International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 8, August - 2014

Resource Matching Job Scheduling Algorithm in Mobile Cloud Computing Environment

Resource Matching Job Scheduling Algorithm

Neha Kapoor
Research Scholar, CSE
DAV Institute of Engineering & Technology
Jalandhar, India

Parveen Kakkar
Assistant Professor, CSE
DAV Institute of Engineering & Technology
Jalandhar, India

***Abstract*— Mobile Cloud Computing (MCC) is a new emerging technology that has revolutionized the way in which mobile subscribers across the globe can enjoy abundant multimedia applications on the go. The scheduling of massive multimedia flows is a complicated task in the mobile cloud environment. The goal of this paper is to propose a model for job-oriented resource scheduling in a mobile cloud computing environment in which the workflow is allocated or scheduled to the process which gives the available resources such as RAM, Bandwidth, MIPS etc. In this paper we did the analysis of job scheduling algorithms i.e. Blind Online Scheduling Algorithm (BOSA) and Resource Matching with Priority (RMP) on the basis of the performance metrics such as TurnAround Time, Waiting Time, Response Time and Throughput. From the results, it has been computed that Resource Matching with Priority Algorithm (RMP) has the lowest time parameters and maximum throughput and proves to be the most efficient algorithm for resource scheduling.**

Keywords*— Job Scheduling, Process, Mobile Cloud Computing.

I. INTRODUCTION

Mobile computing is based collectively on three major concepts: hardware, software and communication. The hardware can be considered as mobile devices, such as Android phone and laptop. The mobile computing software is comprised of the numerous mobile applications installed the devices, such as the anti-virus software, browser, g-mail and games. The communication is comprised of the infrastructure, protocols and data delivery techniques which must be transparent to the end users.

MCC is defined as "a mobile computing technology that makes use of elastic resources of various clouds and network technologies providing unrestricted functionality, storage, and mobility to mobile devices anywhere, anytime based on the pay-as-you-go principle."

Job scheduling is defined as the process to schedule the jobs to the available resources by finding out a sequence order in which the jobs can be executed considering various transaction logic constraints. The effective scheduling algorithm is one that spreads the load evenly on the available processors thus maximizing the processor's utilization while minimizing the total execution time.

II. SCHEDULING CRETERIA

CPU scheduling [8] is the process of allocating a process to perform its task. Scheduling of the process is based on certain criteria of the algorithm. Criteria include:

- **CPU utilization:** CPU should be kept as busy as possible. That is, it should involve a maximum usage.
- **Throughput:** Time in one measure of work that the numbers of processes are completed per unit time, is supposed to be maximum.
- **Turn Around time:** The time that is involved from the time of submission of a process till its completion, is supposed to be minimum.
- **Waiting time:** The addition of the time that the process is waiting for its execution, is supposed to be minimum.
- **Response time:** The time that the process is submitted for a request until the first response is produced, is supposed to be minimum.

III. PROBLEM FORMULATION

The previous work done in this area assumes that the all the service requests in the cloud computing are linear in nature even if the request load is heavy , It assumes it is linearly increasing and there is not much non–linearity however , this is not the case in real life scenarios. Any service request will have some peak and off peak time with request to day hours, week hourly and may even have seasonal peaks and valleys as a matter of fact, but at the same it is also known fact it will have some pattern which may be recognizable to some extend by using machine learning algorithms or by using some statistical tests etc. There is strong case where there may be need to find upper and lower bounds to find sub optimal solution locally of the complete dataset sample considered at particular time and then come the some logical conclusion especially when there is need to find new correlation based finding is to be determined as in the previous work which works on considering service rate , resource allocation , service routing to match resources in mobile cloud environment , therefore , there is need to work in realistic function designs which can incorporate the patterns that arise of real time rather than linear.

IV. METHODOLOGY ADOPTED

The methodology that we will follow is that first of all the cloud simulated environment would be developed using simulator CloudSim in NetBeans IDE 7.4. After that workload distribution policy is to be designed. Then workload will be distributed according to the policy framed considering the following attributes such as arrival rate, service request rate, and available resources. After the distribution of the workload the next step is to design the scheduling policy and an algorithm. Then the workload will be executed using the scheduling algorithm proposed. At last the performance of a designed algorithm will be evaluated.

