Research Inventy: International Journal of Engineering And Science
Issn (e): 2278-4721, Issn (p):2319-6483, PP: 38-45
www.researchinventy.com

Overcoming Load Imbalance of Unbalanced Matrix with the RowSum Method

Payel Ray*[1], Debabrata Sarddar[2], Subhajit Roy[3]
Department of Computer Science & Engineering, University of Kalyani[1,2]
Bengal Institute Of Engineering & Management, MAKAUT[3]
payelray009@gmail.com[1], dsarddar1@gmail.com[2], jroy395@gmail.com[3]

Abstract: *Cloud computing is turning into one of the main rising technologies based on the internet.In a simple terms we can say, cloud computing by which we will be able to store and gain acces to data and the application in excess of the internet as an alternative to your computer's hard drive. Cloud computing is mainly a on-demand basis service structure from appliances to storage space and compacts with the authority usually above the internet and its main advantage is pay as you use. There is a wide range of network servers joined to online computing to provide different types of online services to cloud clients. There are some fewer figures of network servers having joined the cloud networks that have to perform greater tasks and the time is limited. Therefore its easy to accomplish the entire works at a specific time. Few systems complete each task, so there is a demand for balancing workloads at a limited time. Here Load balancing reduces the time limit of each task in an exacting process.*

There are invariably no chances to continue an equal quantity of network servers to accomplish the same quantity of jobs in the same period. All tasks to be performed would be higher than the connection to the servers. There are inadequate servers that have to hand over a million jobs in an instant.

We will propose an algorithm that a few nodes perform the jobs here;some jobs are more than the nodes and balance each node to exploit the quality of services in cloud computing.

Keywords:*Cloud Computing, Load balancing, Unbalanced Matrix ,Minimum Completion Time.*

I. Introduction:

Cloud computing is the exploitation of online resources received as a support over a system [1]. The list emerges from the conventional make support of a diminished-shaped sign as an abstraction for the system communications containing in system figures [2]. Cloud system entrusts distant services with a consumer's data, program, and estimation.Cloud computing comprises sources made existence on the network as managed third-party services. These functions present the standard of entrance to modern software industries and strong-end systems of physical workstation machines [3]. The purpose of cloud computing is to concern traditional device or great-performance computing application, made to work by analysis tools, to execute trillions of calculations per unit term, in custom-oriented functions for example financial portfolios, to give personalized guidance, to give data cache or to power extensive, immersive physical machine games. The cloud computing uses techniques of extensive stores of centers running low-cost customer PC technology with specific relations to expand data-handling chores across them. This distributed infrastructure encloses large pools of processes bound. often, virtualization techniques are required service to manipulate the influence of cloud computing.

1.1 Virtualization

Virtualization is the structure of a virtual comb rather than exact combs, such as an OS, a desktop computer, a machine, or systems [4]. Virtualization software aids us in passing the same working techniques and diverse operations on the same center or the same system at the same moment. Considering the cloud system, it is easy to be dealt with to as experienced computing abilities. Program and data are presented on-ordered for benefits through the network. Virtualization is capable to live without the cloud, but the cloud system cannot endure without virtualization. Cloud computing uses virtualization to provide supplies to the consumer. On virtualization, higher than one process is capable to go on a particular substantial system, so system application is ready to be raised.

1. There are two groups of virtualization area.

Full-Virtualization- In this virtualization, the full operations of one system remain over another system.The real-system is functionality living in a virtual system in the virtual form [19][13].

Para-Virtualization- In this virtualization, the full establishment is not entirely living;supplies are given partly.

2. Benefits of virtualization

a. The distribution of systems works in cost reduction.

b. Hardware independent.
c.Virtual systems can be supplied between different hosts.

2. Cloud Computing & Load Balancing

2.1Cloud system: Cloud system Cloud computing is a set for facilitating desirable, on-used structure all right of entrance to a shared pool of configurable sources that is capable to be instantly stipulation and delivered with the shortest management attempt [5][12]. Cloud system is based on virtualization. Virtualization is the key feature of the cloud system. Cloud system virtualizes a particular system into a collection of virtual systems. Fundamentally, a virtual system is a program implementation of the real system. A moderate-standard computing called virtual system monitor is effective for the distribution of a particular physical machine among individual sources.
The purpose of cloud computing is to have stronger function of the systemand to obtain the most worthwhile source expenditure, best throughout, least return time, and evading extra load.

