Volume 7, Issue 1, January 2017 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Energy Aware Resource Allocation in Cloud Computing

Gagandeep Kaur, Arshdeep Singh
Bhai Maha Singh College of Engineering, Kotkapura Road, Sri Muktsar Sahib,
Punjab, India

Abstract— This implementation aims towards the establishment of performance qualitative analysis on make span in VM task allocation and process according to their deadline, then implemented in CloudSim with Java language. Here major stress is given on the study of dead line based task scheduling algorithm with heterogeneous resources of the cloud, followed by comparative survey of other algorithms in cloud computing with respect to scalability, homogeneity or heterogeneity and process scheduling. A previous study also indicates change of MIPS will affect the response time and increase in MIPS versus VM decreases the response time. When image size of VM is implemented against the VM bandwidth then no significant effect is found on response time and it remains constant for which these parameters are investigated. But in case of Cloudlet long length versus Host bandwidth a pattern is observed in which response time increases in proportionate manner. Using the modified approach the reduction in the down time of the various processes are achieved as shown in results.

Keywords— cloud computing, task scheduling, virtual machine.

I. INTRODUCTION

In cloud computing, load balancing technique intends for allocating the load among the physical servers to advance the scalability of servers and increasing the resource consumption, energy overheads etc for high availability and throughput.

The virtual to virtual migration is the technique of migrating the load of overloaded virtual machine from one physical machine to the under loaded machine having high speed while virtual machine is still running.

Cloud computing is moderately prevailing technology. Cloud computing has newly appeared as one of the exhortation in the ICT industry. It is a innovative technology of computing that is broadly utilized in today's industry and society. It is newest technology for sharing resources. Cloud computing is the Internet-based computing that permits to shared resources, software's and information can be offered to computers and other devices on requirement.

The user can access applications anywhere that runs in the cloud, through an Internet-enabled mobile device or an associated computer and existing 7x24x365. With a cloud app, open a browser, log in, customize the app, and start using it. Amazon was the creator of the clouds and proposed cloud services to public like Amazon Elastic Compute Cloud (EC2), Simple Storage Service (S3). Cloud computing is not an entirety latest concept; it is initiated from the large-scale Grid environment. It's pay as you go model means that users pay the services and use the cloud. Cloud Computing give 3 sorts of services like PAAS (Platform As A Service)[1], SAAS (Software As A Service), IAAS (Infrastructure As A Service)[1]. Cloud User use the services of the cloud tho' net. Currently a day's range of user will increase to the usage of cloud therefore range of user will increase within the cloud. Therefore at a time accessibility of the resources and satisfy the client demand programming is important. The programming main goal is to balance the load on distributed system, most utilization of the resources with minimum completion time. During this we have a tendency to study programming parameters like performance, time interval, waiting time, throughput, make span etc.[1]

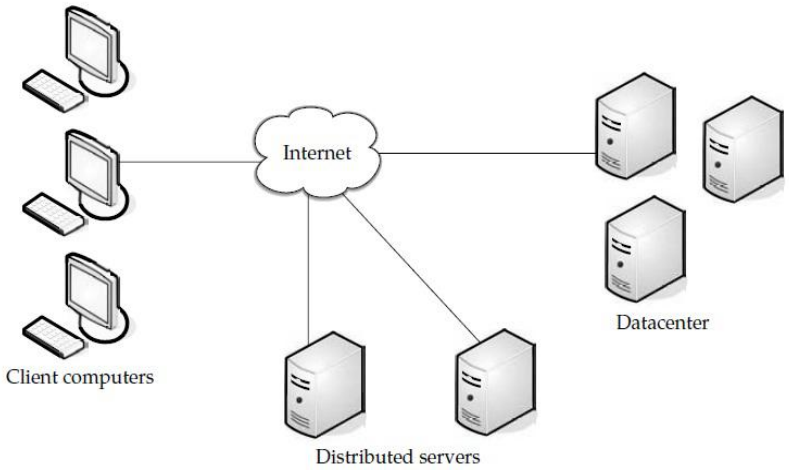

Fig 1: Cloud Computing [15]

© 2017, IJARCSSE All Rights Reserved

The services provided by the cloud area unit categorized into the following three cloud service models [1]:

Software as a Service (SaaS): It provides ability to cloud users to access and use the applications of cloud provider on pay-per-use basis. User can access these applications simply through browser whereas cloud provider manages the underlying infrastructure required for running such applications.

Platform as a Service (PaaS): throughout this service model, cloud service provider distributes computing platform so as that users can develop their own applications exploitation programming languages whereas not having any overhead of managing underlying hardware and package layers. The provided computing platform would possibly contains software package, setting to support program execution, direction systems etc.

