Task – Heterogeneous Resource Mapping Algorithm for Grid Environment

G.K.Kamalam, B.Anitha

Abstract- In grid, scheduling algorithms play vital role of mapping a set of tasks to the available heterogeneous resources. Extant literatures have shown that the task mapping problem is an NP-Complete problem. Heuristic scheduling algorithm aims to obtain the minimum overall execution time of the set of tasks. In this paper, we address the problem of scheduling a set of n tasks with a set of m resources, such that the makespan is minimized. The proposed task scheduling algorithm (RTS) is based on the well-known optimization algorithm, called Hungarian algorithm. (RTS) algorithm considers an equal number of tasks and resources, and maps the tasks to the resources and makes an effective scheduling decision. We simulate the (RTS) algorithm and compare it with the Min-min heuristic scheduling algorithm. The performance evaluation shows that (RTS) produces minimized makespan and better resource utilization in comparison to existing Min-min.

Keywords: Hungarian method, Heuristic algorithm, Optimization, Task Scheduling

I. INTRODUCTION

Grid environment is a distributed resource sharing infrastructure. Grid computing is emerged for processing large data in scientific computing community. To satisfy the computational need of the organization to execute advanced applications, distributed computers are connected through internet in a Grid-like manner. One of the key issues in the heterogeneous network is to achieve efficiency in resource sharing. Resource discovery, resource allocation and task execution are the three phases of Grid scheduling. Scheduling of tasks to the available resources plays a vital role in dynamic grid environment. Scheduling deals with assigning of incoming tasks to the grid resources depending on the constraints to minimize the overall execution time. Different types of schedulers based on different grid characteristics like dynamicity and adaptability would be expected to solve computationally hard problems.

Task scheduling on distributed grid environment is class of NP-Complete problem. To get an optimal execution time over the dynamicity of the grid resources, a heuristic scheduling algorithm is needed. Task scheduling plays an important role in grid computing. A task is divided into number of subtasks which are assigned to various machines for execution on a computation grid. There are large numbers of scheduling algorithms available to minimize the overall execution time and to maximize resource utilization [4]. Researchers have proposed several heuristics task scheduling algorithms to improve overall system performance. Heuristic approach has two categories. In on-line mode, whenever a task arrives it is assigned to the first free machine. In batch mode, the tasks are collected in a batch and are mapped at prescheduled time. The proposed heuristic scheduler in grid computing environment reduces the execution time by allocating the task to suitable resources.

II. LITERATURE REVIEW

A well known Min -min algorithm calculates the expected execution time of each unassigned task on different resources. Then it selects the task with minimum execution time and designates it to the corresponding resource on which the minimum execution time is achieved. The selected task is removed from the pool. Same procedure is repeated until all the tasks get mapped. The algorithm results in larger makespan when longer tasks are more compared to shorter one [4,5,7].

In Max-min algorithm, priority is given to the task having larger completion time. The algorithm seems to be better than the Min -min algorithm when the number of shorter tasks is more. In the Max-min algorithm, the task with short execution time may wait for large ones to be executed [4,5,7].

Opportunistic Load Balancing (OLB) algorithm schedules each task to the next available resource in random order without considering the expected task completion time on that resource. Its goal is to make all resources engaged. Since the algorithm does not consider the execution time for the submitted tasks, it results in long makespan [4,5,7].

The Minimum Execution Time (MET) grid scheduling algorithm, assigns each task to the machine that has the minimum execution time. It allocates based on first come first served basis. It allocates without considering whether the machine is available and also the current load of the machine. MET assigns each task based on minimum execution time that causes load imbalance between the machines [4,5,7].

Minimum Completion Time(MCT) algorithm designates the task to the resources based on the minimum expected completion time of that task on that resource in random order. Completion time is computed by adding the resource availability time and the execution time of the task on that resource. Each task is assigned to the resource that has minimum completion time [4,5,7].