3. Classification of Algorithms:

It has subdivided The load balancing algorithms based on the new expounding of a technique and who introduced the process
1. Depending on which initiates the process [6]:
A. Sender-Initiated: Sender or client initiates the completion of the load balancing algorithm on analyzing the demand for load balancing.
B. Receiver-Initiated: Receiver or workstation center initiates the conclusion of a load balancing algorithm on diagnosing the wish for load balancing.
C. Symmetric: This section of the algorithm is a combination of sender-initiated category and receiver-initiated category algorithms.
2. Depending on the present state of the Static system algorithm: In the static algorithm, there is a suited circulation of traffic among the workstations. This algorithm requires a preceding knowledge of the sources of the scheme so that the knowledge of moving the loads not depend on the performing condition of the system.
3. Dynamic Algorithm: In the dynamic result, for balancing the load, the smallest workstation center in the entire process is considered upon and selected. For this, real-time connection with the Internet is required that is competent to broaden the data progress in the scheme. Here to conclude on maintaining the load, the standing construct of the process is exploited.

4. Load Balancing in Cloud

Load balancing is a method of reassigning the full load to the dissimilar nodes of the shared system to develop resource utilize well-formed and to have a corrected response time of the operation, alongside removing circumstances where a few of the nodes are extra loaded whereas a collection of alternative nodes are simply packed. A load-balancing method strives to heighten the application of resources with a tiny load or idle resources, thereby clearing the resources with an extraload. The method tries to deal with the load amongs teach obtainable resource.At the same time; it pursues to shorten the make-span of the fruitful exploitation of resources.

In dispersed schemes, produced of homogeneous and devoted systems, load balancing processes have been effectively examined. On the separate part, these method will not achieve fresh in cloud architecture owing to its heterogeneity, scalability, and sovereignty. This creates load-balanced scheduling for cloud computing extra tough and an interesting issue for many analysts.

The Non-conventional methods vary from the classical process in that it causes the most suitable results in a small term. There is no most excellent scheduling process for a cloud system. The alternative choice is to establish on a proper approach to use in known cloud surroundings owing to the components of the tasks, servers, and network heterogeneity.

5. Related Works

LBMM [7][14] is a static balancing algorithm. This algorithm employs load balancing amongst machines considering it. The Min–Min algorithm chooses the lowest completion time for all tasks. Then it supports the works with the least completion time amongst all the works. The algorithm continues by selecting the tasks to the system planning the shortest make span. Min - Min states similar programs in apprehension of all tasks are organized.The algorithm executes the Min - Min schedule and creates a decision of the node with the excellent make span. Consequent to the system needs the work with the smallest execution point. The execution time of the selected job is dealt with for each source. The peak execution time of the chosen job is graded with the make-span. If the execution time is smallest, thus the selected job is assigned to the machine, including the maximum make span. Else, the then maximum make span of the job is chosen, and the strides are constant. The method ends if each appliance with each job is charged. In conditions where the model of short jobs is bigger than the estimate of sufficient jobs in meta-job, these algorithms have improved appearance. This algorithm does not think small and elevated structures, heterogeneity, and jobs.

Two-Phase Scheduling [8] is the structure of OLB and LBMM algorithms to drive superior completion competency and continues on with the process load balancing. OLB scheduling keeps on each machine in operating status to obtain hold of the purpose of load balancing with the LBMM algorithm is enlisted to lessen the conclusion of the time of all tasks on the appliance by this means decreasing the full execution time. This algorithm was performing on having rescued from the process of sources and chose up the effort competency.

Min-Min [9] assumes MM with a load balancing method. In this procedure, all tasks are split into subtasks. The completion times of each subtask on every server, since completely as the threshold, are classified. All subtasks, along with the server to performs the shortest volume of execution time for the subtask goes into Min time. The subtask having the slightest volume of execution time amongst all subtasks is wished and allotted to the linked server.

LB3M [10[15]] scheme determines the average completion time of all subtasks on all servers. The subtask has the largest average completion time,and the server is providing the least completion time taken. At present, the chosen subtask will be performed by the assigned server. If the server is before assigned, formerly the procedure evaluates the integration time of the server. For the given server, make-span is the summation of the make span of appointed tasks and integration time for the current subtask. For the unallocated server, make-span is the execution time of the remaining task. This growth is recurrent in by the subsequently unassigned subtasks.

