# Survey of Algorithm: Scheduling Systems and Distributed Resource Management in Grid

Nabeel Zanoon, Ph.D
Al- Balqa' Applied University
(BAU), Department of Applied
Science, Aqaba-Jordan

Nashat Al Bdour, Ph.D
TafilaTechnical University
Tafila - Jordan 66110
P.O.Box 179

Evon Abu-Taieh, Ph.D,
Faculty of Computer
Information Systems, Jordan
University -Aqaba
Aqaba-Jordan

## ABSTRACT

Grid computing is the form of distributed computing where the resources of various computers are shared to solve a particular problem. Grid can be used for a variety of purposes. Job scheduling is used to schedule the user jobs to appropriate resources in grid environment. The resources in the Grid are heterogeneous and geographically distributed with varying availability, and a variety of usage and cost policies for diverse users at different times vary with time. The management of resources and application scheduling in such a large and distributed environment is a complex task. In this paper, a survey of various job scheduling algorithms is made. The job scheduling algorithms are compared and contrasted based on the makes pan, flow time, resource utilization and completion time. They contribute to developing more efficient scheduling algorithms. This will help interested researchers carry out further work in this thrust area of research.

## General Terms

Your general terms must be any term which can be used for general classification of the submitted material such as Pattern Recognition, Security, Algorithms et. al.

## Keywords

Grid Environment, Resource Management, algorithm scheduling, task Scheduling..

## 1. INTRODUCTION

Grid computing is a collection of resources that are distributed and utilized for the execution of the request submitted by users from geographically multiple distributed points. Because the grid system was found to meet the requirements of the users, materials management is considered the active and main ingredient in the grid system. It stands on the acceptance of the applications from users to match between demands and available resources. [1]

The resources are geographically distributed and connected with a computing network environment, individuals and organizations that have their own access policy, cost and mechanism. The owners of resources management and control have to use their favorite resources scheduling system which is called local scheduling. It is expected that the users of the network will honor that by ensuring that it does not contradict with the policies of the owners of the resources. When users find a difference in the prices of two networks, global resources management and systems will fix that. The roles played by scheduling include planning and organizing tasks in distributed resources [2]. In the grid environment, scheduling process is classified into several stages: first, detect the resource. Second, select the resource and schedule under certain mechanism. Third, forward the request to the corresponding source [3].

The architectural model of resource management systems is affected by the scheduling structure. The hierarchy of scheduling depends on the number of resources with jobs and computer scheduled operations [4] the task order for hardware resources and the arrangement of communication between tasks such as scheduling, function and task share concepts common to resource management and scheduling. It is part of the work required to implement part of the functional or application [5]

Grid resources management system consists of grid scheduling and other elements. Grid scheduling works to detect and choose the most appropriate resources for the job. It affirms matching in the scheduling characteristics of resources in the application and characterizes the owner of the resource in the resource owner's policy. The resources are delivered to resources' policy owner via the information service. The candidates were selected from the matching process. The final winners will be selected based on the algorithm of schedule. The final grid scheduling stands for the reservation of the resources that need future appointments [6]. Grid scheduling reveals and chooses the most appropriate resources for the job based on important information collected about the status of the resources, especially in an environment like grid environment that is geographically distributed. Grid information system (GIS) provides information about the resources required to appoint and choose appropriate resources. This information represents the capacity of the central processor, memory and network capacity [7].

## 2. Related Work

S.Parsa et al., [8] through their studies show that the Max-min seems to do better than the Min-min algorithm whenever the number of shorter tasks is much more than the longer ones, but in the other cases, early execution of large tasks might increase the total response time of the system. Also, in the Max-min algorithm, the small tasks may wait for large ones to be executed. For example, if there is only one long task, the Max-min algorithm executes many short tasks concurrently with the long task. In this case, the makespan of the system is most likely determined by the execution time of the long task. However, since the Min-min algorithm attempts to assign the short tasks before the long ones, the makes pan increases compared with the Max-min.

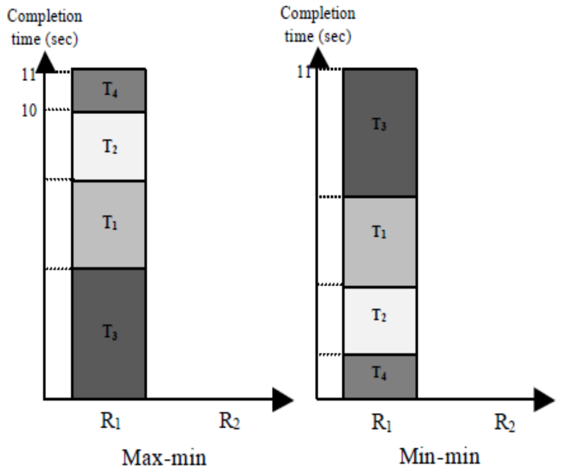

**Fig 1. Gantt chart of the Max-min and Min-min algorithms**

In Figure 1, Grant chart was used to represent Max-Min and Min-Min algorithm as both achieve scheduling with equal makes pan 11 seconds while the algorithm (RASA) achieve scheduling with the value of the makes pan 10 seconds.

The chart shows that RASA is even with other algorithms, or may be worse than described. In general, if tasks about the same size were provided, Min-Min Min-Max algorithms may work better than the RASA.

K.Etminani et al., [9] have studied some algorithms and held several experiments on the basis of measurement in terms of the number of factors that affect the performance of the algorithm as shown in the table (1).

**Table 1. Comparison between algorithms (Max-Min, Min-Min)**

| | Max-Min algorithm | Min- Min algorithm |
|---|---|---|
| A few short tasks along with many long tasks | | Min-Min outperforms Max-Min |
| A few long tasks along with many short tasks | Max-Min outperforms Min-Min | |
| A few long tasks along with many short tasks | Max-Min outperforms Min-Min | |
| load balancing level | Max-Min has better load balancing level than Min-Min | |
| average utilization rate | Max-Min is better than Min-Min algorithm | |

H.Nagda et al. [10] compared the utilization irate of resources and Makes pan by conducting several experiments for a number of tasks between 500 to 5,000. It was found that the Max –Min algorithm is better than the algorithm Min-Min based on the measured data, the utilization rate of resources and Makes pan.

The graphs clearly show that the value of makes pan rate and the utilization of resources in Max-min is better than in Min-Min algorithm, and that Max-Min is superior to Min-Min in terms of makes pan value.

We have previously noted that the Max-Min algorithm is better than the Min-Min algorithm. The larger the number of tasks, the more difficult it is to predict the results. When the number of resources is relatively more than the number of tasks, some of the resources may be idle, and some of the tasks may not be intended. In light of this, the lost time and management time are almost the same for both algorithms. Our analysis of the final results of the experiments shows that the Max-Min algorithm is better than the Min-Min algorithm.

G.Sharma, et al. [11] in his paper shows that the pace code for each of the algorithms (suffrage, max-min, min-min) which are compared by posting the examples in the form of multiple scenarios. The first example about the number of individual tasks shows that suffrage begins scheduling before Max begins. The remaining tasks are assigned to resources through one of two strategies: if the first task is assigned to a resource through a strategy of Max-Min, the next task shall be assigned by suffrage. In the next round, the task is assigned by using strategies that are different from those used in the final round.

K.Gupta et al., [12] used a simulation program that compares several algorithms in terms of the conditions of endurance in the framework of the difference in the number of tasks as shown in the table (2).

**Table 2. Comparison between algorithms (Max-Min, Min-Min, Suffrage)**

| Makespan | Max-Min | Min-Min | Suffrage |
|---|---|---|---|
| Performance under low load conditions | No. of Task= 30 No. of Machines = 10 | | |
| | 59.3 | 46 | 38.9 |
| Performance under medium load conditions | No. of Task= 70 No. of Machines = 10 | | |
| | 130.1 | 95.1 | 79.5 |
| Performance under high load conditions | No. of Task= 120 No. of Machines = 10 | | |
| | 248.1 | 159.7 | 152.5 |

By analyzing the data, we determined that the Suffrage algorithm in performance is better than in other algorithms.

M.Maheswaran et al. [13] compared natural Makes pan in terms of the difference of natural (normal) value of the interval estimated 10 seconds. He showed that the suffrage algorithm is superior to both Max-Min and Min-Min based on the value makes pan as shown in Figure (2.2)

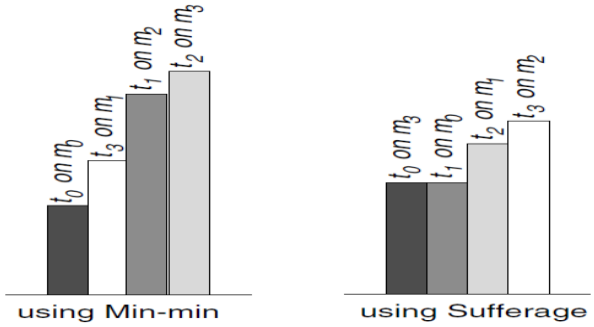

**Fig 2. An example of a scenario where the Suffrages gives a shorter makes pan than the Min-min**

Makes pan value in the Suffrage algorithm is less than Min-Min value. Makes pan value measures the amount of productivity in computerized networks. The more the value of the Makes pan, the better the performance of the algorithm. It is through value Makes pan that we can evaluate the performance of Makes pan.

## 3. GRID SCHEDULING SYSTEMS

Grid scheduling plays an important role in grid environment which is geographically distributed. Scheduling identifies resources to implement the application tasks. Grid scheduling system has more complex environments than traditional environments such as cluster and magic computer. Since the environment is geographically distributed, the resources are distributed over multiple span of domains where there is a difference in connection speeds in the network [14]. The nature of the grid environment is considered a challenge for task scheduling, as remote task competes with the local job or even with the same level of tasks. The factor that determines the degree of load is not only the execution time but also the number of existing resources and scheduling systems [15]. The grid schedulers can be classified under three categories. In the first category, there are schedulers for individual applications. In the second category, there are meta-schedulers that take many applications and try to resolve the contention effects between applications. In the third category of schedulers, there are schedulers for each resource. The resource can be anything from a single SMP or cluster to an enterprise-wide grid [16].

## 4. SCHEDULING ALGORITHMS

In order to make the best use of the geographical distribution of the grid network, role of the scheduling algorithms should be activated. Algorithms play an important role in sending tasks to grid resources, so that applications are performed independently and distributed through the sharing of resources, regardless of the owner [17]. The acceleration in the development of grid computing is a must since to demonstrate the high performance of the rid network the problem task scheduling should be bypassed. All tasks should be implemented in the shortest time possible [18]. In grid computing, the resource performs the operation, the wizard processes data for compatibility between the supplier and the task of gathering information about the available resource. The mechanism of resources identification and tasks assignment is the responsibility of grid resources management system [19]. Task scheduling achieves two goals: high-performance computing to reduce the execution time for each request and high productivity computing that aims to schedule a group of tasks to increase the processing capacity of the systems over a long period of time [20].

In grid environment, scheduling algorithms are divided into sections: prompt scheduling and batch scheduling method. Scheduling assigns tasks according to resources. In the immediate scheduling algorithm, the task is specified to the resource as soon as it is scheduled. However, in batch scheduling method, the algorithm groups tasks to be designated in specific times and scheduling roles [21]

## 4.1. Min-min algorithm

The min-min algorithm works on the basis of the estimation of execution and the end times for each task and each resource. It selects the task with the shortest completion time, so that the task is allocated to the resource that is able to implement the least time [22]. The min-min algorithm works by the policy of the fastest and the available resources, so it receives the least job. In order to understand that, we believe that the slowest nodes will receive the largest job. This method leads to a loss of execution time and to an inconsistency between the load balance and the speed of the node. In the light of that, load balancing will break down. To understand the min-min algorithm, see fig 3 [23].

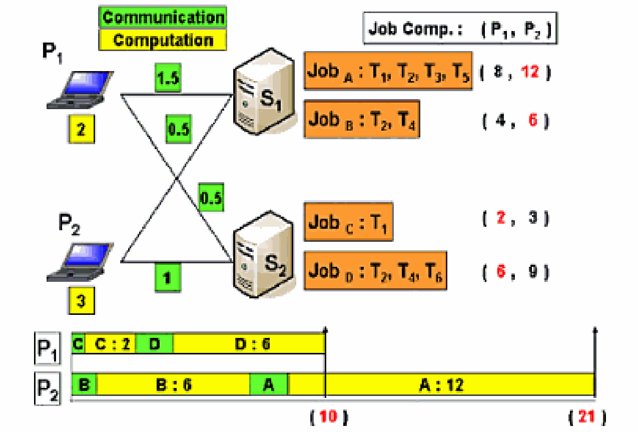

**Fig 3. Min-Min Job scheduling in a heterogeneous network**

The min-min algorithm could be used in multiprocessor systems and in homogeneous and heterogeneous distributive systems [24].

## 4.2. Max-min algorithm

The first phase in the max-min algorithm is similar to the first phase in min-min algorithms, but in max-min algorithm the tasks that are specified need the maximum time to perform and are set to the corresponding resource and matching with task [25] The difference between the two simple algorithms as in Figure 4 is that once the machine that provides the earliest completion time is found for every task, the task tk that has the maximum earliest completion time is determined and then assigned to the corresponding machine.

Algorithm 1. Min-Min Algorithm

```
For all task $t_i$ in meta-task $Mv$ (in an arbitrary order)
    For all machines $m_j$ (in a fixed arbitrary order)
        $C_{ij}=e_{ij}+r_j$
Do until all tasks in $Mv$ are mapped
    For each task in $Mv$ find the earliest completion
        time and the machine that obtains it
    find the task $t_k$ with the minimum earliest
        completion time
    assign task $t_k$ to the machine $m_j$ that gives the
        earliest completion time
    delete task $t_k$ from $M_v$
    update $r_j$
    update $C_{ij}$ for all $i$
end do
```

Algorithm 2. Max-min Algorithm

```
For all task $t_i$ in meta-task $Mv$ (in an arbitrary order)
    For all machines $m_j$ (in a fixed arbitrary order)
        $C_{ij}=e_{ij}+r_j$
Do until all tasks in $Mv$ are mapped
    For each task in $Mv$ find the earliest completion
    time and the machine that obtains it
    find the task $t_k$ with the maximum earliest
    completion time
    assign task $t_k$ to the machine $m_j$ that gives the
        earliest completion time
```

```
(9)  delete task $t_k$ from $M_v$
(10) update $r_j$
(11) update $C_{ij}$ for all $i$
(12) end do
```

**Fig 4. The difference between Min-min and Max-min heuristic**

That is, in Line (6) of Fig 4, (minimum) would be changed to (maximum). The Max-min has the same complexity as the Min-min. But if we give more importance to short tasks than to long tasks, the max-min algorithm will have better performance than the min-min [26].

### 4.3. Suffrage algorithm

This algorithm operates on the basis of the difference in completion time between the first and second time for that task [27]. This algorithm finds the minimum completion time for each task. The difference between the values of two tasks represents the suffrage value. Then the tasks with higher suffrage values are assigned to available resources [28]. Suffrage algorithm assigns tasks to resources according to job requirements. If the appropriate resource is not assigned, the suffrage value for each task is defined through the following equation [29]:

$$SV = \text{second MCT} - \text{MCT} \qquad (1)$$

The MCT (minimum completion time)

The performance and efficiency of the scheduling algorithms can be evaluated based on several criteria including differences among scheduling algorithms. The most important of them are makes pan, flow time and resource utilization. Makes pan is the time when HC system finishes the latest task, and flow time is the sum of execution times of all the tasks. An optimal scheduling will be the one that minimizes the flow time and makes pan. The objective of scheduling algorithm is to increase the system throughput measure [30].

## 5. CONCLUSION AND FUTURE WORK

Grid computing can solve complex tasks in shorter time and utilize the hardware efficiently. To make the grid work efficiently, best job scheduling strategies have to be employed. Job scheduling is the foremost step in grid computing where the users' jobs are scheduled to different machines. The various strategies have been studied and classified. The comparison of the algorithms has been studied. The future work will be concerned with the development of a better scheduling algorithm that is heterogeneous and works in dynamic environments.

## 6. REFERENCES

[1] K.Krauter, R.Buyya, M.Maheswaran, A Taxonomy and Survey of Grid Resource Management Systems, Software Practice and Experience, (2002).

[2] R.Buyya, D.Abramson, J.Giddy, Grid Resource Management, Scheduling and Computational Economy, International Workshop on Global and Cluster Computing (2000).

[3] Amir M Bidgoli, Zahra Masoudi Nezad, "A new scheduling algorithm design for grid computing tasks", 5th SASTech 2011, Khavaran Higher- education Institute, Mashhad, Iran. May 12-14.

[4] R.Buyya, D.Abramson, J.Giddy, An Economy Driven Resource Management Architecture for Global Computational Power Grids, international conference on parallel and distributed processing techniques and applications (2000).

[5] P .K. Vargas, I.Dutra, C.Geyer, Application partitioning and hierarchical management in grid environments, 1st international doctoral symposium on Middleware (2004).

[6] A.Anjum, Data Intensive and Network Aware (DIANA) Grid Scheduling, University of the West of England, (2007).

[7] P.Bansal, Different Scheduling Techniques in Grid Environment, International Journal of Latest Trends in Engineering and Technology (IJLTET), (2012).

[8] S.Parsa, R.Entezari- Maleki, RASA: A New Grid Task Scheduling Algorithm. International Journal of Digital Content Technology and its Applications. Volume 3, Number 4, (2009).

[9] K.Etminani, M. Naghibzadeh, N. A. Raeeji Yanehsari, "A Hybrid Min-Min Max-Min Algorithm with Improved Performance", Ubiquitous Computing and Communication Journal, 2009.

[10] H.Nagda, D.Visariya, Task Scheduling in Grid Computing Team Coda, Distributed Project Final Report, http://www.cs.rit.edu.

[11] G.Sharma,P.Banga, Task Aware Switcher Scheduling for Batch Mode Mapping in Computational Grid Environment, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, 2013.

[12] K.Gupta, M.Singh, Heuristic Based Task Scheduling In Grid, International Journal of Engineering and Technology (IJET), Vol 4 No 4, 2012.

[13] M.Maheswaranو S.Ali, Dynamic mapping of a class of independent tasks onto heterogeneous computing systems, Journal of Parallel and Distributed Computing, Volume 59, 1999.

[14] L.Thabet, C.Hanachi, K.Ghédira: Towards an Adaptive Grid Scheduling: Architecture and Protocols Specification. KES-AMSTA 2009.

[15] M.Wu, X. Sun, the GHS grid scheduling system: implementation and performance comparison, IPDPS'06 Proceedings of the 20th international conference on Parallel and distributed processing, IEEE, (2006).

[16] S.Prabhakar, Zone Based Scheduling: A Framework for Scalable Scheduling of SPMD parallel programs on the Grid, Thesis for the degree of Master of Science in Computer Science, 2003.

[17] S.Parsa, R.Entezari-Maleki, "RASA: A New Task Scheduling Algorithm in Grid Environment " World Applied Sciences Journal 7 (Special Issue of Computer & IT): 2009.

[18] Y.Zhang , Y.Inoguchi , Influence of Inaccurate Performance Prediction on Task Scheduling In A Grid Environment, IEICE TRANSACTIONS on Information and Systems Vol.E89-D.No.2, 2006.

[19] L.Zhang, Y.Chen, R.Sun, S.Jing, B.Yang,Task Scheduling Based on PSO Algorithm in Computational Grid, International Journal of Computational Intelligence Research,Vol.4, No.1 (2008)

[20] P.Raga Priya, S.Prathiba, Improving Grid Resource Allocation to monitor the task Scheduling and Binding, IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, 2013.

[21] R.Entezari-Maleki, A.Movaghar, A Genetic Algorithm to Increase the Throughput of the Computational Grids, International Journal of Grid and Distributed Computing Vol. 4, No. 2, (2011).

[22] S.Parsa, R.Entezari-Maleki, RASA: A New Grid Task Scheduling Algorithm. International Journal of Digital Content Technology and its Applications. Volume 3, Number 4, (2009).

[23] T.Chen, C.Hsu, Sh.Chen, “Scheduling of Job Combination and Dispatching Strategy for Grid and Cloud System”, Advances in Grid and Pervasive Computing: 5th International Conference, CPC 2010.

[24] R.Entezari-Maleki, A.Movaghar, A Genetic Algorithm to Increase the Throughput of the Computational Grids, International Journal of Grid and Distributed Computing, Vol. 4, No. 2, (2011).

[25] W.Sun, Y.Zhu, Z.Su, D.Jiao, M.Li, "A Priority-Based Task Scheduling Algorithm in Grid,", pp.311-315, 2010 3rd International Symposium on Parallel Architectures, Algorithms and Programming, 2010.

[26] M.Maheswaran, S. Ali, H. J. Siegel, D. Hensgen, R. F. Freund. "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems", Journal of Parallel and Distributed Computing, no. 59(2), pp. 107 – 121, 1999.

[27] A.Chaturvedi, R.Sahu, New Heuristic for Scheduling of Independent Tasks in Computational Grid, International Journal of Grid and Distributed Computing Vol. 4, No. 3, September, 2011.

[28] K.Gupta, M.Singh, Heuristic Based Task Scheduling In Grid, International Journal of Engineering and Technology (IJET), Vol 4 No 4, 2012.

[29] K. Etminani, M. Naghibzadeh, N. A. Raeeji Yanehsari, “A Hybrid Min-Min Max-Min Algorithm with Improved Performance”, Ubiquitous Computing and Communication Journal, 2009.

[30] D. Sharmila, M. Suguna ,S. Sivavidya , “ A study on Multi project Resource constrained project Scheduling using Metahewristic Approach”, in IJCSMC Volume 2, Issue 5, May2013.