www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 4 April 2017, Page No. 20942-20945
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i4.26

Implementation of Multiqueue Interlacing Peak Scheduling based on Task Classification in Cloud Computing

B.Nandan[1], A,Nikhil[2], A.Rahul[3] B.Sai Kiran Reddy[4]

Associate Professor, Department of CSE, Guru Nanak Institutions, Ibrahimpatnam, Hyderabad, India[1]
B Tech Student, Department of Computer Science & Engineering, Guru Nanak Institutions, Hyderabad, India[2] B Tech Student, Department of Computer Science & Engineering, Guru Nanak Institutions, Hyderabad, India[3] B Tech Student, Department of Computer Science & Engineering, Guru Nanak Institutions, Hyderabad, India [4]

Abstract . The benefit load information, for instance, CPU, I/O, and memory use, is every so often accumulated and redesigned, and the errand information as to CPU, I/O, and memory is accumulated. Second, resources are sorted into three lines as demonstrated by the piles of the CPU, I/O, and memory: CPU heightened, I/O genuine, and memory concentrated, as showed by their solicitations for resources. Finally, once the endeavors have been reserved, they need to interlace the benefit load peak. Some composes of assignments ought to be facilitated with the benefits whose loads contrast with a lighter sorts of assignments. So to speak, CPU intensive assignments should be facilitated with resources with low CPU use; I/O-genuine assignments should be facilitated with resources with shorter I/O hold up times; and memory-genuine assignments should be facilitated with resources that have low memory use. The ampleness of this technique is exhibited from the speculative viewpoint. It has also been ended up being less mind boggling regarding time and place. Four examinations were planned to check the execution of this procedure. Tests impact four estimations: 1) typical response time; 2) load modifying; 3) due date encroachment rates; and 4) resource use. The test outcomes exhibit that this procedure can alter stacks and improve the effects of benefit part and utilization sufficiently. This is especially bona fide when resources are compelled. Along these lines, various errands will pursue the same resources. In any case, this system shows advantage over other similar standard computations. In this paper we show the implementation of Multi-queue Interlacing Peak scheduling based on Task Classification in Cloud Computing in Ellipse IDE.

Key Words: *Cloud computing, load balancing , multi queue, task classification, taskscheduling.*

INTRODUCTION

Cloud computing involves a diverse range of application tasks . Their demands for resources differ accordingly some require large amounts of storage, and some require CPUs with a powerful computing capacity; others are data intensive and they have considerable I/O. These all cause load imbalance. For example, real-time temperature measurements, and bank deposits and withdrawals involve greater CPU demands and need immediate responses. For these application tasks, the task requests increase sharply as the amount of user access increases. The system's processing speed will be slow, and it may crash, unable to continue service. Additionally, some application tasks require reading and storing data from a database, requiring frequent disk reading and writing to the server, i.e., they require a lot of I/O.

GENERAL

The ampleness of this technique is exhibited from the speculative viewpoint. It has also been ended up being less mind boggling regarding time and place. Four examinations were planned to check the execution of this procedure. Tests impact four estimations: 1) typical response time; 2) load modifying; 3) due date encroachment rates; and 4) resource use. The test outcomes exhibit that this procedure can alter stacks and improve the effects of benefit part and utilization sufficiently.

OBJECTIVE

The scope of the project is to provide Average response time, load balancing ,deadline violation rates and resource utilization. Can be easily checked and provide dead line for each resources. Task scheduling makes the user to check the status of server also.

EXISTING SYSTEM

In Existing system CPU Speed, CPU load, Memory usage is not calculated. Because of lack of resource allocation resource can't be fetched effectively.

1. PROPOSED SYSTEM:

The method is based on the diversity of tasks and the dynamic factors ofresources. First, it collects resource and task information,which includes the loads for the CPU, I/O, and memory ofresources, and the remands for the CPU, I/O, and memoryof tasks. Second, it sorts resources into three queues accordingto the loads of the CPU, I/O, and memory .

