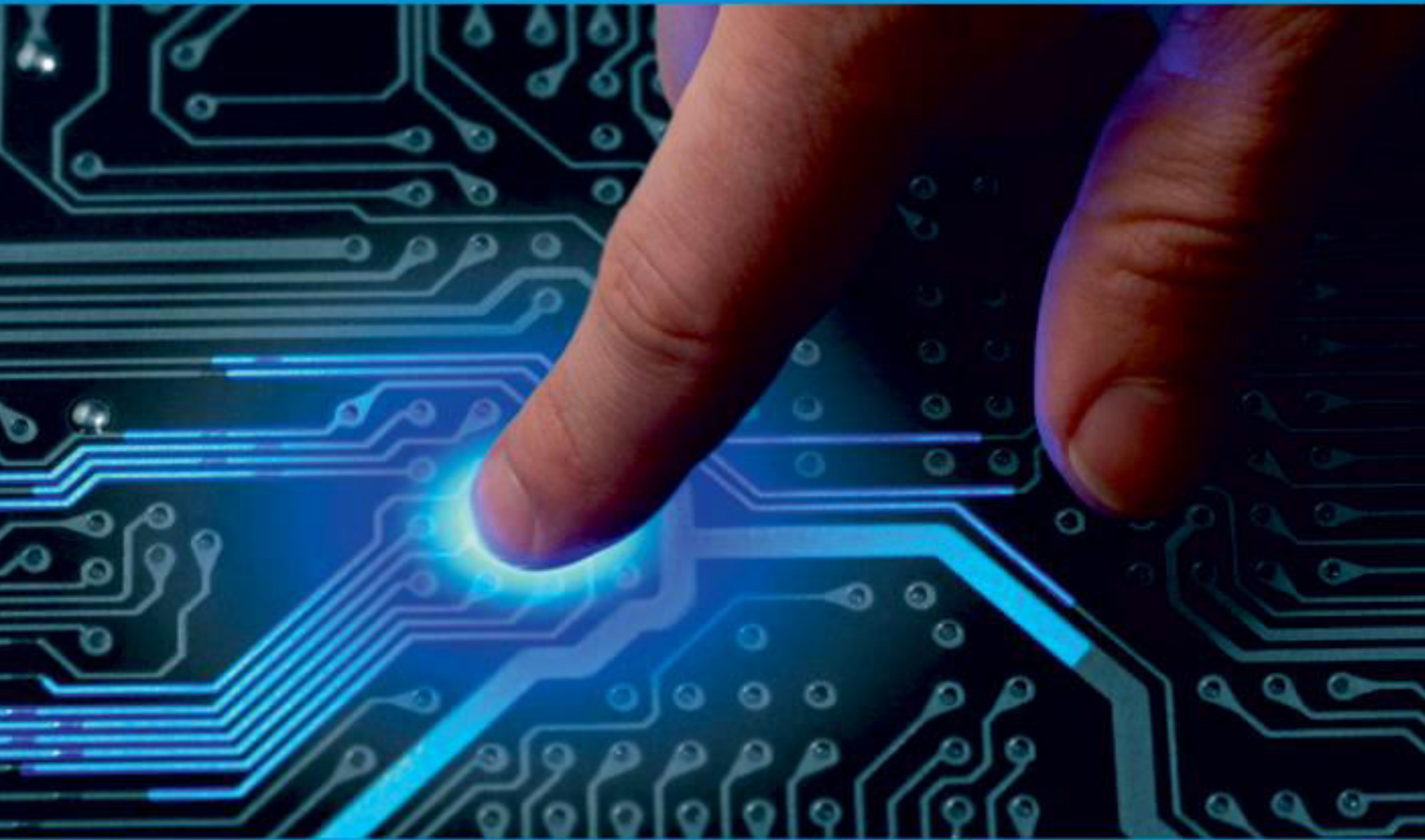
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 2, February 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

Credit Score Tasks Scheduling Algorithm for Mapping a Set of Independent Grid Environment

DR.R.GEETHA, LOKESHWARASARAVANAN P

Associate Professor, Department of Computer Applications (MCA), K.S.R. College of Engineering (Autonomous), Tiruchengode, India

