IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-6, Issue-3 E-ISSN: 2347-2693

Priority Based Least Waiting Time Load Balancing Algorithm Applied in Cloud Computing

A.B. Majumder [1], S. Sil [2], S. Das [3], A. Mondal [4]

[1] Dept. of Information Technology, JIS College of Engineering, Autonomous (Under WBUT), Kalyani, India
[2] Dept. of Information Technology, JIS College of Engineering, Autonomous (Under WBUT), Kalyani, India
[3] Dept. of Information Technology, JIS College of Engineering, Autonomous (Under WBUT), Kalyani, India
[4] Dept. of Information Technology, JIS College of Engineering, Autonomous (Under WBUT), Kalyani, India

Corresponding Author: annwesha.banerjee@gmail.com, Tel.: +91-90512-55006

Available online at: www.ijcseonline.org

Received: 20/Feb//2018, Revised: 26/Feb2018, Accepted: 19/Mar/2018, Published: 30/Mar/2018

Abstract— Cloud computing is a term, which involves virtualization, distributed computing, networking, software and web services. Cloud computing changes the paradigm of computing by providing computing as service. To provide scalable services to a user, load balancing a key requirement for customer satisfaction and proper work management. This paper proposes an effective load balancing algorithm using various parameters to distribute the load efficiently among various processors enabling better resource utilization and system response time. This proposed method assigns priority to the server and balance the load to servers according to propriety and also considering the waiting time. This proposed method almost guarantees the maximum throughput in minimum response time, and thus the user has to wait for minimum amount of time to get the job done.

***Keywords*—** Cloud Computing, Virtualization, distributed computing, load balancing

I. INTRODUCTION

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [1].

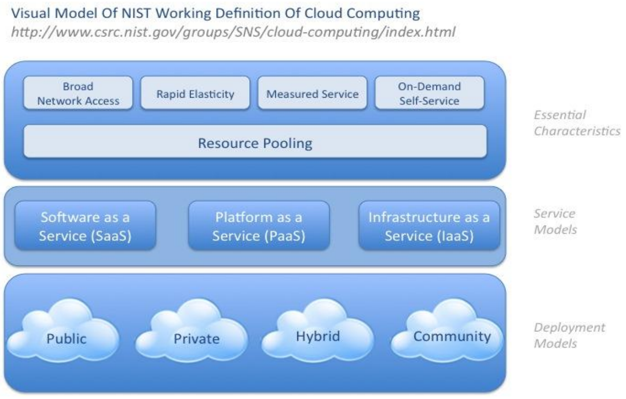

Visual Model Of NIST Working Definition Of Cloud Computing
http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html
Broad Network Access
Rapid Elasticity
Measured Service
On-Demand Self-Service
Resource Pooling
Essential Characteristics
Software as a Service (SaaS)
Platform as a Service (PaaS)
Infrastructure as a Service (IaaS)
Service Models
Public
Private
Hybrid
Community
Deployment Models

Fig: NIST Cloud Model.

Cloud load balancing is the process of distributing workloads and computing resources in a cloud computing environment. Load balancing allows enterprises to manage application or workload demands by allocating resources among multiple computers, networks or servers. Cloud load balancing involves hosting the distribution of workload traffic and demands that reside over the Internet.

Cloud load balancing helps enterprises achieve high performance levels for potentially lower costs than traditional on-premises load balancing technology. Cloud load balancing takes advantage of the cloud's scalability and agility to meet rerouted workload demands and to improve overall availability. In addition to workload and traffic distribution, cloud load balancing technology can provide health checks for cloud applications.

The important things to consider while developing such algorithm are: estimation of load, comparison of load, stability of different system, performance of system, interaction between the nodes, nature of work to be transferred, selecting of nodes and many other ones. This load considered can be in terms of CPU load, amount of memory used, delay or Network load. Load balancing improves the response time also. Load balancing performs two major tasks: resource allocation and scheduling. [2]. The proposed method has focused on the resource allocation to a certain server.

