IMPROVED SCHEDULING ALGORITHM FOR LOAD BALANCING IN CLOUD ENVIRONMENT

POOJA MANGLA[a1] AND SANDIP KUMAR GOYAL[b]

[ab]Department of CSE, Maharishi Markandeshwar University, Mullana, Ambala, Haryana, India

ABSTRACT

Cloud computing is the new technology which is totally based on virtualization. Load balancing is one of the most important issues which play a great role in scheduling of resources, and even in task management. In this research paper, we have developed a new scheduling technique by studying existing techniques.

KEYWORDS: Cloud Computing,

INTRODUCTION TO CLOUD COMPUTING

Cloud computing is a new technology which uses virtual machine instead of physical machine to host, store and network the different components. Cloud Computing refers to manipulating, configuring, and accessing the applications online. It offers online data storage, infrastructure and application. A Cloud is a kind of parallel and distributed system possessing a group of inter-connected and virtualized computers that are dynamically scheduled and highlighted as one or more unified computing resources based on service-level agreements established through conciliation between the service provider and consumers [1].

Cloud Computing is a way of managing large number of highly virtualized resources such that, from a management perspective, they resemble a single large resource. There is a greater need for IT to help address business challenges and cloud computing can help you do all of these:

Doing More With Less

Reduce capital expenditures and operational expenses.

Higher Quality Services

Improve quality of services and deliver new services that help the businesses to grow and reduce costs.

Reducing Risk

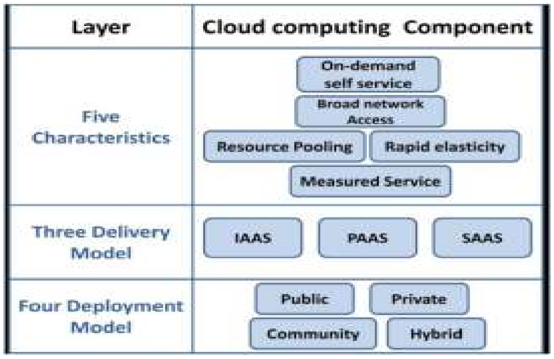
Ensure the right levels of security and resiliency across all businesses data and processes.

Breakthrough Agility

Increase ability to quickly deliver new services to capitalize on opportunities while containing costs and managing risks.

Cloud computing is the provision of dynamically scalable and often virtualized resources as a service over the Internet (public cloud) or intranet (private cloud).

MODEL OF CLOUD COMPUTING



Layer	Cloud computing Component
Five Characteristics	On-demand self service Broad network Access Resource Pooling Rapid elasticity Measured Service
Three Delivery Model	IAAS PAAS SAAS
Four Deployment Model	Public Private Community Hybrid

Figure 1.1: Model for Cloud Computing

LOAD BALANCING

Load balancing is a generic term used for distributing a larger processing load to smaller processing nodes for enhancing the overall performance of system. Load Balancing is done with the help of load balancers where each incoming request is redirected and is transparent to client who makes the request.

Load Balancing helps in improving the performance substantially, maintenance of system stability, cost effectiveness, etc. Load balancing is a process of reassigning the total load:

- to the individual nodes of the collective system
- to make resource utilization effective
- to improve the response time of the job
- simultaneously removing a condition in which some of the nodes are over loaded while some others are under loaded.

HEURISTIC LOAD BALANCING IN CLOUD COMPUTING

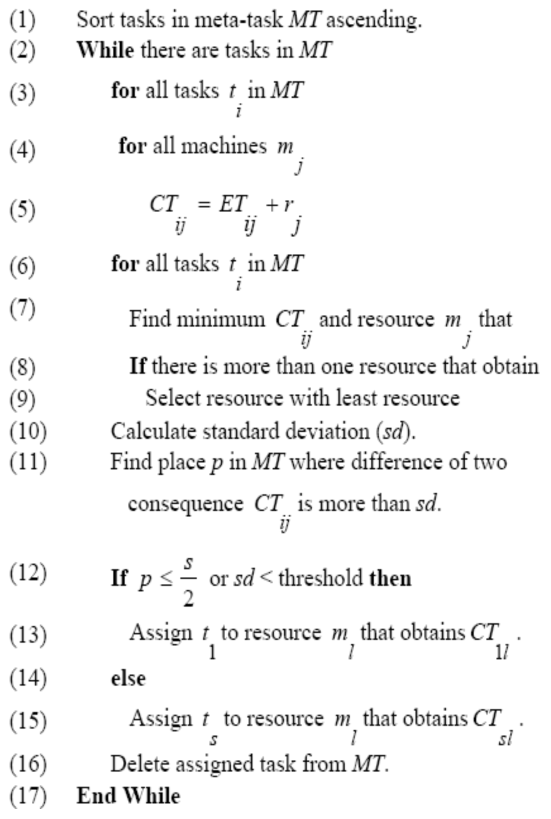
To better use tremendous capabilities of this large scale distributed system, effective and efficient