Infrastructure as a Service (IaaS): It offers ability to the users for exploitation the infrastructure (Physical resources) like processors, storage disks, RAMs, routers etc provided by the service provider on pay-per-use basis. exploitation this model, little organizations can avoid the big worth of buying such infrastructure. These physical resources area unit virtualized therefore on share them among multiple cloud users. jointly there area unit following four cloud preparation models that show the means that through that cloud services area unit utilized by its users [13].

Private Cloud: personal cloud is build for the exclusive use by single organization. that means all the resources provided by the non-public cloud are accessed and used only by users of the organization World Health Organization owns that cloud. Main sensible issue regarding exploitation personal cloud is its security since its resources are shared at intervals utterly totally different users of same organization. Another advantage of non-public cloud is its ability to supply customization that allows organization to mildew it in step with demand. but the matter with personal cloud is that it provides less measurability. Community Cloud: Community Cloud permits for sharing its resources among the users of multiple organizations that are having same desires and objectives. This cloud divides initial establishment worth among several organizations. These clouds provide somewhat extra measurability of resources than personal cloud. Public Cloud:

Public cloud provides unlimited storage, services and computing setting to the users all over world through net on pay-per-use basis. Public clouds area unit built and managed by third party agencies. Public clouds provide extra measurability, accessibility and flexibility than personal clouds. but inadequate security might be a serious disadvantage with the final public cloud since the resources provided by public cloud area unit shared among sizable quantity of worldwide users from utterly totally different organizations.

Hybrid Cloud: Hybrid clouds area unit built by combining the non-public and public clouds. Hybrid cloud thus aggregates the properties of every personal and public clouds like measurability, flexibility and security. Throughout this model, users of non-public clouds use the resources of public cloud once its own resources become inadequate. The extra required resources area unit taken from public cloud on pay-per-use basis.

II. TASK SCHEDULING IN CLOUD

Scheduling is that the cluster of strategies that manage the order of execution of multiple tasks on the processors therefore on decrease the time and value required to execute of those tasks. at intervals the cloud atmosphere, task hardware plays vital role of allocating cloud provider's resources among the massive type of users. Task designing deals with distribution of the tasks among the cloud servers that technique or execute these tasks for user (or client). associate economical task designing policy provides correct utilization of resources, load deed and improvement of execution value and time. therefore these days task designing is main analysis topic at intervals the area of cloud computing. There ar various varieties of designing like static, dynamic, pre-emptive, non pre-emptive, centralized and distributed designing.[1]

A lot of scheduling algorithms are available today for cloud computing. But their performance is questionable. Different parameters are to be considered for scheduling. Two major parameters are task size and priority considered in the proposed approach. But these are not the only ones. There are some other parameters which influences the scheduling of tasks and utilization of resources. Lot of studies are taking place in this area.

a) Task Size (Min Min Algorithm): The effective utilization of resources can be increased by using a load balancing algorithm. This is achieved by making use of resources of an unused(idle) processor while release the resources of processors having heavy load. The load balancing algorithm distributes the load among all the resources which are available. This type of algorithm also minimize the makespan with the valuable use of resources3,16,17. Min Min algorithm starts with a set of tasks. There is a task set T. Along with the task set there are some resources too. After implementing the algorithm, task set will be allocated or mapped to resource set. The algorithm chooses the task having minimum size.
b) Priority Based: This technique works based on the user's priority of tasks. But in such a case, other problems may arise. In optimal conditions, the cloud system gets good results when shortest job is executed first. But this won't always be the case in this technique. Sometimes task with larger length is associated with a highest priority. This is the big issue in this algorithm.
c) Cost Based Resource Scheduling: Each resource is associated with a cost. The resource with lowest cost will be assigned to the task. The practical difficulty of this concept is that when a user needs a resource having a highest price. In this situation user will not get the resource. So scheduling based on the resource cost has limitations.
d) Resource Scheduling Based on Energy Efficient Methods: The cloud computing architecture used today has number of hardware components to accomplish the user needs. Cloud providers are trying to reduce the energy consumption13. There is no direct relationship between energy consumption and task scheduling. The paper doesn't tell how to do actual scheduling of tasks.

e) Scheduling Based on Reliability Factor: Proper scheduling eliminates overloaded conditions during the utilization of resources. It also explains how to handle a failure. This factor(Failure) is not relevant because failure may happen to any system. The parameters that are more important need more thought.
f) Scheduling Based on Activity Based Costing: In order to execute a task, different resources have to be used. Cloud service provider charges a cost based on the usage of resources. The paper doesn't tell about the parameters that are essential for scheduling. Proper scheduling reduces the cost on each resources.
g) Context Aware Scheduling: This situation can be avoided by utilizing the resources effectively10. The method used to solve the situation is to execute the client request in the client side itself without moving into the server.
h) Dynamic Slot Based Scheduling: Hadoop is a concept that deals with big data. MapReduce function is used for handling this big data. Each task gets a time slot11. During that period task gets executed. This approach checks the utilisation of resources. This is a complex approach because it needs migration of tasks. Task migration is difficult when we deal with big data.
i) Credit Based Scheduling: This approach considers two parameters: (1) Task Length and (2) User Priority. The algorithm is based on credit system. Each task is assigned a credit based on their task length and priority. In the actual scheduling of the task, these credits will be considered.