Department of Computer Applications (MCA), K.S.R. College of Engineering (Autonomous), Tiruchengode, India

ABSTRACT: Heterogeneous grid environments are well suited to solve the scientific and engineering applications that require large computational demands. The problem of optimally mapping, that is selecting the appropriate resource and scheduling the tasks in an order onto the resources of a distributed heterogeneous grid environment has been shown, in general to be a NP-Complete problem. NP-Complete problem requires the development of heuristic techniques to identify the best possible solution. In this paper, a new heuristic scheduling algorithm called Credit Score Tasks Scheduling Algorithm (CSTSA) is proposed. It aims to maximizing the resource utilization and minimizing the makespan. The new strategy of Credit Score Tasks Scheduling Algorithm is to identify the appropriate resource and to find the order in which the set of tasks to be mapped to the selected resource. The order in which the tasks to be mapped is identified based on the Credit Score of the task. Experimental results show that the proposed Credit Score Tasks Scheduling Algorithm outperforms the Min-min heuristic scheduling algorithm in terms of resource utilization and makespan.

KEYWORDS*:* Computational Grid, Grid scheduling, Heuristic, Makespan .

I. INTRODUCTION

Grid is an infrastructure and builds various functions and it helps to involve integrated and collaborative use of various technologies like computers, networks, database and scientific instrument which are owned and managed by multiple organizations. It is globally distributed and consists of heterogeneous and loosely coupled data and resources. Grid is the dynamic environment, so it has the ability to change the resource frequently. Middleware is one of the important strategies in grid computing which divides program into number of pieces among several computers [2,4].

Computational grid is defined as the distributed infrastructure that appears to an end user who divides the job among individual machines and run the calculations in parallel and returns the results to the original machine. Scheduling has direct impact on performance of grid application. One important challenge in task scheduling is to allocate the optimal resources to the job in order to minimize the task computation time. Several heuristic task scheduling

algorithms have been developed for task scheduling. Dynamically tasks are entered and scheduler must allocate the resource effectively but it is a tedious process [9,10,11].

Opportunistic Load Balancing (OLB) algorithm assigns the job in an arbitrary order based on the shortest schedule to the processor without considering the ETC of that processor and also it assigns task in arbitrary order to the next available machine regardless of its expected execution time of the machine [5,6].

Minimum Execution Time (MET) algorithm based on the minimum execution time of the task it is assigned to the machine without considering the resource availability of that machine and also it assigns job to the machine in arbitrary order regardless of the current load on the processor in order to improve the performance and faster execution [1,7].

Minimum Completion Time (MCT) algorithm with the earliest completion time and minimum expected completion time of the job each task is assigned arbitrarily to the processor. The ETC of the job j on the processor p is added to the p's current schedule length which is the completion time of the job j on the processor p [1,7].

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165|

|| Volume 11, Issue 2, February 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1102011 |

Min-min algorithm calculates the expected completion time of each task with all the processors then it assigns the task to the resource with the minimum expected completion time [5,6,8].

Max-min algorithm is similar to the Min-min algorithm; first it calculates the minimum completion time of the entire task and selects the machine with the minimum expected completion time. Then allocates the job with maximum minimum completion time is assigned to the corresponding processor [5,6].

Suffrage Heuristic works as follows, the first step is to calculate the minimum and second minimum completion time for each task, the difference between the two values is defined as the suffrage value. The second step is the task with higher suffrage value is assigned to the machine with minimum completion time. The mapping of task to the machine suffers in terms of ECT according to the suffrage value [3].

II. MATERIALS AND METHODS

A. *Problem Definition*

An application consists of ‘n’ independent meta-task and a set of ‘m’ heterogeneous resources. The problem of mapping the ‘n’ meta-tasks to the set of ‘m’ heterogeneous resources in a grid computing environment is an NP-Complete problem [9, 12]. This paper proposes a new algorithm Credit Score Tasks Scheduling Algorithm for solving the scheduling problem in a grid computing environment.

The order in which the tasks to be mapped to a set of resources determines the efficient scheduling which results in the reduced makespan. The proposed new algorithm Credit Score Tasks Scheduling Algorithm provides an ordered set of tasks, which specifies the order in which the tasks to be scheduled to the set of ‘m’ resources. The proposed Credit Score Tasks Scheduling Algorithm provides reduced makespan than the existing Min-min heuristic scheduling algorithm.

