Becerra, Y., Whalley, I., Steinder, M., Torres, J., Ayguadé, and E.: Resource-Aware Adaptive Scheduling for MapReduce Clusters. In: Kon, F.,Kermarrec, A.-M. (eds.) Middleware 2011. LNCS, vol. 7049, pp. 187–207. Springer, Heidelberg (2011).

[7] Naoufal Er-raji, Faouzia Benabbou., Mirela Danubianu., Amal Zaouch.,” Supervised Machine Learning Algorithms for Priority Task Classification in the Cloud Computing Environment”., https://www.researchgate.net/publication/331844265., November 2018

[8] Abdul Razaque., 2Nikhileshwara Reddy Vennapusa., Nisargkumar Soni.,” Task Scheduling in Cloud Computing”., Cleveland State University, OH, USA.

[9] Shamsollah Ghanbari, Mohamed Othman, “A Priority Based Job Scheduling Algorithm
[10] in Cloud Computing”, Article in Procedia Engineering · January 2012.
[11] Tai-Lung Chen et al, Scheduling of Job Combination and Dispatching Strategy for Grid and Cloud System, GPC,(2010) 612–621.
[12] Kyriaki Z, Multi-Criteria Job Scheduling in Grid Using an Accelerated Genetic Algorithm, J Grid Computing, 10 (2012) 311–323.
[13] Monir Abdullah, Mohamed Othman et al, Optimal Workload Allocation Model for Scheduling Divisible Data Grid Applications, Future Generation Computer Systems 26 (2010) 971-978.
[14] Amin Shokripour , Mohamed Othman et al, New Method for Scheduling Heterogeneous Multi-Installment Systems, Future Generation Computer Systems 28 (2012) 1205–1216.
[15] Naoufal Er-Raji, Faouzia Benabbou, Ahmed Eddaoui, Task Scheduling Algorithms in the Cloud Computing Environment: Survey and Solutions , (2016) International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), 1 (6).
[16] Abdul Razaque, Nikhileshwara Reddy Vennapusa, Nisargkumar Soni, Task Scheduling in Cloud Computing, Systems, Applications and Technology Conference (LISAT), Long Island – (IEEE-2016).
[17] Varinder Kaur, Gurjot Kaur, Cloud Computing Scheduling Algorithms: A Review, (2015) International Journal of Computer Science and Network (IJCSN) 2 (4).
[18] Arnav Wadhonkar , Deepti Theng, A Survey on Different Scheduling Algorithms in Cloud Computing, International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB16), (IEEE-2016).
[19] Vigindra, Sudhir Shenai, Survey on Scheduling Issue in Cloud Computing, international conference on modeling optimization and computing – (ICMOC-2012).
[20] Polo, J., Castillo, C., Carrera, D., Becerra, Y., Whalley, I., Steinder, M., Torres, J.,Ayguadé, E.: Resource-Aware Adaptive Scheduling for MapReduce Clusters. In: Kon, F.,Kermarrec, A.-M. (Eds.) Middleware 2011. LNCS, vol. 7049, pp. 187–207. Springer,Heidelberg (2011).
[21] Naoufal Er-raji., Faouzia Benabbou., Mirela Danubianu., Amal Zaouch Supervised Machine Learning Algorithms for Priority Task Classification in the Cloud Computing Environment.,
[22] Kaur, P.D., Chana, I. ―Unfolding the distributed computing paradigm, In: International Conference on Advances in Computer Engineering, pp. 339-342 (2010)
[23] Rizvi, Syed, Abdul Razaque, and Katie Cover. "Third-Party Auditor (TPA): A Potential Solution for Securing a Cloud Environment." Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on. IEEE, 2015