6. Background

Tasks scheduling and provision of systems are the essential traces of cloud computing, which influence the movement of the server [11]. To get peak throughput, various task scheduling algorithms have been elected by the analysts for scheduling and mount of resources. The conversion of the profitable algorithm sets on the attitude of the server.

A. Task Scheduling in Cloud Computing

In cloud systems, precise task scheduling is desired to build prowess and to decrease down the execution space. The purpose of task scheduling is to prove an exceptional origin from all the delivered systems so that carry out the performance of the computing position develops.

B. Makespan

The total time rated for all the tasks in meta-task to get solved Make span. Meta-task is a distribution of the queue in which jobs, which are ready to perform, is stored. It is a evaluate the throughput of the differing computing machines.

7. Problem definition

Suppose a non-balanced matrix consisting of machines M = {M_{11}, M_{12}, ., M_{1n}}. The jobs J = {J_{11}, J_{12},. . ., J_{1n}} is considered to be distributed for execution on 'm' presented machines. The execution cost of all jobs on all machines is recognized and stated in the matrix of n*m. The objective is to achieve the best result. A method is devised to get the said costs in such a way that each job is to be designated on the presented machines.

The proposed algorithm as given below:

Considered a problem where a set of machines M = {M_{11}, M_{12}, M_{13}, M_{14}, M_{15}}, and a set of jobs J = {J_{11}, J_{12}, J_{13}, J_{14}, J_{15}, J_{16}, J_{17}, J_{18}}. The assignment matrix contains the execution time of allworks to all machines.

Steps-1 to 2: Input: 5*8.

Ji/Mj	**M_{11}**	**M_{12}**	**M_{13}**	**M_{14}**	**M_{15}**
J_{11}	310	300	290	300	220
J_{12}	250	310	290	310	210
J_{13}	180	200	310	200	190
J_{14}	330	190	190	250	180
J_{15}	280	220	190	260	170
J_{16}	200	210	220	200	150
J_{17}	230	310	230	190	170
J_{18}	270	200	260	220	200.

Step-3: To gain the sum of all rows and all columns of the matrix, i.e.,.,the sum of all rows and all columnsare as follows:

Sum Row

J_{11}	**J_{12}**	**J_{13}**	**J_{14}**	**J_{15}**	**J_{16}**	**J_{17}**	**J_{08}**
1420	1370	1080	1140	1120	980	1130	1150

Sum Column

M_{11}	M_{12}	M_{13}	M_{14}	M_{15}
2050	1940	1980	1930	1490.

To partition the matrix to classify the first sub-problem by selecting rows equivalent to J_{13}, J_{14}, J_{15}, J_{16}, J_{17},and second sub-problem by selecting rows related to the jobs J_{11}, J_{12}, $J_{18}$and by deleting columns related to M_{11}, M_{12}.Then the modified matrices are as follows:

Sub-Problem-I:

Ji/Mj	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}
J_{13}	180	200	310	200	190
J_{14}	330	190	190	250	180
J_{15}	280	220	190	260	170
J_{16}	200	210	220	200	150
J_{17}	230	310	230	190	170

Moreover, Sub-Problem-II:

Ji/Mj	M_{12}	M_{14}	M_{15}
J_{11}	300	300	220
J_{12}	310	310	210
J_{18}	200	220	200

The final result is as follows:

Machine	→Job	→Cost
M_{11} →	J_{13}	→180
M_{12} →	J_{14}*J_{18}	→190+200=390
M_{13} →	J_{15}	→190
M_{14} →	J_{11}*J_{17}	→300+190=490
M_{15} →	J_{12}*J_{16}	→200+150=350

The paper suggested a modified way of solving unbalanced assignment problems. An assignment method called Hungarian [6] offers overall assignment cost of the matrix that is 1600, along with the additional three jobs assigned to the dummy machines, in further statements.These three tasks are overlooked for supplementary execution, while the unique matrix is divided into the sub-problems, balanced assignment problems in the natural world.

