ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.7

Dynamic Real Time Scheduling Technique for Overloaded Jobs to Improve the Performance in Mobile Cloud

Shilpa B Kodli[1], Pooja P Fulari[2]
Assistant Professor[1], PG Scholar[2]
Department of MCA
Visvesvaraya Technological University, Post Graduate Centre, Kalaburagi, India

Abstract:
Cloud computing platforms provide computational resources such as CPU, storage, physical machines etc. for running users' applications. Often, the same application can be implemented in various ways, each with different resource requirements or with different priorities. Taking advantage of this flexibility when allocating resources to users by using different priorities can both greatly benefit users and lead to much better resource utilization and low task delay. In this paper we are presenting two such mechanisms which use prioritization: one in which forked tasks are given complete precedence over newly arrived tasks, and the other in which a threshold is fixed to control the priority so that complete precedence is given to the forked responsibilities if their range exceeds a predefined threshold , the possibility to use code/task offloading techniques between mobiles and clouds in order to reduce the delays of workflows deployed on mobile devices .With mobile devices increasingly able to connect to cloud servers from anywhere at any time, resource-constrained devices can potentially perform offloading of computational tasks to either improve resource usage or to improve performance.

Keywords: Cloud infrastructure, Mobile cloud computing, Resource allocation, Offloaded job, Priority differentiation.

I. INTRODUCTION

In this work, we recommend a solution which allocates resources for job requests in the cloud. Jobs are offloaded by the mobile devices which accomplish their requests by the virtual machines (VMs) which are hosted on physical machines (PMs) in cloud. These jobs also called as primary tasks which can divide new, secondary tasks; a job is completed only when all the forked tasks complete their service. The secondary tasks need to interact with the primary task as well as with each other. However, the PM may not have the resources required to execute the secondary tasks, which is then queued as an 'overflow' tasks in order to find a new 'home'. Secondary tasks, and an overflow tasks, need to be serviced primarily, subsequently to avoid a disruption of the request as per the resulting user dissatisfaction. The proposed solution manages the two types of tasks that uses a queuing model based on addition of multi-dimensional Markov system and Birth-Death queuing systems. We also considering a priority differentiation among the tasks, which is implemented using the two mechanisms. In the first mechanism, overflow tasks are always serviced before any regular tasks. In the second, threshold is set for the number of overflow tasks. As long as the number of overflow tasks is below the threshold, a Weighted Fair Queuing is used; otherwise, only overflow tasks are serviced until their number drops below the threshold. . We investigate the complete routine of both mechanisms in which the method of a Markovian-Multiserver queuing system with two priority ranges to form the useful service allocation method and the multi-dimensional Markov scheme which used to model virtual machine provisioning. Our overall performance effects, suggest that the edge-based priority scheme performs better, also it can be measured to acquire the desired performance level. Mobile devices might also offload their requests to a virtual machine which is working on a cloud data center. The requests may divide new tasks which require virtual machines of their own. Attaining reasonable performance level in such situation needs flexible service allocation mechanisms inside the cloud data center.

II. LITERATURE SURVEY

The executive as well as the exploitation of the VM in request dealing out is estimated in [1], through examining the completing time of the submissions along with entirety effecting time of the recreation. It has need of added resources on the calculating mass. The course of devolving over the elegant portable procedures using the VM consumption is as a result considered as a tough means. The presumption of queuing is functional in favor of assessing reaction time at usual, the substitution involving the presentations plus the price tag in the fusion cloud is discovered here. The online verdict approach is planned with the help of optimization procedures for the circulation of appeals to make the standard time for replying are closer to the tentatively finest along with that to direct the outsourcing outlay relying over a prearranged financial plan [2].

In [3] numerical as well as the dynamical plans in support of delegating is formed by this form as per to the intention task and their disparities. The concern over the effectiveness of the pass on of the hardware terms is given away from the consequences. The pass on algorithm applied here be able to advance appreciably the competence of the force along with the rapidity of the finishing of the movable effort streams.

