Athens Institute for Education and Research
ATINER

ATINER's Conference Paper Series
COM2014-0912

A Study on Task scheduling Algorithms in Grid Computing

Abdulaziz Omar Alsadhan
Department of Software Engineering,
Collage of Information and Computer Sciences,
King Saud University, Riyadh, Saudi Arabia

Mohd Mudasir Shafi
Deanship of Distance and E-Learning,
King Saud University, Riyadh, Saudi Arabia

Athens Institute for Education and Research
8 Valaoritou Street, Kolonaki, 10671 Athens, Greece
Tel: + 30 210 3634210 Fax: + 30 210 3634209
Email: info@atiner.gr URL: www.atiner.gr
URL Conference Papers Series: www.atiner.gr/papers.htm
Printed in Athens, Greece by the Athens Institute for Education and Research.

ISSN: **2241-2891**

28/5/2014

An Introduction to ATINER's Conference Paper Series

ATINER started to publish this conference papers series in 2012. It includes only the papers submitted for publication after they were presented at one of the conferences organized by our Institute every year. The papers published in the series have not been refereed and are published as they were submitted by the author. The series serves two purposes. First, we want to disseminate the information as fast as possible. Second, by doing so, the authors can receive comments useful to revise their papers before they are considered for publication in one of ATINER's books, following our standard procedures of a blind review.

Dr. Gregory T. Papanikos
President
Athens Institute for Education and Research

This paper should be cited as follows:

Alsadhan A.O. and Shafi M.M. (2014) **"A Study on Task scheduling Algorithms in Grid Computing"** Athens: ATINER'S Conference Paper Series, No: **COM2014-0912.**

A Study on Task scheduling Algorithms in Grid Computing

Abdulaziz Omar Alsadhan
Department of Software Engineering,
Collage of Information and Computer Sciences,
King Saud University, Riyadh, Saudi Arabia

Mohd Mudasir Shafi
Deanship of Distance and E-Learning,
King Saud University, Riyadh, Saudi Arabia

Abstract

Grid computing technology is a form of distributed computing which presents freedom for the users to connect various remotely distributed computer resources to work in tandem to achieve a common goal or to solve a particular problem. A grid computing network can be made from a diverse processor types and different computers resources like storage drives, printers, RAM etc. connected by a superfast network spread across the world. Grid computing uses resources from unrelated computers which can be geographically scattered to carry out a substantial task at higher computing level making all these distributed computer resources to perform at a super computing level. The main idea behind grid computing is use these geographically dispersed resources efficiently so as to minimize the computational cost and operational complexity of very large and complex problems. Scheduling the tasks efficiently plays a vital role in achieving the results in a grid computing environment. To carry out the task and achieve the goal of grid computing, it's extremely important to use an efficient scheduling algorithm. Selecting a particular algorithm will determine the runtime of the operation and utilization of the resources in grid environment. This paper will first provide an overview of grid computing. Secondly, components of a typical grid will be discussed. Third, grid scheduling algorithms will be classified. Fourth, task scheduling algorithms in grids will be discussed. Based on this comprehensive study of grid computing and task scheduling algorithms, a conclusion and future course of study will be provided.

Key Words: Grid computing, distributed computing, task scheduling, grid algorithms.

