Improving Performance of Job Scheduling Algorithms in Cloud Computing Environment

Siddhardha Tagirisa

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

May 2015

Improving Performance of Job Scheduling Algorithms in Cloud Computing Environment

Thesis submitted in partial fulfilment
of the requirements for the degree of

Bachelor of Technology

In

Computer Science and Engineering

By

Siddhardha Tagirisa

(111cs0427**)**

Under the supervision of

Prof. Pabitra Mohan Khillar

NIT Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Rourkela-769 008, Orissa, India
May 2015

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769008, Odisha, India.

Certificate

This is to ensure that the work in the proposition entitled Improving Performance Of Job Scheduling Algorithms In Cloud Computing Environment by Siddhardha Tagirisa (111cs0427), is a record of a unique examination work did by him under my watch and direction in complete satisfaction of the necessities for the grant of the level of Bachelor of Technology in the branch of Computer Science and Engineering, National Institute of Technology Rourkela. Neither this theory nor any piece of it has been submitted for any degree or scholastic grant somewhere else.

(Professor. **Pabitra Mohan Khillar**)

Professor,CSEDepartment NITRourkela,Odisha

Acknowledgement

"I want to express my most profound and sincere appreciation to my task guide, Prof. P.M.Khilar for having confidence in my capacity. His significant bits of knowledge have helped me complete my project work. The adaptability of work and significant proposals that he has offered me has profoundly empowered me and helped me in the fruitful finish of my venture. I'm paying off debtors to every one of our teachers, classmates and companions at National Institute of Technology Rourkela for their participation and backing too. My full dedication to the work would have not been conceivable without their gifts and moral support."

Siddhardha Tagirisa

Authors' Declaration

I therefore proclaim that all the work contained in this report is my own work unless generally recognized. Likewise, the majority of my work has not been beforehand submitted for any scholarly degree. All wellsprings of cited data have been recognized by method for fitting references.

Siddhardha Tagirisa
National Institute of Technology, Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India
May 2015

Abstract

Scheduling in cloud computing is a system which is utilized to enhance general execution time of the job. A good scheduling algorithm can help in load balancing as well. Scheduling in cloud could be possible in three territories i.e. scheduling jobs within the virtual machine, Scheduling Virtual Machine on the host, and scheduling jobs to the Virtual Machine.

The last scheduling i.e. scheduling the cloudlets to the virtual machine is implemented on both static and dynamic environments in this thesis. In this the jobs are intelligently scheduled to the best possible virtual machine so that the overall execution time can be reduced. The outcome proves that the proposed algorithm gives better result in comparison to the already used sequential algorithm.

Keywords: Cloud Computing, Scheduling, Load Balancing, Execution Time Improvement, Algorithm

List of Figures& Tables:

Contents

Chapter 1

Introduction

Chapter 1

Introduction

1.1 Overview

Cloud computing is the conveyance of figuring administrations over the Internet. Cloud administrations permit people and organizations to utilize programming and equipment that are overseen by outsiders at remote areas. Illustrations of cloud administrations incorporate online file storage, social networking sites, web-mail, and online business applications. The cloud computing model permits access to data and PC assets from anyplace that a system association is accessible. Cloud computing gives an imparted pool of assets, including information storage room, systems, PC processing power, and particular corporate and client applications.

This administration can tackle numerous issues, for example:-

- Organizations require not to purchase the entire arrangement of servers. They can simply have administrations from the administration suppliers. It’s easy to handle as many responsibilities are given to cloud provider.
- It's anything but difficult to handle the same number of obligations are given to cloud supplier.
- It serves best for the flexible applications (Applications which scale very fast for example - Facebook)

Scheduling the jobs in cloud is an enormous test these days. As though jobs i.e. Cloudlets [1] are booked legitimately then the general execution time can be decreased essentially. Yet, Scheduling is a NP Hard issue so it is vital to apply a few heuristics to get best result commonly.

Cloud computing is the fate of innovation said by different CEO of IT industry. It deals with the rule that the client must pay agreeing the time for which it is utilizing the assets from different cloud suppliers. It is no doubt understood that it is not simple to handle extensive assets. It is excessive regarding cash too. So Cloud Computing is the best response to these testing issues.

Cloud computing is answer to each issue confronted by numerous Industries which are specifically or in a roundabout way identified with the Information Technology. This field is becoming unfathomably the same number of the huge players in the Information Technology field like Microsoft, Google, Amazon, SAP are contributing a ton of cash to get enhanced results. They feel that it is the eventual fate of innovation.

1.2 Cloud Computing Architecture

The Cloud Computing structural scheduling takes a shot at a layered methodology. It has a few on-screen characters like Cloud Consumer, Cloud Provider, Cloud Auditor, Cloud Broker, and Cloud Carrier. The mix of every one of these performing artists makes the building design of Cloud Computing [3].It is an exceptionally boundless region that is the reason it is not straightforward how everything is going on but rather this reference construction modelling gives an extraordinary review.