PROPOSED SYSTEM TECHNIQUE

Technique: MIPSM.(Multi queue Interlacing Peak Scheduling)

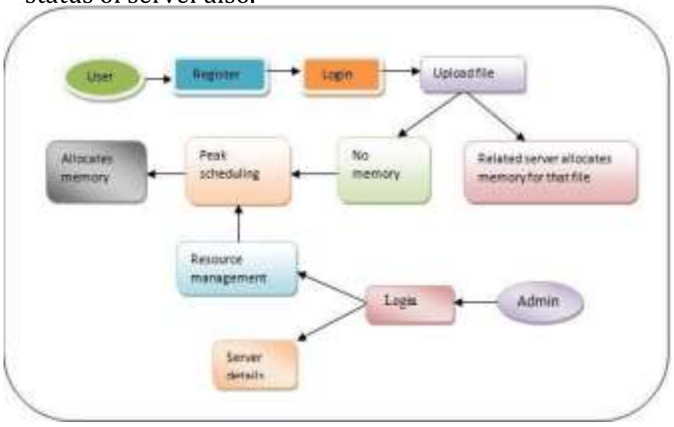
This method is based on the diversity of tasks and the dynamic factors of resources. First, it collects resource and task information, which includes the loads for the CPU, I/O, and memory of resources, and the remands for the CPU, I/O, and memory of tasks. Second, it sorts resources into three queues according to the loads of the CPU, I/O, and memory

PROPOSED ADVANTAGES

Task scheduling makes the user to check the status of server.

SYSTEM ARCHTECTURE

In this paper it is discussed mainly about the requirements, architecture. The scope of our project is to provide Average response time, load balancing ,deadline violation rates and resource utilization. Can be easily checked and provide dead line for each resources. Task scheduling makes the user to check the status of server also.

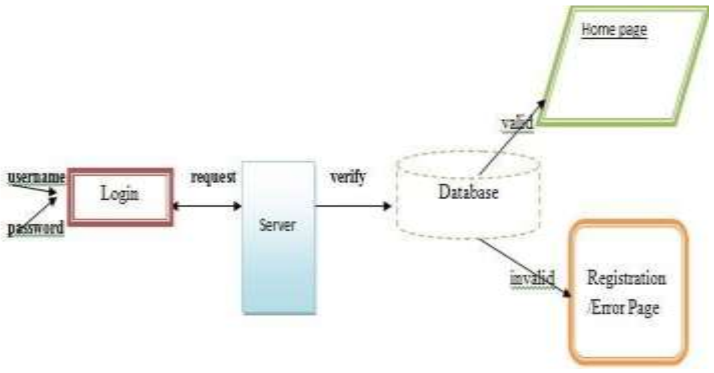


In cloud computing, resource load dynamics, task diversity, and the differences in resource task demand can lead to load imbalance and affect the efficiency of task scheduling and resource utilization. In order to solve these problems, a MIPSM is here proposed. First, tasks were divided into three queues: 1) CPU intensive; 2) I/O intensive; and 3) memory intensive. Second, the resources were sorted according to CPU utilization loads, I/O wait times, and memory usage. Last, three queues of tasks were scheduled to those resources whose loads for the corresponding metric (CPU, I/O, or memory) were lighter than the others. This method was found to balance loads effectively through theoretical verification.

