ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse07932020.pdf
https://doi.org/10.30534/ijatcse/2020/07932020

Dual Objective Task Scheduling Algorithm in Cloud Environment

O.S. Abdul Qadir[1], **Dr. G. Ravi** [2]
[1]Research Scholar in Computer Science, Jamal Mohamed College (Autonomous)
(Affiliated to Bharathidasan University)
Tiruchirappalli, Tamil Nadu, India
abdulqadir@jmc.edu
[2]Associate Professor & Head, PG and Research Department of Computer Science
Jamal Mohamed College (Autonomous), (Affiliated to Bharathidasan University),
Tiruchirappalli, Tamil Nadu, India
ravi_govindaraman@yahoo.com

ABSTRACT

Cloud computing takes into account to permit the sharing of resources like networks, servers, storage, applications, and services to achieve access of these resources from any computer in the world through the Internet. The task scheduling algorithms are of NP-hard in nature. The importance of task scheduling is to map the tasks with the appropriate resources for execution. Makespan is the difference in time from start to end of the scheduling. Load balancing is the sharing of workload among the resources. The objective of any scheduling algorithm is to reduce the makespan with proper utilization of resources. In this paper a new algorithm (DOTS) is developed with the realistic dual-objective criteria that minimize the makespan and balance the load across the resources. The distribution of tasks among the resources is also evaluated using the coefficient of variation. The dual objectives are transformed into a single score using weighted sum method. The results clearly indicate that the proposed DOTS technique performs well when compared with seven other scheduling algorithms.

Key words: Cloud Computing, Task Scheduling, Makespan, Load balancing, ETC Matrix and Coefficient of Variation.

1. INTRODUCTION

1.1 Cloud Computing

Cloud computing is a model that share the resources like servers, storage, applications and services in the Internet to provide various on demand services. Cloud computing is one of the fastest growing technology and is applied in everywhere business operations. The services of cloud computing are usually classified as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

1.2 Task Scheduling

Tasks are a form of incoming request. Task Scheduling is the concept of allocating the tasks to the resources based on some principles. The length of scheduling is the time taken to finish all the tasks. Similarly, the schedule length of a particular resource is the time taken to execute all the tasks assigned to it. Different resources will have different schedule length depending on the scheduling techniques used.

1.3 Makespan

Any scheduling algorithm will try to execute all the tasks in the shortest possible time by appropriately mapping with the available resources. Hence every resource will be allocated with a specific set of tasks based on the scheduling policy. The length of execution of a particular resource is the total execution of all the tasks allocated to it. The maximum value in the set of the total execution time of every resource is termed as the makespan. In other words, makespan is the length of the schedule of a resource having the maximum total execution time.

1.4 Load balancing

Load balancing is the process of reassigning the total load of the entire system to individual resources in order to have an equal or near to equal utilization of all the resources. This will lead to the reduction of the response time of the tasks. Load balancing removes the situation where some resources are overloaded in comparison with some other resources which are underloaded. The main goal of load balancing is the effective utilization of resources there by reducing the overall execution time (makespan).

1.5 ETC Matrix

The expected time to compute (ETC) matrix model defines the specification of the execution time of all the tasks across the available resources. For example, a value of [3,4] in the ETC matrix represents the execution time for the 3rd task on the 4th resource. In general, a value of [i, j] in the matrix represents the execution time for the ith task on the jth resource. The table 1 shows a sample ETC matrix. It consists of 17 tasks and 4 resources. The purpose of input is served by the ETC matrix for any scheduling techniques.

