International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 20 Issue 2 – FEBRUARY 2016.

QOS GUIDED PROMINENT VALUE TASKS SCHEDULING ALGORITHM IN COMPUTATIONAL GRID ENVIRONMENT

Dr.G.K.Kamalam[1] , B.Anitha[2] and V.Gokulnathan[3]

[1]*M.E.,Ph.D, Department of Information Technology, Kongu Engineering College, Perundurai, Erode, Tamil Nadu 638052, India.*
[2]*M.E.,Department of Information Technology, Kongu Engineering College, Perundurai, Erode, Tamil Nadu 638052, India.*
[3]*B.TECH II IT—A Department of Information Technology, Kongu Engineering College, Perundurai, Erode, Tamil Nadu 638052, India.*

Abstract-- **Grids enable large-scale coordinated and collaborative resource sharing. Grid resources owned and managed by multiple organizations for solving scientific and engineering problems that require the large amount of computational resources. Scheduling of the tasks to the distributed heterogeneous grid resources belongs to the class of NP-Complete problems. To achieve high performance in the heterogeneous grid environment requires an efficient mapping of the tasks to the appropriate resources is essential. The order in which the tasks are scheduled to the resources is very critical criterion in scheduling which results in reduced makespan. This paper proposes a heuristic scheduling technique QoS Guided Prominent Value Tasks Scheduling Algorithm that determines the order in which the tasks are to be scheduled to the appropriate resources to optimize the completion time of the tasks. The comparison study shows that the proposed QoS Guided Prominent Value Tasks Scheduling Algorithm deals with the efficient resource mapping to the tasks and provides overall optimal performance with reduced makespan. The experimental results reveal that the order of mapping heuristic strategy depends on the parameters such as (a) QoS value (b) Prominent value and (c) execution time of the tasks.**

Index Terms- **Task Scheduling, Heterogeneous, QoS, NP-Complete**

I. INTRODUCTION

An emerging trend in network technology led to the possibilities of interconnection of diverse set of geographically distributed heterogeneous resources which supports executing computationally intensive applications. The high performance of the grid applications can be achieved by an efficient scheduling strategy. The key strategy for achieving high performance is the efficient mapping of the meta-task to the available computational resources. The fundamental criterion for obtaining optimal task scheduling is the reduced makespan [3,9]. Meta-task can be defined as a collection of independent, non-communicating tasks. Makespan can be defined as the overall completion time of all the computational tasks. The problem of optimally mapping the computational tasks to the diverse set of geographically distributed heterogeneous grid resources has been shown to be NP-Complete [2,4]. The grid scheduler needs to consider the task and QoS constraints to identify a better mapping between the tasks and the grid resources. The proposed QoS

Guided Prominent Value Tasks Scheduling Algorithm based on the task requirement of QoS classifies the tasks into high QoS tasks and low QoS tasks. The grid resources based on the task constraints are classified into high QoS provision resources and low QoS provision resources. The proposed QoS Guided Prominent Value Tasks Scheduling Algorithm performs the better mapping between the tasks and the grid resources by computing the Prominent Value (PV) for the task. The tasks are ordered into the Prominent Value Set (PVS) from minimum to the highest prominent value of the task. The proposed algorithm achieves optimal scheduling with reduced makespan compared to that of the Min-min heuristic scheduling algorithm.

II. RELATED WORKS

Opportunistic Load Balancing (OLB) algorithm does not consider the minimum execution time or minimum completion time of the tasks and schedules the tasks in the arbitrary order to the available grid resources. The grid resources are also selected in an arbitrary order [5,6].

Minimum Execution Time (MET) algorithm assigns each task to the resource with the minimum expected execution time for that task [1,7].

Minimum Completion Time (MCT) algorithm assigns each task to the resource with the minimum completion time for that task. The disadvantage of the algorithm is that some tasks do not have the minimum execution time [1,7].

Min-min algorithm starts with the set U of all unmapped tasks. The set of minimum completion time for each task in the set U is calculated. Then, the task with the overall minimum completion time is selected and scheduled to the particular resource. The newly allocated task is removed from the set U and the process repeats until all tasks in the set U are mapped [5,6,8].

The Max-Min algorithm starts with the set U of all unmapped tasks. The set of minimum completion time for each task in the set U is calculated. Then, the task with the overall maximum completion time is selected and scheduled to the particular resource. The newly allocated task is removed from the set U and the process repeats until all tasks in the set U are mapped [5,6].