[1]Corresponding author

scheduling algorithms are needed. Many such algorithms have been designed and implemented. We introduce a new scheduling algorithm based on two conventional scheduling algorithms, Min-Min and Max-Min, to use their cons and at the same time, overcome their pros. This heuristic scheduling algorithm, called min-min min-max selective, is evaluated using a Cloud simulator called CloudSim by comparing to its performance against the two basic heuristics which it comes from.

Reviewing Min-Min and Max-Min heuristics, it can be seen that depending on the length of unassigned tasks in MT (meta-task), one of these heuristics has better results than the other one [7]. For example, if there is only one long task and too many short tasks, Max-Min will execute long task first and allows short tasks to be executed concurrently with the long task, resulting better makespan and even better resource utilization rate and load balancing level, compared to Min-Min that executes all short tasks first and then executes the long task.

BASIC STEPS FOR PROPOSED ALGORITHM



```
Sort tasks in meta-task $MT$ ascending.
While there are tasks in $MT$
    for all tasks $t_i$ in $MT$
        for all machines $m_j$
            $CT_{ij} = ET_{ij} + r_j$
    for all tasks $t_i$ in $MT$
        Find minimum $CT_{ij}$ and resource $m_j$ that
        If there is more than one resource that obtain
            Select resource with least resource
    Calculate standard deviation ($sd$).
    Find place $p$ in $MT$ where difference of two
        consequence $CT_{ij}$ is more than $sd$.
    If $p \le \frac{s}{2}$ or $sd <$ threshold then
        Assign $t_1$ to resource $m_l$ that obtains $CT_{1l}$ .
    else
        Assign $t_s$ to resource $m_l$ that obtains $CT_{sl}$ .
    Delete assigned task from $MT$.
End While
```

PERFORMANCE METRICS

- **Makespan**
- Makespan is a measure of the throughput of the heterogeneous computing systems, such as cloud. It can be calculated as the following relation:

makespan = max(CTi) , where ti belongs to MT

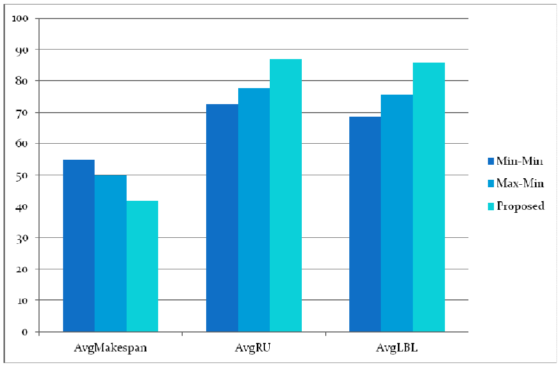
The less the makespan of a scheduling algorithm, the better it works.

- **Average resource utilization rate**
- It is one of the metrics that is used in [14]. Average resource utilization of each resource can be calculated

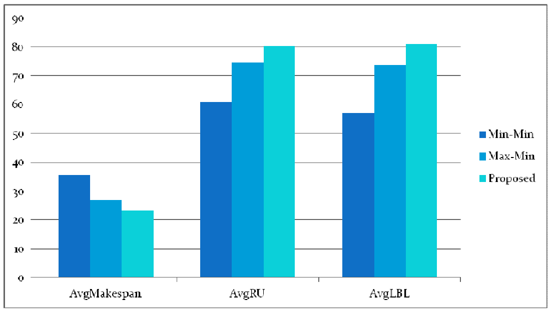
- **Load balancing level**
- The mean square deviation of ru is defined as : The best and most efficient load balancing level is achieved if d equals zero and β equals 1. So, scheduling algorithm will have better performance if d is close to 0 and β is close to 1. It is the other metric that is used in [14].