Table 1: ETC Matrix

R / T	R1	R2	R3	R4
T1	22	29	14	21
T2	38	54	16	13
T3	36	19	18	29
T4	31	43	28	36
T5	22	30	21	25
T6	9	31	47	38
T7	54	8	22	30
T8	38	52	49	19
T9	16	36	18	5
T10	48	6	35	45
T11	32	38	51	54
T12	45	52	46	15
T13	33	32	37	36
T14	47	11	46	29
T15	52	48	18	26
T16	19	31	28	27
T17	43	15	14	18

1.6 Relative Scheduling Algorithms

To compare and analyze the standard of the proposed DOTS methodology, seven different scheduling algorithms are considered. These algorithms have usually been referred by many of the researchers in their respective proposals. The algorithms include Opportunistic Load Balancing (OLB), Minimum Execution Time (MET), Minimum Completion Time (MCT), Min-Min, Max-Min, Resource Aware Scheduling Algorithm (RASA) [1] and Novel Heuristic Based Task Scheduling (NHBTS) [2]. The aim of any scheduling algorithm is that they mainly focus on the two factors, makespan and resource utilization. The rest of the paper is organized as follows. In Section 2, some of the relevant work scheduling algorithms are discussed. The Section 3 defines the three mathematical model used in the paper. The DOTS methodology is illustrated in Section 4 with the algorithm and flow chart. A detailed working of DOTS technique is illustrated in the experimental analysis of Section 5. In section 6 the results are compared with the algorithms in section 1.6. Finally, the conclusion is given in Section 6.

2. RELATED WORKS

In task scheduling algorithms, the tasks have different execution times from each other on the available resources. Researchers develop a fair scheduling algorithm that avoids imbalance in the length of the resource execution and to attain a minimum makespan. In [3], Meenakshi Sharma and Pankaj Sharma, developed a new algorithm that reduces the response time of the virtual machine. This algorithm has three phases; initialization phase to determine the response time of virtual machine, identifying the appropriate virtual machine and returning this virtual machine to controller. In [4], Santhosh B. and Manjaiah D.H., proposed an algorithm, by modifying Max-min approach. In this approach the average execution time for tasks are calculated and the tasks having just above the average execution time is chosen and mapped to the resource that provides the minimum completion time. In [5], Abhay Kumar Agarwal and Atul Raj, proposed a new algorithm that has a better load balancing when compared with similar algorithms. In [6] Shubham Mittal and Avita Katal has introduced an optimal task scheduling algorithm that adapts the advantages of the various methods currently available to suit the situation. This algorithm affords a method in which the scheduling scheme is automatically chosen from the existing algorithms of Max-min, Min-min, RASA, Improved Max-Min, and Enhanced Max-Min. In [7], Mir Salim Ul Islam and Bhawana Rana, proposed an algorithm based on the length and priority to have improve load balancing. In [8], Mao-Lun Chiang et al., developed a new algorithm that performs a good completion time and attains load balancing. This algorithm works by comparing the sufferage vale, mean sufferage value and the finish time of tasks on the server nodes. In [9], Priyanka Dhurvey and Nagendra Kumar developed a new scheduling technique to reduce the makespan and increase the utilization of resources. The makespan and resource utilization values are compared with existing algorithms and proved to show good results. In [10], Davneet Singh Chawla, Dr. Kanwalvir Singh Dhindsa, proposed a new scheduling technique in which the tasks are allocated based on some calculated priorities to attain load balancing. In [11], Mohit Kumar and S.C. Sharma, developed an algorithm whose goal is to continuously monitor all virtual machines and improve the utilization rate of the cloud resource, thus increasing the application's execution speed. In [2], O.S. Abdul Qadir and Dr. G. Ravi, designed a new scheduling algorithm called NHBTS that had a better reduction on makespan. The NHBTS algorithm executed the task by mapping it to, either to its best resource or to the next best resource.

3. MATHEMATICAL MODELS

The DOTS technique makes use of the following three mathematical models, Coefficient of Variation (CV), Multi-Criteria Decision Making and Weighted Sum Method.

