Resource Allocation with improved Min-Min Algorithm

Rajwinder kaur
Lovely Professional University
Phagwara, Punjab

Prasenjit Kumar Patra
Lovely Professional University
Phagwara, Punjab

ABSTRACT

A distributed system is a software system in which components located on networked computers communicate and coordinate their actions by passing messages. Most of the existing solutions on task scheduling and resource management in distributed computing environment are based on the traditional client/ server model, enforcing a homogeneous policy on making decisions and limiting the flexibility, unpredictable reliability and scalability of the system. Thus, we need well organized system architecture to provide high system availability with task scheduling scheme for distributed system especially on Grid and Cloud. In this paper, we propose an efficient rescheduling based task scheduling algorithm named improved Min-Min Algorithm (I Min-Min) which performs scheduling in order to enhance system performance in any distributing system. The proposed method has two-phases. In the first phase the traditional Min-Min algorithm is executed and in the second phase the tasks are rescheduled to use the unutilized resources effectively.

Keywords

Distributed computing, Min-Min algorithm, Scheduling algorithm, Resource allocation technique, Cloud computing, Grid computing

1. INTRODUCTION

In distributed systems schedule a task on an appropriate resource is an difficult task because distributed resources are geographically dispersed lies in different domains, different policies, and different administrative .It become necessity to deal an efficient way to make a schedule for coming tasks by using a task scheduling technique which cope up with the situation when m number of tasks come and n number of resources are available ,mapping these tasks in that manner by which it produce a minimum makespan and flow time.

Now for any system task came through a sequential flow. Workflow is concerned with the automation of procedures whereby files and data are passed between participants according to a defined set of rules to achieve an overall goal [1]. A task is a (sequential) activity that uses a set of inputs to produce a set of outputs. Processes in fixed set are statically assigned to processors, either at compile-time or at start-up. There are two types of scheduling: static and dynamic. [3].

Scheduling is nothing but a set of task versus set of processors and a workflow scheduling can be defined as the automation in scheduling of workload. A scheduling can be categories into two categories: Job Scheduling and Job Mapping and Scheduling. Job Scheduling is what in which independent jobs are scheduled among the processors of distributed computing for optimization. A Job Mapping and Scheduling requires the allocation of multiple interacting tasks of a single parallel program in order to minimize the completion time on parallel computer system [1].

Makespan is defined as the overall finish time, which is equal to the actual finish time of the exit task. So reduce in makespan can consider as improvement in performance of a system [2].

The management of processor time, memory, network, storage and other Components in a distributed system especially on Grid and Cloud is clearly very important. the overall aims to efficiently and effectively schedule the application that need to utilize the available resource in the distributed system environment especially on Grid and Cloud environment .from an users point of view, resource management and scheduling should be transparent; their interaction with it being confined to a manipulating mechanism for submitting their application. It is important in a distributed system that is a resource management and scheduling service can intact with those that may be installed locally. Load balancing procedure facilitated connection building between schedulers and resource managers [1]. Tasks are fetched from the scheduler when resource is available which fulfill the user’s requirements then work is loaded on the resource.

2. RELATED WORK

***Original Min-Min* Algorithm**

Min-Min algorithm [4] is a heuristic algorithm, it first finds the minimum expectation time of all tasks .After that it select the tasks with the minimum execution time among all the tasks and assign that task on that resource. Same step are repeated by Min-Min until all tasks are not scheduled. The limitation of Min-Min algorithm is that it select the smaller tasks first and occupy the resource which have high computation power. The tasks which are large it waits the time until smaller tasks [16]. The Min-Min is not produced optimal schedule when tasks are greater than number of available resource.

To avoid the drawbacks of the Min-Min algorithm many algorithm are which are discussed sequentially here under.

D.Doreen Hephzibah Miriam and K.S.Easwarakumar proposed a Double Min Min Algorithm for Task Meta scheduler on hyper cubic P2P Grid Systems. They integrate Grid with P2P on to the extended Hypercube topology for task scheduling and load balancing. This algorithm gives optimal makespan and balances the load in order to enhance system performance. To enhance Load balancing, the algorithm finds the Mean CT for the scheduled results and reselects the tasks whose values are greater than Mean CT. For the purpose to reselect task, the SPA based Min Min is computed. According to the experimental result the proposed algorithm gives better results than the conventional algorithms [5].