Task length credit:

The cloud system executes tasks having different length. If the tasks are arranged based on the increasing order of length, tasks having shorter length will reside at the beginning of the array and the task having highest length will be present at the last.

III. RELATED STUDY

Deshmukh A. et.al. [1] have discussed/described different load balancing strategies, algorithms and methods and by studying pros and cons of different techniques used for load balancing, authors are specifically giving priority to Dynamic load balancing method rather than Static load balancing. They investigate that comparative behavior of load balancing with different parameters; dynamic load balancing is more reliable and after that they conclude that efficient load balancing can clearly provide major performance benefit.

Alam, B. et.al. [2] states that in Round Robin Scheduling, the time quantum is fixed and then processes are scheduled such that no processes get CPU time more than one time quantum in one go. If time quantum is too large, the response time of the processes is too much which may not be tolerated in interactive environment. If time quantum is too small, it causes unnecessarily frequent context switch leading to more overheads resulting in less throughput. In this paper, a method using fuzzy logic has been proposed that decides a value that is neither too large nor too small such that every process has got reasonable response time and the throughput of the system is not decreased due to unnecessarily context switches.

Hongchao Hu et.al. [7] presented that the limitations in complexities and extensibilities of CICQ scheduling policies are first analysed. Then, based on this analysis, the guidelines for designing high extensible scheduling policies and the concept of virtual channel are proposed. Based on the guidelines and virtual channel, it comes up with a dynamic round robin scheduling algorithm-FDR (fair service and dynamic round robin), which is simple, high efficiency and fair service. FDR is based on round robin mechanism and its complexity is of order one(O(1)). It allots the scheduling share for each virtual channel according to its current states. Thus, FDR has good dynamic and real-time performance, and it can adapt to unbalanced traffic load network environment. Simulation results under SPES show that FDR performs good delay, throughput and anti-burst performance, which can be applied in high performance routing and switching devices.

Berra, P.B. et.al. [11] have discussed for policies for request assignment that with High-performance servers and high-speed networks will form the backbone of the infrastructure required for distributed multimedia information systems. A server for an interactive distributed multimedia system may require thousands of gigabytes of storage space and a high I/O bandwidth. In order to maximize the system utilization, and thus minimize the cost, it is essential that the load be balanced among each of the server's components, viz. the disks, the interconnection network and the scheduler. Many algorithms for maximizing retrieval capacity from the storage system have been proposed in the literature. This paper presents techniques for improving the server capacity by assigning media requests to the nodes of a server so as to balance the load on the interconnection network and the scheduling nodes. Five policies for request assignment-round-robin (RR), minimum link allocation (MLA), minimum contention allocation (MCA), weighted minimum link allocation (WMLA) and weighted minimum contention allocation (WMCA)-are developed. The performance of these policies on a server model developed by the authors (1995) is presented. Authors also consider the issue of file replication, and develop two schemes for storing the replicas: the parent group-based round-robin placement (PGBRRP) scheme, and the group-wide round-robin placement (GWRRP) scheme.

Jingnan et.al. [12] have presented that with the advent of powerful network processors (NPs) in the market, many computation-intensive tasks such as routing table lookup, classification, IP sec, and multimedia transcoding can now be accomplished more easily in a router. An NP consists of a number of on-chip processors to carry out packet level parallel processing operations, ensuring good load balancing among the processors increases throughput. However, such type of multi processing also gives rise to increased out-of-order departure of processed packets. This work first proposes an Ordered Round Robin (ORR) scheme to schedule packets in a heterogeneous network processor assuming that the workload is perfectly divisible. The processed loads from the processors are ordered perfectly. This finding analyzes the throughput and derives expressions for the batch size, scheduling time and maximum number of schedulable processors.

© 2017, IJARCSSE All Rights Reserved

Guofei Sun et.al. [13] states that the current virtual machine(VM) resources scheduling in cloud computing environment mainly considers the current state of the system but seldom considers system variation and historical data, which always leads to load imbalance of the system. In view of the load balancing problem in VM resources scheduling, this paper presents a scheduling strategy on load balancing of VM resources based on genetic algorithm. According to historical data and current state of the system and through genetic algorithm, this strategy computes ahead the influence it will have on the system after the deployment of the needed VM resources and then chooses the least-affective solution, through which it achieves the best load balancing and reduces or avoids dynamic migration. This strategy solves the problem of load imbalance and high migration cost by traditional algorithms after scheduling. Experimental results prove that this method is able to realize load balancing and reasonable resources utilization both when system load is stable and variant.