B. *Proposed CSTSA Algorithm*

The mapping of the ‘n’ meta-tasks to the set of ‘m’ heterogeneous resources is made based on the following assumptions [1,7]:

- A set of independent, non-communicating tasks called meta-tasks is being mapped.
- Heuristics originate a static mapping.
- Each resource executes a single independent task at a time.
- The number of tasks to be scheduled and the number of heterogeneous resources in the grid computing environment are static and known a priori.
- ETC (Expected Time to Compute) matrix represents the expected execution time of a task on a resource.
- ETC matrix of size n*m, where ‘n’ represents the number of meta-tasks and ‘m’ represents the number of heterogeneous resources.
- ET_{ij}- represents the expected execution time of a task t_i on a resource r_j.
- Task set is represented as $T=\{T_1,T_2,T_3......T_n\}$
- Resource set is represented as $R=\{R_1,R_2,R_3......R_m\}$
- The accurate estimate of the expected execution time for each task on each resource is contained within an ETC matrix
- TCT_{ij} –expected completion time of task Ti on resource R_j
- RT_j-ready time of resource R_j
- Makespan = $\max(TCT_{ij})$
- ETC matrix is computed by the formula

  $$ETC_{ij} = \frac{Task\ length_i}{Power_j}$$

  where Tasklength$_i$ represents the length of the task T_i in MI and power$_j$ represents the computing power of the resource R_j in MIPS
- The ready time of the resource R_j, is the time at which the resource R_j completes the execution of the previously assigned tasks and is defined as

$$RT_{j=\sum_{i=1}^{n} ET_{ij}}$$

The proposed Credit Score Tasks Scheduling Algorithm considers two criteria for scheduling the meta-tasks onto the resources. The two criteria considered for efficient scheduling are,

1) Task Execution Time Credit
2) Unique Value Credit for the meta-task

The proposed algorithm schedules the task with the highest credit score value to the resources that provides the minimum completion time of the task.

C. Task Execution Time Credit

The steps involved in calculating the task execution time credit for a meta-task is listed below:

1. From the ETC matrix, the maximum execution time of a task is identified.

$$MAXET=\max(ET_{ij}),\ 1\leq i\leq n,\ 1\leq j\leq m$$

2. Credits are assigned to each task using the following formula:

$$CV1 = \frac{MAXET}{2}$$

$$CV2 = \frac{MAXET}{3}$$

$$CV3 = CV1 + CV2$$

$$CV4 = CV2 + CV3$$

The algorithm for finding the Task Execution Time Credit Score is shown below:

Find the maximum value in the ETC matix

```
    Assign MAXET=0
    for i=1 to n do
      for j=1 to m do
        if(ET_ij> MAXET)
          MAXET=ET_ij
    end for
    end for
```

Compute the following

$$CV1 = \frac{MAXET}{2}$$

$$CV2 = \frac{MAXET}{3}$$

$$CV3 = CV1 + CV2$$

$$CV4 = CV2 + CV3$$

For all submitted tasks T_i, in the task set T, find the maximum execution time of each task,

if $MAXET_i < CV_1$

$CS_i = 4$

else if $CV_1 \leq MAXET_i \leq CV_3$

$CS_i = 3$

else if $CV_3 \leq MAXET_i \leq CV_4$

$CS_i = 2$

else

$CS_i = 1$

end if

end for

D. Unique Value Credit for Task

The unique value is an important criterion for scheduling meta-task onto the heterogeneous resources in a grid environment. A unique value is assigned to each task. The proposed algorithm schedules the task to the resources based on the total credit score of the task.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165|

|| Volume 11, Issue 2, February 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1102011 |

Algorithm for finding the unique credit value is shown below,

```
for all submitted tasks in the task set T,
    Assign unique value UVi for each task ti
    Find out the task with highest unique value
Choose denominator value, dv
for each tasks ti in the task set T,
```

$$Compute\ tUVC_i = \frac{UV_i}{dv}$$

$Compute\ tUVC_i = \frac{UV_i}{dv}$ $Compute\ tUVC_i = \frac{UV_i}{dv}$

```
end for
end for
```

The denominator value dv is determined as shown below:

If the highest unique value given to a task is a two digit number, then dv=100. If the highest unique value given to a task is a three digit number, then dv=1000 and so on.