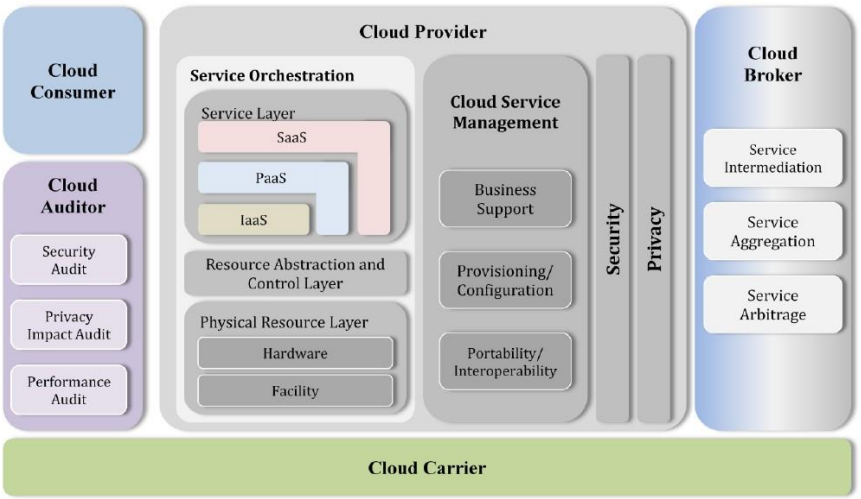


Figure1.1: Cloud Computing Architecture (adapted from [1])

Every on-screen character assume a huge part in the cloud computing working model. They need to have influence to get the job done and fulfil the client [1].

The diverse cloud buyers are characterized underneath:-

Cloud Consumer

Cloud Consumers are the performers which expends the administrations gave by the cloud supplier. They can specifically profit administrations from the suppliers or they can contact the cloud merchant to get the administrations. They will pay as per the time for which they have the administrations from the supplier [4]. They can have distinctive administrations like SaaS, Paas, Iaas.

Cloud Provider

Cloud Provider are the performing artists which gives the administrations to the purchasers. The administrations can be SaaS, PaaS, IaaS. The huge players like google, amazon and so forth have the capacity to give Infrastructure as an administration. Else all other little players are just dealing with giving SaaS or to the some degree PaaS. The Cloud Provider likewise charges the cash as per the time for which the administrations are benefitted by the buyers.

In SaaS administration cloud gives the customers to host the administration i.e the product at their end (on their assets). This helps the purchasers for not to purchase the assets to run the product. All the security perspectives are additionally taken care of by the suppliers itself.

In PaaS the supplier gives a toolbox to the customers like distinctive IDE or SDK to order their product and host them on the supplier's assets [5]. This helps the purchaser to have a reasonable control over the assets gave by the Cloud Provider.

In IaaS the shopper has the aggregate control over the assets gave by the Cloud Provider [6]. The low level programming could be possible in IaaS. A percentage of the issues like reaction time, memory use and Disk Access can undoubtedly be tended to utilizing IaaS as an administration.

Cloud Auditor

Cloud Auditor is the performer which deals with all the evaluating and bookkeeping parts of the entire cloud computing situation [7]. In the event that any kind of contention happen as far as valuing then it is determined by Cloud Auditor [8].

Cloud Broker

Cloud Broker is the performer which acts in the middle of the Cloud Consumer and Cloud Provider [9]. It helps the clients to pick the proficient and the best cloud supplier. It assumes an extremely basic part in arranging the arrangement between the supplier and buyer. In the event that any sort of contention happens then that contention is determined by the Broker [10].

Cloud Broker gives benefits in the accompanying way:-

- Service Inter-mediation It goes about as a middle in giving administrations from supplier to the customer.
- Service Aggregation it consolidates the administrations gave by distinctive Cloud Providers and afterward conveys those administrations to the Cloud Consumers.
- Service Arbitrage it improves the administrations gave by the cloud supplier and afterward give that as a support of the distinctive Cloud Consumers [11].

Cloud Carrier

Cloud Carrier is simply a medium through which the cloud administrations are given. It is essentially the association supplier for instance airtel, MTNL, BSNL, Vodafone and so forth [12]

1.3 Division of Thesis

Chapter 1 gives the introduction of the theory and brief introduction about the project. Chapter 2 states the Review of related work &Motivation. Chapter 3 briefs the proposed work. Chapter 4 shows the results & implementation. Chapter 5 gives the conclusion.

Chapter 2

Literature Review

Review of related work

Motivation

Chapter 2
Literature Review

Cloud computing is still another zone of exploration so very little has been done in this field as of not long ago. Yet at the same time I have checked on a percentage of the exploration papers and books to increase some foundation and enhance the officially created Scheduling algorithms till now. This field is developing quickly in the later past so the measure of examination going on is concentrated.

2.1 Review of related work

The analysts of this paper did a study on the effectively created and characterized load balancing algorithms [23]. They gave an itemized diagram of the effectively characterized load balancing algorithms. As we realize that load balancing and scheduling are by one means or another related in the field of cloud computing consequently I am characterizing this paper as a piece of my postulation in the Literature Overview segment. As indicated by the creators of this paper the load balancing algorithms are of two sorts' i.e. static load balancing algorithms and dynamic load balancing algorithms. The static load balancing algorithms are those which don't alter or change states at runtime yet dynamic load balancing algorithms unquestionably do that at run time. As we all realize that the in Cloud Environment new assets continue including and the general structure of the cloud continue changing hence static load balancing algorithms don't assume an enormous part. To adapt up to the changing cloud environment the dynamic load balancing algorithms assume a huge part. This paper additionally characterized the difficulties behind the load balancing in cloud environment.