Milan Sokile et.al. [15] in this paper, author has studied that the comparison are made between various techniques but static load balancing algorithm are more stable and it is also easy to predict their behavior, but at a same time dynamic distributed algorithm are always considered better than static algorithm. Experimental results of performance modeling show that dynamic load balancing is better than static load balancing in a dynamic environment, which manifest in frequent clients' object creation requests and in short objects' lifetimes.

IV. EXISTING SCHEMES

The following task designing algorithms space unit presently established among the cloud environments

A. hymenopteran Colony improvement (ACO)-inspired:

A replacement Cloud hardware supported hymenopteran Colony improvement is that the one resented by Cristian Mateos. The goal of our hardware is to attenuate the weighted flowtime of a gaggle of PSE jobs, whereas collectively minimizing Makespan once using a Cloud. among the ACO formula, the load is calculated on each host taking into consideration the equipment utilization created by all the VMs that unit of measurement punishment on each host. This metric is helpful for Associate in Nursing hymenopteron to choose the tiniest quantity loaded host to assign its VM.Parameter Sweep Experiments (PSE) may well be a spread of numerical simulation that involves running Associate in Nursing outsized sort of freelance jobs and frequently desires lots of computing power. These jobs ought to be with efficiency processed among the utterly totally different computing resources of a distributed setting like those provided by Cloud. Consequently, job designing throughout this context therefore plays a basic role. throughout this formula, Makespan and flowtime unit of measurement evaluated. analysis results of this metrics show that ACO performance more than two totally different (Random and Best effort) algorithms.[8]

B. Min-Min Algorithm:

Min-Min begins with a gaggle of tasks that unit of measurement all unassigned. First, it computes minimum completion time for all tasks on all resources. Then among these minimum times the minimum value is chosen that's that the minimum time among all the tasks on any resources. Then that task is regular on the resource on it it takes the minimum time and thus the on the market time of that resource is updated for all the alternative tasks. it's updated throughout this manner; suppose a task is assigned to a machine and it takes twenty seconds on the assigned machine, then the execution times of all the alternative tasks on this assigned machine square measure planning to be increased by twenty seconds. once this the assigned task is not thought of and thus a similar technique is perennial until all the tasks unit of measurement assigned resources.

C. Max-Min formula:

Max-Min is kind of same as a result of the min-min formula except the following: throughout this once looking for the completion time, the minimum execution times unit of measurement famed for each and every task. Then among these minimum times the most value is chosen that's that the most time among all the tasks on any resources. Then that task is regular on the resource on it it takes the minimum time and thus the on the market time of that resource is updated for all the alternative tasks. The modification is finished among a similar manner as for the Min-Min.

D. Particle Swarm improvement (PSO) Algorithm:

Particle Swarm improvement (PSO) as a meta-heuristics technique may well be a self-adaptive international search based totally improvement technique introduced by Kennedy and Eberhart [5]. The PSO formula is alike to totally different population-based algorithms like Genetic algorithms (GA) but, there is not any direct recombination of individuals of the population . The PSO formula focuses on minimizing the worth of computation of Associate in Nursing application advancement. As a live of performance, Authors used worth for complete execution of application as a metric. the target is to attenuate the worth of execution of application workflows on Cloud computing environments. Results show that PSO primarily based task-resource mapping square measure ready to do a minimum of thrice worth savings as compared to Best Resource selection (BRS) based mapping for our application advancement. in addition, PSO balances the load on cipher resources by distributing tasks to on the market resources.[5]

E. spherical Robin Algorithm:

The spherical Robin formula in the main focuses on distributing the load equally to any or all the resources. Victimization this formula, the broker allocates one VM to a node throughout a cyclic manner. The spherical robin

designing within the cloud computing is unbelievably rather like the spherical robin designing utilized within the technique designing. The hardware starts with a node and moves on to future node, once a VM is assigned to that node. this can be often perennial until all the nodes square measure assigned a minimum of 1 VM then the hardware returns to the first node over again. Hence, throughout this case, the hardware does not stay awaken for the exhaustion of the resources of a node before moving on to future. Though spherical robin algorithms unit of measurement supported straightforward rule, lots of load is planned on servers and thus unbalancing the traffic. Results of spherical Robin formula shows higher amount and payload deed as compared to the alternative formula.[7]