Our previous work, Min-mean heuristic scheduling algorithm works in two phases. In the first phase, Min-mean heuristic scheduling algorithm starts with a set of all unmapped tasks. The algorithm calculates the completion time for each task on each resource and finds the minimum completion time for each task. From that group, the algorithm selects the task with the overall minimum completion time and allocates to the appropriate resource. Removes the task from the task set. This process repeats until all the tasks get mapped. The algorithm calculates the total completion time of all the resources and the mean completion time. In phase 2, the mean of all resources completion time is taken. The resource whose completion time is greater than the mean value is selected. The tasks allocated to the selected resources are reallocated to the resources whose completion time is less than the mean value [10, 11].

In QoS guided min-min heuristic, tasks are classified into the high QoS and low QoS tasks. High QoS tasks are given the highest priority and are first scheduled using Min-min heuristic scheduling algorithm. Low QoS tasks are also scheduled using Min-min heuristic scheduling algorithm [14].

In QoS priority grouping algorithm, the tasks are classified into 'n' groups based on the number of resources on which the tasks can execute. Tasks from each group are scheduled using sufferage algorithm independently [13].

In QoS sufferage heuristic algorithm, tasks are grouped into two groups based on high QoS and low QoS requirements. Tasks are scheduled based on the highest sufferage value assigned to the resource. Sufferage value is the difference between the earliest completion time and the second earliest completion time [3].

In QoS based predictive max-min, min-min switcher algorithm, tasks are grouped into high QoS request tasks and low QoS request tasks. The algorithm schedules the tasks using the two conventional algorithms, Max-min and Min-min based on standard deviation of minimum completion time of unassigned tasks [15]. A task with high QoS request can only be executed on a resource with high QoS provision [16].

III. MATERIALS AND METHODS

Meta-task is defined as the collection of independent tasks with no inter-task data dependencies. An application consists of 'n' independent meta-task and 'm' heterogeneous resources. An optimal order of mapping the meta-tasks to the set of heterogeneous in a grid environment is an NP-Complete problem [12]. An efficient scheduling algorithm which determines an optimal order in mapping the meta-tasks to the set of heterogeneous resources is proposed.

In the proposed QoS Guided Prominent Value Tasks Scheduling Algorithm, meta-tasks are grouped into two groups: High QoS and Low QoS. The tasks with high QoS requirements can only be executed on resources with high QoS provision. High QoS task assigned with low QoS value. Low QoS tasks are assigned with high QoS value based on the execution time of the tasks. The low QoS tasks which have the highest execution are given the high priority.

The proposed algorithm QoS Guided Prominent Value Tasks Scheduling Algorithm considers a scheduling criteria 'Prominent Value of the meta-task' for efficient scheduling. The task order in which the tasks to be scheduled are based on the Prominent Value. The tasks are ordered from the minimum Prominent Value to the highest Prominent Value. The ordered tasks are scheduled to the resources with the minimum completion time of the task. The common objective function of the task scheduling algorithm is the makespan. Makespan is defined as the total time required for executing the meta-task. The proposed algorithm provides an optimal schedule with reduced makespan.

A. *Notations And Definitions*

The notations and definitions used in this paper are shown in Table 1.

Table 1 Notation and Definition

Notation	Definition
$Tasklength_i$	Length of the i^{th} task in MI
$Computing\ speed_j$	Computing speed of the j^{th} resource in MIPS
ETC	Expected Time to Compute matrix
ETC_{ij}	Expected Execution Time of the task t_i on resource r_j
RT_j	Ready Time of resource r_j after having completed the previously assigned tasks
METC	Maximum value in the ETC matrix.
TV_1	METC/2
TV_2	METC/3
TV_3	TV_1+TV_2
TV_4	TV_2+TV_3
P_i	Credit Point for each task t_i
QV_i	QoS Value for each task t_i
QCV_i	QoS Credit Value for each task t_i
PV_i	Prominent Value for each task t_i
PVS	Lists the tasks in the order of minimum to highest value of PV_i

The pseudocode for finding Credit Point for each task is given below:

```
Algorithm Credit Point
Initialize MAXET =0
  for i=1 to n
    for j=1 to m
        if ETC_ij > METC
        METC= ETC_ij
        end if
```

```
    end for
end for
Compute TV1 = METC/2
Compute TV2= METC/3
Compute TV3 = TV1+TV2
Compute TV4= TV2+TV3
for all submitted task ti in the meta-task Mt
Find the maximum execution time of each task
  if METCi < TV1
      Pi = 4
  else if TV1 ≤ METCi ≤ TV3
      Pi = 3
  else if TV3 ≤ METCi ≤ TV4
      Pi = 2
  else
      Pi = 1
  end if
end for
```

B. QoS Value

The resources may not have the capability to execute all the tasks due to its low QoS provision. The tasks that can be executed in only one resource or few resources are grouped into high QoS tasks. The tasks that can be executed in all resources are grouped into low QoS tasks.