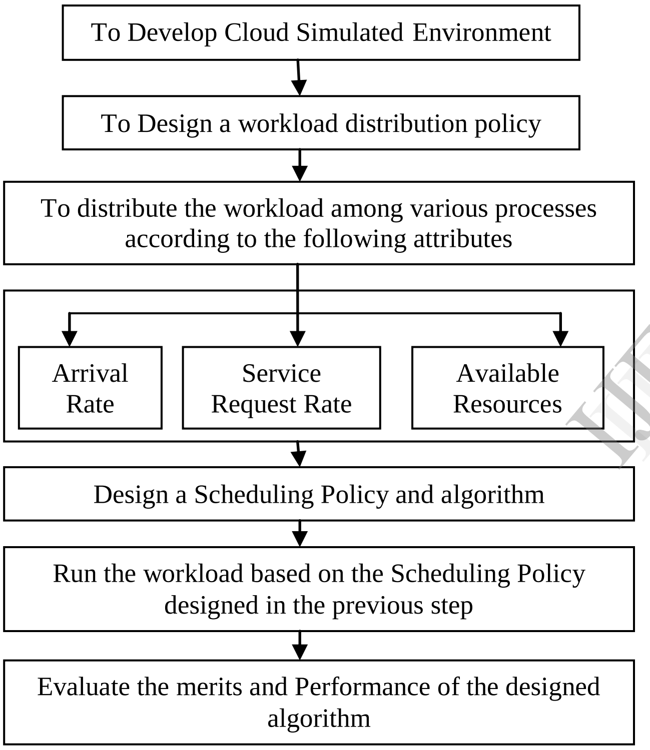

Fig1: Flow Diagram of Research Methodology Adopted

A. *Blind Online Scheduling Algorithm (BOSA)*

BOSA algorithm is based on a multiple-channel queuing system in which two or more channels (servers) are available to handle jobs that arrive for service. In the base paper, author has considered the most basic multiple channel system that contains parallel data centers serving a single queue on a first-come first served (FCFS) basis. The data centers all provide the same service. The single queue may separate into shorter queues in front of the respective data centers. BOSA algorithm is based on the following assumptions:

1) The input of jobs is infinite.
2) The service is on first-come first served basis.
3) The arrival of jobs follows Poisson probability law and service time has an exponential distribution.
4) There are 'k' virtual machines in 'n' data centers in r region, each of which provides identical service.
5) The arrival rate is smaller than the combined service rate of all k service facilities.

ALGORITHM 1: BOSA

STEP 1: Consider u = {1, ……., U} represent set of user classes and s = { 1, ……, S} represent set of server classes.

STEP 2: Get the Task List according to an independent Poisson Processes in the first-come first-served basis based on the calculations done by applying Poisson-Exponential Multiple Channel Queuing Model where, $P_{u,}$ u ϵ U.

STEP 3: Now, considering the assumption 3 & assumption 4, BOSA satisfies an exponential function and the server from the same class i.e. s ϵ S has the same average service time.

STEP 4: Let γ be a specific scheduling scheme and t >= 0, be a time slot, the waiting time of a class u ϵ U is given by W_u (t; γ).

STEP 5: Find the appropriate cloudlet/task providing the service for each R in Regions, DC in Data Center, and JOB in Task List.

STEP 6: Get the state of a virtual machines say VM from a list of a virtual machines.

STEP 7: If VM_STATE = IDLE, then add the job to the scheduled list.

STEP 8: Calculate the service time for the job and compare it with the minimum service time given by the formula in [3]

$$\min_{\gamma} \sum_{u=1}^{U} \sum_{k=1}^{P_u(T)} \frac{W_{u,k}(\gamma)}{P_u(T)} \quad (1)$$

$$s.t.\ P\left(\frac{Y_s(t;\gamma)}{\sum_{Y_s} Y(t;\gamma)} \right) \geq 1-\eta$$

Where
η ϵ (0,1) is the venture level
P_u(T) shows the recorded user request over time period T
$W_{u,k}$ represents the waiting time up to T by the k^{th} class u that has requested the service at the beginning of time.

STEP 9: If the service time of a job < Min Time Formula as given above in the equation (1), add the job to the cloudlet.

STEP 10: If no Virtual Machine available, job must wait in the waiting queue.

B. *Resource Matching With Priority Scheduling Algorithm (RMP)*

Priority of jobs is an important issue in scheduling because some jobs should be serviced earlier than other those jobs can't stay for a long time in a system. So, we have proposed a prioritized resource matching scheduling algorithm and implemented in a simulated environment.

ALGORITHM 2: RMP

STEP 1: Find the appropriate Task List for each 'R' in regions and for each 'DC' in data centers.

STEP 2: Find the state of each virtual machine.

STEP 3: If (VM == Idle), find the minimum sized cloudlet ().

STEP 4: Calculate the Score for a Current available resources as given by formula:
Score of Current available resources = Current Total Million Instructions per Second * Current RAM * Current Bandwidth;

STEP 5: Calculate the Score of Required Resources as given by formula:
Score of Required Resource = Required Total MIPS * Required RAM * Required Bandwidth;

STEP 6: If the Score of current available resources is less than Score of the requested resources then assign the first task to the first available Virtual Machine.

STEP 7: If Virtual Machine is busy then wait in the waiting queue, else perform step 6.

V. EXPERIMENTAL SETUP

The scheduling algorithms have been executed and resulted in a simulated environment using a simulator named CloudSim. CloudSim is a java based tool especially for the execution of cloud based application or simulation.

A. *Simulation Description*

CloudSim version 3.0.2 is used to implement scheduling algorithms in Mobile Cloud Computing. The computer running Window 7 operating system is used for simulation.
We have implemented the "Prioritized Resource Matching Scheduling" model and compared it with Blind Online Scheduling Algorithm (BOSA) initially by taking into account the various parameters like Turnaround time, Waiting time, Response Time and Throughput.

B. *Configuration of Data Centers*

1) Configuration of Processing Elements

TABLE I: CONFIGURATION OF PROCESSING ELEMENTS

PARAMETERS	VALUE
No of Processing Element	2
Processing Power	2000

2) Configuration of Hosts

TABLE II: CONFIGURATION OF HOSTS

PARAMETERS	VALUE
No of Host	1
RAM	2048 MB
Storage	1000000 MB
Bandwidth	10000
VM SCHEDULING	Time-Shared

3) Configuration of Virtual Machines

TABLE III: CONFIGURATION OF VIRTUAL MACHINES

PARAMETERS	VALUE
No of VM	5
Image Size	10000 MB
RAM	512 MB
VM name	"XEN"

VI. RESULTS AND ANALYSIS

We have done the comparison of turnaround time, waiting time, response time and throughput of jobs when executed by applying "Resource Matching with Priority Scheduling Algorithm (RMP)" versus the "Blind Online Scheduling Algorithm (BOSA)". Figure 2 shows a graphical representation of comparison between two algorithms considering waiting time.

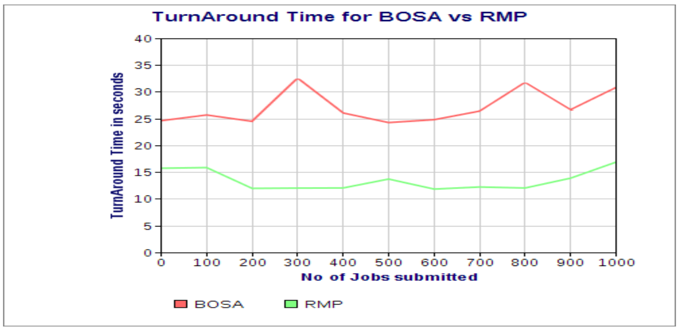

Fig.2. Comparison of BOSA and RMP Algorithm based on their TurnAround Time

Fig. 3 shows a graphical representation of comparison between two algorithms considering waiting time.

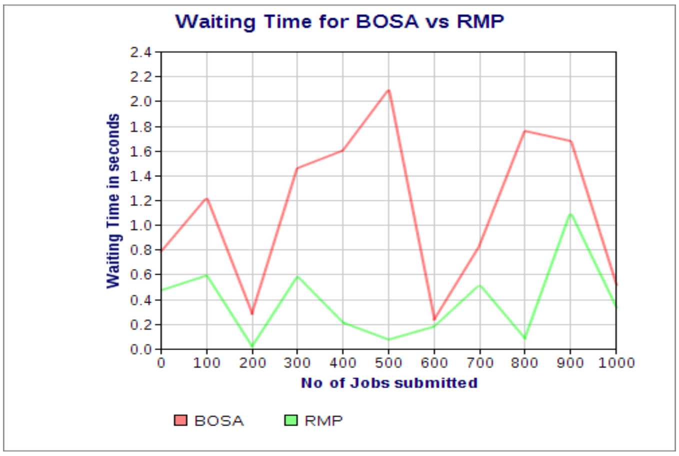

Fig.3. Comparison of BOSA and RMP Algorithm based on their waiting Time

Fig. 4 shows a graphical representation of comparison between two algorithms considering response time.

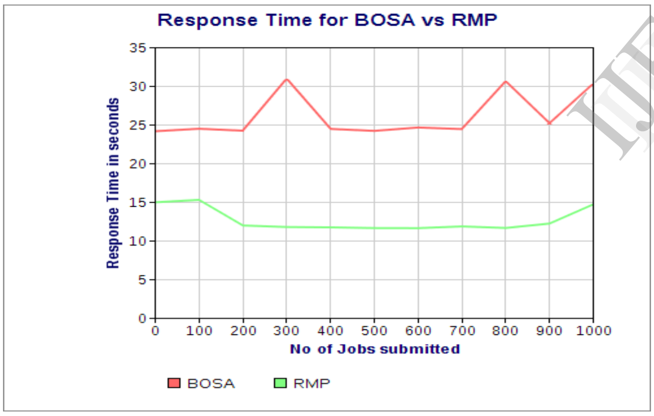

Fig.4. Comparison of BOSA and RMP Algorithm based on their Response Time

Fig 5 shows a graphical representation of comparison between two algorithms considering Throughput per unit time.

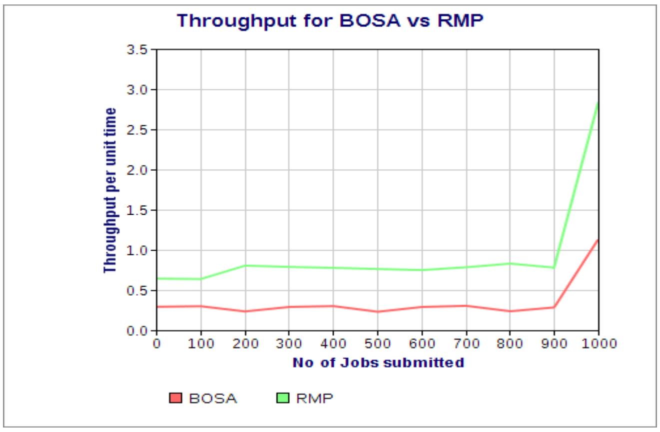

Fig.5. Comparison of BOSA and RMP Algorithm based on their Throughput

VII. CONCLUSION

Scheduling is one of the most important tasks in mobile cloud computing environment. In this paper, we have analyzed two scheduling algorithms i.e. Blind Online Scheduling Algorithm (BOSA) and Resource Matching with Priority Algorithm (RMP) and tabulated various parameters.

From the observed results, it is concluded that:

1. The Resource Matching with Priority (RMP) in mobile cloud computing has reduced the turnaround time of jobs when compared with Blind Online Scheduling Algorithm (BOSA).
2. The waiting time for the jobs has been reduced in case of Resource Matching with Priority Scheduling Algorithm as compared to Blind Online Scheduling Algorithm (BOSA).
3. The response time has also been improved using Resource Matching with Priority Algorithm.
4. The throughput per unit time has been maximized in case of Resource Matching with Priority Scheduling Algorithm (RMP) as compared to Blind Online Scheduling Algorithm (BOSA).

Hence, RMP Algorithm proves to be a more efficient Scheduling algorithm for allocating mobile cloud computing resources to the jobs submitted by the users. Despite of giving least execution time and cost effective, the existing scheduling algorithms do not consider reliability issue. In future, we will enhance our algorithm by considering reliability issue using failure generator and failure monitor component of workflow management system for resource scheduling.

VIII. ACKNOWLEDGEMENT

This research paper is made possible through the help and support of my guide and mentor Mr. Parveen Kakkar. I would like to thank him for his utmost support and encouragement.

REFERENCES

[1] Xiaocheng Liu, Chen Wang, Bing Bing Zhou, Junliang Chen, Ting Yang, Albert Y. Zomaya. "Priority-Based Consolidation of Parallel Workloads in the Cloud." IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 9, Sep 2013.

[2] Susheel Thakur, Arvind Kalia, Jawahar Thakur. "Server Consolidation Algorithms for Cloud Computing Environment: A Review." International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 9, Sep 2013.

[3] Liang Zhou And Zhen Yang, Nanjing University Of Posts And Telecommunications Joel J. P. C. Rodrigues, Instituto De Telecomunicac,O~ Es, University Of Beira Interior Mohsen Guizani, Qatar University. "Exploring Blind Online Scheduling For Mobile Cloud Multimedia Services." IEEE Wireless Communications, Jun 2013.

[4] Sung-Min Jung, Nam-Uk-Kim, Tai-Myoung Chung. "Applying Scheduling Algorithms with QOS in the Cloud Computing." in proc. of International Conference on Information Science and Applications, Jun 2013, pp. 63-64.

[5] L. Zhou and H. Wang. "Toward Blind Scheduling in Mobile Media Cloud: Fairness, Simplicity, and Asymptotic Optimality," IEEE Transactions on Multimedia, vol. 15, no.4, Jun 2013.

[6] Niroshinie Fernando, Seng W. Loke , Wenny Rahayu. "Mobile cloud computing: A survey." Future Generation Computer Systems, vol.29, no.1, pp. 84-106, Jan 2013.

[7] N.Usha , Bhavana. V, Manjunath.A.E. "Mobile Applications: Delivery Technologies In Multimedia Cloud Computing." International Journal of Engineering Research & Technology (IJERT), vol. 2, no.1, pp. 2278-0181, Jan 2013.

[8] Silberschatz, A. P.B. Galvin and G. Gagne (2012), Operating System Concepts, 8th edition, Wiley India.