3.1 Coefficient of Variation (CV)

Standard deviation is the measure of deviation of the data in any distribution from the mean which is used as the reference point. The Coefficient of Variation is the percentage deviation of the data with respect to the mean. The mean is the sum of data considered divided by the number of observations. CV is used to check how the distribution of data around the mean value for a set of concerned situations. It helps to compare the consistency of two or more collections of data. A CV value of 0 reflects a perfect distribution. The acceptable CV differs based on the research sector. As per Dr. Ehsan Ebrahimi [12], basically CV is very good for any value less than or equal to 10. The CV value between 10 and 20 is considered to be good. Even the value of CV up to 30 is acceptable. A CV value above this results in the poor distribution of data.

CV is calculated using the formula:

$$CV = \frac{\text{Standard Deviation } (S)}{\text{Mean}} \text{ X } 100 \quad (1)$$

$$S = \sqrt{\frac{\sum(\text{xi} - \bar{\text{x}})^2}{n}} \quad (2)$$

The proposed DOTS methodology applies the concept of CV on the data set containing the execution length of the two extreme resources i.e. the resource having the maximum overall execution time and the resource having the minimum overall execution time.

3.2 Multi-Criteria Decision Making

The multiple objectives in a problem can be solved by applying Multiple-criteria decision-making (MCDM) or multiple-criteria decision analysis (MCDA) [13]. It acts as an interface if there exist multiple criteria in taking a decision. The objectives may be conflicting in nature. The MCDM evaluates the multiple criteria and produce the result. There are many MCDM methods [14]. The most widely used and the simplest is the Weighted Sum Method or Model (WSM).

3.3 Weighted Sum Method

Weighted Sum Method [15] is a multi-criterion decision-making method which is applied when there are multiple alternatives with multiple objectives and a decision has to be taken. It is used to transform multiple objectives into a single objective score. In this method the objectives are assigned weights depending on the role of their importance. The equation of WSM is

$$f = (w1 \text{ X } f1) + (w2 \text{ X } f2) \quad (3)$$

Where, f is the transformed single objective.

$f1$ and $f2$ are the two different objectives and

$w1$ and $w2$ are corresponding weights to $f1$ and $f2$

The proposed DOTS methodology applies the concept of Weighted Sum Model MCDM to arrive at a decision of selecting a scheduling technique as each scheduling algorithm has a considerable variation in both the criteria: the makespan and load balancing.

4. PROPOSED METHODOLOGY

A task scheduling is effectual if the available resources are employed in best possible ways. The best uses of resources reduce the makespan and also increases its utilization. The DOTS methodology applies the technique for the effective utilization of resources.

The DOTS methodology aims to reduce the makespan by simultaneously improving the utilization of resources.

Objectives

The methodology has dual objectives.

(i) To achieve minimum makespan i.e. to reduce the overall completion time of all tasks.
(ii) To achieve better utilization of resources.

The two objectives can be evaluated into a mono objective using the Weighted Sum Method. Weighted Sum Method (WSM) is a multi-criterion decision-making method in which there will be multiple alternatives and have to determine the best alternative based on multiple criteria.

Since the two objectives are inverse in nature i.e. a minimum objective for makespan and maximum objective for load balancing,

Makespan (MS) = min f(X)
Resource Utilization = max f(X)

Then it is conventional to perform a scaling operation on makespan and resource utilization to convert them to their respective dimensionless forms of f1 and f2 [16]. This operation maximizes the objectives i.e. a higher value is desired after scaling. The scaling operation performed on makespan for any scheduling technique produced by different scheduling techniques is defined below:

$$f = \left\{ \frac{\text{minimum (makespan values produced by different techniques)}}{\text{makespan of a scheduling technique}} \text{ X } 100 \right\} \quad (4)$$

The resource utilization value for any resource is

$$f2 = \frac{\text{Total Execution Time of the Resource}}{\text{Execution Time of the Makespan Resource}} \text{ X } 100 \quad (5)$$

The resource utilization value for any scheduling technique is calculated as:

$$f3 = \frac{\text{sum of Resource Utilization}}{\text{Number of Resources}} \quad (6)$$