The task that can be executed in only one resource is given the QoS value 1. The task that can be executed in only two resources is given the QoS value 2 and so on. The tasks that can be executed on all resources are given the QoS value based on their execution time. The task that has maximum execution time is given high QoS value and so on.

The pseudocode for finding the QoS Credit Value is shown below:

```
        Algorithm QoS Credit Value
For all submitted tasks in the meta-task Mt
        Compute QCVi =QVi / dv
End for
```

The value dv is determined as follows:

If the highest QoS value assigned for a task is a two digit number, dv=100, if it is a three digit number, dv=1000, and so on.

C. QoS Guided Prominent Value Tasks Scheduling Algorithm

```
for all submitted tasks ti in the meta-task Mt
        Calculate Credit Point for each task using
Credit Point Algorithm
        Calculate QoS Credit Value for each task
using QoS Credit Value Algorithm
end for
for each task ti in the meta-task Mt
        Compute PVi = Pi * QCVi
end for
Order the tasks in the Prominent Value set PVS in
the ascending order of PVi.
for all tasks in the Prominent Value Set (PVS)
   for all resources Rj
        Compute TCTij=ETij+RTj
end for
end for
    do until all tasks in PVS are mapped
    for each task in PVS find the earliest
completion time and the resource that obtains it.
        Find the task tk with the minimum
earliest completion time.
        Assign task tk to the resource Rj that
gives the earliest completion time
        Delete task tk from CSS
        Update RTj
        Update TCTij for all i
end for
end do
Compute makespan = max(TCTij) for all i, j
```

A simple example is given below to illustrate the execution of the proposed algorithm QoS Guided Prominent Value Tasks Scheduling Algorithm and to compare its efficiency with the existing Min-min heuristic scheduling algorithm. Table 1 shows the execution time of 9 tasks on 5 resources. The entry 'X' in the table denotes that the resource does not have the capability to execute that particular task due to its low QoS provision.

Table 1 ETC Matrix

Tasks	R_1	R_2	R_3	R_4	R_5
t_1	X	X	X	X	10.5
t_2	X	X	X	7.5	11.0
t_3	X	X	X	5.2	6.2
t_4	X	X	14.7	8.0	12.8
t_5	X	X	4.4	13.3	7.2
t_6	X	3.8	5.9	5.8	3.4
t_7	X	7.8	4.0	17.3	6.8
t_8	15	9.0	5.6	7.1	6.4
t_9	5.1	4.8	4.1	16.1	13.2

The maximum value in the given ETC matrix is,

METC=17.3
TV=17.3/2=8.7
TV=17.3/3=5.8
TV=14.5
TV=23.2

The Credit Point for each task is computed and is shown in Table 2.

Table 2 Credit Point for each Task

Tasks	Pi
t_1	3
t_2	3
t_3	4
t_4	2
t_5	3
t_6	4
t_7	2
t_8	2
t_9	2

The task t_1 can be executed only on one resource R_5. Task t_1 is called high QoS task. So, the task t_1 is given the low QoS value 1.Next, the task t_2 can be executed on two resources R_4 and R_5. The task t_1 is given the low QoS value 2 and so on. The tasks t_8 and t_9 are called low QoS task, since they can be executed on all resources. The tasks t_8 and t_9 are given high QoS value. The task t_9 has maximum execution time and is given the high priority and the credit for task t_9 is 5 and for task t_8 is 6. The QoS value, QoS Credit Value for each task is computed and is shown in Table 3. The Prominent Value for each task t_i is computed and is shown in Table 3.

Table 3 Prominent Value for each Task

Tasks	Pi	QV_i	QCV_i	PV_i
t_1	3	1	0.1	0.3
t_2	3	2	0.2	0.6
t_3	4	2	0.2	0.8
t_4	2	3	0.3	0.6
t_5	3	3	0.3	0.9
t_6	4	4	0.3	1.6
t_7	2	4	0.4	1.6
t_8	2	6	0.5	1.0
t_9	2	5	0.6	1.2

The tasks are ordered in the Prominent Value Set (PVS) in the ascending order of PV_i.

$$PVS = \{t_1,t_2,t_4,t_3,t_5,t_8,t_9,t_6,t_7\}$$

The high QoS tasks are scheduled to the resources that have low QoS provision and the low QoS tasks are scheduled to the resources that have the high QoS provision. The tasks are scheduled in the order specified in the task set PVS. The makespan obtained for the Min-min algorithm, QoS sufferage algorithm and the proposed QoS Guided Prominent Value Tasks Scheduling Algorithm is shown in Table 4.