MODULES

➢ USER INTERFACE DESIGN

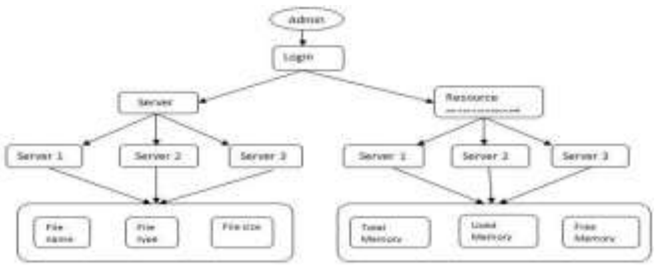
This is the first module of our project. The important role for the cloud user is to move login window to cloud user window. This module has created for the security purpose. In this login page we have to enter login user id and password. It will check username and password is match or not (valid user id and valid password). If we enter any invalid username or password we can't enter into login window to user window it will shows error message. So we are preventing from unauthorized user entering into the login window to user window. It will provide a good security for our project. So server contain user id and password server also check the authentication of the user. It well improves the security and preventing from unauthorized user enters into the network. In our project we are using JSP for creating design. Here we validate the login user and server authentication.



➢ ADMIN

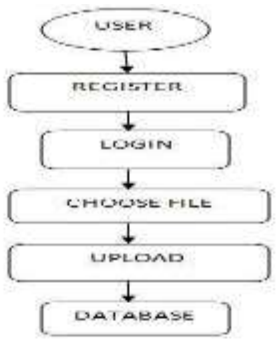
This is the second module of our project. In this the cloud provider (admin) has rights to view the server usage and resource management according to memory utilization. Here admin maintains three different types of servers such as server1 (.pdf files), server2 (.docx files) and server3

(.jpg/jpeg files) .The internal memory allocated to each server is 3145728 bytes.



➢ USER UPLOAD

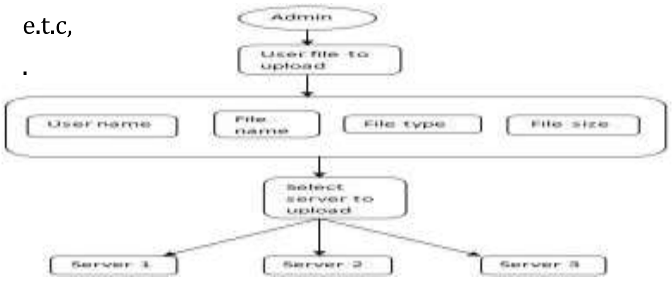
This is the third module used for uploading consumer's files or documents into the virtual machines. These constraints serve a dual purpose as they can introduce high-level policies which are the flexible file access policies and assist in administration tasks. When user uploads any file of particular size that file will be uploaded into one of the virtual machines according to the file parameters of that login user.



➢ TASK SCHEDULING

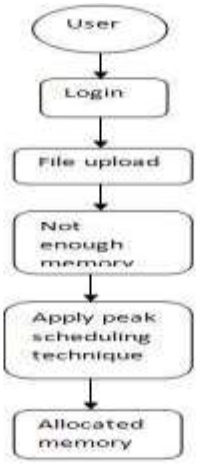
This is the fourth module of our project. In this the admin internally acts to store the user uploaded file in to the virtual machines based on the file parameters. At the time of uploading the file parameters are retrieved such as file type, file size, file key, server, upload time e.t.c,

.



➢ LOAD BALANCING

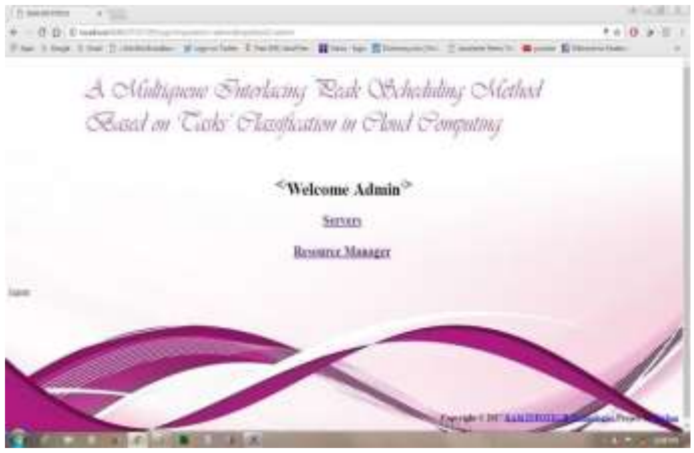
This is the fifth module of our project. In this the admin undertake the load control. If cloud user upload file is huge that it means the uploaded file memory is more than the available server memory. At that situation the temporary storage space is activated and the resources sorted, finally allocated memory to store properly.