8. Proposed method

To determine the matrix cost in addition to acombination of thejob(s) vs. machine(s) of anunbalanced matrix, toconcentrate on a problem consisting of machines $M=\{M_1,M_2,...,M_m\}$.The jobs $J=\{J_1,J_2,...,J_n\}$, is considered to be assigned for execution on the 'm' obtainable machines with the implementation cost C_{ij}, where i=1,2,...,m and j=1,2,...,n are mentioned in the matrix in which mis greater than n. First of all, we get the sum of all rows and all columns of the matrix, store the results in the array, Row-sum,and Column-sum. Then tochoose the first n rows byRow-sum, i.e., starting with smallest to next smallest to the array Row-sum and deleting rows related to the remaining (r) jobs. Store the results in the new array that should be the array for the first sub-problem. Do this process again until remaining jobs become less than amachine when remaining jobs are less than n, then, deleting (c) columns byColumn-sum, i.e., corresponding to thevalue(s) most higher to next higher to form the last sub-problem. Accumulate the results in the new array that will be the array for the last sub-problem. Apply the Hungarian method [4] to get the best possiblesolution of all sub-problems, which is now becoming abalanced problem. Lastly, rearrange allthe sub-problems to acquire the best possiblecost.

8.1. Computational algorithm

The method in the paper is to verify the following components:
– Identifythe condition to assign the overload jobs.
– Establish the procedure of the assignment.
– Compute the least cost.

8.2.Algorithm

To present an algorithmic illustrationof the method, consider a problem that consists of 'm' Machines M={M_1, M_2, ...,M_m}. Moreover,'n' jobs J={J_1, J_2,...,J_n} is considered to be assigned for execution on 'm' obtainable machines with the implementation cost C_{ij}, where i=1,2,...,m and j=1,2,...,n, where m>n, i.e., the jobs is greater than machines.

Step1: *Input: m*n matrix.*
Step 2:*To calculate the* ***Row Sum*** *of alltasks, correspondingly.*
Step 3: *To divide the problem into two components. The first component will be the n*n matrix based on the highestrow-sum,andthe remaining components would be then*m format where n>m. To calculate easy, we follow this rule. (First, thecomponent will be followed by a single machine with a single job with executes shortest execution time based on highestrow-sum, and thesecondcomponent will be followed byadevice that performs minimum execution time based on maximum row-sum.).*
Step 4: *In second component, find least cost of unassigned jobs J_i of all already assigned machines from highest Row-sum, and it is to be attached to therelated machine and remove from the assigned job and related machine from the matrix and find next least cost of unassigned jobs from next maximum row-sum and assign the task to its matching machine and so on in anticipation ofeachtask assigned to their consequentdevice.*
Step 5: *So each jobassigned as a minimum to its related machine and more than two jobs are not allocated to any device.*
Step 8:*Stop.*

9.Illustration of an example

Let us a setthathas5 machines M={M_1,M_2,M_3, M_4, M_5}, and 8 jobsJ={J_1, J_2, J_3, J_4, J_5, J_6, J_7, J_8}.The set contains the execution costs of every job toeachmachine.

Step-1: Input: 5*8matrix

J_n/M_m	M_1	M_2	M_3	M_4	M_5
J_1	151	277	185	276	321
J_2	245	286	256	264	402
J_3	246	245	412	423	257
J_4	269	175	145	125	156
J_5	421	178	185	425	235
J_6	257	257	125	325	362
J_7	159	268	412	256	286
J_8	365	286	236	314	279

Step 2 It is to calculate the **Row Sum** of each task, respectively.

J_n/M_m	M_1	M_2	M_3	M_4	M_5	Row-sum
J_1	151	277	185	276	321	**1210**
J_2	245	286	256	264	402	**1453**
J_3	246	245	412	423	257	**1583**
J_4	269	175	145	125	156	**870**
J_5	421	178	185	425	235	**1444**
J_6	257	257	125	325	362	**1326**
J_7	159	268	412	256	286	**1381**
J_8	365	286	236	314	279	**1480**

Step 3: We split the matrix into two parts. The first part will be the n*n matrix based on the maximum row-sum, and the remaining parts would be then*m format.Easy to calculate, we follow this rule.First, the part will be followed by one machine with one job with executes minimum execution time based on maximum row-sum,and thesecond part will be followed byadevice that performs minimum execution time based on maximum row-sum.

J_n/M_m	M_1	M_2	M_3	M_4	M_5	Row-sum
J_2	**245**	286	256	264	402	**1453**
J_3	246	**245**	412	423	257	**1583**
J_5	421	178	185	425	**235**	**1444**

J_7	159	268	412	**256**	286	**1381**
J_8	365	286	**236**	314	279	**1480**

Step 4: Like all machines assigned by jobs by now,thus we want to remain thelowest cost of unassigned jobs inthe employedsystem. Find minimum value of unassigned jobs J_i of all already assigned devices from maximum Row-sum,and it is to be attached to the corresponding machine and remove from the assigned job and similar machine from the list and find next minimum cost of unassigned jobs from next maximum row-sum and allocate the task to its corresponding machine and so on until all tasks assigned to their corresponding device.