A percentage of the fundamental reasons are:-

Spatial Distribution of the Cloud Nodes As we realize that cloud is a dispersed framework subsequently the cloud hubs are spatially appropriated over the topographical areas along these lines it turns into an incredible test for the creators of load balancing algorithms to outline the algorithms remembering the land area of the hubs. The Fault Tolerance must be considered.

Capacity/ Replication to enhance the adaptation to internal failure of the framework information replication is the route we to go. It is not simple to oversee such an immense measure of information. This tremendous measure of information putting away is not simple and getting it additionally exceptionally troublesome. Thusly, it is essential to deal with this immense measure of information legitimately to give satisfactory measure of administration.

Algorithm Complexity If the multifaceted nature of the algorithm is more than the general turnaround time will most likely increment. Subsequently it is critical for the load balancing algorithm planners to do a fitting exchange off between the algorithm many-sided quality and the general unpredictability of the application.

Point of Failure If Cloud is utilizing the brought together approach then one purpose of disappointment turns into an enormous test subsequently the answer for this issue is that the structural scheduling of cloud must get to be decentralized.

In this paper the researchers presented another kind of scheduling algorithm which is known as Meta Scheduling algorithm [24]. In this paper they isolated the jobs into four unique classes i.e. short tight, short wide, long thin and long wide. With this division they set the need more to the shorter jobs in such a path, to the point that they don't need to trade off with the longest end time of the bigger job.

In the event that we see in an alternate manner this takes a shot at the standard of slag time. They will slag the longest job to the greatest degree and afterward they will postpone its transforming as needs be. It is critical that starvation does not and that is the primary objective of the algorithm thusly they don't defer the job more than there slag time.

This exploration work the scientists studied the effectively created and utilized scheduling algorithms [24]. As we realize that cloud computing uses the effectively characterized and utilized conventions and algorithms yet as a part of a totally diverse way which surely an alternate construction modelling hence a hefty portion of these Scheduling algorithms which are utilized here are as of now being utilized as a part of operation framework.

Scheduling Algorithm	Scheduling Method	Scheduling Parameter	Scheduling Factor	Findings	Environment
Resource-Aware-Scheduling algorithm (RASA)	Batch Mode	Make Span	Grouped task	1. It is used to reduce makespan	Grid environment
RSDC (RELIABLE SCHEDULING DISTRIBUTED IN CLOUD COMPUTING)	Batch Mode	processing time	Grouped task	1. It is used to reduce processing time. 2. It is efficient for load balancing.	Cloud environment
An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing	Batch Mode	Quality of Service, Service request time	An array of workflow instances	1. High QoS 2.High throughput	Cloud environment

2.1SchedulingAlgorithmComparison

The above figure demonstrates the distinction among the few Scheduling algorithms. It recognizes them on the premise on Scheduling algorithm, scheduling system, Scheduling parameter, scheduling component, discoveries and nature. As we realize that cloud computing uses the effectively characterized and utilized conventions and algorithms yet as a part of a totally diverse way which in fact an alternate structural scheduling subsequently large portions of these Scheduling algorithms which are utilized here are now being utilized as a part of operation framework. As indicated by the creators of this paper the Scheduling algorithms are of two sorts' i.e. static scheduling algorithm and dynamic Scheduling algorithm. The static scheduling algorithms are those which don't alter or change states at run-time yet dynamic Scheduling algorithms doubtlessly do that at run time. As we all realize that the in Cloud Environment new assets continue including and the general structure of the cloud continue changing along these lines static load balancing algorithms don't assume an enormous part. To adapt up to the changing cloud environment the static scheduling algorithms assume a huge part. On the off chance that we nearly perceive the above characterized Scheduling we ran over a conclusion that each algorithm performs better in a predefined situation.

The researchers of RSDC Paper did bunches of work and enhanced PPDD (Processor-set Partitioning and Data Distribution Algorithm) algorithm which focuses on scheduling in VMs [17].

Scheduling parameter of this algorithm is Processing Time.

- It is utilized to lessen preparing time.
- It is proficient for load balancing.

2.2 Motivation

- The job Scheduling issue in cloud computing situations is of more significance because of its prerequisite in different applications.
- Efficient Scheduling algorithms as far as turnaround time are vital.
- The existing algorithms over virtual machines are widely available in the literature
- However, most of the algorithms are static in nature.
- Dynamic scheduling of jobs are necessary particularly for real time system environment where the jobs need to be schedules as soon as they arrive to a data center.
- Criteria needs to be distinguished with a specific end goal to calendar the jobs proficiently.

2.3 Objective

Propelled by the prerequisite of enhanced job Scheduling algorithm in cloud environment, the accompanying targets were attempted:

- To enhance the turnaround time in cloud computing scheduling algorithms
- To compare with existing algorithms.
- To implement the proposed algorithm IRSDC using language C++ and standard simulator CLOUDSIM.

Chapter 3

Proposed Work

Chapter 3

3.1 Proposed Work

Scheduling in cloud computing is a procedure which is utilized to enhance the general execution time (make compass) of the jobs. A decent Scheduling algorithm can help in burden adjusting also. Scheduling in cloud could be possible in three territories i.e. Scheduling jobs inside the virtual machine, Scheduling Virtual Machine on the host, and Scheduling jobs to the Virtual Machine. The last scheduling i.e. scheduling the jobs to the virtual machine is added to RSDC which takes after scheduling jobs to the virtual machine and actualized.