ALGORITHM DEFINITION PEAK SCHEDULING ALGORITHM

This method is based on the diversity of tasks and the dynamic factors of resources. First, it collects resource and task information, which includes the loads for the CPU, I/O, and memory of resources, and the remands for the CPU, I/O, and memory of tasks. Second, it sorts resources into three queues according to the loads of the CPU, I/O, and memory.

RESULTS



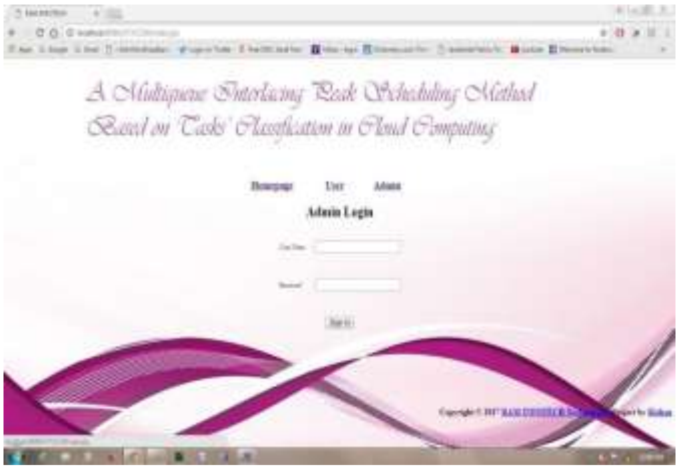
A Multiqueue Interlacing Peak Scheduling Method Based on Tasks' Classification in Cloud Computing

Welcome Admin

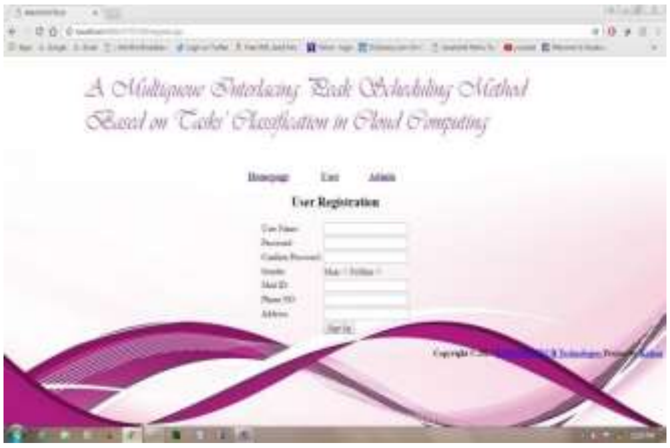
Servers

Resource Manager

DESCRIPTION: This is the main page which displays on the browser which as user, admin , user registration options.

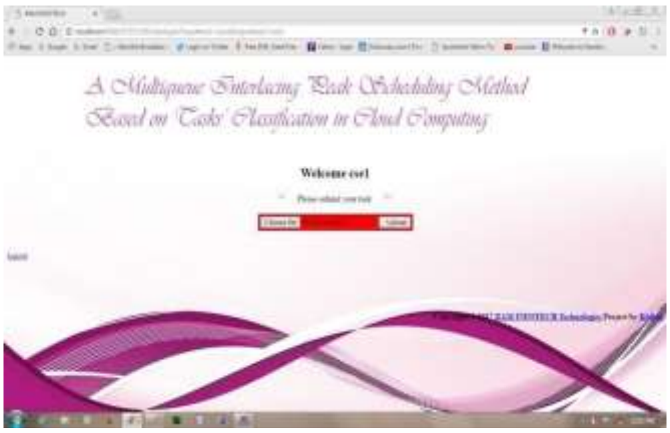

A Multiqueue Interlacing Peak Scheduling Method
Based on Tasks' Classification in Cloud Computing
Homepage
User
Admin
Admin Login
Sign In

DESCRIPTION: Here Admin user will login by giving his authenticated user identification number and a highly secured password.