The assignment of weight to objectives in WSM is a voluntary choice. Here the weight for the objectives i.e. makespan and load balancing are assigned the values of 60% and 40%

respectively, since the study mainly focusses on makespan. Now apply the WSM (multiplying $f1$ and $f2$ with their respective weights and adding them) to give the resultant single objective value for every scheduling technique.

$$f = \left(\frac{60}{100} \text{ X } f1\right) + \left(\frac{40}{100} \text{ X } f1\right) \tag{7}$$

Since both the objectives has been normalized using a scaling operation a higher value for f is desired. This is termed as Weighted Sum Score (WSS).

Coefficient of variation is the percentage variation of data from the mean. The DOTS methodology determines the CV between the resources of highly loaded and lightly loaded i.e. the resources having maximum and minimum length of execution. Since we calculate the CV for only the two extreme resources the CV value is fixed at 10. A CV value of 0 reflects a perfect distribution or the optimal distribution of tasks among resources with equal load balancing. Then resources with the maximum and minimum length of execution can deviate by 10 percentages from the average of time taken between them. For example, if the mean of the above said resources happens to be 50 then the heavily loaded resource value will range from 50 to 55 and lightly loaded resource value will range from 45 to 50.

The use of CV in the DOTS methodology emphasizes that the execution length of the remaining resources falls between these values. It also avoids unnecessary movement of tasks between resources once a fair scheduling is arrived. Further relocation of tasks among the resources may not yield a better case or may only have a minimal improvement at the cost severe timing.

Algorithm

A formal presentation of the proposed DOTS methodology is given below:

```
1.  From the ETC Matrix allocate the tasks to its best resource i.e. the resource having the minimum execution time for the respective tasks. The tasks are allocated in the order they arrive.
2.  Calculate the makespan (MS) and the total execution time (TET) for the maximum and minimum resource.
3.  Find the coefficient of variation (CV) for TET.  If CV is less than 10 (the resource utilization is good and the load is almost evenly balanced) then Exit from the algorithm.
4.  For every Resource Set Flag[R]to 0.
5.  Repeat steps 6 to 13
    until ( (∀ R, Flag[R] = 0) OR (CV > 15))
    6. Identify the resources having the highest total execution time (RH) and lowest total execution time (RL) (Flag [RH] = 0 and Flag[RL] = 0.
    7.  Make the Set S empty.
    8.  For all the tasks allocated to RH,
        Find the range (RAi) between RH and RL.
        Add all tasks to set S.
        9.  Select the task Ti from the set S, having minimum RAi value.
        10. Calculate Total Execution Time of RL (TETRL)
        11. If (TETRL is less than MS) Then
            Ti is removed from RH and assigned to RL.
            Calculate MS.
            Calculate CV.
            If (CV <=10)
            Exit
            Identify (RH) and (RL)
        12. Delete this task from S.
        13. Repeat steps 9 to 12 until S becomes empty
            OR RH changes to another resource.
14. If RH changes to another resource set Flag[RH] = 1
    Else exit from the algorithm.
```

Working Principle

To formulate an algorithm that generates a solution for better makespan and resource utilization, the proposed work starts with an early scheduling by assigning the tasks to the resources that execute them faster.

Calculate the CV for the total execution time of every resource. If the CV is less than or equal to 15 then the distribution of tasks among the resources are good. Further movement of tasks may not lead to a better solution.

Determine the resources having the maximum and minimum values of total execution time from the set of R resources. The corresponding resources are marked as RH and RL. Identify all the tasks that are mapped to RH. Theses task are assigned to a set S. For every task in set S, the difference in execution time between RH and RL is determined. From the set S the task Ti having low valued range is chosen. For the makespan to be minimized, it is calculated by removing the allocation of Ti is from RH and allocating to RL. If the calculated makespan is less than the previous one then this task relocation is performed.

With this new change of execution time of resources, the CV value is again determined. If the value falls below 15, the scheduling process stops. If either the calculation of reallocation does not yield a decrease in makespan value, or if the corresponding RH resource remains the same after the reallocation of the task Ti, the process is continued by taking the next task from the set S in their nondecreasing order.

If RH changes to another resource, all the above said are repeated by identifying the set of tasks assigned to RH. The RH is flagged before changing it to another resource, as it should not be considered again to avoid the cyclic infinite loop.

The DOTS methodology will exit if anyone of the following situation occurs:

If the CV value is less than or equal to10

If there is no improvement in makespan by relocating the tasks from the maximum length resource to minimum length resource and

If the makespan resource repeats again.

The figure 1 represents the flowchart of the algorithm presented for the DOTS methodology.

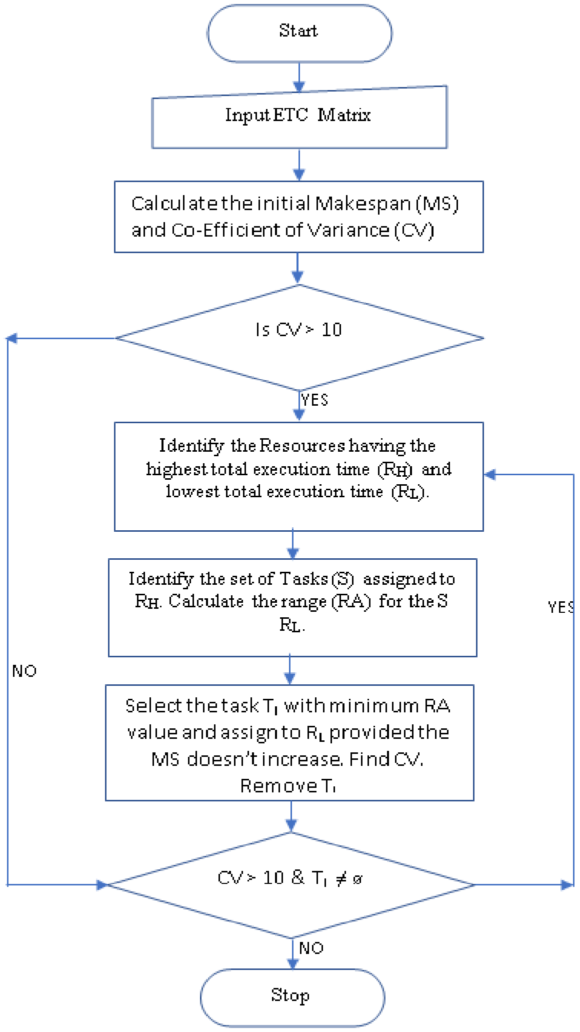

Figure 1: Flowchart of the proposed DOTS methodology

5. EXPERIMENTAL ANALYSIS

Consider the ETC matrix of table 1. The matrix has 4 resources and 17 tasks.

Initially the arrived makespan is 113 and the respective resource is R3. The resource with minimal length of execution is R4 and its value is 52. The initial calculation of CV stands at 36.97.

In the first Iteration of the loop of line 8, the tasks associated with R3 are T1, T3, T4, T5, T15 and T17. Their difference in execution time with R4 is 7, 11, 8, 4, 8 and 4. So the minimum range is 4 and the corresponding task is T5. The allocating of the task T5 to R4 from R3 effects in the change of makespan to 92, a whopping 20% reduction in the makespan value. The makespan resource remained the same. Now the resource with minimal length of execution is R2 and its value is 57. Now the CV has also reduced to 23.49, making better utilization of resources.

In the second Iteration, since there is no change in the makespan resource the task T5 from the above said set of tasks is discarded and for the remaining tasks the difference in execution time with R2 is calculated as 15, 1, 15, 30 and 1. Hence the task T3 is allocated to R2. It further reduces the makespan to 77 from 92 which is nearly 32% reduction for the makespan value. Now the makespan resource is R4 and R1 with a value of 60 is the resource with minimal length of execution. Again, the CV also reduced to 12.41, making the utilization of resources more consistent.

In the third Iteration, as there is a new makespan resource R4, its associated tasks are determined. The set of tasks include T2, T5, T8, T9 and T12. Their difference in execution time with R1 is 25, 3, 19, 11 and 30. In this case, the task T5 is rejected as its selection increases the makespan. Next task T9 is tried and its allocation to R1 reduces the makespan to 76. As the CV is 2.70 the resources are almost evenly balanced with their overall execution time.

Using the equation 4 the scaling operation is performed on the makespan and the normalized value is obtained

The Weighted Sum Score is also calculated using equation 7.

The final overall execution time for the 4 resources stands at 76, 76, 74 and 72 and makespan value is 76.

The results for the input ETC matrix of table 1 are summarized below:

Makespan value	:	76
Normalized Makespan Value	:	100
Resource utilization	:	98 %
Weighted Sum Score (WSS)	:	99.20

6. RESULTS AND DISCUSSION

The DOTS technique is compared and analyzed with the scheduling techniques mentioned in Section I. The parameters considered for analysis and discussion include the Makespan, Resource Utilization and Weighted Sum Score. The makespan value produced by DOTS, NHBTS, OLB, MET, MCT, Min-Min, Max-Min and RASA techniques are 76, 79, 140, 113, 90, 82, 118 and 101. These values are tabulated in table 2.

The figure 2 displays the performance of the various scheduling techniques for the data presented in table 2 with respect to the makespan objective.

Table 2: Makespan of Various Scheduling Algorithms

RESULT	
Scheduling Techniques	**Makespan**
DOTS	76
NHBTS	79
OLB	140
MET	113
MCT	90
Min-Min	82
Max-Min	118
RASA	101

The graphical representation (figure 2) clearly indicates the DOTS further reduces the makespan produced by NHBTS and other scheduling techniques.

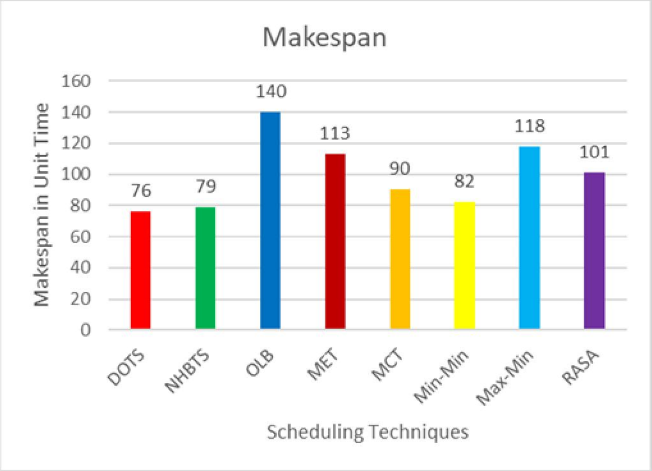

Figure 2: Performance of Various Scheduling Techniques

The Resource Utilization of the above said techniques are tabulated in the table 3. The resource utilization of the scheduling techniques was calculated using the equation 6. The values obtained for the DOTS, NHBTS, OLB, MET, MCT, Min-Min, Max-Min and RASA are 98, 92, 95, 62, 88, 87, 98 and 87.

Table 3: Resource Utilization of Various Scheduling Algorithms

RESULT	
Scheduling Techniques	**Resource Utilization**
DOTS	98
NHBTS	92
OLB	95
MET	62
MCT	88
Min-Min	87
Max-Min	98
RASA	87

The figure 3 displays the graphical representation of the data provided in table 3. From the pictorial representation it is understood that the best load balancing is provided by the DOTS as well as Max-Min scheduling. The poor load balancing is produced by MET.

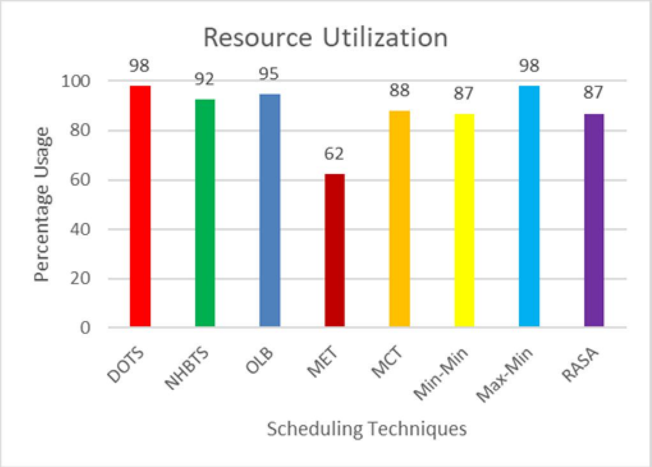

Figure 3: Resource Utilization of Various Scheduling Techniques

The tables 2 and 3 displayed the results of the both objectives makespan and resource utilization separately. As indicated previously the two objectives are combined by assigning appropriate weights to them. For this the two objectives should not be conflict in nature. Hence, the makespan values are normalized to a maximum function by applying the scaling operation given in the equation 4. The scaled makespan values for DOTS, NHBTS, OLB, MET, MCT, Min-Min, Max-Min and RASA techniques are 100, 96, 54, 67, 84, 93, 64 and 75. These values are tabulated in table 4. Now for both the objectives a maximum value will result in the better solution.

Table 4: Normalized Makespan of Various Scheduling Algorithms

RESULT	
Scheduling Techniques	**Normalized Makespan**
DOTS	100
NHBTS	96
OLB	54
MET	67
MCT	84
Min-Min	93
Max-Min	64
RASA	75

Finally, the Weighted Sum Method defined in the equation 3, is applied on both the objectives makespan and resource utilization and transformed into a single value. The results are displayed in the table 5. The calculated WSM scores for the

DOTS, NHBTS, OLB, MET, MCT, Min-Min, Max-Min and RASA techniques are 99.20, 94.40, 70.40, 65.00, 85.60, 90.60, 77.60 and 79.80.

Table 5: Weighted Sum Score of Various Scheduling Algorithms

Weighted Sum Method			
Scheduling Techniques	**NMS**	**RU**	**WSM Score 60% NMS + 40 % RU**
DOTS	100	98	99.20
NHBTS	96	92	94.40
OLB	54	95	70.40
MET	67	62	65.00
MCT	84	88	85.60
Min-Min	93	87	90.60
Max-Min	64	98	77.60
RASA	75	87	79.80

The graphical representation for the table 5 is given in figure 4. The DOTS technique has the maximum value followed by NHBTS and Min-Min scheduling techniques.