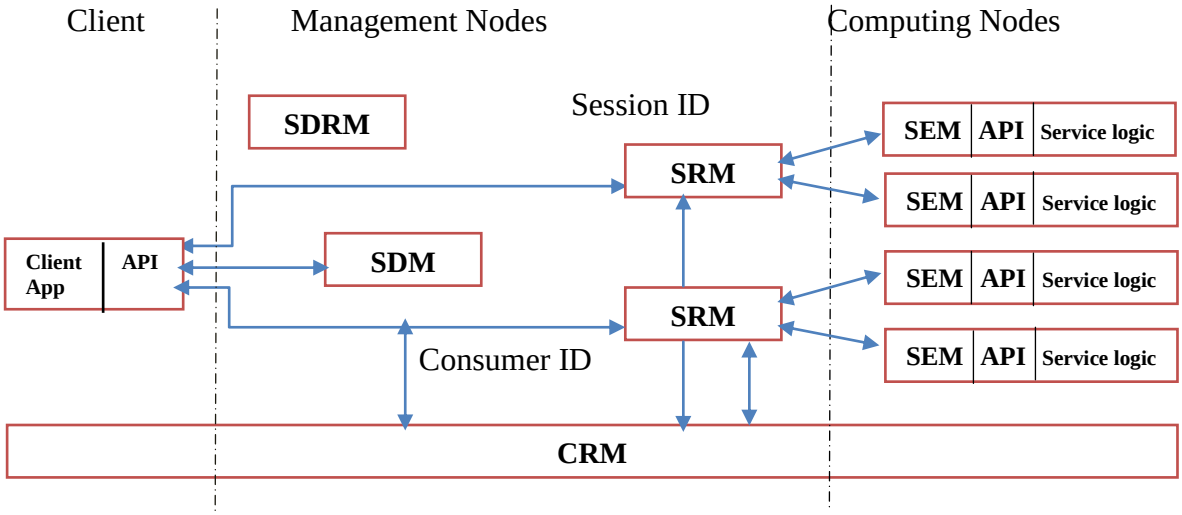
Contact Information of Corresponding author:
Mohd Mudasir Shafi, Deanship of Distance and E-Learning,
King Saud University, Riyadh, Saudi Arabia
Email: mmudasir@ksu.edu.sa

Introduction

The constant and persistent access to high speed internet as well as the possibility of having powerful computers at a relatively lower cost has changed the way computers are used these days. These advances in the field of computers and networks has provided us with the opportunity to use geographically scattered and platform independent computer resources to solve very complex problems (Dong & Akl 2006). The availability and use of these resources have led to the evolution of new superfast computing paradigm called grid computing (Buyya et.al 2002).

Grid computing is the new contemporary field of distributed computing which is mainly employed to solve the complex problems of science and engineering (Sha fan 2010). A Grid architecture consists of hardware and software framework based in different locations which provide a low-cost, steady and constant access to sophisticated problem solving (Singh & Sonia 2013; Buyya et.al 2005). Globus, Condor, Netsolve and Nimrod are some of the famous grid computing systems (Wu, Ming & Xian-He Sun 2003).