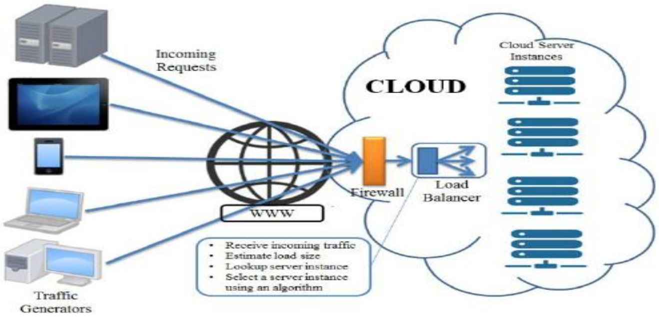


Fig 2: Cloud Load Balancing [3]

II. Related Work

There is plenty of established and well functioned load balancing algorithms. Load balancing algorithms are categorized as static, dynamic, random, centralized, and distributed. Round Robin is a random sampling based load balancing algorithm [4]. M. Nakai et al.has propsed a load balancing algorithm to choose closest server[5]. J. Hu et al. has applied Genetic Algorithm in load balancing[6]. Central Load Balancing Policy for Virtual Machines (CLBVM) is a load balancing algorithm that enhanced performance[7]. A two-level task scheduling mechanism that obtained high resource utilization has been proposed by Y. Fang et al.[8]. Honey Bee Load Balancing is a bio inspired load balancing algorithm proposed by Obaid Bin Hassan, A Sarfaraz Ahmad[9]. Throttled Load Balancing Algorithm is a load balancing algorithm in which client requests for virtual server[11]. Agarwal, M., Srivastava, D have also proposed a genetic algorithm based load balancing algorithm.[12]. Particle Swarm Optimization. Based load balancing algorithm has been proposed by Ramezani, F., Lu, j., Hussain, F[13] Another GA based cloud computing load balancing algorithm has proposed by Dasgupta, K. et al.[14]. Patel, R., Patel, S., Patel, D., Patel, T have proposed an advanced GA based load balancing algorithm.[15]. Task Scheduling based on multi Population Genetic Algorithm in cloud computing was proposed by Bei, W., Jun, L.[16]. Sandeep kaur and Sujhwinder Sharma have introduced a Enhanced Optimal scheduling Cost which reduces waiting and turn around time.[17]. Stochastic Hill climbing is used in cloud computing task scheduling propsed by Brototi Mondala et al.[18]. Neighbour awareness and prediction mechanisms has been applied for task scheduling in a proposed paper [19]. Dynamic well Organised load balancing algorithm for virtual machines is another cloud computing task scheduling algorithm.[20].

Ekta Rani and Harpreet Kaur have used Raven Roosting Optimization Algorithm (RRO) in cloud load balancing.[21]. Annwesha B. Majumder et al. have proposed a simple load balancing algorithm considering speed and load.[22].

III. Methodology

We started off with studying the various load balancing algorithm. One day we realized all are efficient enough to distribute the loads equally but neither of them ensured the task will be completed in least amount of time.

Hence we started to work on such an algorithm that ensures the task to be completed in least amount of time.

IV. PROPOSED METHOD

Our proposed method considers time and priority as key factors for designing the algorithm. First, we need to enroll different processor. Enrolment of the processor will include the processor speed. Each processor will have a field that will contain the amount of time it will take to complete its allocated task. To start with all value on the above said field will have 0(zero) since at the start neither processor is assigned with any task

After the processors are assigned, the user can enroll their task. After the tasks are enrolled we need to find out time to completion of the task by each and every processor. Say a particular task need to process ‘X’ amount of data.

So, time to complete by a processor with speed ‘S’ currently having ‘T’ time to complete current allocated process will be,

$$W1 = \frac{X}{S} \tag{1}$$

$$W2 = Priority * W1 \tag{2}$$

$$Total\ Time = T + W2 \tag{3}$$

Then we need to check which processor is taking the least amount of time to complete the task along with the current allocated task. The processor that is taking the least amount of time the process gets allocated to that processor.

1.1 Algorithm

```
START
Step 1: Initialize the number of processor in an array (say p[])
Step 2: Initialize the number of process needed to execute in an array (say q[])
Step 3: Input the processing speed of each processor and store in p[]
Step 4: Input the amount of data to be processed for each process and store in q[]
```

Step 5: Call the function that would execute the proposed algorithm (and pass both the arrays)

Step 6: Initialize 2 arrays (say cs&ap) of length equal to number of processors and no of processes (that would contain the current state of the processors and process allocate to processor)

Step 7: Initialize 2 variable (say x and z)

Step 8: is X>no of processes? If yes go to step 21 else go to Step 9

Step 9: Create a random number

Step 10: Fetch the value corresponding to the index x from q[ ]

Step 11: Multiply the value with the random number

Step 12: Calculate time taken to complete by the first processor along with existing process (if any)

[Time taken = cs[1]+ value from Step 11]

Step 13: Initialize 3 variables (say t &tc& p) and allocate the time required from Step 11 to t and p=1

Step 14: is z>no of processor? If yes goto Step 8 else goto Step 15

Step 15: Calculate time taken to complete by the z'th processor along with existing process (if any)

[Time taken = cs[z] + value from Step 9/p[z]]

Step 16: Allocate the time required from Step 15 and allocate to tc

Step 17: is t<tc+? If yes t=tc and p=z. cs[z] = cs[z] + tc.

Step 18: goto Step 14

Step 19: ap[x] = p

Step 20: goto Step 8

Step 21: Display ap[ ]

STOP

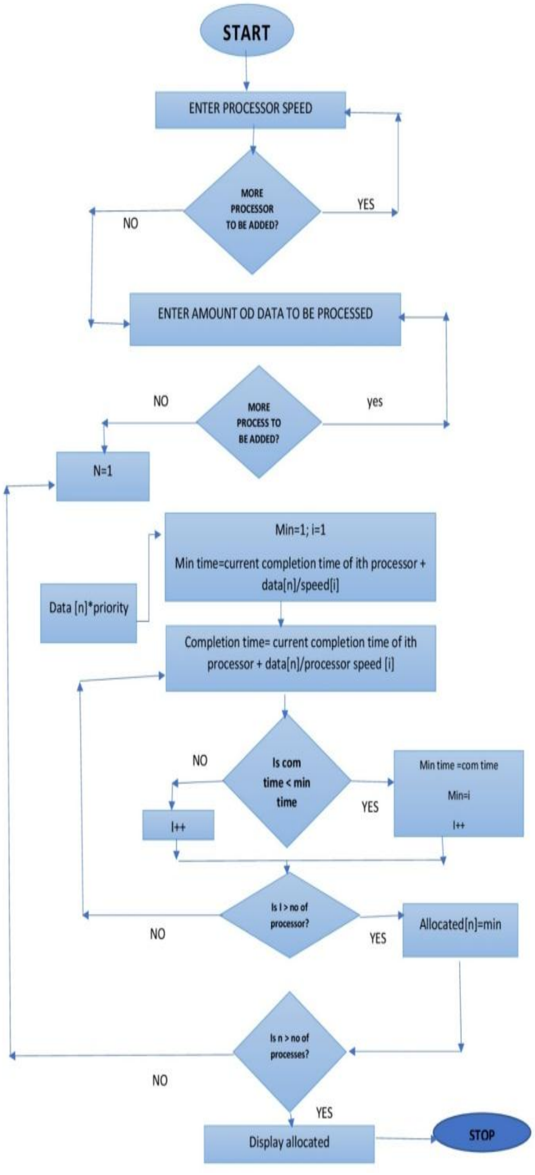
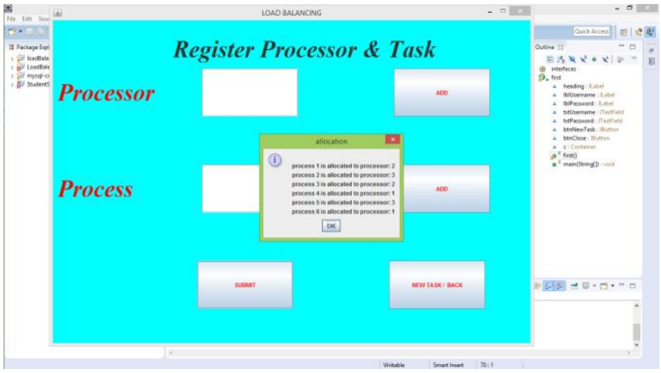

START
ENTER PROCESSOR SPEED
MORE PROCESSOR TO BE ADDED?
NO
YES
ENTER AMOUNT OD DATA TO BE PROCESSED
NO
MORE PROCESS TO BE ADDED?
yes
N=1
Min=1; i=1
Min time=current completion time of ith processor + data[n]/speed[i]
Data [n]*priority
Completion time= current completion time of ith processor + data[n]/processor speed [i]
NO
Is com time < min time
YES
Min time =com time
Min=i
I++
I++
Is I > no of processor?
NO
YES
Allocated[n]=min
Is n > no of processes?
NO
YES
Display allocated
STOP

Fig:7 Algorithm Flow Chart

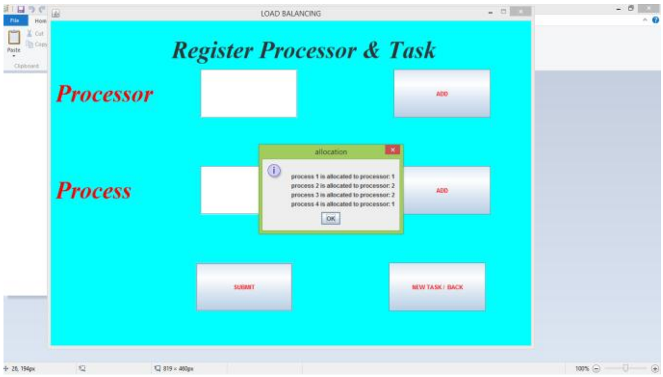
V. Results and Discussion

It showed promising signs to be 0ne of the best algorithm in the coming years. This almost guarantees that the task will be completed in least amount of time.



process 1 is allocated to processor: 2
process 2 is allocated to processor: 3
process 3 is allocated to processor: 2
process 4 is allocated to processor: 1
process 5 is allocated to processor: 3
process 6 is allocated to processor: 1

Fig 4: Execution Snapshot 1



process 1 is allocated to processor: 1
process 2 is allocated to processor: 2
process 3 is allocated to processor: 2
process 4 is allocated to processor: 1

Fig 5: Execution Snapshot 2

We consider 7 different processors and their probability to be chosen for job execution

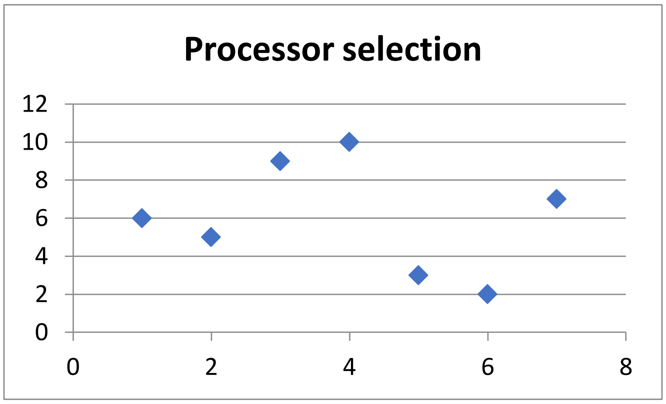


Fig 6: Selection of Processor

VI. Conclusion

Load balancing is all about equal distribution of load among systems but obviously, we aren't going to assign task in such a way that a processor which can complete the current task along with existing task faster than other processor remains idle just because when the task arrived it was busy and other was idle. Hence we might find that a processor can have 'n' number of task at a point and the other has none at that instance. But eventually we will observe it will be the fastest to complete all task. Future scope of this project will be halting current executing task and assigning a task to that processor that arrived at a later stage but with highest priority. Cloud Computing is a vast concept and load balancing plays a very important role in case of Clouds. There is a huge scope of improvement in this area. The performance of the given algorithms can also be increased by varying different parameters.

References

[1] The NIST Definition of Cloud Computing , Peter Mell Timothy Grance, NIST Special Publication 800-145

[2] A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment Mayanka Katyal, Atul Mishra http://www.publishingindia.com.

[3] R.Piplode, P. Sharma and U.K. Singh, "Study of Threats, Risk and Challenges in Cloud Computing", International Journal of Scientific Research in Computer Science and Engineering, Vol.1, Issue.1, pp.26-30, 2013.

[4] Enhanced Equally Distributed Load Balancing Algorithm For Cloud Computing, Shreyas Mulay, Sanjay Jain, IJRET: International Journal of Research in Engineering and Technology ISSN: 2319-1163.

[5] A. M. Nakai, E. Madeira, and L. E. Buzato, "Load Balancing for Internet Distributed Services Using Limited Redirection Rates", 5th IEEE Latin-American Symposium on Dependable Computing (LADC), 2011, pages 156-165.

[6] J. Hu, J. Gu, G. Sun, and T. Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud ComputingEnvironment", Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), 2010, pages 89-96

[7] A. Bhadani, and S. Chaudhary, "Performance evaluation of web servers using central load balancing policy over virtual machines oncloud", Proceedings of the Third Annual ACM Bangalore Conference (COMPUTE), Jan 2010.

[8]Y. Fang, F. Wang, and J. Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", Web InformationSystems and Mining, Lecture Notes in Computer Science, Vol. 6318, 2010, pages 271-277.

[9] Anitha H M, P. Jayarekha , "Security Challenges of Virtualization in Cloud Environment", International Journal of Scientific Research in Computer Science and Engineering, Vol.6, Issue.1, pp.37-43, 2001 .

[11] Shridhar G.Domanal and G.Ram Mohana Reddy, "Load Balancing in Cloud Computing Using Modified Throttled Algorithm", IEEE Randles, M.Bendiab, A. T. & Lamb, D. (2008).

[12] Agarwal, M., Srivastava, D.: A Genetic Algorithm inspired task scheduling in Cloud Computing. In : International Conference on Computing, Communication and Automation (ICCCA2016) (2016)

[13] Applied Soft Computing (2013) Ramezani, F., Lu, j., Hussain, F.: Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization. International journal of Parallel Programming (2014)

[14] Dasgupta, K., Mandal, B., Dutta, P., Mondal, J., Dam, S.: A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing. First International Conference on Computational

Intelligence: Modeling Techniques and Applications (CIMTA) 2013 (2013)

[15] Patel, R., Patel, S., Patel, D., Patel, T.: Improved GA Using Population Reduction Load Bancing in Cloud Computing. 2016 Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI) (2016)

[16] Bei, W., Jun, L.: Load balancing Task Scheduling based on multi Population Genetic Algorithm in cloud computing. In : 35th Chinese control Conference, China (2016) .

[17] Sandeep kaur and Sujhwinder Sharma Load Balancing in Cloud Computing with Enhanced Optimal Cost Scheduling Algorithm. Imperial Journal of Interdisciplinary Research (IJIR) (2016)

[18]B. Mondal,., K. Dasgupta, P. Dutta, P.: Load Balancig in Cloud Computing using Stochastic Hill Climbing-A soft Computing Appproach. ELEVIER (2012)

[19] Ariharan, V., Manakattu, S.: Neighbour Aware Random Sampling (NARS) algorithm for load balancing in Cloud computing. 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) (2015)

[20] Vanithaa, M., Marikkannu, P.: Effective resource utilization in cloud environment through a dynamic well Organised load balancing algorithm for virtual machines. Computers and Electrical Engineering (2017)

[21] Ekta Rani , Harpreet Kaur Efficient Load Balancing Task Scheduling in Cloud Computing using Raven Roosting Optimization Algorithm International Journal of Advanced Research in Computer Science Volume 8, No. 5, May-June 2017.

[22] Annwesha Banerjee Majumder Dipak Kumar Shaw and Sourav Majumder " A Load Balancing Algorithm for Selection of Competent Server in Cloud Environment Based on Capacity, Load and Energy" Indian Journal of Computer Science and Engineering (IJCSE) Vol. 8 No. 4 Aug-Sep 2017

Authors Profile

Mrs. Annwesha Bnaerjee Majumder has Completed her Master in MCNT from WestBengal University of Technology. Currently she is working as an Assistance Professor in the department of Information Technology , JIS College of Engineering. She has five years of research experiences.

Mr. Subhojit Sil was born on March 3, 1995 in Kolkata, West Bengal. He did his schooling from St. Stephen's School, Kolkata and pursued Bachelor of Technology from JIS College of Engineering, Kalyani, West Bengal. He is currently working at Capgemini. He is an OCA (Oracle Certified Associate). Certified in JEE 7 (Spring, JPA, SOAP, REST, Web Services).

Ms. Shilpi Das was born on September 1, 1994 in Tinsukia, Assam. She did her schooling from Carmel School, Digboi, Assam and pursued Bachelor of Technology from JIS College of Engineering, Kalyani, West Bengal. She is currently working at Amazon. She is certified for participating in the grand finale of 'Smart India Hackathon, 2017. She has done certification trainings in 'Web based Enterprise Application Development using JEE 7' and 'Programming with PHP 5.0 & MySQL'.

Ms. Aditi Mondal was born on September 9, 1995 in Burdwan, West Bengal. She did her schooling from Harisava Hindu Girls' High School, Burdwan and pursued Bachelor of Technology from JIS College of Engineering, Kalyani, West Bengal. She is currently working at ICICI Securities. She is certified for participating in the grand finale of 'SMART INDIA HACKATHON, 2017. She has done certification trainings in 'Web based Enterprise Application Development using JEE 7' and 'Programming with PHP 5.0 & MySQL'.