In [4] work suggests a framework for honest resource allocation that captures such implementation tradeoffs via allowing customers to put up more than one "resource desires". we show that Nash-Bargaining scheme has many best houses, which include pare to optimality and envy freeness, in a wide form of environments while the apparently tons less attractive Lexicographically-Max-Min-Fair fares higher, and is even proof against manipulations, in constrained settings of interest .

III. EXISTING SYSTEM

In the current system, a framework for moving smart phone usage dealing out to the cloud centers which is based on the concept of smart phone virtualization in cloud and reports lack of scalability by way of growing virtual machine of an entire phone device on the cloud. CMcloud is a mobile-to-cloud offloading policy which attempts to minimalize equally the server costs and user provision charge by means of unburdening various mobile applications to a particular server while demanding to accomplish the objective of overall performance of all programs. Also in this existing system, there are some problems with power consumption, high task delay and it also results in a lesser amount of battery existence, quality of check.

3.1 Problem Statement

The problem statement which describes that there are some issues in the existing system such as poor performance, task delay and less battery lifetime. So to overcome these problems we have proposed solution which reduces the waiting time and also we proposed two different priority levels one in which full priority is given to overflow tasks and the other one is threshold based priority to avoid disruption of the application and to ensuring user dissatisfaction.

IV. PROPOSED SYSTEM

In this system we proposed two such mechanisms which uses prioritization: one in which forked tasks are given complete priority over newly arrived tasks, and the other in which a threshold is fixed to control the priority so that complete precedence is given to the forked tasks if their range exceeds a predefined threshold. These proposals help in dealing with the trouble for solving the specific iteration to reach any preferred inaccuracy level. Also in the proposed system the delay of tasks is low and provides a great support for power effectiveness.

METHODOLOGY

In this paper we applied a method which uses Two mechanisms, which works on the prioritization; the first one includes the bifurcated tasks that are given full priority in excess of lately arrived responsibilities whereas, in second one a threshold is acknowledged to direct the priority so that occupied priority is given to the tasks that are diverse if their number exhausted an earlier termed threshold. The Integrated Model is the algorithm that is employed in this work. It takes into account the two stochastic parts that are interactive. The iteration will conclude in the situation where the ideals of the likelihood within the dive of the iterations that are consecutively at variance. Here, the amount of the shifting is achieved at one only in the opening overtake of the VMM.

SYSTEM ARCHITECTURE

As shown in below architecture data the user will get register as data owner with the cloud. After registration he can login, after logging in he will purchase the cloud space where he can upload files to the allocated space. Then data owner will send job request to Cloud the cloud facility supplier (CFS) will provide space for the data file which are uploaded by data owner. Cloud service provider will allocate resources i.e.it will create virtual machine for each resources requested by user. Cloud server stores all the data owner and users information. (RAM) Resource allocation module finds the (PM) physical machine that can accommodate the task. If PM found, VM will be instantiated with the queued task. If not task is rejected. Tasks are distributed into two types i.e. first is resource allocation with complete precedence of excess tasks and the second is resource allocation with threshold-based priority of overflow task.

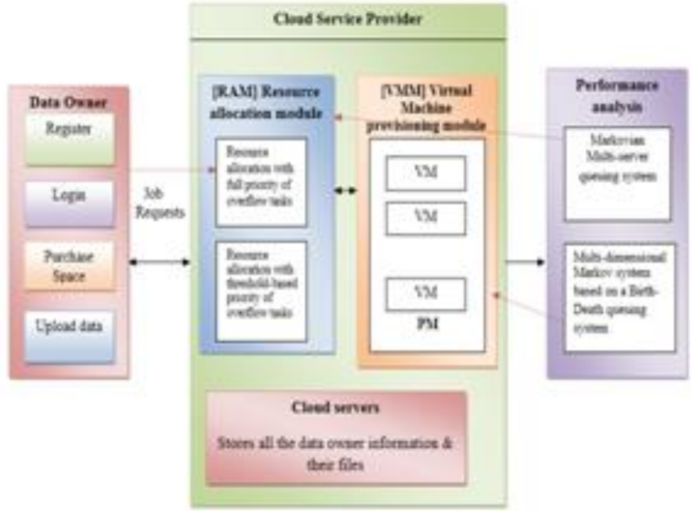

Cloud Service Provider
Data Owner
Register
Login
Purchase Space
Upload data
Job Requests
[RAM] Resource allocation module
Resource allocation with full priority of overflow tasks
Resource allocation with threshold-based priority of overflow tasks
[VMM] Virtual Machine provisioning module
VM
VM
VM
PM
Performance analysis
Markovian Multi-server queuing system
Multi-dimensional Markov system based on a Birth-Death queuing system
Cloud servers
Stores all the data owner information & their files