F. Genetic Algorithm:

Genetic formula may well be a way of coming up with among that the tasks unit of measurement assigned resources per individual solutions (which unit of measurement called schedules in context of scheduling), that tells regarding that resource is to be assigned to it task. Genetic formula relies on the biological construct of population generation. The foremost terms utilized in genetic formula are[6]

a. Initial Population

Initial population is that the set of all the those that unit of measurement utilized within the genetic formula to hunt out the optimum resolution. every resolution among the population is called as a private. and every individual is pictured as a body for making it acceptable for the genetic operations. From the initial population the folks unit of measurement elect and a number of operations unit of measurement applied on those to form future generation. The coupling chromosomes unit of measurement elect supported some specific criteria.[6]

b. Fitness operate

A fitness operate is used to measure the quality of the folks among the population per the given improvement objective. The fitness operate could also be utterly totally different for numerous cases. In some cases the fitness operate could also be supported purpose in time, whereas in cases it's going to be supported budget constraints.

c. Selection

We use the proportion selection operator to ascertain the prospect of various folks genetic to future generation in population. The proportional selection operator suggests that the prospect that's chosen and genetic to next generation groups is proportional to the dimensions of the individual's fitness.

d. Crossover

We use single-point crossover operator. Single-point crossover suggests that only one intersection was started among the individual code, at that point a district of the strive of individual chromosomes is modified.[8]

e. Mutation

Mutation means that the values of therefore sequence locus among the body cryptography series were replaced by the alternative sequence values so on get a replacement individual. Mutation is that negates the price at the modification points with reference to binary coded folks.

V. RESEARCH GAPS

Outages. Have you ever been unable to access your email due to your provider being down? Now, imagine if you needed a document for an important business meeting or presentation and your storage prov. Remote server security makes it harder, but not impossible, for hackers to reach your data. If there is a compromise of the server(s) where your data is stored, your personal information may be exposed to the world. There's also a good chance that more than just your information may be affected ider's site was down. Believe me it happens, and it happens at the most inconvenient times.

Storage limits. While your local hard drive may be able to hold 500GB or more of data, unfortunately a remote server may only allow you to freely store about 5GB. If you want more room, you'll have to pay. Still, even with a paid account, it can't begin to touch the amount of room you have locally. There also may be a limit on the size of the data that can be stored.

Limited features. If you use remote software that's provided by the storage service to manipulate and modify your data, it usually lacks the features of a program running locally.It is one of the simplest scheduling techniques that utilize the principle of time slices. Here the time is divided into multiple slices and each node is given a particular time slice or time interval i.e. it utilizes the principle of time scheduling.

Throttled: In this algorithm the client first requests the load balancer to find a suitable Virtual Machine to perform the required operation.The process first starts by maintaing a list of all the VMs each row is individually indexed to speed up the lookup process. If a match is found on the basis of size and availability of the machine, then the load balancer accepts the request of the client and allocates that VM to the client. If, however there is no VM available that matches the criteria then the load balancer returns -1 and the request is queued.

Execution Load: It is spread spectrum technique in which the load balancer spread the load of the job in hand into multiple virtual machines. The load balancer maintains a queue of the jobs that need to use and are currently using the services of the virtual machine. The balancer then continuously scans this queue and the list of virtual machines. If there is a VM available that can handle request of the node/client, the VM is allocated to that request [17]. If however there is a VM that is free and there is another VM that needs to be freed of the load, then the balancer distributes some of the tasks of that VM to the free one so as to reduce the overhead of the former VM. The jobs are submitted to the VM manager, the load also maintains a list of the jobs, their size and the resources requested. The balancer selects the job that matches the criteria for execution at the present time.

© 2017, IJARCSSE All Rights Reserved

The existing methods will evenly distribute the traffic, but does not take into account the current load or responsiveness of the nodes.

The existing methods are intelligent, but also do not consider the current load or responsiveness of the nodes.

This can be a problem if you need to add power quickly, but can be beneficial if you have a server that is experiencing problems keeping up with its inbound traffic.