Figure 1. *Components of Grid (Sha Fan, 2010)*



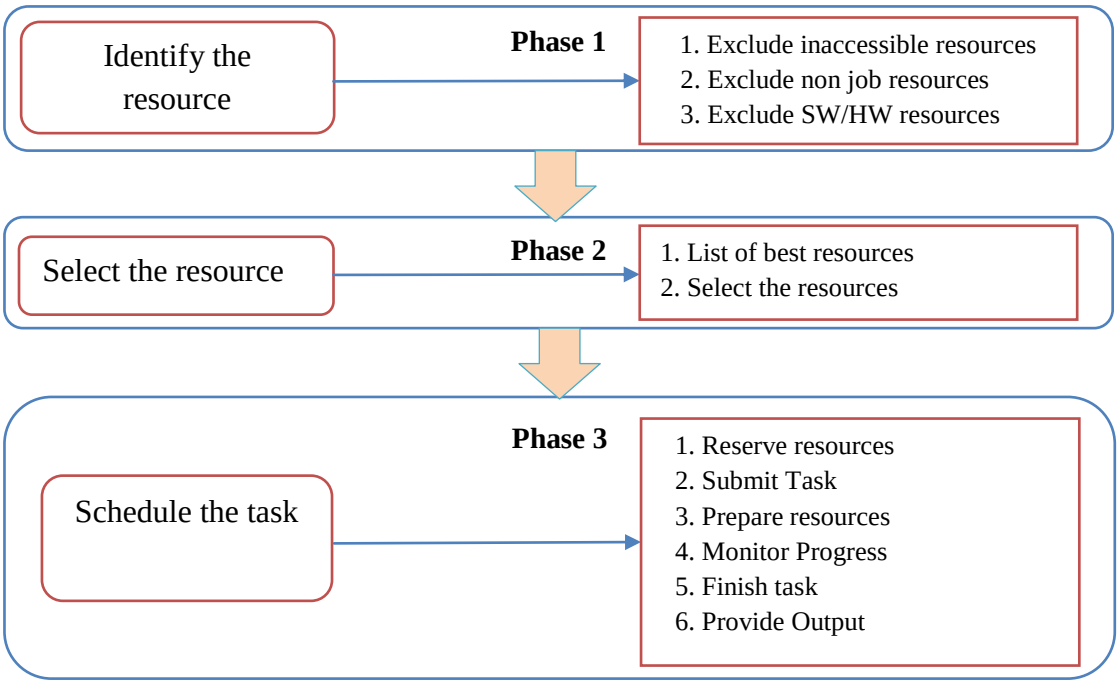
Components of a Grid Architecture

According to Sha fan (2010), a typical enterprise grid system consists of following five components (fig 2):

- *SDM (Service Direction Manager):* A client makes request for a service are made through SDM. It provides access to clients and clients can access other components of system through SDM.
- *SRM (Service Runtime Manager):* A SRM process is initiated after clients have gained access through SDM. A group of tasks are received by SRM and resource allocation is done by it.

- *SEM (Service Execution Manager):* A SEM is responsible for managing and monitor a service occurrence. Stopping a service when an error is encountered and reporting this error to SRM is done by SEM.
- *SRDM (Service Registration and Deployment Manager):* SRDM registers a service and sends the service to the management nodes.
- *CRM (Computing Resource Manager):* CRM regulates the resource allocation to the requests made by services according to the defined scheduling algorithm. A request for resources is made by Services to CRM and it's the responsibility of CRM to allocate the available resources.

Figure 2. *Task Scheduling Process*



Classification of scheduling Algorithms in Grids

Grid is a unique kind of distributed computing system (Dong & Akl 2006). Casavant & Kuhl, 1988 classified the following types of scheduling algorithms in distributed computing:

Hierarchical Algorithms

Local and Global

In local scheduling, native processes are allocated to CPU for certain time interval. Local scheduling methodology can be applied to grids. In global

scheduling, different process are allocated to different CPU's at different time intervals.

Static and Dynamic

In static scheduling, information about the task and resources available are already known and each task is assigned to the resources once only. Dynamic scheduling is applied when the execution time and resources need for completion of a job are not know. Resources are allocated to the tasks after making request for the resource for a certain amount of time.

Optimal and Suboptimal

If the information about the jobs and resources needed for execution are known, then optimal scheduling policy can be applied. Due to the complicated nature of grids, it's not always possible to assume the resource availability and optimization. In such cases, a suboptimal kind of scheduling strategy is feasible. Suboptimal can be further classified into two types which are given below in 4.

Approximate and Heuristic

Approximate algorithms does not look for an optimal solution but are contended with an acceptable "good" solution. These types of algorithms rely on some metric to choose the solution. The time taken to find a "good" solution is an important factor in choosing these algorithms.

Heuristic algorithms are based on making assumptions about the operation and the system attributes. These algorithms fall under the category of static algorithms and are dependent on special parameters. These kind of algorithms are used in very disparate and highly dynamic grids.

Distributed and Centralized

In centralized scheduling, as the name itself implies, the scheduling decisions are made by a single processor which makes them fairly easy to implement whereas in distributed, scheduling decisions are made by multiple processors.

Cooperative and Non-cooperative

In non-cooperative scheduling, each processor act alone, independent of other processors and make scheduling decision about the resources associated with it. In cooperative scheduling, each processor schedules its own task but communicates with other processor in order to achieve a common goal.

There are still some features of scheduling on grids that are not covered by hierarchical algorithms. Casavant & Kuhl, 1988 classified these algorithms as *"Flat Classification Characteristics"*.