Figure.1. System Architecture

6.1 SYSTEM MODULES

There are 3 main modules which are as follows:

1. Data Owner
2. Cloud Service Provider
   In Cloud Service Provider there are 2 sub-modules which are as follow:
   i. Resource Allocation Module
   ii. Virtual Machine Provisioning Module
3. End User

1. Data Owner:

The job of the possessor is to login, in support of the application with the suitable user name and a code that must be a secret one. The next procedure is to act upon the relevant activities that include seeking history of all the records and the records that need to be uploaded and so forth. The most important task of the possessor is to have a look at all the processes that are undergoing in the application.

2. Cloud Service Provider:

Offers resource appealed by each client from virtual machines that are formed by Cloud service provider. Accumulating the facts about the holder and the user is looked after by CSP, furthermore the right to use the information is done all the way through the IP network.

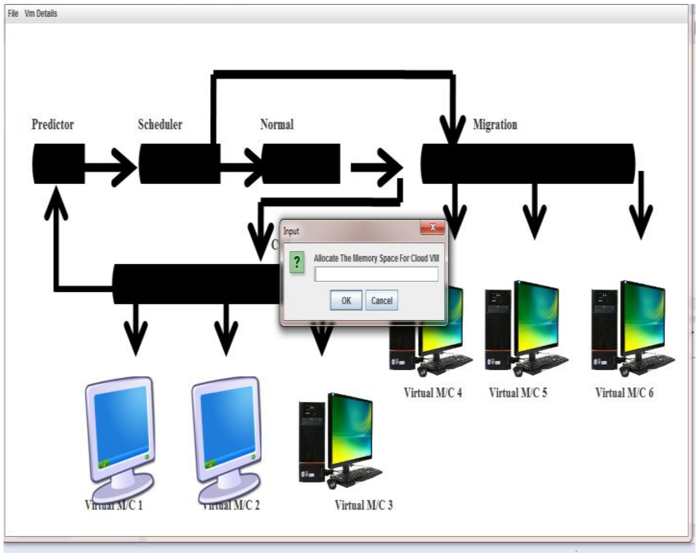
Resource Allocation Module:
The needs of the job when arrived at the data centre, it is then amassed in the recent queue assignment. When a packed queue of tasks are about to get in, in that case they need to be blocked.

Virtual Machine Provisioning Module:
It handles to start up the procedure plus specification and the deployment of Virtual machines as well.

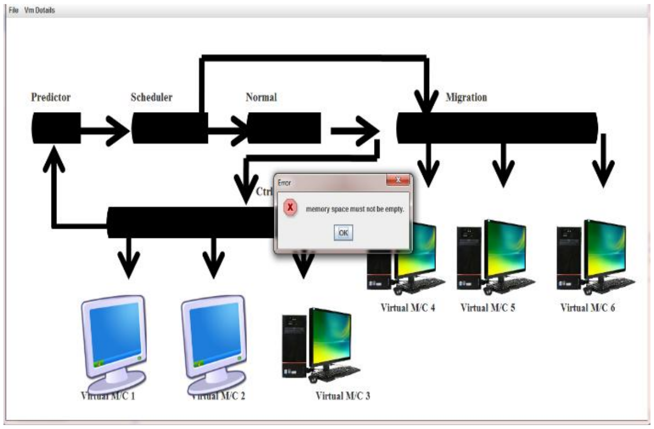
3. End User:
The client specifically need to get registered so as to log in to the cloud and then start their processing's that include downloading the records, analyse and have a right of entry to the uploaded files.