VI. METHODOLOGY

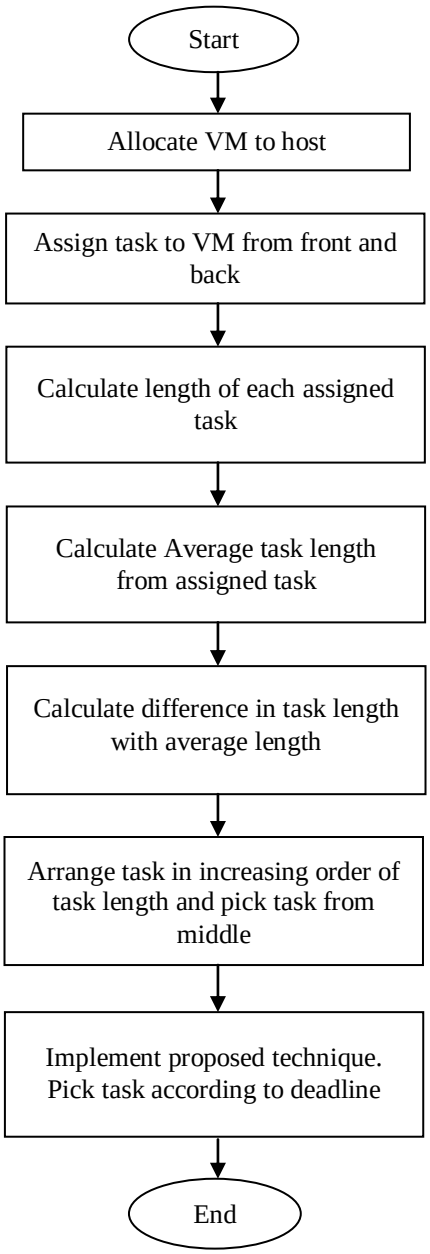

Fig 2: Flowchart

1) In the very first step the vm are to be placed on the host according to the resource vector and assign privileges to vm.
2) Assign task lost on the virtual machines from both sides front and back.
3) Then one has to calculate the task length of each task and then calculate the average value of each task.
4) Calculate the difference of average value and individual task length for each task that is assigned to each vm.
5) Arrange task from the middle of the queue generated after calculating the difference.
6) Add the task deadline and calculate the priority of each task.
7) Generate and validate results for both approach with deadline and without deadline.

VII. RESULTS AND DISCUSSION

Fig 3 is the comparative study for priority/length vs Cloudlet Id. In this graph priority of tasks are rescheduled according to the deadline provided to the task. From this figure it is clear that the priority/length ratio against cloudlet id is better in case of proposed approach as compared to existing one.

© 2017, IJARCSSE All Rights Reserved

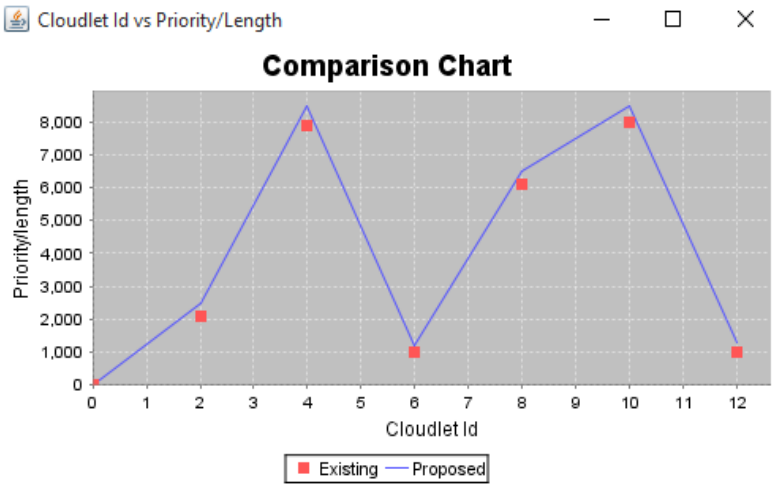

Fig 3: Comparison of Priority/length vs Cloudlet Id

The value of proposed scenario in case of priority/length vs cloudlet id is more than 8000 where as in case of existing scenario it is only 8000. From these results it may be justified that the priority/length ratio is better in case of proposed scenario as compared to the existing one.

Fig 4 is the comparative study for Make Span vs Cloudlet size. Make span is the condition when the system is in ideal case when it is not doing any kind of operation so lagging occur due to it.

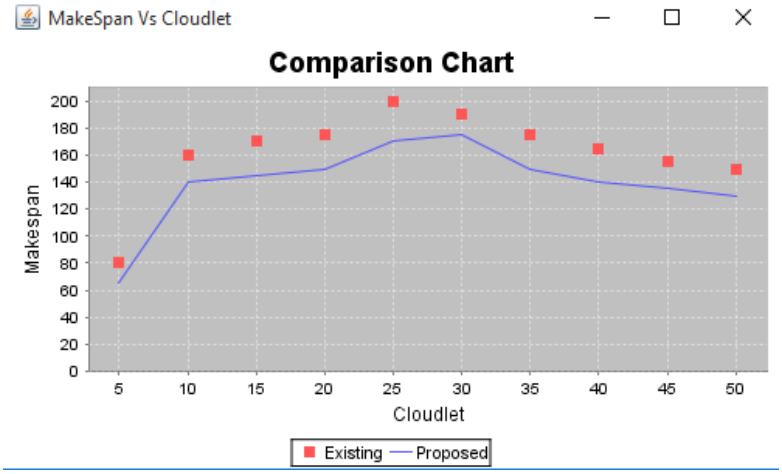

Fig 4: Comparison Chart for Cloudlet Size vs Make span

From this figure it is clear that the Make Span against cloudlet size is better in case of proposed approach as compared to existing one. The value of proposed scenario in case of cloudlet size vs make span is more than 200 where as in case of existing scenario it is only 8000. From these results it may be justified that the make span is better in case of proposed scenario as compared to the existing one.