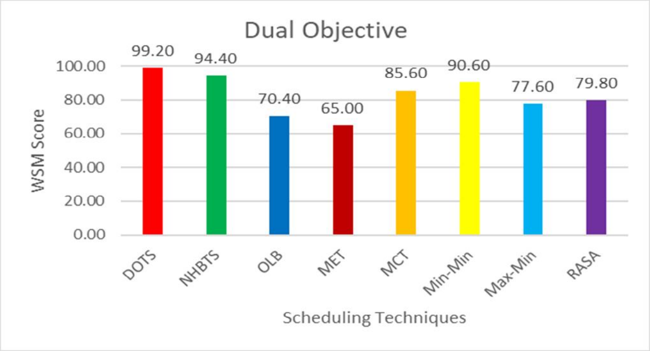

Figure 4: Weighted Sum Score of Various Scheduling Techniques

7. CONCLUSION

This research article proposed a new task scheduling (DOTS) by considering dual objectives, the makespan and load balancing. The Min-Min scheduling technique is good for producing a decent makespan but performed poorly in resource utilization. The Max-Min scheduling technique is good while considering the resource utilization factor. The DOTS performed exceptionally well on both the objectives independently. These objectives are combined using a famous MCDM technique called Weighted Sum Method, to produce resultant score by assigning appropriate weights. The DOTS technique has the highest score of 99.20 followed by NHBTS and Min-Min having a score of 94.40 and 90.60. The results clearly indicate that the proposed DOTS technique performs well when compared with the standard algorithms.

REFERENCES

1. O.S. Abdul Qadir and Dr. G. Ravi. **A Survey on Task Scheduling Algorithms in Cloud Computing**, *International Journal of Innovations in Engineering and Technology (IJIET)*, Volume 15 Issue 4 March 2020
2. O.S. Abdul Qadir and Dr. G. Ravi. **A Novel Heuristic Based Task Scheduling Algorithm to Minimize the Makespan in Cloud Environment**, *International Journal of Advanced Science and Technology*, Vol. 29, No. 5, (2020), pp. 3737- 3746.
3. Prof. Meenakshi Sharma and Pankaj Sharma". "**Performance Evaluation of Adaptive Virtual Machine Load Balancing Algorithm**," *(IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 3, No.2, 2012. https://doi.org/10.14569/IJACSA.2012.030215
4. Santhosh B. and Manjaiah D.H. "**An Improved Task Scheduling Algorithm based on Max-min for Cloud Computing**", *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, vol. 2, no. 2, pp. 84-88, May 2014.
5. Abhay Kumar Agarwal and Atul Raj. "**A New Static Load Balancing Algorithm in Cloud Computing**," *International Journal of Computer Applications*, December 2015. https://doi.org/10.5120/ijca2015907285
6. Shubham Mittal and Avita Kata. **An Optimized Task Scheduling Algorithm in Cloud Computing**, *2016 IEEE 6th International Conference on Advanced Computing*.
7. Mir Salim Ul Islam and Bhawana Rana. **Task Scheduling in Cloud Computing," International Journal of Advance Research**, Ideas and Innovations in Technology, 2017.
8. Mao-Lun Chiang, Hui-Ching Hsieh, Wen-Chung Tsai and Ming-Ching Ke. "**An Improved Task Scheduling and Load Balancing Algorithm under the Heterogeneous Cloud Computing Network**," *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST 2017)*. https://doi.org/10.1109/ICAwST.2017.8256465
9. Priyanka Dhurvey and Nagendra Kumar. "**Resource Optimization Based Scheduling Algorithm in Cloud Environment**," *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 5, Issue 5, May 2017.
10. Davneet Singh Chawla, Dr. Kanwalvir Singh Dhindsa. "**A Load Balancing Based Improved Task Scheduling Algorithm in Cloud Computing**", *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*," Volume 5 Issue IX, September 2017.
11. Mohit Kumar and S.C. Sharma. "**Load balancing algorithm to minimize the makespan time in cloud environment**," ISSN 1 746-7233, England, *UK World*

Journal of Modelling and Simulation, Vol. 14 (2018) No. 4, pp. 276-288.

12. https://www.researchgate.net/profile/Ehsan_Ebrahimi5
13. https://nptel.ac.in/content/storage2/nptel_data3/html/mhrd/ict/text/110107115/lec1.pdf
14. https://en.wikipedia.org/wiki/Multiple-criteria_decision_analysis.
15. Aarushi Singh and Sanjay Kumar Malik**, Major MCDM Techniques and their application-A Review**, IOSR Journal of Engineering (IOSRJEN), Vol. 04, Issue 05 (May. 2014).
16. www.researchgate.net/post/how_to_put_Weight_parameter_for_a_Multi_Objective_Function.