E. An Illustrative Example

Consider the following example for a grid system with ten tasks and three resources. The ETC matrix is given in Table1.

Table1 ETC Matrix

Task	R_1	R_2	R_3
T_1	9.62	10.42	8.33
T_2	10.58	11.46	9.17
T_3	10.77	11.67	9.33
T_4	11.15	12.08	9.67
T_5	12.5	13.54	10.83
T_6	13.27	14.38	11.5
T_7	14.42	15.63	12.5
T_8	15	16.25	13
T_9	15.96	17.29	13.83
T_{10}	18.27	19.79	15.83

The maximum execution time in the given ETC matrix is

MAXET=19.9
CV1=9.9
CV2=6.6
CV3=16.5
CV4=23.1

Credit Score (CS_i) for each task t_i is computed using Algorithm1 and the result is shown in Table2.

Table 2 Credit Score for each Task

Task	CS
T_1	3
T_2	3
T_3	3
T_4	3
T_5	3
T_6	3
T_7	3
T_8	3
T_9	2
T_{10}	2

A Unique Value (UV) for each task is assigned in random in the range 1 to 10. Unique Value Credit (UVC) for each task is computed using the Algorithm 2 and is shown in Table3.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165|

|| Volume 11, Issue 2, February 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1102011 |

Table 3 Unique Value Credit for each Task

Task	CS	UV	UVC
T_1	3	1	0.01
T_2	3	7	0.07
T_3	3	9	0.09
T_4	3	2	0.02
T_5	3	6	0.06
T_6	3	10	0.1
T_7	3	3	0.03
T_8	3	8	0.08
T_9	2	5	0.05
T_{10}	2	4	0.04

The total credit score for each task t_i is computed using the formula,

$$TCS_i = CS_i * UVC_i$$

and the result is shown in Table4.

Table4 Total Credit Score for each Task

Task	CS	UV	UVC	TCS
T_1	3	1	0.01	0.03
T_2	3	7	0.07	0.21
T_3	3	9	0.09	0.27
T_4	3	2	0.02	0.06
T_5	3	6	0.06	0.18
T_6	3	10	0.1	0.3
T_7	3	3	0.03	0.09
T_8	3	8	0.08	0.24
T_9	2	5	0.05	0.1
T_{10}	2	4	0.04	0.08

The tasks to be scheduled are ordered in the Credit Score Set ‘CSS’ in the descending order of TCS_i.

$$CSS=\{T_6,T_3,T_8,T_2,T_5,T_9,T_7,T_{10},T_4,T_1\}$$

Now, the tasks are scheduled to the resource with minimum completion time. The makespan is 43.96 sec.
The order in which the tasks are scheduled, and the makespan obtained for Min-min algorithm and the proposed Credit Score Tasks Scheduling Algorithm is shown in Table5.

Table 5 A Comparisons between Min-min Algorithm and Credit Score Tasks Scheduling Algorithm in makespan and task schedule order.

Algorithm	R1	R2	R3	Makespan
Min-min	T2,T5, T8	T3,T6, T9	T1,T4, T7, T10	46.33
CSTSA	T3,T5, T10	T8,T9, T1	T6,T2, T7, T4	43.96

F. Credit Score Tasks Scheduling Algorithm (CSTSA):

For all submitted tasks in the task set T
    Calculate Task Execution time credit using Algorithm 1
    Calculate Task unique value credit using Algorithm 2
For each task t_i
    Compute $TCS_i = CS_i*UVC_i$
Order the tasks in the credit score set CSS in the descending order of TCS_i .
For all tasks T_i in the credit score set CSS
    For all resources R_j
        Compute $TCT_{ij}=ET_{ij}+RT_j$
end for
end for

```
Do until all tasks in CSS are mapped
   for each task in CSS find the earliest completion time and the resource that obtains it.
          Find the task t_k with the minimum earliest completion time.
              Assign task t_k to the resource R_j that gives the earliest completion time
              Delete task t_k from CSS
              Update RT_j
              Update TCT_ij for all i
end for
end do
       Compute makespan = max(TCT_ij) for all i, j
```

III. RESULTS AND DISCUSSION

This section presents the experimental results computed for the benchmark model by Braun et al [1,4,7].