J_n/M_m	M_1	M_2	M_3	M_4	M_5	Row-sum
J_1	151	277	185	276	321	1210
J_4	269	175	145	125	156	**870**
J_6	257	257	125	325	362	1326

Step 5: So all jobs assigned at least to its corresponding machine, and more the two jobs are not allocated to any device.

J_n/M_m	M_1	M_2	M_3	M_4	M_5
J_1	***151***	277	185	276	321
J_2	**245**	286	256	264	402
J_3	246	**245**	412	423	257
J_4	269	175	145	***125***	156
J_5	421	178	185	425	**235**
J_6	257	257	***125***	325	362
J_7	159	268	412	**256**	286
J_8	365	286	**236**	314	279

Final Result:

M1→ J1*J2→ 151+245=396
M2→J3→245
M3→J6*J8→125+236=361
M4→J4*J7→125+256=381
M5→J5→235

Our approach: 1618

M_m	M_1	M_2	M_3	M_4	M_5
Costs	396	245	361	381	235

10. ResultAnalysis:

Following tableExecution time (ms) of each task at all nodes.
The experiments were performed. The version of the system is Intel Core i3 4th Generation processor, 3.4 GHz CPU and 4GB RAM running on Windows 7 platform. We have considered makespan. We have conducted twosets of simulation scenarios as follows.

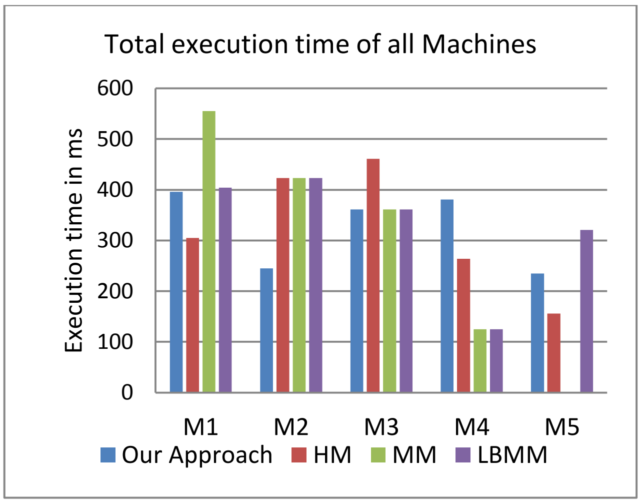

Figure 1: Execution time (ms) of each task at different computing nodes.

Figure 1 explains the implementation time for eachtask at several computing nodes. To compute the presentation of our proposed work is evaluated with other processes shown in Figure 1. Figure 1 displays the evaluation of theexecution time of all nodes among in our approach. Forcompleting each task by using theproposed algorithm, LBMM, HM, and MM, the makespanis396, 423, 461, and 555 ms, correspondingly. Our approach attained the lowestcompletion time and improved balancing ofthe load than other algorithms.

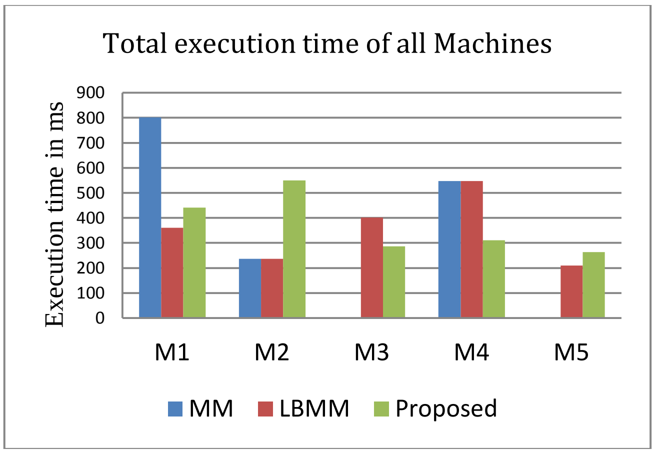

Figure 2: Execution time (ms) of eight tasks at five nodes.