VII. RESULTS

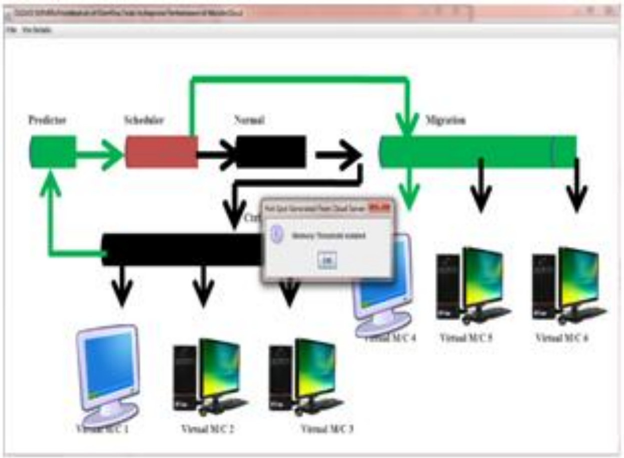
The below images represent the output results of virtual machine creation and how we assigning different tasks to virtual machine. To upload file to cloud server first we need to create virtual machine after creation we need to assign threshold limit to that virtual machine if threshold exceeds limit it will migrate those tasks to another virtual machine if there is sufficient space then it will upload those files to available virtual machine.



Allocate The Memory Space For Cloud VM



memory space must not be empty.



Memory Threshold exceeded

VIII. CONCLUSION

In this paper we have developed two priority schemes for resource allocation in a server pool, based on giving different priorities to the overflow tasks which includes: the one where a full priority is given to an overflow tasks and the other threshold-based priority of overflow tasks. Also, we have examined the effect of task arriving rate, service time and the size of job on the performance metrics for both priority schemes. Also, we have calculated the effect of threshold location on the edge-based priority scheme. Our outcomes authorize that threshold-based priority presents better system performance than full priority of overflow tasks. In future, the modification of the threshold-based priority scheme is the idea find the best location with respect to the threshold is one of performance, which are adopted by the cloud computing provider's that belongs to different policies performance, which are adopted by the cloud computing provider's that belongs to different policies.

IX. REFERENCES

[1]. E.Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R Chandra and P.Bahl. "MAUI: making smartphones last longer with code Offload", (June 2010).

[2]. D. Chae, J. Kim, J. Kim, J. Kim, S. Yang, Y. Cho, Y. Kwon and Y. Paek "CMcloud: Cloud Platform for Cost-Effective Offloading of Mobile Applications", (May 2014).

[3]. M. Shiraz and A. Gani. "Mobile Cloud Computing: Critical Analysis of Application Deployment in Virtual Machines", (February 2012).

[4].Y. Niu, b. Luo, F. Liu, J. Liu and B. Li. "When Hybrid Cloud Meets Flash Crowd: Towards Cost-Effective Service Provisioning", pp. 1044–1052, Hong Kong, China, (April 2015.)

[5]. B. Gao. L. He, L. Liu, K. Li and S. A. Javis." From Mobiles to Clouds: Developing Energy Aware Offloading Strategies for Workflows", (September 2012).

[6]. D. Zarchy, D. Hay and M. Schapira. "Capturing Resource Tradeoffs in Fair Multi- Resource Allocation". pp. 1062–1070, Hong Kong, China, (April 2015)

[7]. X. Zhang, C. Wu, Z. Li and F. C. M. Lau. A Truthful Optimal Mechanism for On-demand Cloud Resource Provisioning“. pp. 1053– 1061, Hong Kong, China, (April 2015).

[8]. M. J. O'Sullivan and D. Grigoras. “Integrating mobile and cloud resources Management using the cloud personal assistant. Simulation Modelling Practice and Theory”, (January 2015).

[9]. Yi-Hsuan Kao and Bhaskar Krishnamachari, Moo-Ryong Ra, Fan Bai. ”Hermes: Latency Optimal Task Assignment for Resource-constrained Mobile Computing”.

[10]. Bo Gao1, Ligang He1, Limin Liu2, Kenli Li and Stephen A. Javis “From Mobiles to Clouds: Developing Energy-aware Offloading Strategies for Workflows “.

[11].Doron Zarchy, David Hay, Michael Schapira. “Capturing Resource Tradeoffs in Fair Multi-Resource Allocation”.