T. Kokilavani, Dr. D.I. George Amalarethinam presents an algorithm to load balancing for Static Meta-Task Scheduling. It addresses some issue such as resource discovery, heterogeneity, fault tolerance and task scheduling. Grid environment faces load balance problem the most. Load Balanced Min-Min (LBMM) algorithm is proposed that reduces the makespan and increases the resource utilization. Firstly they work on Min-Min algorithm and finds minimum execution tasks. In second round they rescheduled the tasks and get minimum makespan. It uses the advantages of Max-Min and Min-Min algorithms and covers their disadvantages [6].

X.he et al represents a new algorithm based on original Min-Min algorithm. This algorithm is also called Qos guided Min-Min algorithm. At the time of scheduling of tasks it require high band width than other algorithm. Meanwhile if the required bandwidth for different tasks various highly the Qos guided min-min provide better result than min-min algorithm. Whenever the bandwidth requirement of all of the tasks is almost the same, the QoS guided Min-Min algorithm works as min-min algorithm [7].

Kamalam et al., [8] presents a new scheduling algorithm named Min-mean heuristic scheduling algorithm for static mapping to achieve better performance. The proposed algorithm reschedules the Min-Min produced schedule by considering the mean makespan of all the resources. The algorithm deviates in producing a better schedule than the Min-Min algorithm when the task heterogeneity increases.

Sameer Singh et al., [9] have presented two heuristic algorithms: 1) QoS Guided Weighted Mean Time-Min (QWMTM) and 2) QoS Guided Weighted Mean Time Min-Min Max-Min Selective (QWMTS). Both algorithms are for batch mode independent tasks scheduling. The network bandwidth is taken as a QoS parameter.

Singh.M et al., [10] present a QoS based predictive Max-Min, Min-Min Switcher algorithm for scheduling jobs in a grid. Before scheduling the next job the algorithm makes a proper selection among the QoS based Max-Min or QoS based Min-Min algorithm on the basis of heuristic applied. The algorithm uses the history information about the execution of jobs to predict the performance of non-dedicated resources. In this algorithm the authors take-care the problem Due to non-dedicated property of resources on the execution time of grid jobs.

Abraham, A., Buyya, R. and Nath, B.,[11] introduced to us with LJFR-SJFR, The algorithm firstly found the set of minimum computing times as Min-Min algorithm then in next step it finds the shortest jobs. The tasks resource with overall minimum completion times from m (SJFR) and also considered the longest job in fastest resource. It is a Nature's Heuristics for Scheduling for Jobs on Computational Grids**.** At starting this method assign m longest tasks on m fastest resource then in next round this method assign the shortest task to fastest resource and the longest task to the fastest resource alternately. Experimental result shows this algorithm introduce good napping rather than OLB.

Saeed Parsa and Reza Entezari-Maleki [12] proposed a new algorithm RASA: A New Task Scheduling Algorithm in Grid Environment by combining two famous algorithms. They are 1) Min-Min and 2) Max-Min. Min-Min algorithm first finds the maximum expectation time then choose the task with the minimum execution time among all and then assign the task on that resource which provide minimum completion time for that task, Because of it Executes small tasks first, large tasks have to wait till small tasks are not completely executed. This is the disadvantage of this particular algorithm. Max-Min algorithm executes larger tasks first causes problem. The author’s proposed RASA algorithm. According to this algorithm at first the scheduler allocates the resource to tasks according to number of available resource. The available resource are add then it choose Min-Min algorithm procedure first .It allocates the resource to tasks in round one time smaller tasks in next round to large tasks it vice-versa. The simulation result shows this algorithm is outperform rather the two conventional algorithms.

El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahamawy [13] proposed an improved version of Max-Min algorithm named Extended Max-Min Scheduling Using Petri Net and Load Balancing. It works on the expected execution time rather than complete time as a selection basis. They used Petri nets which are well suited for modeling the concurrent behavior of distributed systems. The result shows that this algorithm achieving schedules with a lower makespan instead than RASA and original Max-min.

C.Kalpana, U.Karthick Kumar and R.Gogulan [14] introduced a method using load balancing algorithm for fair scheduling over max min. This algorithm tries to provide optimal solution and also tries to reduce the execution time and expected price for the execution of all jobs. The performance of this algorithm is better than Deadline First, Simple Fair Task Order, Adjust Fair task order and Max Min.