Figure[2]displays the implementation time for eight tasks at five nodes. For completing all tasks by using theproposed algorithm, LBMM, and MM, the makespan is550,547, and 802 ms, correspondingly. Our approach attained the not lowest completion time,but it is better than others algorithms because of MM executes three nodes only and its Maximum completion time is 802 ns and minimum completion time is 236 ns as well as in case LBMM where Maximum completion time is 547 ns and minimum completion time is 210 ns wherein our method, maximum completion time is 550 ns and minimum completion time is 263 ns. That is, there is a minimum difference between maximum execution node and minimum execution node.

11. Conclusion

None of the jobs are assigned to dummy machines. The solution to the unbalanced matrix problem, thematrix obtained with the support of the Hungarian method,is mentioned.Allsubtaskswith makespanare assigned toits relatedmachine so that the result is best possiblefor the system. Although some devices execute more than one job,makespan is most favorable, and loads are balancing with our proposed work.

The technique is presented in an algorithmic shape and implemented on the numerous sets of input data to test the act and usefulness of the algorithm.Applying to load balancing is straightforward.

References:

[1]. Chan, M. C., Jiang, J. R., & Huang, S. T. (2012, August). Fault-tolerant and secure networked storage. In Digital Information Management (ICDIM), 2012 Seventh International Conference on (pp. 186-191). IEEE.

[2]. Jeevitha, M., Chandrasekar, A., &Karthik, S. (2015). Survey On Verification Of Storage Correctness In Cloud Computing. International Journal Of Engineering And Computer Science, 4(09).

[3]. Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. Wireless communications and mobile computing, 13(18), 1587-1611.

[4]. Dive, S. R., Ingale, A. D., &Shahade, M. R. (2013). Virtual Machine and Network-Performance, Study, Advantages,AndVirtualisation Option. International Journal of Advanced Research in Computer Science,4(6).

[5]. Moura, J., &Hutchison, D. (2016). Review and analysis of networking challenges in cloud computing. Journal of Network and Computer Applications, 60, 113-129.

[6]. Eager, D. L., Lazowska, E. D., &Zahorjan, J. (1985). A comparison of receiver-initiated and sender-initiated adaptive load sharing. ACM SIGMETRICS Performance Evaluation Review, 13(2), 1-3.

[7]. Kokilavani, T., &Amalarethinam, D. G. (2011). Load balanced min-min algorithm for static meta-task scheduling in grid computing. International Journal of Computer Applications, 20(2), 43-49.

[8]. Mishra, S. K., Sahoo, B., &Parida, P. P. (2018). Load Balancing in Cloud Computing: A big Picture. Journal of King Saud University-Computer and Information Sciences.

[9]. Shaw, S. B., & Singh, A. K. (2014, September). A survey on scheduling and load balancing techniques in a cloud computing environment. In Computer and Communication Technology (ICCCT), 2014 International Conference on (pp. 87-95). IEEE.

[10]. Madni, S. H. H., Latiff, M. S. A., Abdullahi, M., & Usman, M. J. (2017). Performance comparison of heuristic algorithms for task scheduling in the IaaS cloud computing environment. PloS one, 12(5), e0176321.

[11]. Ray P., Sarddar D. (2018) Load Balancing with Inadequate Machines in Cloud Computing Networks. In: Mandal J., Sinha D. (eds) Social Transformation – Digital Way. CSI 2018. Communications in Computer and Information Science, vol 836. Springer, Singapore

[12]. Sarddar, D., & Ray, P. (2017). A New Effort for Distributed Load Balancing. INTERNATIONAL JOURNAL OF GRID AND DISTRIBUTED COMPUTING, 10(8), 1-9.

[13]. Ranjan Kumar Mondal, Payel Ray, Enakshmi Nandi, Biswajit Biswas, Manas Kumar Sanyal, and Debabrata Sarddar. "Load Balancing of Unbalanced Matrix with Hungarian Method." In International Conference on Computational Intelligence, Communications, and Business Analytics, pp. 256-270. Springer, Singapore, 2017.

[14]. Ranjan Kumar Mondal, Payel Ray, Enakshmi Nandi, Priyajit Sen, and Debabrata Sarddar. "Load balancing of the unbalanced cost matrix in a cloud computing network." In Computer, Communication and Electrical Technology: Proceedings of the International Conference on Advancement of Computer Communication and Electrical Technology (ACCET 2016), West Bengal, India, 21-22 October 2016, p. 81. CRC Press, 2017.