A. Benchmark Descriptions

To evaluate the proposed algorithm, the benchmark model instances are divided into twelve different types of ETC matrices. The size of the ETC matrix is 512*16, where 512 represent the number of tasks and 16 represents the number of resources. Twelve combinations of ETC matrices were based on the three metrics: Task heterogeneity, Resource heterogeneity, and Consistency. For each twelve different type of ETC matrix, the results were averaged over 100 different ETC matrices of the same type. The benchmark instances are labelled as u-x-yyzz.k where,

u-uniform distribution in generating ETC matrices

x-consistency(c-consistent,i-inconsistent,

s-semi-consistent or partially consistent)

An ETC matrix is said to be consistent if a machine m_j executes any task t_i faster than resource r_k, then resource r_j executes all tasks faster than resource r_k.

An ETC matrix is said to be inconsistent if a resource r_j executes some tasks faster and some tasks slower than resource r_k.

Semi-consistent ETC matrices are the matrices that includes a consistent sub-matrix.

Task heterogeneity is the amount of variation in the execution time of tasks in the metatask for a given resource.

yy-task heterogeneity(hi-high, lo-low)

Resource heterogeneity is the amount of variation in the execution time of a given task among all the resources.

zz- Resource heterogeneity(hi-high, lo-low)

Twelve combinations of ETC matrices comprises three groups of four instances each. The first, second and third group corresponds to consistent, inconsistent and Semi-consistent ETC matrices each of them having high and low combinations of task and resource heterogeneity.

B. Evaluation Parameters

Makespan

Makespan is the important optimization criteria for grid scheduling. Makespan is calculated as makespan=max(TCT_{ij})

Table 1 shows the 12 different types of instances in the first column, the makespan value obtained by Min-min in the second column, CSTSA in the third column. Graphical representation of Table 1 in Figure shows that the CSTSA provides better makespan than Min-min Heuristic Scheduling Algorithm.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165|

|| Volume 11, Issue 2, February 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1102011 |

Table 1: Comparison based on makespan (in sec)

Instances	Min-min	CSTSA
u-c-hihi-0	8298107	6801626
u-c-hilo-0	79940.04	65523.65
u-c-lohi-0	267044.9	218886
u-c-lolo-0	2600.802	2131.773
u-ic-hihi-0	3565661	2922630
u-ic-hilo-0	32412.49	26567.22
u-ic-lohi-0	125061.7	102508.1
u-ic-lolo-0	1062.335	870.7535
u-s-hihi-0	4602970	3772870
u-s-hilo-0	44979.51	36867.91
u-s-lohi-0	169090.7	138596.9
u-s-lolo-0	1586.498	1300.389

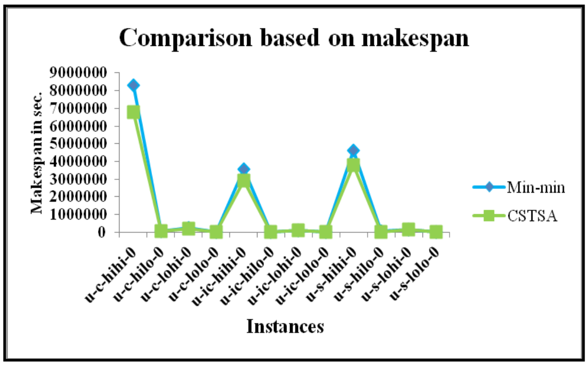


Figure 1: Comparison based on makespan

Table 2, 3, 4, 5 show the comparison of the makespan values obtained by Min-min and CSTSA in all the four instances which comprises High Task High Resource, High Task Low Resource, Low Task High Resource, Low Task Low Resource. The four instances are represented for consistent, inconsistent, semi-consistent or partially consistent heterogeneous computing systems. Figure 2, 3, 4, 5 shows the graphical representation of all the four instances for three different consistencies.