Manuscript published on November 30, 2019.
* Correspondence Author

Dr.G.K.Kamalam*, Assistant Professor(SLG) in Kongu Engineering College Erode

Ms.B.Anitha, Assistant Professor in Kongu Engineering College Erode

A Min-mean heuristic scheduling algorithm has been proposed for static meta-tasks. The proposed algorithm reschedules the tasks by considering the mean completion time of all the resources [4].

The author discussed the theoretical elements of the Hungarian algorithm and the algorithm is based on the Konig's theorem of graph theory. Author represented this theorem in the form of graph instead of matrix. The theorem is applied only if there is a zero in each row and column [3].

A Method of Classification for Customer Expectations is proposed. Relationship Marketing is performed dynamically. The proposed method considers the customer's perspective only [1].

The algorithm, A Novel Economy Model Based on Grid Resource Supermarket is proposed. Static Task Scheduling is performed. The model is tested in small scale situations only [9].

The author proposed A Transmission Mode Assignment Algorithm which is based on Hungarian algorithm and the static allocation of tasks to resource is performed. The algorithm results in improved overall system throughput [2].

The author proposed An Improved Hungarian Algorithm. Static job scheduling is performed. It results in least total time. The drawback of this algorithm is that it consumes more time for the evaluation process [10].

The Load Balancing of Unbalanced Matrix with Hungarian Method is proposed. Static job scheduling is performed. The objective of the algorithm is to balance the load efficiently. The disadvantage of this algorithm is, the jobs and machines are mapped in different phases [11].

The optimization algorithm aims to minimize the makespan. The proposed Resource Task Scheduling algorithm (RTS) is based on the well-known optimization algorithm, called Hungarian algorithm. The proposed algorithm (RTS) considers an equal number of tasks and resources, and schedules the tasks to the resources and makes an effective scheduling decision.

III. PROPOSED METHOD

A. Resource Task Scheduling (RTS) Algorithm

(RTS) algorithm is based on a combinatorial optimization algorithm, called Hungarian algorithm [14], which was originally developed by Harold W. Kuhn in 1955. RTS is an assignment or a scheduling algorithm, which arranges the tasks in such a manner that it is well-suitable to resources in the grid.

Algorithm step:

Input: Resources (R) [i=1....m]
Task (T) [j=1....n]

Step 1: For all resources, Processing Capacity of each resource
RPC[i] = Resource Processing Capacity in MIPS

Step 2 : For all Task, Length of each Task
TL[j] = Length of Tasks in MI

Step 3: Find Execution Time for each task to resource
ETC[i][j]= TL[j]/RPC[i] Seconds

Step 4 : Construct Expected Execution Time Matrix ETC[Task, Resource].

B. Resource-Task Scheduling Procedure based on Hungarian method.

Step I (A) Row reduction: From all the values of each row subtract the minimum value of the corresponding row in the ETC matrix.

(B) Column reduction: From all the values of each column subtract the minimum value of the corresponding column in the ETC matrix

Step II Zero assignment: (A) After completing the row reduction and column reduction, in the resulting ETC matrix starting from the first row, examine the rows one by one until a row containing exactly one zero is found. Then an assignment by the symbol is marked to that zero. Now cross all the □ros in the column in which the assignment is made. This procedure is repeated for each row. **(B)** Starting with column 1, examine the columns one by one until a column containing exactly one zero is found. Then an assignment by the symbol
□ is marked and cross all the other zeros in the row in which the assignment was made. Repeat Step II (A) and (B) operations on rows and columns until all zero's have either been assigned or crossed-out.

Consider the two cases: **(a)** either all the zeros are assigned or crossed out, i.e., we get the maximal assignment. Or **(b)** At least two zeros are remained by assignment or by crossing out in each row or column. If either all the zeros are not assigned or not crossed out some of the zeros are excluded by trial and error method. This step leads to two situations.

(i) If the total assigned zero's is equal to n then the assignment is said to be optimal. Stop the process.

(ii) If the total assigned zero's is less than n. Use step III and onwards.