3.2 Assumptions

The Assumptions behind my proposed procedures are:

Static Implementation:

- The cloud is a private cloud having less number of VMs.
- At runtime no more cloudlets too assets are included i.e., a static domain.
- The jobs can be either scheduled prior to VMs or to the VMs.

Dynamic Implementation

- There is no security issues and shortcomings.

3.3 Problem Domain

The problem is to schedule or map the cloudlets to the Virtual Machine so that the overall turnaround can be slightest.

- Let there be set of jobs C=t1, t2, t3 …
- Let there be an arrangement of Virtual Machines V=v1, v2, v3 …
- It is assumed that the environment is static and all the jobs are autonomous.
- Output that which Cloudlet must be executed to which VM to have the least Turnaround time.

3.4 ARCHITECTURE

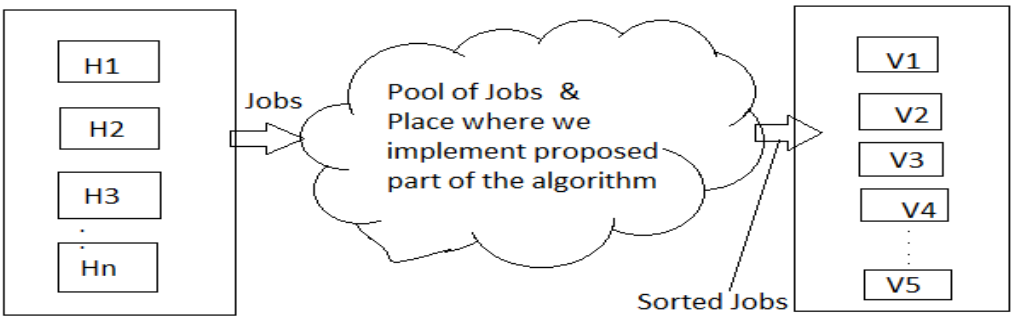


Figure 3.1 Architecture of implemented VMs

The service model is PaaS.

H1, H2, ...,Hn. Are hosts and V1, V2…Vn are virtual Machines.

3.5 Proposed Algorithm

- Let there be a set of Jobs
- Let there be a set of VM's.
- Scheduling Process (elaborated under this section).
- Output will be the set of pairs of virtual machine's and jobs.
- Execution of jobs in VMs.
- Turnaround time algorithm.

3.6 Scheduling Process & Execution

The job scheduling algorithm is given below.

1. Take Job Length **L** & Processing speeds **S** & Storage Capacity of VMs **C** &deadlines of job **D**.
2. Arrange jobs according to the length.
3. Rank Jobs according to deadline & length.
4. Higher the deadline higher the rank.
5. Lesser the length higher the rank.
6. Assign higher rank to VM having high processing speed.

7. Check whether the job will fit into VM's storage capacity.
8. If not assign it to the VM with storage capacity >= job length with speed next to the first VM we have choose to assign.
9. Dead line is more preferable than length.
10. After assigning jobs to VM's we will follow RSDC and will calculate the turnaround time for analysing the algorithms.

RSDC Flowchart [2]

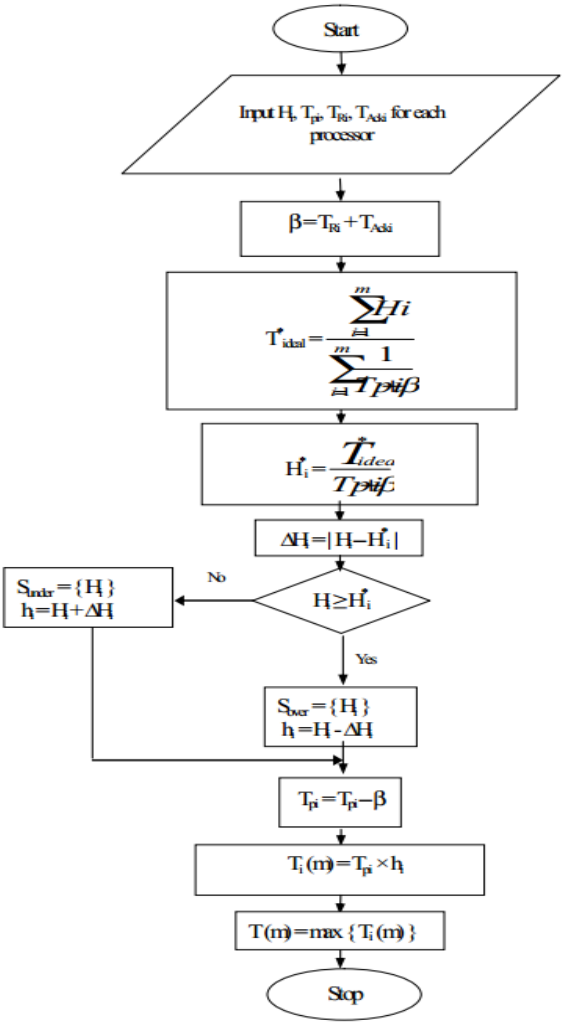


Figure 2. RSDC Flowchart

Figure 3.2 RSDC Flowchart

m : number of processors
H_i : initial loads for each processor
TP_i : processing time for each processor
TR_i : request time for each processor
T_{Acki} : acknowledge time for each processor
H^*_i : average processing for each processor
$\Delta Hi : H_i - H^*_i$ the deviation from the average
S_{under} : the set of under load processors
S_{over} : the set of overload processors
h_i : the amount of balanced loads for each processor
T_i (m): the balanced processing time for each processor
T (m): the final time of processing the processors

3.7 Generic Terms

The generic terms which are used in this work as follows:

- Set of VM's& number of VMs
- Set of Jobs: set of cloud lets in the private cloud
- Waiting Time: time spent in the queue i.e., the difference between arrival time and the start time of job
- Job length: Number of instructions
- Power of VM: the storage capacity of VM
- Dead line: time before which jobs must finish their execution
- Turnaround Time: The time between the end of execution & arrival time.
- processing time for each VM
- request time for each process
- acknowledge time for each processor
- The final time of processing the Job.