Table 2: Comparison based on makespan (in sec)

Instances	Min-min	CSTSA
u-c-hihi-0	8298107	6801626
u-ic-hihi-0	3565661	2922630
u-s-hihi-0	4602970	3772870

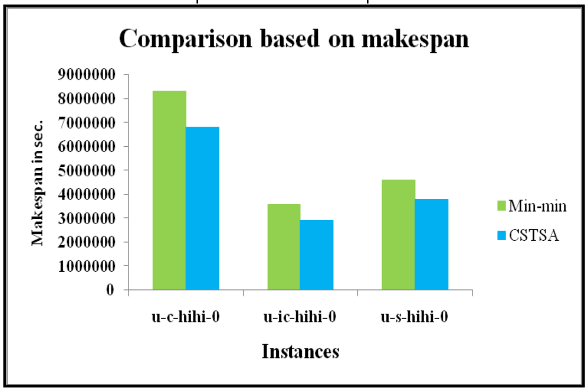


Figure 2: Comparison based on makespan for High

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165|

|| Volume 11, Issue 2, February 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1102011 |

Task High Resource Heterogeneity

Table 3: Comparison based on makespan (in sec)

Instances	Min-min	CSTSA
u-c-hilo-0	79940.04	65523.65
u-ic-hilo-0	32412.49	26567.22
u-s-hilo-0	44979.51	36867.91

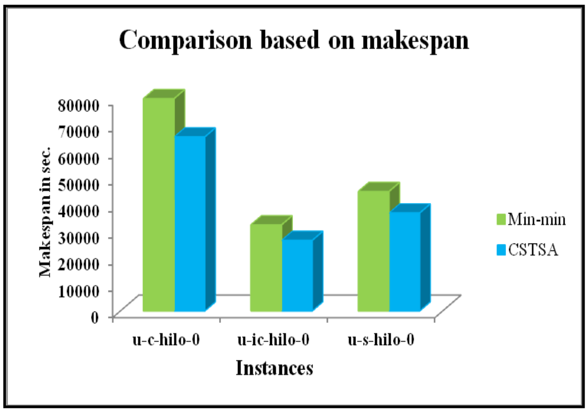


Figure 3: Comparison based on makespan for High Task Low Resource Heterogeneity

Table 4: Comparison based on makespan (in sec)

Instances	Min-min	CSTSA
u-c-lohi-0	267044.9	218886
u-ic-lohi-0	125061.7	102508.1
u-s-lohi-0	169090.7	138596.9

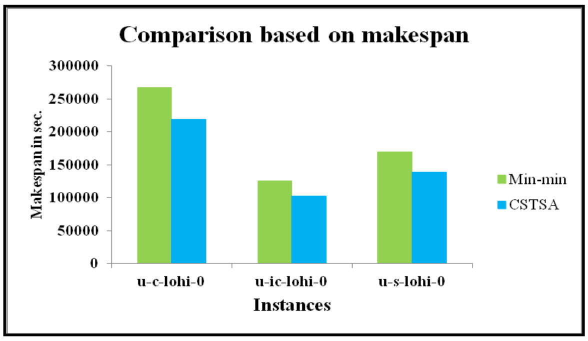


Figure 4: Comparison based on makespan for Low Task High Resource Heterogeneity

Table 5: Comparison based on makespan (in sec)

Instances	Min-min	CSTSA
u-c-lolo-0	2600.802	2131.773
u-ic-lolo-0	1062.335	870.7535
u-s-lolo-0	1586.498	1300.389

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165|

|| Volume 11, Issue 2, February 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1102011 |