Step III: In order to cover the entire zero's at least once the following sequences may be adopted. **(i)** Mark (√) to all rows in which the assignment has not been done. **(ii)** For the marked rows, mark (√) to the corresponding column. **(iii)** For the marked columns, find the position of assigned zero's in the corresponding column and then mark (√) to the corresponding rows which are not marked till now. **(iv)** Repeat the procedure (ii) and (iii) till the completion of marking. **(v)** Draw the lines through unmarked rows and marked columns. Note: If the above method does not work then make an arbitrary assignment and then follow step IV.

Step IV: Select the least value from the uncovered values. **(i)** Subtract this least value from all those values which are not covered. **(ii)** Add this least value to all those values which are at the intersection of two lines.

Step V: Now, the number of zero's has been increased in the ETC matrix. Again repeat Step II and find the required assignment.

IV. RESULTS AND DISCUSSION

A. Example

For illustration a set of five tasks, namely T={T1,T2,T3,T4,T5} and a set of five resources, namely R={R1,R2,R3,R4,R5} is considered.

Task length, resource processing capacity, namely TL and RPC is shown in Table I and II.

Table – I Task Length (MI)

Tasks	Task Length (TL)
T1	25100
T2	30800
T3	242700
T4	68000
T5	6400

Table - II Resource Processing Capacity (MIPS)

Resources	Resource Processing Capacity (RPC)
R1	5000
R2	3000
R3	1000
R4	2000
R5	1500

ETC matrix for five tasks and five resources is shown in Table III.

Table - III ETC Matrix

Tasks	R1	R2	R3	R4	R5
T1	5.0	8.4	25.1	12.6	16.7
T2	6.2	10.3	30.8	15.4	20.5
T3	48.5	80.9	242.7	121.4	161.8
T4	13.6	22.7	68.0	34.0	45.3
T5	1.3	2.1	6.4	3.2	4.3

Row Reduction ETC matrix is shown in Table IV.

Table - IV Row Reduction Matrix

Tasks	R1	R2	R3	R4	R5
T1	0.0	3.4	20.1	7.6	11.7
T2	0.0	4.1	24.6	9.2	14.3
T3	0.0	32.4	194.2	72.9	113.3
T4	0.0	9.1	54.4	20.4	31.7
T5	0.0	0.8	5.1	1.9	3.0

Column Reduction ETC matrix is shown in Table V.

Table - V Column Reduction Matrix

Tasks	R1	R2	R3	R4	R5
T1	0.0	2.6	15.0	5.7	8.7
T2	0.0	3.3	19.5	7.3	11.3
T3	0.0	31.6	189.1	71.0	110.3
T4	0.0	8.3	49.3	18.5	28.7
T5	0.0	0.0	0.0	0.0	0.0

Finally, the task-resource assignment matrix is shown in Table VI.

Table - VI Task – Resource Assignment Matrix

Tasks	R1	R2	R3	R4	R5
T1	~~0.0~~	~~0.0~~	0.8	~~0.0~~	$\boxed{0.0}$
T2	~~0.0~~	~~0.0~~	7.9	$\boxed{0.0}$	~~0.0~~
T3	$\boxed{0.0}$	31.6	189.1	71.0	110.3
T4	~~0.0~~	$\boxed{0.0}$	41	10.2	20.4
T5	~~0.0~~	~~0.0~~	$\boxed{0.0}$	~~0.0~~	~~0.0~~

The resulted makespan value, resource, and the corresponding task allocated combination shown in Table VII. The table values depict clearly the proposed (RTS) algorithm results with good makespan and better resource utilization compared to Min-min.

Table - VII makespan Comparison

Algorithm	R1	R2	R3	R4	R5	makespan
Proposed Algorithm (RTS)	T3	T4	T5	T2	T1	48.5
Min-min Algorithm	T1, T3	T2, T4	T5	--	--	53.5

B. Evaluation Parameters:makespan

Optimization criteria for grid scheduling is makespan.

(RTS) algorithm is compared with Min-min using different datasets in terms of makespan as shown in Fig.1. The Fig.1. clearly proves (RTS) algorithm produces good makespan than Min-min.