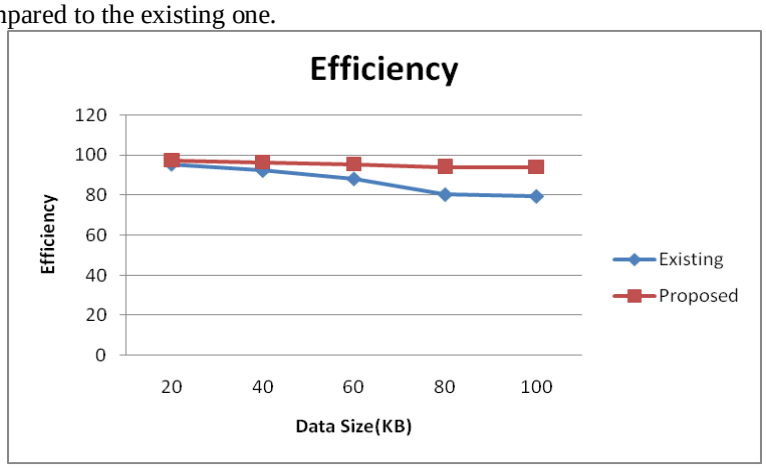

Fig 5: Efficiency

Efficiency: This may be defined as the ratio of output to the input. Efficiency of a system may define the performance of it. From the graph it is clear that the efficiency in proposed system is more than that of existing system.

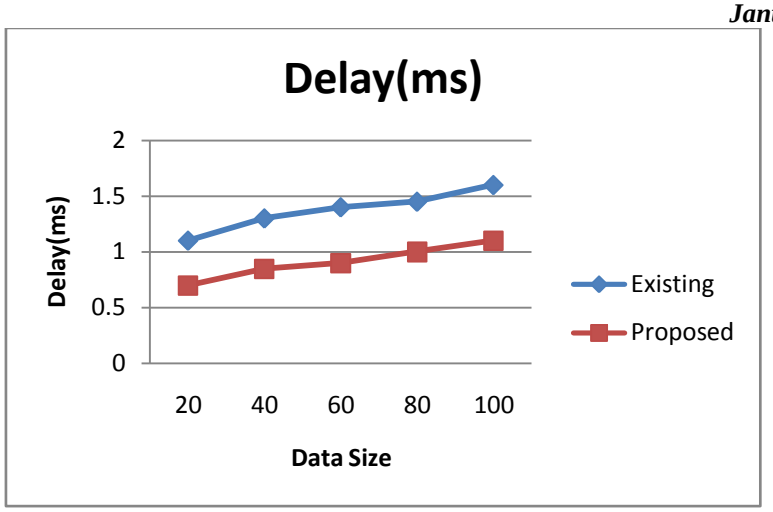

Fig 6: Delay

Delay: This may be defined as the time taken for a task to be completed. From the graph it is clear that the delay in proposed system is less than that of existing system.

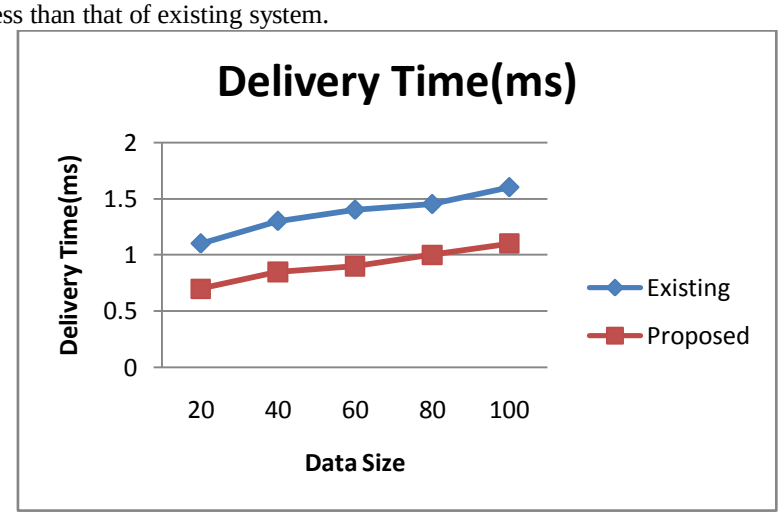

Fig 7: Delivery Time (ms)

Delivery Time(ms): This may be defined as the time taken for a task to be delivered. From the graph it is clear that the delay in proposed system is less than that of existing system.