RESULTS AND DISCUSSION

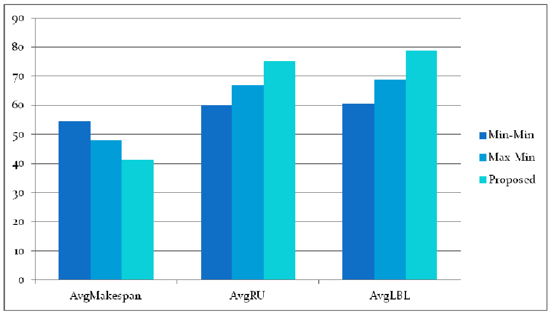
Scenario-1: (More Lenghty Tasks):



Scenario-2: (Less Lenghty Tasks)



Scenario-3: (Mixed Lengthy)



CONCLUSION

In this research work, a new model is developed and analyzed. In this method, three parameters i.e., makespan, average resource utilization and load balancing level, are calculated and the corresponding results are shown. It has been observed that the results generated from proposed model are better than the existing algorithms.

REFERENCES

Moharana S.S., Ramesh R.D. and Powar D., 2013. Analysis Of Load Balancers In Cloud Computing, International Journal Of Computer Science And Engineering, **2**(2):101-108.

Alakeel A.M., 2010. A Guide to Dynamic Load Balancing in Distributed Computer Systems‖, International Journal of Computer Science and Network Security, **10**(6).

Moraga R.J., DePuy G.W. and Whitehouse G.E., 2006. Metaheuristics: a solution methodology for optimization problems. Handb Ind Optim Probl Handb Ind Syst Eng AB Badiru. http://dx.doi. org/10.1201/9781420038347.

Pop F., Dobre C. and Cristea V., 2009. Genetic algorithm for DAG scheduling in grid environments. In: 5th IEEE int conf intell comput commun process, pp. 299–305.

Sharma T. and Banga V.K., 2013. Efficient and Enhanced Algorithm in Cloud Computing‖, International Journal of Soft Computing and Engineering, **3**(1).

Ahluwalia et. al., 2016. International Journal of Advanced Research in Computer Science and Software Engineering, **6**(6):340-347.

Rani A. and Garg K., 2016. “Multi-objective Differential Evolution for Task Scheduling in Cloud Computing”, International Journal of Innovative Research in Computer and Communication Engineering, **4**(5).

Mitchell M., 2012. An Introduction to Genetic Algorithms‖, International Journal of Communication Network Security ISSN: 2231 – 1882, **1**(4).

Sharma P. and Khurana N., 2013. “Study of Optimal Path Finding Techniques”, International Journal of Advancements in Technology, **4**(2).

Freund R.F., Gherrity M., Ambrosius S., Camp-bell M., Halderman M., Hensgen D., Keith E., Kidd T., Kussow M., Lima J. D., Mirabile F., Moore L., Rust B. and Siegel H.J., 1998. Scheduling resources in multi-user, heterogeneous, computing environments with Smart Net, 7th IEEE Heterogeneous Computing Workshop (HCW’98), pp. 184–199.

Casanova H., Legrand A., Zagorodnov D. and Berman F., 2000. Heuristics for Scheduling Parameter Sweep Applications in Grid Environments, In Proc. 9th heterogeneous Computing Workshop (HCW'00), Cancun, Mexico, pp. 349-363.

He X., Sun X. and Laszewski G., 2003. A QoS Guided Min-Min Heuristic for Grid Task Scheduling, Journal of Computer Science and Technology. Special Issue on Grid Computing, **18**(4):442-451.

Wu M., Shu W. and Zhang H., 2000. Segmented Min-Min: A Static Mapping Algorithm for Meta-Tasks on Heterogeneous Computing Systems, In Proc. 9th Heterogeneous Computing Workshop (HCW'00), Cancun, Mexico, pp. 375-385.

Cao J., Spooner D.P., Jarvis S.A. and Nudd G.R., 2005. Grid Load Balancing Using Intelligent Agents, Future Generation Computer Systems, **21**(1):135-149.

Buyya R. and Murshed M., 2002. GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computting, Journal of Concurrency and Computation: Practice and Experience, **14**(13-15):1175–1220.

Rafsanjani M.K. and Bardsiri A.K., 2016. A Hyper-Heuristic Method For Scheduling The Jobs In

Cloud Environment‖, Informatics Engineering, an International Journal, **4**(1).

Alkhanak E.N., Lee S.P. and Ling T.C., 2015. A Hyper-Heuristic Approach using a Prioritized Selection Strategy for Workflow Scheduling in Cloud Computing‖, The 4th International Conference on Computer Science and Computational Mathematics (ICCSCM 2015).

Pinjari A. and Mandre B.R., 2015. Cloud-Scheduling Algorithm using Hyper heuristic Approach: Study and Implementationl, International Journal of Computer Systems (ISSN: 2394-1065), **2**(6).

Velte A.T., Velte T.J. and Elsenpeter R., 2010. Cloud Computing: A Practical Approachl, TATA McGRAW-HILL Edition.