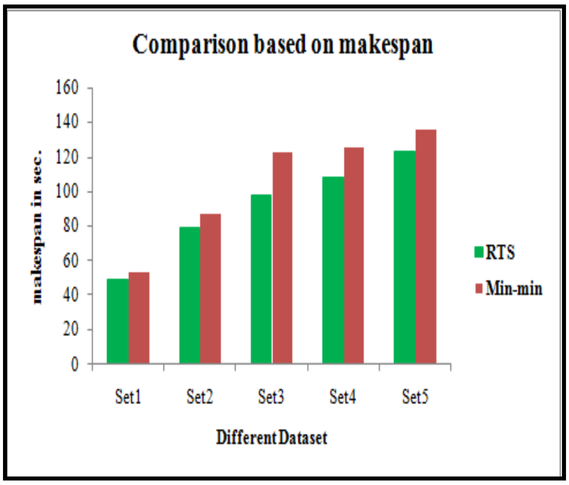

Fig.1. makespan comparison

IV. CONCLUSION AND FUTURE SCOPE

The proposed (RTS) algorithm identifies the Resource-Task mapped pair based on a combinatorial optimization algorithm, called Hungarian algorithm. (RTS) algorithm and Min-min are tested using different dataset values. The experimental results show that (RTS) algorithm compared to Min-min brings good makespan value and better resource utilization. The proposed (RTS) algorithm can be enhanced to schedule dependent tasks in future.

REFERENCES

1. C.Bassa, and Gil-Lafuente., “The Hungarian algorithm for specific customer needs”, Soft Comput. Manage. Business Econ. Stud. Fuzziness and Soft Comput.Springer, 2012, pp. 363-379.
2. R.Chithra, R.Bestak, and S. Patra., “Hungarian method based joint transmission mode and relay selection in ‚device-to-device communication”, In: 8th IFIP Wireless and Mobile Networking Conference, IEEE.2015, pp. 261–268.
3. J.Gil-Aluja., “Theoretical elements of the Hungarian algorithm. In: The Interactive Management of Human Resources in Uncertainty”, Applied Optimization, Springer, 1998, pp. 158–170.
4. 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, 2010, pp. 24-31.
5. 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, No. 4, 2010, pp. 85-91.
6. G.K.Kamalam, and Dr.V.Murali Bhaskaran., “An Efficient Hybrid Job Scheduling for Computational Grids”, International Journal of Computer Applications, 2011.
7. 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, 2012, pp. 423-430.
8. H.Kuhn., “The Hungarian method for the assignment problem. 50 Years of Integer Programming 1958–2008”, Springer, 2009, pp. 29–47.
9. M.Li, N.Xiong, B.Yang, Z.Li, J.Park, and C.Lee., “Posted price model based on GRS and its optimization for improving grid resource sharing efficiency”, Telecommun. Syst. Springer, Vol. 55 No.1, 2014, pp.71–79.
10. T.Li, Y.Li, and Y.Qian., “Improved Hungarian algorithm for assignment problems of serial-parallel systems”, J. Syst. Eng. Electron. IEEE, Vol.27 No.4, 2016, pp. 858–870.
11. R.Mondal, P.Ray, E.Nandi, B.Biswas, M.Sanyal, and D.Sarddar., “Load balancing of unbalanced matrix with Hungarian method”, In International Conference on Computational Intelligence, Communications and Business Analytics, Communications in Computer and Information Science, Springer,2017, pp. 256–270.

AUTHORS PROFILE

Dr.G.K.Kamalam is working as an Assistant Professor (SLG) in the Department of Information Technology, Kongu Engineering College, Tamil Nadu, and India. Her research area is Grid and Cloud Computing. She has presented 35 papers in national and international conferences and published 25 papers in international journals in her research and other technical areas. Her Area of Interest includes Algorithm Analysis and Optimization Techniques.

B.Anitha received BE degree in Computer Science and Engineering from Madras University and M.Tech degree from Anna University. She is currently working as an Assistant Professor in the Department of Information Technology, Kongu Engineering College, Tamil Nadu, India. Her research area is Grid Computing and Data Analysis. She has published papers in national and international journal. She is a life member of Computer Society of India (CSI). Her research interest includes Machine Learning and Deep Learning.