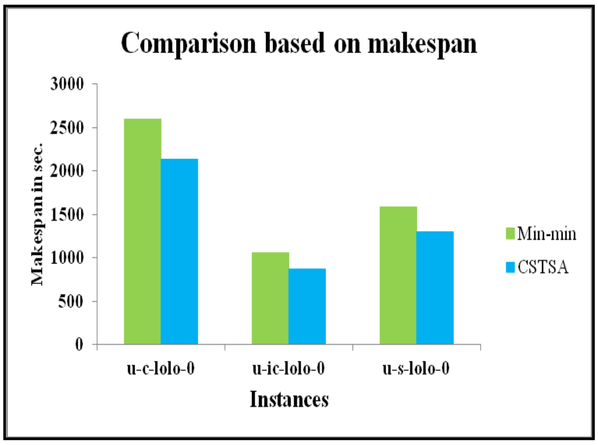


Figure 5: Comparison based on makespan for Low Task Low Resource Heterogeneity

IV. CONCLUSION AND FUTURE WORK

Grid environment can accommodate users with high computational tasks. Scheduling the tasks to the appropriate resources to achieve minimum completion time of the tasks is one of the challenging scenarios in grid computing environment. The current work emphasizes on selecting the tasks in the order in which it is to be scheduled to the resources to achieve reduced makespan. Based on the experimental study using 12 different types of ETC matrices with various characteristics such as task heterogeneity, resource heterogeneity, and consistency, the Credit Score Tasks Scheduling Algorithm significantly outperformed the Min-min heuristic scheduling algorithm in achieving reduced makespan. Because of its robust performance, Credit Score Tasks Scheduling Algorithm is a viable solution for static scheduling problem on heterogeneous grid environment.

REFERENCES

[1] T.Braun, H.Siegel, N.Beck, L.Boloni, M.Maheshwaran, A.Reuther, J.Robertson, M.Theys, B.Yao, D.Hensgen, and R.Freund, "A Comparison Study of Static Mapping Heuristics for a Class of Meta-tasks on Heterogeneous Computing Systems", In 8th IEEE Heterogeneous Computing Workshop(HCW'99), pp. 15-29, 1999.

[2] I.Foster and C. Kesselman, "The Grid: Blueprint for a Future Computing Infrastructure", Morgan Kaufmann Publishers, USA, 1998.

[3] E.U.Munir, J.Li, and S.Shi, "QoS Sufferage Heuristic for Independent Task Scheduling in Grid", Information Technology Journal 6(8), pp. 1166-1170, 2007.

[4] TD. Braun, HJ. Siegel, N.Beck, " A Taxonomy for Descriging Matching and Scheduling Heuristics for Mixed-machine Heterogeneous Computing Systems", IEEE Workshop on Advances in Parallel and Distributed Systems, West Lafayette, pp. 330-335, 1998.

[5] R.Armstrong, D.Hensgen, and T.Kidd, "The Relative Performance of Various Mapping Algorithms is Independent of Sizable Variances in Run-time Predictions", In 7th IEEE Heterogeneous Computing Workshop(HCW'98), pp. 79-87, 1998.

[6] R.F.Freund and H.J.Siegel,"Heterogeneous Processing", IEEE Computer , 26(6), pp. 13-17, 1993.

[7] T.D.Braun, H.J.Siegel, and N.Beck, "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing 61, pp.810-837, 2001.

[8] R.F.Freund, and M.Gherrity, "Scheduling Resources in Multi-user Heterogeneous Computing Environment with Smart Net", In Proceedings of the 7th IEEE HCW, 1998.

[9] G.K.Kamalam, and Dr. V..Murali Bhaskaran, "A New Heuristic Approach:Min-Mean Algorithm For Scheduling Meta-Tasks On Heterogeneous Computing Systems", IJCSNS International Journal of Computer Science and Network Security, Vol.10 No.1, pp. 24-31, 2010.

[10] G.K.Kamalam, and Dr. V..Murali Bhaskaran, "An Improved Min-Mean Heuristic Scheduling Algorithm for Mapping Independent Tasks on Heterogeneous Computing Environment", International Journal of Computational Cognition, Vol. 8, N0. 4, pp. 85-91, 2010.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165|

|| Volume 11, Issue 2, February 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1102011 |

[11] G.K.Kamalam, and Dr. V..Murali Bhaskaran, "New Enhanced Heuristic Min-Mean Scheduling Algorithm for Scheduling Meta-Tasks on Heterogeneous Grid Environment", European Journal of Scientific Research, Vol.70 No.3, pp. 423-430, 2012.

[12] H.Baghban, A.M. Rahmani, " A Heuristic on Job Scheduling in Grid Computing Environment", In Proceedings of the seventh IEEE International Conference on Grid and Cooperative Computing, pp. 141-146, 2008.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING