

Load Balancing of Unbalanced Matrix Problem of the Sufficient Machines with Min-Min Algorithm



Ranjan Kumar Mondal, Payel Ray, Enakshmi Nandi,
Biswajit Biswas, Manas Kumar Sanyal and Debabrata Sarddar

1 Introduction

We know Cloud Computing is an online-based facility on the Internet environment [1]. It transfers the application and information away from the physical machine into large information centers. Even this type of computing has also changed Information Technology companies utilized to intend software [2]. As it is in its developing phase, therefore there are a lot of complications stay on in its environment [3]. Example:

- To ensure capable access control.
- Migration for the necessity of minimum cost.
- For information move stage and resting stage supply right safety.
- Information Availability.
- Information marsh and transitive cases.
- Information security, the disclosure of susceptible information is feasible, and the vital difficulty in the environment is load balancing. In the event of load balancing,

R. K. Mondal (✉) · P. Ray · E. Nandi · D. Sarddar
Department of Computer Science and Engineering, University of Kalyani, Kalyani, India
e-mail: ranjan@klyuniv.ac.in

P. Ray
e-mail: payelraycse18@klyuniv.ac.in

E. Nandi
e-mail: enakshmicse18@klyuniv.ac.in

D. Sarddar
e-mail: dsarddar1@gmail.com

B. Biswas · M. K. Sanyal
Department of Business Administration, University of Kalyani, Kalyani, India
e-mail: biswajit.biswas0012@gmail.com

M. K. Sanyal
e-mail: manassanyal123@gmail.com

© Springer Nature Singapore Pte Ltd. 2018
J. K. Mandal et al. (eds.), *Methodologies and Application Issues of Contemporary Computing Framework*, https://doi.org/10.1007/978-981-13-2345-4_7

it has altered varieties of information, for example, a number of jobs those are waiting in the queue, and processing time at each processor and neighboring systems also might be exchanged amongst the systems to get the proficient and improved total work. So, many algorithms have proposed for Load Balancing purpose. In this article, we will say an innovative algorithm for proper load balancing to find the developed processing time for issues of improved superiority performance.

Task scheduling process can meet up clients' necessities, and get better the uses of resource, thereby developing the total performance of the cloud. However, the scheduling in a cloud environment is frequently about the static job requirements, and use of the resources rate is small. As a novel characteristic of the cloud, this article presented a two stages task scheduling technique based on load balancing [4].

Task scheduling is a multi-objective combinatorial optimization example in the web, among tasks is spanning and nodes balancing are vital features. Load balancing get-togethers service suppliers' condition. Scheduling determination inevitably reduces the job spanning. That is not interrelated with job finishing time. This article gives a scheduling approach that does not t work merely out of a scheduling series with little task runtime and average task makespan but convinces load balancing also [5].

2 Task Scheduling Optimization

Task scheduling is associated with the competence in cloud computing services and is of dominant significance. Scheduling in distributed methods typically has the objective of distributing the loads and makes the best use of their operation, while reducing the overall runtime [6]. Task scheduling has an essential role in developing elastic and dependable distributed systems. Its central principle is to schedule the tasks over resources with dynamism, taking into account the resources existing for implementation involving the innovation of an appropriate [7]. In a cloud environment, the number of tasks of the workflow can modify extremely rapidly, particularly when resources are allocated. The use of the heuristic algorithm makes sure a satisfactory runtime of the scheduling because it considerably decreases the difficulty of the search space. This offers a negotiation among the jobs to processors at execution time and the minimization of the assignment [8]. In prior works, heuristic optimization, GA, fuzzy-genetic algorithm, multi-objective GA, swarm optimization, and regular best-oriented ant colony have been applied to minimize task scheduling, with two primary objectives: (1) to decrease execution time, and (2) to decrease the result of tasks in cloud surroundings [9].

3 Cloud Computing

Now we see what cloud computing is? It is the most demand for web computing and a delivery mechanism nowadays. Still, it is not a new idea, but rather than today's advanced age, it has turned out to be universal because of the growth of the web vastly, broadband, and versatility requirements for users are it consumers, endeavors. The concentration is on the obvious slant of lesser scale and little ventures on the way to cloud computing and the benefits obtained by them. The investigation applies few components impacting the cloud exploit by this commerce group, whose requirements and industry fundamentals are in sum dissimilar from generous endeavors.

Cloud is simple to users and information; they are able inbuilt in different ways marked items, preventive open source, tools or programming. As a rule, they are standing on groups of servers to Open Source programming attached with in-house applications with framework programming. The actual difficulty that is anticipating cloud related to a standard phase is security and displaying cost [10]. Cloud computing has one of a type pay-as-you-go gain express; throughout which relationships pay for operation [11].

3.1 *Virtual Machine Migration*

Since cloud computing is an expressed structure, while the overall load is enlarged in an exact server farm, VM movement evades implementation corruption of the structure. The price of distribution and testing a sound cloud requires a bundle of exertion including apparatus assets and in the lengthy execution cost for its moved frameworks arrangement [10]. Be that as it may, this disadvantage can be destroyed with the support of Cloud Simulator. It displays the evaluation of the cloud computing. At this point, the structure, phase, and the resources are come into view [12].

3.2 *Cloud Service Models*

The model is throughout of some essential characteristics, three administration models, and some sending models.

A. SaaS

It is a service where an application is assisted by an administration offered to users. This type of service mitigates a load of encoding maintenance yet consumers give up manage over encoding forms and fundamentals. This evades disbursements of the concern, maintenance, patches.

B. PaaS

It provides a computing phase and a reply stack as supervision. The purchaser builds the product utilizing instruments with libraries from the dealer. The client likewise manages encoding business and agreement settings. The dealer provides the systems, servers, and diverse administrations. It presents a phase for constructing applications.

C. IaaS

It dealers present virtual machines, and document-based capacity, firewalls, virtual neighborhood, and encoding packs, etc. Collections of hypervisors can balance profits all over as signified by user' differing requirements. All structures are rearranged on demand. This supplier delivers basic system. They withdraw from purchasing machines, and evaluating benefit require [13].

4 Scenario

We outline the pictures for virtual machine load balancing algorithms in the various cloud. Under various pictures, the algorithms possibly will dissimilar restrictions.

Public Cloud: This cloud demotes to while a Cloud is accessible in a pay-per style [14]. Some main profit to facility contributors is suggested by this cloud, including no primary investment in communications and shifting of threats to communications suppliers. Though, this clouds lack fine-grained manage over information, network, and security settings, which gets in the way their efficiency in a lot of business developments [15]. Due to the lack of consistency, a variety and regularly altering APIs compose it hard to imprison all the virtual machines and hosts applications in this picture. Additionally, inconsistent loads is another face for virtual machine load balancing algorithms. So, a number of researchers have approved historic information to forecast the prospect load to conquer this challenge [16, 17].

Private Cloud: This cloud word denotes to inside information centers of a business not made obtainable to the common public. Though a public Cloud has the advantage of reduced assets to deal and improved operation speed, private Clouds are even further admired among activities along with a survey by IDG in [18]. The revision exposed that companies have a tendency to minimize active infrastructure with the performance of a private cloud which effects in an entire minor cost of possession. In some research, the entity clouds with small volume are executed to estimate virtual machine load balancing presentation. Within a cloud, the intra-information center system frequently has moderately unique properties compared to the inter-information center system [19]. So, dealing with the virtual machine load balancing trouble in a private cloud, the presentation like throughput would be regarded as a limitation.

Hybrid clouds: This cloud is a mixture of other two cloud models that make an effort to attend to the drawbacks of all move towards. In this cloud, a division of the facility communications runs in this clouds when the left behind portion runs in

public clouds. These hybrid clouds deliver additional elasticity than other clouds. Exclusively, they supply control and security over information compared to public clouds, while still assisting on-demand facility. The drawback, planning a hybrid cloud demands suspiciously determining the paramount dividing among other both cloud components [20]. Under this circumstance, the messaging cost will be the main limitation for virtual machine load balancing algorithms. Such as, in a distributed cloud, asks for may have the restriction that these demands are needed to be assigned to an official information center. As well, in a multi-cloud involving more than one clouds [21], the immigrations operations may be associated to load immigration from a cloud to another cloud (private to public) [22].

5 Load Balancing Definition

Load balancing is the first tenure utilizes to give out substantial progression loads to slighter progression systems to improve the tremendous work of the system [23]. In a system, this is a procedure of distributing total loads among different systems of a dispersed system to progress operation and performance time. A model load balancing algorithm should keep away from any exact system [24].

In cloud background, the choice of algorithm is not effortless; this is because it occupies additional constraints like security, reliability, throughput, etc. So, the most important objective of an algorithm in this environment is to develop the response time by sharing a whole load of a system. The algorithm should additionally make sure it is not overloading any particular system [25].

Load balancers efforts in two different ways: one is cooperative, and another is noncooperative. In a collaborative approach, the systems work at the same time in order to attain the general objective to minimize the total response time. And another mode, the tasks execute separately to develop the response time of limited functions [26].

Algorithms of load balancing are categorized as static algorithm and dynamic algorithm. The static algorithms are typically appropriate for uniform and steady surroundings and create great consequences in this atmosphere. Though, they are typically not elastic and could not equivalent the dynamic changes to the quality at some point in the implementation time. Dynamic algorithms are further bendable and get into concern dissimilar categories of attributes in the system together prior to and during execution time [22].

These algorithms can get used to modify and give superior results in various and dynamic settings. On the other hand, the distribution attributes become more composite and dynamic. Consequently, some algorithms could become useless and cause more overhead than compulsory resulting in substantial degradation of the presentation of the service [27].

A. **The Basic Idea of Algorithms:**

Load balancing makes a decision the task that how to decide the subsequent system and to transport a new demand to relocate the load from the process to under full process. It extends the incoming requesting load among the available systems to develop the performance considerably by the Cloud manager. There are two different kinds of an algorithm based on their functioning process; they are static and dynamic.

- B. **Static LB Algorithm:** It does not depend on the current circumstances. It makes a decision in the host; the demand would be implemented before setting up the demand.
- C. **Dynamic LB Algorithm:** The load balancer analyzes the present condition of load approaches at all obtainable host and performs the demand at the suitable host.

RR, WRR (Weighted Round Robin) algorithm and so on are an illustration of load balancing algorithm, but among them, RR is the most simple algorithm, utilized in the case when all the machines in the cluster have the similar processing facility. First-Come-First-Served, Throttled, Honey Foraging Algorithm and so on are the example of dynamic scheduling algorithm are superior to a static algorithm and appropriate for a lot of demands, which can carry the various workload, which would be unable to predict.

6 Related Works

Cloud computing offers a range of services to the user example resource sharing, online tools, and online storage [28]. In surroundings, each system performs a task or a subtask [29]. The Minimum Completion Time algorithm allocates tasks to the systems having the expected lowest amount of execution time of this job over other systems [25]. The Min-Min assumes the similar work as the MCT [30] algorithm to allocate a task and the scheme to finish this task with least amount execution time over other systems [31]. The LBMM [32] accepts Min-Min and load balancing strategy. It evades the unnecessarily copied assignment. Load Balancing with Job Switching [33] minimize loads of heavy loaded machine to under loaded machine by switching a particular job. Load balancing of Unbalanced Matrix with Hungarian Method [34] uses the Hungarian algorithm where jobs are greater than all nodes. Load balancing of the unbalanced cost matrix [35] is the same as the previous algorithm.

6.1 *Criteria for Performance*

Our proposed load balancing algorithm is considered to get together all scheduling criterion example greatest CPU operation, lowest turnaround time, utmost through-

put, smallest waiting time, and context switches. We discuss some definitions as follows: [36].

- Arrival Time:** Entrance time is the time of entrance of the process in the memory.
- Response Time:** This means the time of arrival minus initial reaction time by the processor.
- Burst Time:** This time is how long the process holds the processor.
- Turnaround Time:** This means the time of arrival minus the time of execution of the task.
- Waiting Time:** This is that how long of a time, the process waiting in the ready queue.
- Throughput:** This is a number of the process executed per unit period

7 Proposed Work

Whenever the matrix of an assignment problem is not a square matrix, that is, while the figure of sources system is not identical to the figure of destinations system, the assignment problem is called an unbalanced assignment problem. This kind of problems, replica rows (or columns) are added to the matrix to complete it to shape a square matrix. The replica rows or columns will contain all elements as zeroes. The Hungarian method [37] could be used to solve this type of crisis.

8 A. Algorithm

To provide an algorithmic demonstration of the technique, consider an example which consists of 'i' systems $i = \{N_1, N_2, \dots, N_i\}$. And 'j' jobs $j = \{J_1, J_2, \dots, J_j\}$ is considered to be allocated for performance on 'm' available computing systems and the performance value C_{mn} , where $m = 1, 2, \dots, i$ and $n = 1, 2, \dots, i$, where $j > i$, i.e., the numeral of jobs is greater than a numeral of computing systems.

The development of the proposed algorithm is presented as follows

Step 1: At first, we evaluate the average run time of each machine for all jobs, in that order.

Step 2: Next, it is to discover the job having the highest average run time.

Step 3: Remove those machine one by one with the highest average run time until a quantity of machines is equal to the quantity of jobs.

Step 4: Now, it is to discover the unallocated job having the least runtime minimum than the highest average run time for the job chosen in Step 3. Then, this job is removed to the chosen job for execution.

Step 5: If there is no unallocated job can be chosen in Step 2, all machines that include unallocated and allocated machines should be reestimated. The least run time of an allocated machine is the summation of the least runtime for an allocated job on this machine and the least run time of the present job. The least run time of an unallocated machine is the present least runtime for the job. It is to locate the unallocated node or allocated machine that has the least runtime minimum than the highest average run time for the job chosen in Step 2. After that, this job is removed to the chosen machine for execution.

Step 6: Do again Step 4 to Step 5, until everyone jobs have been executed thoroughly.

Example: Suppose an IT company has six machines that are used for four jobs. All jobs can be allocated to one system at the same time.

The cost of every job on every machine is given in the Table 1.

Assignment Problem

See Table 1.

Final Result

See Table 2.

Table 1 The execution time for each task in various systems

Jobs/Machines	M ₁₁	M ₁₂	M ₁₃	M ₁₄	M ₁₅	M ₁₆
J ₁₁	18	19	18	15	36	18
J ₁₂	22	18	17	17	42	23
J ₁₃	13	20	12	14	42	22
J ₁₄	22	24	19	18	38	22
J ₁₅	14	17	10	19	24	21

Table 2 Assigned each task at each computing systems respectively

Jobs/Machines	M ₁₁	M ₁₂	M ₁₃	M ₁₄	M ₁₆
J ₁₁	18	19	18	15	18
J ₁₂	22	18	19	17	23
J ₁₃	13	20	12	14	22
J ₁₄	22	24	19	18	23
J ₁₅	14	17	10	19	21