A Multiqueue Interlacing Peak Scheduling Method
Based on Tasks' Classification in Cloud Computing
Homepage
User
Admin
User Registration

DESCRIPTION: Here user can upload the file of different formats like .doc, .pdf, .jpg.

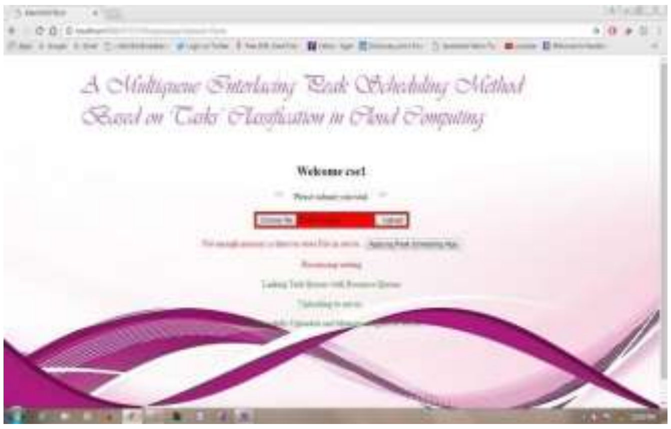
DESCRIPTION: Here when user upload a large size of file, it pop ups a message that " there is no enough space to store the file in the server"


A Multiqueue Interlacing Peak Scheduling Method
Based on Tasks' Classification in Cloud Computing
Welcome cse1
Choose file
Upload

DESCRIPTION : After applying the TASK SCHEDULING ALGORITHM

CONCLUSION

In cloud computing, resource load dynamics, task diversity, and the differences in resource task demand can lead to load imbalance and affect the efficiency of


A Multiqueue Interlacing Peak Scheduling Method
Based on Tasks' Classification in Cloud Computing
Welcome cse1
Choose file
Upload

DESCRIPTION: Here user will login by giving his authenticated user identification number and a highly secured password.

task scheduling and resource utilization. In order to solve these problems, a MIPSM is here proposed. First, tasks were divided into three queues: 1) CPU intensive; 2) I/O intensive; and 3) memory intensive. Second, the resources were sorted according to CPU utilization loads, I/O wait times, and memory usage. Last, three queues of tasks were scheduled to those resources whose loads for the corresponding

REFERENCES

[1] C. Zhu, V. C. M. Leung, X. Hu, L. Shu, and L. T. Yang, "A review of key issues that concern the feasibility of mobile cloud computing," in *Proc. IEEE Int. Conf. Cyber, Phys., Soc. Comput. (CPSCom)*, 2013, pp. 769–776.

[2] A. Beloglazov and R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1366–1379, Jul. 2013.

[3] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1107–1116, Jun. 2013.

[4] J. Luo, L. Rao, and X. Liu, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 775–784, Mar. 2014.

[5] J. Cao, K. Li, and I. Stojmenovic, "Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers," *IEEE Trans. Comput.*, vol. 63, no. 1, pp. 45–58, Jan. 2014.

[6] B. Prabavathy, K. Priya, and C. Babu, "A load balancing algorithm for private cloud storage," in *Proc. 4th IEEE Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, 2013, pp. 1–6.

[7] R. A. M. Razali, R. A. Rahman, N. Zaini, and M. Samad, "Virtual machine migration implementation in load balancing for Cloud computing,"in *Proc. 5th Int. Conf. Intell. Adv. Syst. (ICIAS*), Kuala Lumpur, Malaysia, Jun. 3–5, 2014, pp. 1–4.