Table 1: Comparative Study for existing and proposed approach

Technique / Parameters	Existing	Proposed
Cloudlet Size vs Make span(ms)	185	160
Cloudlet Id vs Priority/length	5000	5500
Efficiency(%)	85	93
Delay(ms)	10.8	9.5
Delivery Time(ms)	8.9	7.8

Table 1 is a comparative study for the proposed and existing approach. From the table it is clear that the cloudlet size vs Make span is lower in case of proposed system as compared to existing one and cloudlet Id vs priority/length is more in case of proposed system as compared to existing one. The other parameters which are delay, efficiency and delivery time are also better in case of proposed system.

VIII. CONCLUSION

This implementation aims towards the establishment of performance qualitative analysis on make span in VM task allocation and process according to their deadline, then implemented in CloudSim with Java language. Here major stress is given on the study of dead line based task scheduling algorithm with heterogeneous resources of the cloud, followed by comparative survey of other algorithms in cloud computing with respect to scalability, homogeneity or heterogeneity and process scheduling.

© 2017, IJARCSSE All Rights Reserved

A previous study also indicates change of MIPS will affect the response time and increase in MIPS versus VM decreases the response time. When image size of VM is implemented against the VM bandwidth then no significant effect is found on response time and it remains constant for which these parameters are investigated. But in case of Cloudlet long length versus Host bandwidth a pattern is observed in which response time increases in proportionate manner.

Using the modified approach the reduction in the down time of the various processes are achieved as shown in results. From the results it is clear that the proposed system used the task deadline as input parameter to improve results.

REFERENCES

[1] A. P. Deshmukh and K. Pamu "Applying Load Balancing: A Dynamic Approach" International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), vol. 2, no 6, June 2012.

[2] B. Alam, M.N. Doja, R. Biswas, "Finding Time Quantum of Round Robin CPU Scheduling Algorithm Using Fuzzy Logic" ICCEE 2008, International Conference, December 22, 2008.

[3] A. Gulati, R. K. Chopra "Dynamic Round Robin for Load Balancing in a Cloud Computing" International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 2, no. 6, June 2013.

[4] B. Yagoubi, Y. Slimani, "Dynamic Load Balancing Strategy for Grid Computing", World Academy of Science, Engineering and Technology 19, 2006.

[5] CloudSim: A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services, The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne, (2011) available from: http://www.cloudbus.org/cloudsim/.

[6] F. Liu, J. Tong, J. Mao, K. Consulting Inc. "NIST Cloud Computing Reference Architecture", version 1, March 30, 2011.

[7] H. Hu , P. Yi , Y. Guo, H. Li, "A Fair Service and Dynamic Round Robin scheduling scheme for CICQ switches" Telecommunications, ICT 2008, International Conference, 16-19 June 2008.

[8] IBM Cloud computing, http://www-07.ibm/in/cloud-computing/.html.

[9] Java software version 1.7 downloaded from: http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html, September 2012.

[10] J. Kaur "Comparison of Load balancing algorithms in a Cloud" International Journal of Engineering Research and Applications (IJERA), vol. 2, no 3, May-June 2012.

[11] D. Jadav, A. N. Choudhary, P. B. Berra, "Techniques for increasing the stream capacity of a high-performance multimedia server" Knowledge and Data Engineering, IEEE Transactions, vol. 11, issue: 2 Apr 1999.

[12] J. Yao, J. Guo, L. N. Bhuyan, "Ordered Round-Robin: An Efficient Sequence Preserving Packet Scheduler" IEEE transactions, vol. 57, no 12, 30 May, 2008.

[13] J. Hu, J. Gu, G. Sun, T. Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment" Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium, 20 Dec. 2010.

[14] M. Sharma, P. Sharma, S. Sharma, "Efficient Load Balancing Algorithm in VM Cloud Environment" International Journal of Computer Science and Technology (IJCST), vol. 3, no 1, March 2012.

[15] Milan E. Soklic "Simulation of Load balancing algorithms" ACM - SIGCSE Bulletin, December, 2002.

[16] H. R. Prakash, M.R. Anala, G Shobha, "Performance Analysis of Transport Protocol During Live Migration of Virtual Machines", IJCSE, Vol.2 5th ed Nov 2011.

[17] S. Rangarajan, J.J. Garcia-Luna-Aceves, "Load-Balanced Routing in Ad hoc Networks" Computer Communications and Networks, 2007. ICCCN 2007, Proc. of Sixteenth International Conference, 16 Aug. 2007.

© 2017, IJARCSSE All Rights Reserved