Yagoubi. B et al., [15] have offered a model to demonstrate grid architecture and an algorithm to schedule tasks within grid resources. The algorithm tries to distribute the workload of the grid environment amongst the grid resources, fairly. Although, the mechanism used here and other similar strategies which try to create load balancing within grid resources can improve the throughput of the whole grid environment, the total makespan of the system does not decrease, necessarily.

MCT (Minimum Completion Time) process assigns the tasks based on their minimum completion time which the resource ready time is available then completion time is calculated by adding the expected execution time of a job the machine with minimum completion time is choose. This algorithm selects only one job at a time. This assigns the tasks in arbitrary order so this algorithm assign tasks to machine that do not have the minimum expectation time for them [4].

MET(Minimum Execution Time) process assigns the tasks to the resource based on minimum execution time of task on that resource MCT(Minimum Completion time) algorithm first checks the availability of resource but MET assign the tasks on the resource without knowing the resource is available or not. It also checks the current load if the resource so it may cause the imbalance [4].

MAX- MIN heuristic method mostly used in task scheduling. This method worked on two parts; in first part calculate set of unmapped tasks with maximum execution time of all assigned tasks. On basis of that result second step performed in which task is choose on basis of the overall minimum finishing time among all tasks, Then the tasks are allocated to corresponding resource and that task is removed from the set .Again both 2 step is repeated on remaining tasks that are in the set[4].

Amid Khatibi Bardsiri1 and Marjan Kuchaki Rafsanjani [17] proposed an algorithm based on differential evaluation. In this paper a different evaluation algorithms presented based on scheduling for efficiently allocated jobs to resource in grid computing system. The goal of this algorithm is minimize the makespan for implementation of this scheduling algorithm various existing all on this are used. The result shows that this algorithm improves performance then other existing algorithm and it delivers improved makespan. The optimization technique of differential evaluation is used for multi objection parameter in grid schedule.

Lei Zhang, Yuehui Chen, Runyuan Sun, Shan Jing and Bo Yang [18] proposed a heuristic approach based on particles swarm optimization to solve high performance computing problems. The goal of this paper is design an algorithm based on PSO behavior of particle swarm optimization to see tasks scheduling problem in grid environment every particle give a possible solution and position vector is outperformed from solution and get minimum makes to complete the tasks.

E. Elmroth et al. [19] have proposed a user oriented algorithm for task scheduling for grid environments, using advanced reservation and resource selection .The algorithm minimizes the total execution time of the individual tasks without considering the total execution time of all of the submitted tasks. Therefore, the overall makespan of the system does not necessarily get small.

Siriluck Lorpunmanee, Mohd Noor Sap, Abdul Hanan Abdullah, and Chai Chompoo-inwai [20] proposed an ant colony optimization technique for dynamic job scheduling to allocate an appropriate resource to a specific task. So in this paper the author proposed an ant colony optimization algorithm to improve scheduling decision at run time.

Adil Yousif, Abdul Hanan Abdullah, Sulaiman Mohd Nor, Adil Abdelaziz [2] Proposed a solution for efficient task scheduling for computational grid such as if m number of tasks come at same time and demand for resources and resources are n at that time then how grid broker do mapping of coming tasks on minimum number of resources, so that processing delay to be less, produce a minimum makespan and flow time. We can schedule it on the basis of first come first serve basis if number of resource is a big number. To overcome this problem they proposed firefly an intelligent Meta heuristic method which is inspired by natural flashing behavior of fireflies. This method is based on population and on fire flies optimization technique which is used to produce global optimal solution on the basis of swarm intelligence, So that all tasks finished within the minimum makespan. In this method all fire flies do effort for searching of relevant resource nodes which produce minimum execution time for a task. They share the collective knowledge among the other fire flies. Firstly this method determine the completion time of all tasks over all fireflies. Then follows the fire flies algorithm (FA). Firefly optimization algorithm steps:-

- ➢ Firefly x attracts all fireflies.
- ➢ Firefly which have less brightness that is attracted by that firefly which is the brightest among those and moved to the bright one.
- ➢ The brightness of any firefly is decreased when distance among fireflies is increased.
- ➢ The firefly that is brightest among all no one can attract it.

3. PROPOSED ALGORITHM

To avoid the drawbacks of the Min-Min algorithm many improved algorithms have been proposed in the literature. All the problems discussed in those methods are taken and analyzed to give a more effective schedule. The algorithm proposed in this paper outperforms all those algorithms both in terms of makespan and load balancing. Thus a better load balancing is achieved and the total response time of the system is improved. The proposed algorithm applies the Min-Min strategy in the first phase and then reschedules by considering the maximum execution time that is less than the makespan obtained from the first phase.

So Proposed Algorithm executes Min-Min in the first round. In the second round it chooses the resources that produce makespan and find the tasks which run on that resource. Find the light load resource. Find the tasks next maximum completion time of every task which run on Rj and algorithm identifies the resource with heavily load by choosing the resources produce makespan by Min-Min and maximum completion of task is compared with the makespan that is produced by Min- Min and also check the new MCT of R_k is less than makespan on which task will be swapped if condition is fulfilled then the task is rescheduled on that resource which fill the condition and produce it and then update the ready time of the both resource. Otherwise the next maximum time of tasks are selected and again repeats the steps. The process stops if all resources and all tasks assigned in those for rescheduling thus the tasks possible resources are rescheduled on the resource which have minimum load. The Flow diagram (Figure 3.1) of the proposed algorithm is described here under.

Proposed Algorithm:

```
1. For all tasks Ti in meta-task Mv
2. for all resources Rj
3. Cij=Eij+rj
4. Do until all tasks in Mv are mapped
5. For each task in Mv find the earliest completion time and the resource that obtains it
6. Find the task Tk with the minimum earliest completion time
7. Assign task Tk to the resource Rl that gives the earliest completion time
8. Delete task Tk from Mv
9. Update rl
10. Update Cil for all i
11. End do
*****************Reschedule************
12. Sort the resources according to completion time
13. Calculate makespan
14. Choose the resource that provide the makespan
15. Choose the tasks run on that resource that provide the makespan
16.  Find next minimum completion time of those tasks and also find the resources provides it
17. Apply following condition one by one on those tasks.
18. If (next completion time of task< makespan and new completion time of machine<makespan)
19.  Reschedule the task to the resource that produced it
20. Update the ready time of both resources
21. End if
```

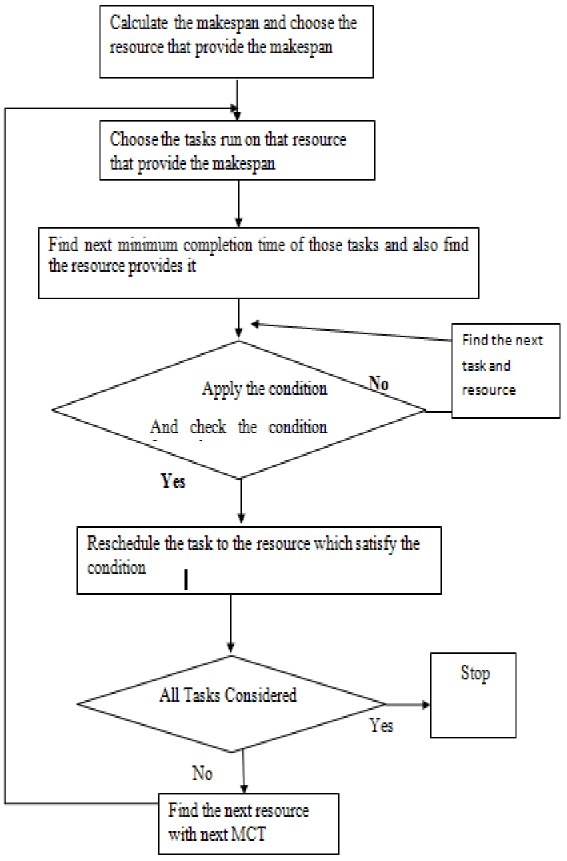

Calculate the makespan and choose the resource that provide the makespan
Choose the tasks run on that resource that provide the makespan
Find next minimum completion time of those tasks and also find the resource provides it
Find the next task and resource
No
Apply the condition
And check the condition
Yes
Reschedule the task to the resource which satisfy the condition
All Tasks Considered
Stop
Yes
No
Find the next resource with next MCT

Figure 3.1: Flow diagram

4. RESULTS

In this section, the experimental result of the proposed algorithm is discussed. The language is java for implementation and to execute both algorithms. The proposed algorithm was run to evaluate its performance for various test cases with different number of workflows with different number of resources. .One of them is discussed below.

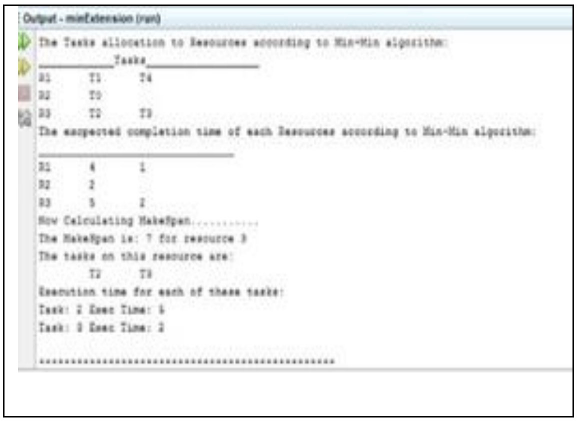
Assume there is a grid environment with three resources and five tasks T0, T1, T2, T3, and T4 are in Meta tasks, scheduler supposed to schedule all the tasks on three resources R1, R2, R3.

This **Figure 4.1** depicts tasks allocation to Resource according to Min-Min algorithm.

- On resource R1 two tasks T1 and T4 execute.
- On resource R2 only one task T0 executes.
- On resource R3 two tasks T2 and T3 execute.

This also shows expected completion time on resources according to Min-Min algorithm.

- On resource R1 two tasks T1 executes with 4 and T4 executes with 1 sec.
- On resource R2 only one task T0 executes with 2 sec.
- On resource R3 two tasks T2 executes with 5 and T3 executes with 2 sec.
- So Makespan that is produced by Min-Min algorithm is 7 sec



```
The Tasks allocation to Resources according to Min-Min algorithm:
__________Tasks_______________
R1      T1      T4
R2      T0
R3      T2      T3
The expected completion time of each Resources according to Min-Min algorithm:
______________________________
R1      4       1
R2      2
R3      5       2
Now Calculating MakeSpan...........
The MakeSpan is: 7 for resource 3
The tasks on this resource are:
        T2      T3
Execution time for each of these tasks:
Task: 2 Exec Time: 5
Task: 3 Exec Time: 2

*******************************************
```

Figure 4.1: Tasks allocation and expected completion time according to Min-Min algorithm

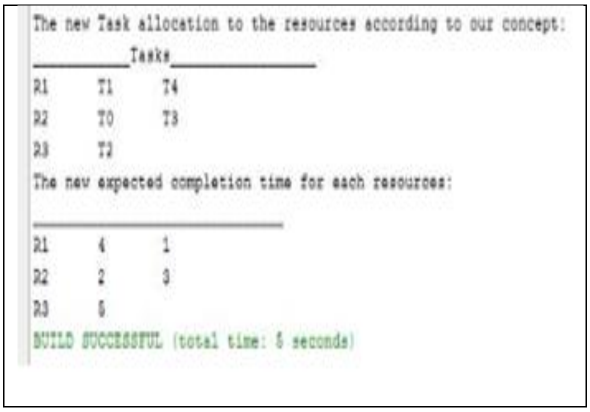
Figure 4.2 shows new tasks allocation to Resource according to proposed algorithm.

- On resource R1 two tasks T1 and T4 execute.
- On resource R2 two tasks T0 and T3 execute.
- On resource R3 only one task T2 executes.

This also shows new expected completion time on resources according to Min-Min algorithm.

- On resource R1 two tasks T1 executes with 4 and T4 executes with 1 sec.
- On resource R2 two tasks T0 executes with 2 and T3 executes with 3 sec.
- On resource R3 only one task T2 executes with 5 sec.

So makespan that is produced by proposed algorithm is 5 sec.

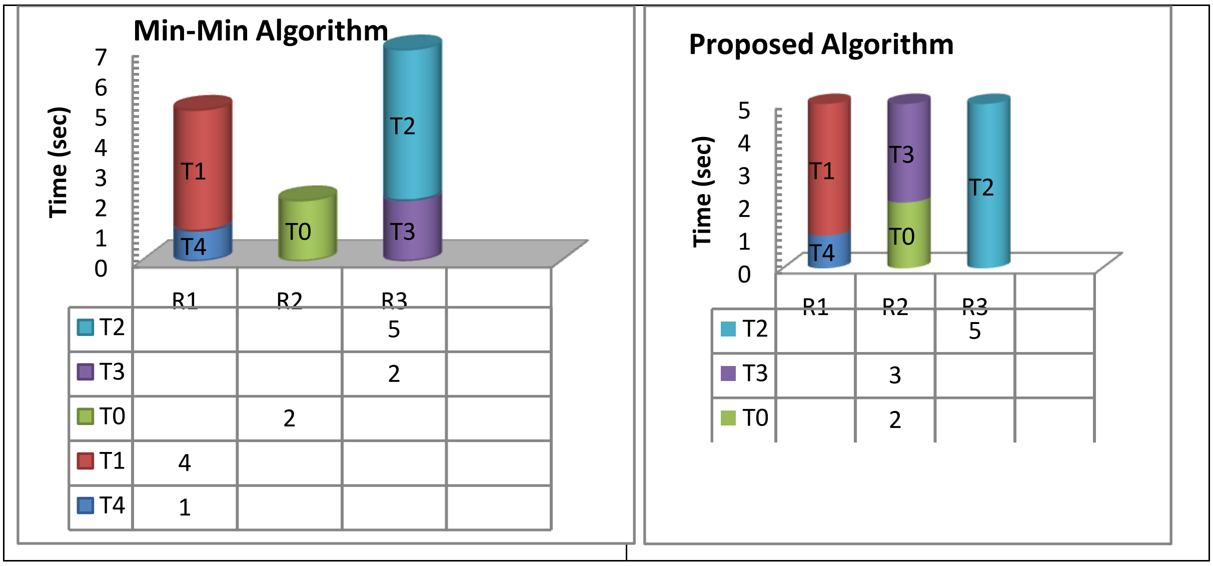


```
The new Task allocation to the resources according to our concept:
__________Tasks_______________
R1      T1      T4
R2      T0      T3
R3      T2
The new expected completion time for each resources:

______________________________
R1      4       1
R2      2       3
R3      5
BUILD SUCCESSFUL (total time: 5 seconds)
```

Figure 4.2: Shows new tasks allocation to Resource according to proposed algorithm

The graph below (Figure 4.3) describes the comparison of both two algorithms for five tasks execution time and produced makespan.



	R1	R2	R3	
T2			5	
T3			2	
T0		2		
T1	4			
T4	1			

	R1	R2	R3	
T2			5	
T3		3		
T0		2		

Figure 4.3: Shows comparison of tasks allocation to Resource according to Min-Min and proposed algorithm

5. CONCLUSION

The simulation result shows that the improved Min-Min scheduling minimizes the makespan with load balancing and guarantees the high system availability in system performance. The Improved Min-Min algorithm is compared with traditional Min-Min, by the experiment evaluation it shows that the new algorithm has a better quality of system load balancing and the utilization of system resources. The proposed method has two-phases. In the first phase the traditional Min-Min algorithm is executed and in the second phase the tasks are rescheduled to use the unutilized resources effectively.

With the help of better resource allocation for tasks and minimization of makespan for the workflow the performance and throughput is increased in the system. Hence the reliability of the system is also increased.

6. ACKNOWLEDGEMENT

In all humility and with much fervor, I owe my deep and sincere gratitude to Lovely Professional University, CSE department, Jalandhar India for the enlightened guidance, continuous encouragement, estimated supervision and paternal affection throughout the period of this research. The assistance of my mentor is very much beneficial for me to carry out the process of research in this field. Key improvements in the proposed research work would not be possible without the valuable suggestion and the feedback of him.

7. REFERENCES

[1] Muhammad K.Dhodhi, Imtiaz Ahmad and Anwar Yatama and Ishfaq Ahmad (2002)" An Integrated Technique for Task Matching and Scheduling onto Distributed Heterogeneous Computing Systems" Journal of Parallel and Distributed Computing 62.pp1338-1361

[2] Adil Yousif , Abdul Hanan Abdullah , Sulaiman Mohd Nor, Adil AbdelazizAdil " Intelligent Task Scheduling for Computational Grid"in ICCIT 2012.

[3] Parisa Rahmani 1, Mehdi Dadbakhsh , and Soulmaz Gheisari. (2012)" Improved MACO approach for grid scheduling"in 2012 International Conference on Industrial and Intelligent Information (ICIII 2012) IPCSIT vol.31

[4] Braun, T.D., Siegel, H.J., Beck, N., Boloni, L.L., Maheswaran, M., Reuther, A.I., Robertson, J.P., et al. (2001)"A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems", Journal of Parallel and Distributed Computing, Vol. 61, No. 6, pp.810–837

[5] D.Doreen Hephzibah Miriam and K.S.Easwarakumar (2010)."A double min min algorithm for task meta scheduler on hypercubic P2P grid system" International Journal of Computer Science Issues, vol 7 ,issue 4, no 5,pp-8-18

[6] T. Kokilavani, Dr. D.I. George Amalarethinam(2011) "Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing" International Journal of Computer Applications (0975 – 8887), Volume20, No.2, pp 43-49,

[7] He. X, X-He Sun, and Laszewski. G.V,(2003) "QoS Guided Minmin Heuristic for Grid Task Scheduling," Journal of Computer Science and Technology, Vol. 18, pp. 442-451,

[8] Kamalam.G.K and Muralibhaskaran.V, (2010),. "A New Heuristic Approach:Min-Mean Algorithm For Scheduling MetaTasks On Heterogenous Computing Systems," International Journal of Computer Science and Network Security, VOL.10, No.1,

[9] Sameer Singh Chauhan,R. Joshi. C, (2010.) "QoS Guided Heuristic Algorithms for Grid Task Scheduling," International Journal of Computer Applications (0975 – 8887), Volume2, No.9, pp 24-31,

[10] Singh. M and Suri. P.K, (2008) QPS A QoS Based Predictive Max-Min, Min-Min Switcher Algorithm for Job Scheduling in a Grid, Information Technology Journal, Volume: 7 Issue: 8, Page No.: 1176-1181.

[11] Abraham, A., Buyya, R. and Nath, B., (2000) "Nature's Heuristics for Scheduling Jobs on Computational Grids". Proceedings of the International Conference on Advanced Computing and Communications

[12] Saeed Parsa and Reza Entezari-Maleki.(2009.)" RASA: A New Task Scheduling Algorithm in Grid Environment" in World Applied Sciences Journal 7 (Special Issue of Computer & IT):pp 152-160,

[13] El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahamawy.(2012)" Extended Max-Min Scheduling Using Petri Netand Load Balancing"in International Journal of Soft Computing and Engineering (IJSCE)Volume-2, Issue-4, pp-198-203

[14] C.Kalpana1 U.Karthick Kumar2 and R.Gogulan3.(2012)" Max-Min Particle Swarm Optimization Algorithm with Load Balancing for Distributed Task Scheduling on the Grid Environment" in IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1,pp365-373

[15] Yagoubi, B. and Y. Slimani, 2007. "Task LoadBalancing Strategy for Grid Computing" in Journal of Computer Science, vol 3 issue-3, pp 186-194.

[16] Geoffrey Falzon, Maozhen Li, "Enhancing list scheduling heuristics for dependent job scheduling in grid computing environments", Journal of Supercomputing, Springer,March 2010

[17] Amid Khatibi Bardsiri1 and Marjan Kuchaki Rafsanjani.(2011) " Scheduling Independent Tasks on Grid Computing Systems by Differential Evolution" in KMITL Sci. Tech. J. Vol. 11 No. 1

[18] ei Zhang1, Yuehui Chen2, Runyuan Sun1, Shan Jing1 and Bo Yang1. (2008)" A Task Scheduling Algorithm Based on PSO for Grid Computing" in International Journal of Computational Intelligence Research (www.ijcir.info.), Vol.4, No.1, pp. 37–43

[19] Elmroth, E. and J. Tordsson, (2008)." Grid resource brokering algorithms enabling advance reservations and resource selection based on performance predictions". Journal of Future Generation Computer Systems, 24: pp-585-593

[20] Siriluck Lorpunmanee, Mohd Noor Sap, Abdul Hanan Abdullah, and Chai Chompoo-inwai(2007)" An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment" International Journal of Computer and Information Engineering, volume 1,no 8, pp-469-476