Table 4 A Comparisons between existing and proposed algorithms in makespan and task schedule order.

	R_1	R_2	R_3	R_4	R_5	Makespan
Min-min		t_9, t_8	t_7, t_5	t_3, t_2, t_4	t_6,t_1	20.7
QoS Sufferage	t_9	t_6,t_7	t_5,t_4	t_2,t_3	t_7,t_8	19.1
QoS Prominent Value	t_9	t_8,t_6	t_4,t_7	t_2,t_3	t_1,t_5	18.7

From Table 4, it is evident that the proposed QoS Guided Prominent Value Tasks Scheduling Algorithm outperforms the QoS sufferage heuristic algorithm and Min-min heuristic algorithm based on makespan. Furthermore, it can be noted that the makespan given by the proposed QoS Guided Prominent Value Tasks Scheduling Algorithm is smaller than the makespan obtained by the QoS sufferage and Min-min heuristic scheduling algorithms.

IV. SIMULATION AND RESULTS

The proposed approach is evaluated with user-defined number of resources and tasks. The execution time of all the tasks is considered for efficient scheduling. The execution time of all the tasks in all the resources is generated using the ETC matrix, a benchmark model designed by Braun et.al [1,4,7]. The rows of the ETC matrix represent the execution time of each task on all given resources.

Figure 1 shows the experimental results corresponding to ETC matrices of 50 Tasks* 5 resources, 100 tasks * 10 resources, 150 Tasks* 10 resources, 200 tasks*10 resources and 250 tasks*10 resources indicate that the proposed QoS Guided Prominent Value Tasks Scheduling Algorithm performs well and outperforms the Min-min heuristic scheduling algorithm. The proposed QoS Guided Prominent Value Tasks Scheduling Algorithm gives reduced makespan for all five cases than the Min-min heuristic scheduling algorithm.

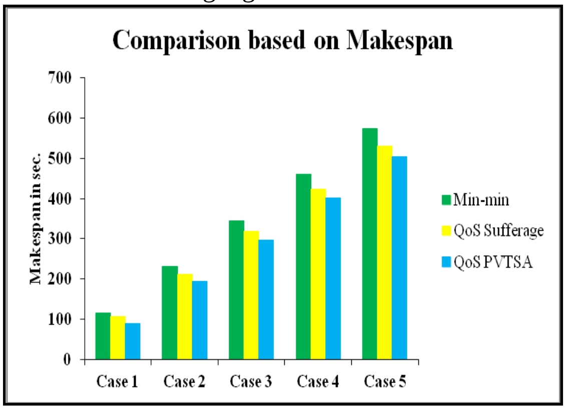

Figure 1: Comparison based on makespan for five different cases

V. CONCLUSION

Task scheduling is an NP-Complete problem in distributed grid environment. This paper proposed a novel heuristic scheduling strategy by considering QoS factor in scheduling the tasks on to the resources. The proposed QoS Guided Prominent Value Tasks Scheduling Algorithm and the Min-min heuristic scheduling algorithm are examined using the benchmark simulation model by Braun et.al [1,4,7]. Presented experimental results prove that the proposed heuristic scheduling strategy QoS Guided Prominent Value Tasks Scheduling Algorithm has a significant improvement in performance in terms of reduced makespan and outperforms Min-min heuristic scheduling algorithm.

VI. 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.

[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.

[13] F.Dong, J.Luo, L.Gao, and L.Ge, "A Grid Task Scheduling Algorithm based on QoS Priority Grouping", In Proceedings of the 5th International conference on Grid and Cooperative Computing, pp.58-61,2006.

[14] H.E.Xiaoshan, X.H.Sun, and G,V.LLaszewski, "QoS Guided min-min heuristic for grid task scheduling", Journal of computer science technology (Special Issue on Grid Computing), pp.442-451,2003.

[15] M.Singh, and P.K.Suri, "Analysis of service, challenges and performance of a grid", International Journal of Computer Science and Network Security, pp.84-88, 2007.

[16] H.Ligang, and A.Stephen, "Dynamic Scheduling of parallel jobs with QoS demands in multiclusters and grids", Proccedings of 5th IEEE/ACM International Workshop on Grid Computing, pp.402-409,2004.