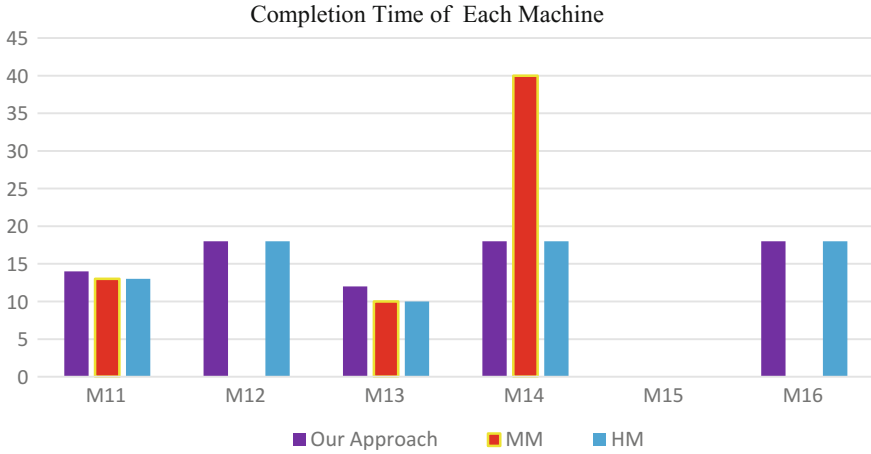


Fig. 1 The judgment of run time (in ms) of all tasks at systems

9 Case Study

Table 1 demonstrates the execution time for all tasks at systems. The threshold is the average execution time of the task t_i in all executing systems. For calculating the performance of our proposed algorithm, our approach is compared with MM and HM shown in Fig. 1. Figure 1 displays the judgment execution time of all executing system among our approach, MM and HM. The execution times for executing all tasks by using our approach, HM and MM are 18, 18, and 42 ms, correspondingly. Our proposed work gets the smallest execution time and greater load balancing than another algorithm, example MM and HM in this environment.

10 Comparison

Our proposed load balancing of unbalanced cost matrix can get improved balancing and execution than other algorithms, example LBMM and MM from the Fig. 1.

11 Conclusion

In this present article, we have proposed a proficient work for the computing network to allocate tasks to computer systems in accordance with their resource potential correspondingly. Our proposed work can complete improved balancing and presentation than other algorithms, example HM and MM from the demonstration.

References

1. B. Hayes, Cloud computing. *Commun. ACM* **51**(7), 9–11 (2008)
2. S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, A. Ghalsasi, Cloud computing—the business perspective. *Decis. Support Syst.* **51**(1), 176–189 (2011)
3. B.P. Rimal, A. Jukan, D. Katsaros, Y. Goeleven, Architectural requirements for cloud computing systems: an enterprise cloud approach. *J. Grid Comput.* **9**(1), 3–26 (2011)
4. Y. Fang, F. Wang, J. Ge, A task scheduling algorithm based on load balancing in cloud computing, in *International Conference on Web Information Systems and Mining* (Springer, Berlin, Heidelberg, 2010), pp. 271–277
5. T. Wang, Z. Liu, Y. Chen, Y. Xu, X. Dai, Load balancing task scheduling based on genetic algorithm in cloud computing, in *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing (DASC)* (IEEE, 2014), pp. 146–152
6. A.Y. Zomaya, T. Yee-Hwei, Observations on using genetic algorithms for dynamic load-balancing. *IEEE Trans. Parallel Distrib. Syst.* **12**(9), 899–911 (2001)
7. C. Zhao, S. Zhang, Q. Liu, J. Xie, J. Hu, Independent tasks scheduling based on genetic algorithm in cloud computing, in *5th International Conference on Wireless Communications, Networking, and Mobile Computing* (2009), pp. 1–4
8. E. Juhnke, T. Dörnemann, D. Böck, B. Freisleben, Multi-objective scheduling of BPEL workflows in geographically distributed clouds, in *4th IEEE International Conference on Cloud Computing* (2011), pp. 412–419
9. F. Ramezani, J. Lu, F.K. Hussain, Task-based system load balancing in cloud computing using particle swarm optimization. *Int. J. Parallel Program.* **42**(5), 739–754 (2014)
10. Y. Wang, J. Li, H.H. Wang, Cluster and cloud computing framework for scientific metrology in flow control. *Cluster Comput.* 1–10 (2017)
11. A. Shawish, M. Salama, Cloud computing: paradigms and technologies, in *Inter-cooperative Collective Intelligence: Techniques and Applications* (Springer, Berlin, Heidelberg, 2014), pp. 39–67
12. A. Shawish, M. Salama, Cloud computing: paradigms and technologies, in *Inter-cooperative Collective Intelligence: Techniques and Applications* (Springer, Berlin, Heidelberg, 2014), pp. 39–67
13. Y. Wang, J. Li, H.H. Wang, Cluster and cloud computing framework for scientific metrology in flow control. *Cluster Comput.* 1–10 (2017)
14. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, et al., Above the clouds: a Berkeley view of cloud computing (2009)
15. L. Zhao, S. Sakr, A. Liu, A. Bouguettaya, *Cloud Data Management* (Springer, 2014)
16. J. Hu, J. Gu, G. Sun, T. Zhao, A scheduling strategy on load balancing of virtual machine resources in cloud computing environment, in *2010 3rd International Symposium on Parallel Architectures, Algorithms, and Programming* (IEEE, 2010), pp. 89–96
17. W.T. Wen, C.D. Wang, D.S. Wu, Y.Y. Xie, An aco-based scheduling strategy on load balancing in cloud computing environment, in *2015 Ninth International Conference on Frontier of Computer Science and Technology* (IEEE, 2015), pp. 364–369
18. G. Roos, Enterprise prefer private cloud: Survey 2013, <http://www.eweek.com/cloud/enterprises-prefer-private-clouds-survey/>
19. A. Li, X. Yang, S. Kandula, M. Zhang, Cloudcmp: comparing public cloud providers, in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement* (ACM, 2010), pp. 1–14
20. Q. Zhang, L. Cheng, R. Boutaba, Cloud computing: state-of-the-art and research challenges. *J. Internet Serv. Appl.* **1**(1), 7–18 (2010)
21. D. Petcu, Multi-cloud: expectations and current approaches, in *Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds* (ACM, 2013), pp. 1–6
22. M. Xu, W. Tian, R. Buyya, A survey on load balancing algorithms for virtual machines placement in cloud computing. *Concurr. Comput. Pract. Exp.* **29**(12) (2017)

23. S. Song, T. Lv, X. Chen, Load balancing for future internet: an approach based on game theory. *J. Appl. Math.* (2014)
24. A.A. Neghabi, N.J. Navimipour, M. Hosseinzadeh, A. Rezaee, Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature. *IEEE Access* **6**, 14159–14178 (2018)
25. H. Mehta, P. Kanungo, M. Chandwani, Decentralized content aware load balancing algorithm for distributed computing environments, in *Proceedings of the International Conference & Workshop on Emerging Trends in Technology* (ACM, 2011), pp. 370–375
26. A. B. Singh, S. Bhat, R. Raju, R. D'Souza, Survey on various load balancing techniques in cloud computing. *Adv. Comput.* **7**(2), 28–34 (2017)
27. K. Al Nuaimi, N. Mohamed, M. Al Nuaimi, J. Al-Jaroodi, A survey of load balancing in cloud computing: Challenges and algorithms, in *2012 Second Symposium on Network Cloud Computing and Applications (NCCA)* (IEEE, 2012), pp. 137–142
28. Y.-T. Wang, Load sharing in distributed systems. *IEEE Trans. Comput.* **100**(3), 204–217 (1985)
29. M. Antoine, L. Pellegrino, F. Huet, F. Baude, A generic API for load balancing in distributed systems for big data management. *Concurr. Comput. Pract. Exp.* **28**(8), 2440–2456 (2016)
30. D. Grosu, A.T. Chronopoulos, M.-Y. Leung, Load balancing in distributed systems: an approach using cooperative games, in *Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, Abstracts and CD-ROM* (IEEE, 2001), pp. 10
31. Robert Fox, Library in the clouds. *OCLC Syst. Serv. Int. Digital Library Persp.* **25**(3), 156–161 (2009)
32. S.-C. Wang, K.-Q. Yan, W.-P. Liao, S.-S. Wang, Towards a load balancing in a three-level cloud computing network, in *2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, vol. 1 (IEEE, 2010), pp. 108–113
33. G. Ritchie, J. Levine, A fast, effective local search for scheduling independent jobs in heterogeneous computing environments. *J. Comput. Appl.* **25**, 1190–1192 (2005)
34. T.D. Braun, H.J. Siegel, N. Beck, L.L. Bölöni, M. Maheswaran, A.I. Reuther, J.P. Robertson, M.D. Theys, B. Yao, D. Hensgen, R.F. Freund, A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *J. Parallel Distrib. Comput.* **61**, 810–837 (2001)
35. S.C. Wang, K.Q. Yan, W.P. Liao, S.S. Wang, Towards a load balancing in a three-level cloud computing network, in *CSIT* (2010), pp. 108–113
36. C.-L. Hung, H.-H. Wang, Y.-C. Hu, Efficient load balancing algorithm for cloud computing network, in *International Conference on Information Science and Technology (IST 2012)*, April 2012, pp. 28–30
37. H.W. Kuhn, The Hungarian method for the assignment problem. *Naval Res. Logist. (NRL)* **2**(1–2), 83–97 (1955)