3.8 Techniques for Performance Evaluation

For Static Environment

- Implementation using C++ language under Linux platform
- VM Environment.
- Analysis of Algorithms

For Dynamic Environment

- Implementation using CLOUDSIM

Chapter 4

Implementation & Simulation Results

Chapter 4

4.1 Implementation in Static Part

```
    while(changed==true){
        changed = false;
        for(int i=0;i<no_jobs-1;i++){
            if(job[i].deadline > job[i+1].deadline){
                Job temp = job[i+1];
                job[i+1] = job[i];
                job[i] = temp;
                changed = true;
            }
        }
    }
    while(changed==true){
        changed = false;
        for(int i=0;i<no_vms-1;i++){
            if(vm[i].speed > vm[i+1].speed){
                VM temp = vm[i+1];
                vm[i+1] = vm[i];
                vm[i] = temp;
                changed = true;
            }
        }
    }
    for(int i=0;i<no_jobs;i++){
        for(int j=0;j<no_vms;j++){
            if(job[i].length < vm[j].storage){
                cout << "Job " << job[i].number << " is assigned to VM " << j+1 << endl;
                break;
            }
        }
    }
        cout << "Input Taken" << endl;
        cout << "Number of Job" << "\t" << "Length of job" << "\t" << "Deadline of Job" << endl;
    for(int i=0;i<no_jobs;i++){
        cout << job[i].number << "\t" << job[i].length << "\t" << job[i].deadline << endl;
    }
}
```

Results and Analysis

```
Number of Job   Length of job   Deadline of Job
1       5       4
3       99      5
5       105     6
2       2       1000
4       101     10000
```

```
Job 1 is assigned to VM 1
Job 3 is assigned to VM 1
Job 5 is assigned to VM 2
Job 2 is assigned to VM 1
Job 4 is assigned to VM 2
```

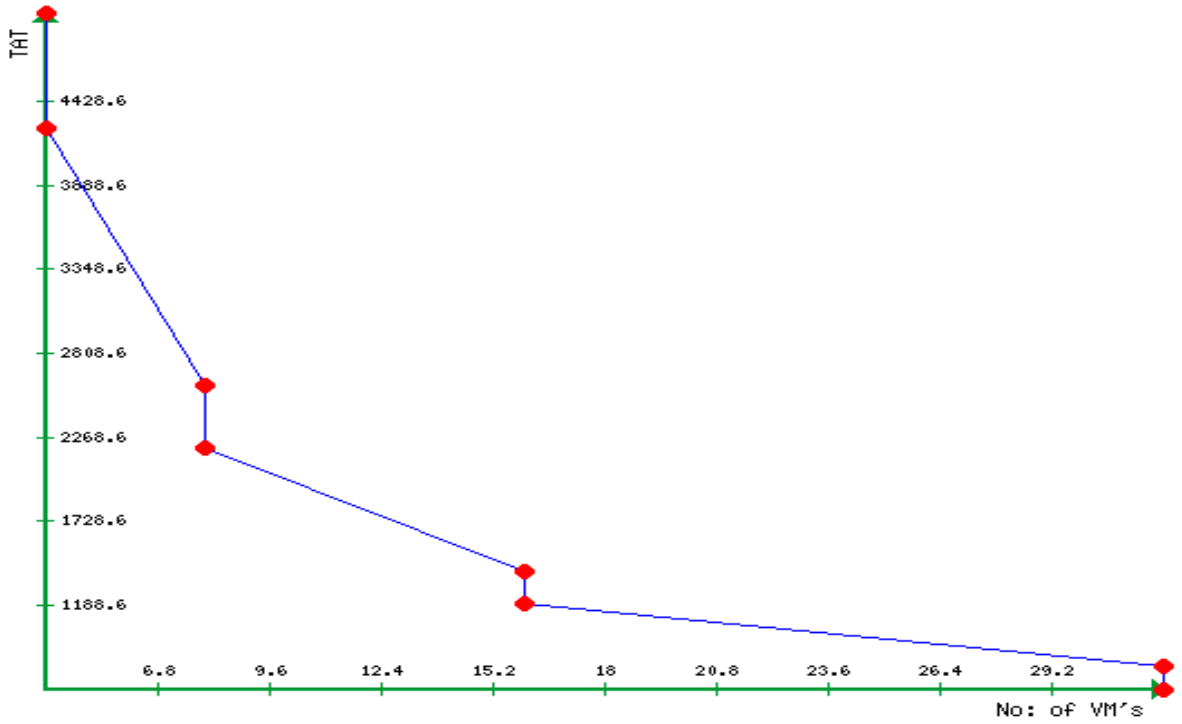
```
Turn around time(exisitng): 5000.1 for 4 VM's
Turn around time(proposed): 4250.09 for 4 VM's
Turn around time(exisitng): 2600.09 for 8 VM's
Turn around time(proposed): 2200.1 for 8 VM's
Turn around time(exisitng): 1400.1 for 16 VM's
Turn around time(proposed): 1200 for 16 VM's
Turn around time(exisitng): 800.1 for 32 VM's
Turn around time(proposed): 648.6 for 32 VM's
Turn around time(proposed): 1200.1 for 50 jobs
Turn around time(existing): 1400 for 50 jobs
Turn around time(existing): 3800.36 for 150 jobs
Turn around time(proposed): 3250 for 150 jobs
```

Summary of Results in Static Environment.

1. No. of jobs taken here is 100

No. of VM's	Existing (TAT)	Proposed
4	5000.1	4250.09
8	2600.09	2200.1
16	1400.1	1200.1
32	800.1	648.6

Graphical Representation

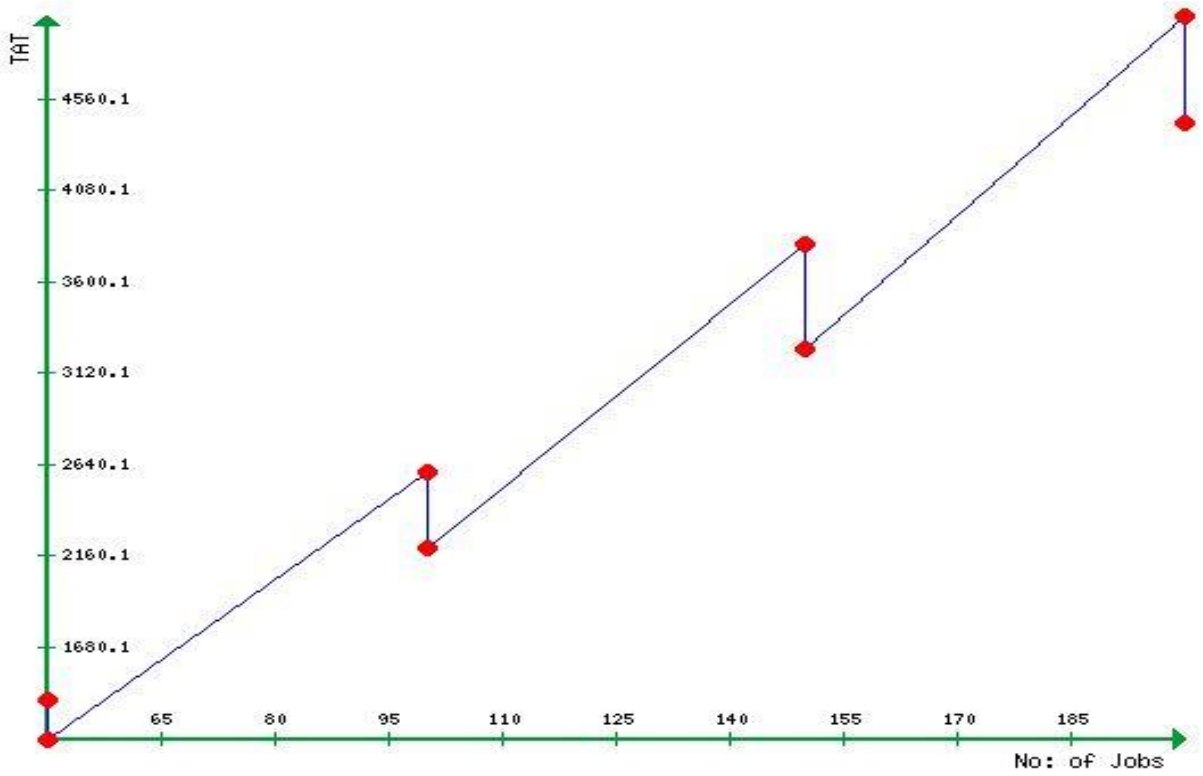


The above algorithm gives less turnaround time of each cloudlet. By utilizing this algorithm we can lessen the execution time of jobs. Else job coordinating is a NP hard issue however by utilizing these heuristics we can coordinate in a polynomial many-sided quality.

2. No. of VM's taken here is 8

No. of jobs	Existing(TAT)	Proposed
50	1400.1	1200
100	2600.09	2200.1
150	3800.36	3250
200	5000.1	4438

Graphical Representation



The result shows that the event that differs the quantity of jobs and keep the quantity of virtual machine consistent all things considered additionally our algorithm beat's the existing algorithm.

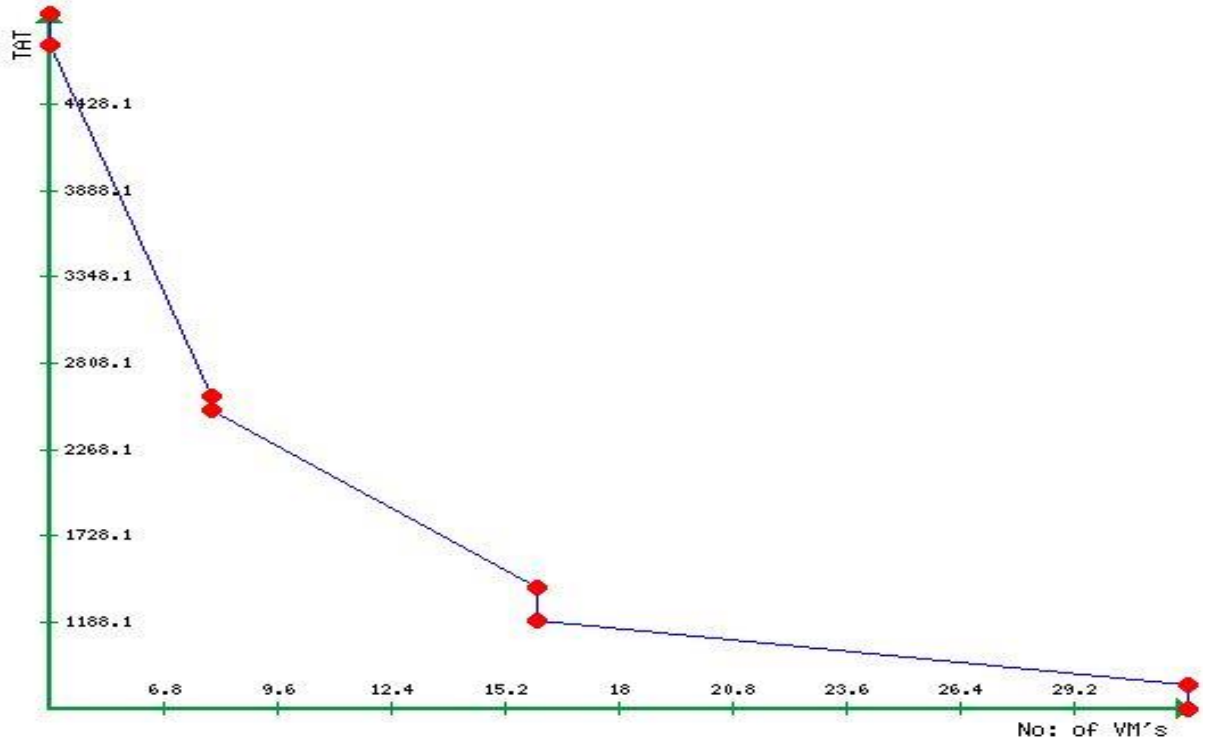
4.2 Implementation in Dynamic Environment

Summary of results in Dynamic Environment.

1. No. of jobs taken here is 100

No. of VM's	Existing(TAT)	Proposed
4	5000.1	4800.1
8	2600.1	2512.1
16	1400.1	1200.1
32	800.1	648.1

Graphical Representation

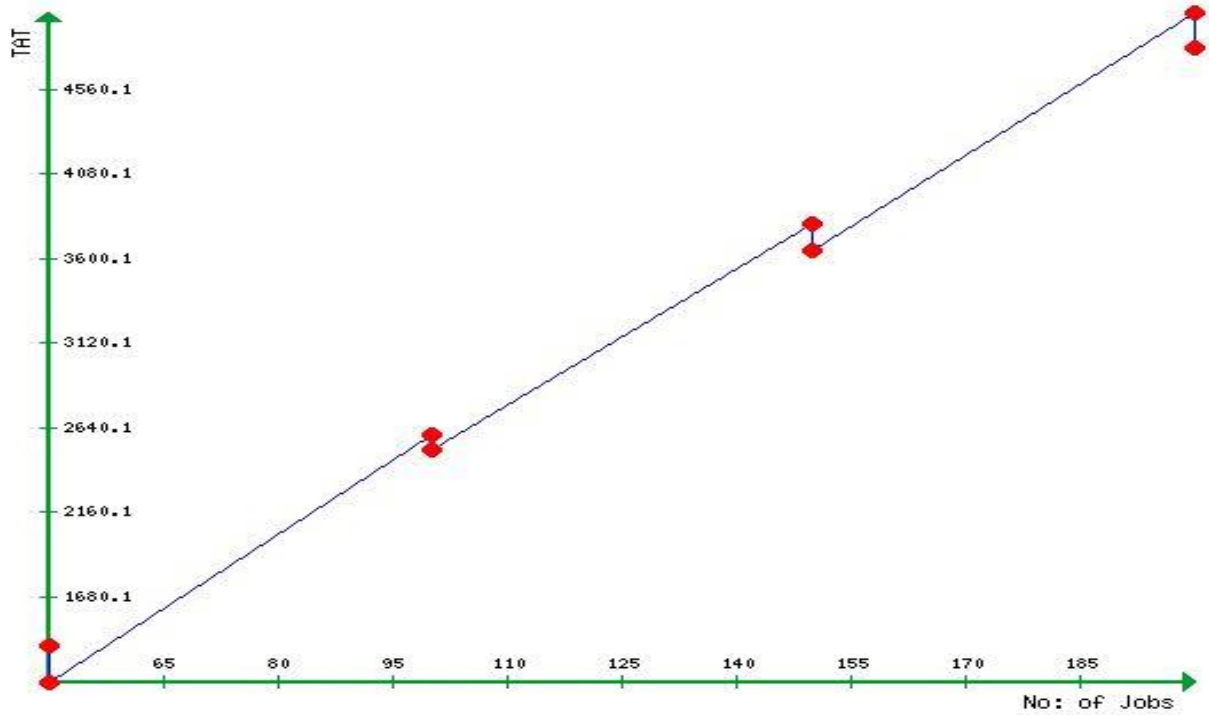


The above algorithm gives the less turnaround time of each cloudlet. By utilizing this algorithm we can decrease the execution time of jobs. Otherwise job coordinating is a NP hard issue yet by utilizing these heuristics we can coordinate in a polynomial multifaceted nature.

2. No. of VM's taken here is 8

No. of jobs	Existing (TAT)	Proposed
50	1400.1	1200.1
100	2600.1	2512.1
150	3800.1	3644.1
200	5000.1	4800.1

Graphical Representation



The result shows that the event that differs the quantity of jobs and keep the quantity of virtual machine consistent all things considered additionally our algorithm beat's the existing algorithm.

Chapter 5

Conclusion

CHAPTER 5

5.1 Conclusion

Most importantly portrayals, results make you feel that a decent scheduling algorithm can truly help to diminish the general execution time of the cloudlets which surely will help the cloud suppliers to draw in more clients. Our outcomes above unmistakably demonstrate that if cloudlets are offered brilliantly to the virtual machines than the general execution can be enhanced effectively.

5.2 Limitations & Future Scope

As this algorithm is tried for static and dynamic situations, this can't be utilized as a part of defective and security conditions. Therefore, it is vital to change the algorithm taking thought of broken and security conditions.

Bibliography

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al., "A view of cloud comput- ing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[2] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm- based cloudlets in mobile computing," Pervasive Computing, IEEE, vol. 8, no. 4, pp. 14–23, 2009.

[3] X. Li, X. Jiang, P. Huang, and K. Ye, "Dartcsim: An enhanced user-friendly cloud simulation system based on cloudsim with better performance," in Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd Interna- tional Conference on, vol. 1, pp. 392–396, IEEE, 2012.

[4] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on, pp. 5–13, Ieee, 2008.

[5] B. T OGRAPH and Y. R. MORGENS, "Cloud computing," Communications of the ACM, vol. 51, no. 7, 2008.

[6] T. Velte, A. Velte, and R. Elsenpeter, Cloud computing, a practical approach. McGraw-Hill, Inc., 2009.

[7] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Computer Security–ESORICS 2009, pp. 355–370, Springer, 2009

[8] M. Hogan, F. Liu, A. Sokol, and J. Tong, "Nist cloud computing standards roadmap," NIST Special Publication, vol. 35, 2011.

[9] S. G. Grivas, T. U. Kumar, and H. Wache, "Cloud broker: Bringing in- telligence into the cloud," in Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, pp. 544–545, IEEE, 2010.

[10] P. Pawluk, B. Simmons, M. Smit, M. Litoiu, and S. Mankovski, "Introducing stratos: A cloud broker service." in IEEE CLOUD, pp. 891–898, 2012.

[11] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, Y. Liu, A. Devarakonda, L. Fong, S. Masoud Sadjadi, and M. Parashar, "Cloud federation in a layered service model," Journal of Computer and System Sciences, vol. 78, no. 5, pp. 1330–1344, 2012.

[12] L. Carrier, G. Cato, and K. Von Essen, "The backscattering and extinction of visible and infrared radiation by selected major cloud models," Applied Optics, vol. 6, no. 7, pp. 1209–1216, 1967.

[13] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities," in High Performance Computing & Simulation, 2009. HPCS'09. International Conference on, pp. 1–11, IEEE, 2009.

[14] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: comparing public cloud providers," in Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, pp. 1–14, ACM, 2010. 23

[15] P. Hofmann and D. Woods, “Cloud computing: the limits of public clouds for business applications,” Internet Computing, IEEE, vol. 14, no. 6, pp. 90–93,2010.

[16] F. Doelitzscher, A. Sulistio, C. Reich, H. Kuijs, and D. Wolf, “Private cloud for collaboration and e-learning services: from iaas to saas,” Computing, vol. 91, no. 1, pp.23–42,2011.

[17] “RSDC (RELIABLE SCHEDULING CLOUD IN CLOUD COMPUTING)” Arash Ghorbannia Delavar,Mahdi Javanmard , Mehrdad Barzegar Shabestari and Marjan Khosravi Talebi “International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, No.3, June 2012”

[18] J. Qiu, J. Ekanayake, T. Gunarathne, J. Y. Choi, S.-H. Bae, H. Li, B. Zhang, T.-L. Wu, Y. Ruan, S. Ekanayake, et al., “Hybrid cloud and cluster com- puting paradigms for life science applications,” BMC bioinformatics, vol. 11, no. Suppl 12, p. S3, 2010.

[19] P. Salot, A survey of various scheduling algorithm in cloud computing environment," International Journal of research and engineering Technology (IJRET), ISSN, pp. 2319{1163, 2013.

[20] M. A Vouk, “Cloud computing–issues, research and implementations,” CIT.Journal of Computing and Information Technology, vol. 16, no. 4, pp. 235–246, 2008.

[21] T. Binz, G. Breiter, F. Leyman, and T. Spatzier, “Portable cloud services using Tosca.” IEEE Internet Computing, vol. 16, no. 3, 2012.

[22] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, “Cloudsim: a toolkit for modelling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, 2011.

[23] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi, and J. Al-Jaroodi, “A survey of load balancing in cloud computing: Challenges and algorithms,” in Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on, pp. 137–142, IEEE, 2012.

[24] S. Sadhasivam, N. Nagaveni, R. Jayarani, and R. V. Ram, “Design and implementation of an efficient two-level scheduler for cloud computing environment,” in Advances in Recent Technologies in Communication and Computing, 2009. ARTCom’09. International Conference on, pp. 884–886, IEEE, 2009.