Flat Classification Characteristics

1. Adaptive and non-Adaptive
   An adaptive scheduling algorithm alters the priority among parameters based on the current and previous state of the system. The priority of the parameter can be altered if it is giving unpredictable results or is deemed less important by the scheduler. A non-adaptive scheduler is the one where the priority or state of the parameters can't be changed or altered.
2. Load Balancing
   These types of algorithms aim for a fair use of resources in a way that load is distributed uniformly among the resources. If a resources is assumed to be overloaded by the scheduler, then load can be transferred to another resource which is relatively less used.
3. Bidding
   These schedulers define the way resources are used and the way tasks are assigned to processors. A call for available tasks is made by nodes called *managers* and a response is sent by nodes called *contractors* to receive the task for execution.
4. Probabilistic
   In these, a task or a process is assigned to the resources randomly. A different number of processes are assigned to resources randomly which causes different set of schedules to be made. Among these set of random schedules, the best resulting scheduling method is employed
5. Static and dynamic assignment
   In static assignment, a job containing different number of tasks is assigned to resources until its completion while as in dynamic assignment different jobs are moved dynamically among resources until completion.

Besides the above given classification, scheduling in grids can also be assorted as:

Objective functions: Two types: - Resource and Application centric (Y. Zhu, 2003; Dong & Akl 2006).

Task Dependent**:** Two types: - Independent and dependent task scheduling (Braun, Siegel et al 2001, Dong & Akl 2006).

Data Based scheduling: Two types: - Replication and no replication contemplated (Dong & Akl 2006).

Nontraditional Scheduling: Two types: - Grid economy and Nature inspired (Dong & Akl 2006; Buyya Abramson & Venugopal 2005; Braun, Siegel et al 2001)

Task Scheduling Process in Grids

Task scheduling is a very vast and an important area of research in grid scheduling. The idea behind scheduling in grids is to commit a task to an accessible and vacant resource with an objective of reducing the cost, execution time and maximizing the resource utilization (Legrand, Marchal & Casanova, 2003). There are three main phases (fig.2) of task scheduling in grids (Schopf, 2002; X. He, X. Sun & Laszewski, 2002; Dong & Akl, 2006; Forti, 2006):

1. Identifying the Resource:

This phase involves ascertaining the potential resource and creating a list of these resources. The list of available resources is stored in Grid Information System (GIS). After a list of resources is collected, following 3 steps are carried out:

- Exclude the resources that are inaccessible to the user.
- Exclude the resources which do not meet job preferences.
- Exclude the software / hardware resources which do not meet the minimum requirements set by the job.

2. Selecting the resource:

This phase involves the resource selection according to the type of task to be executed. Following two steps are carried out in this phase:

- Narrow down the list of best set of resources for job execution.
- Select the resources which ensure high performance execution of the task.

3. Scheduling the task:

In this phase, a task is assigned and executed on the selected resource according to the parameters defined in the scheduling algorithm. This phase finishes with following steps:

- Reserve the resources needed for task execution.
- Submit the task for execution.
- Prepare the system for task execution.
- Observe the progress of execution.
- Finish the task execution and provide results.
- Provide output and release the used resources to be available for other tasks.

Task scheduling Algorithms in Grids

Task scheduling algorithms in grids define the order in which each task is executed. Task scheduling in grids can be either reliant of other tasks i.e.

dependent task scheduling or it may also be autonomous of the other tasks in the system called independent task scheduling (Dong & Akl 2006).

Dependent Task Scheduling

In this type of scheduling, a task to be executed has preeminence over the other tasks in a particular job. That implies that a task to be assigned to the resources have priority over the other tasks in a particular job or a system (Dong & Akl 2006). Given below are some of the most commonly used algorithms that fall under this category:

- Highest Level First with Estimated Times (HLFET) Algorithm

HLEFT is one of the elementary and earliest algorithms suggested for parallel distributed systems (Pop, Dobre, Cristea, 2008). HLEFT works by assigning priority to each task at a static level based on computing the cost of all the tasks (Singh and Sonia 2013). HLEFT work in following three steps (Sharma, Singh and Kaur 2012):

1. Conclude the static level of each task
2. Arrange the list of tasks in descending order.
3. Schedule the task to the processor which allows earliest start time for execution

- Fast Critical Path (FCP)

This algorithm was proposed by Radulescu & Gemund, 1999. FCP is a compile time list scheduling algorithm. The main ideas behind FCP are:

1. Sorting limited number of tasks at any given time
2. Selecting the processor from which the last message was received or which becomes idle at first.

This algorithm works by maintaining a sorted list of tasks of fixed size. Each reach task is then added to this sorted list. If there isn't any space available in this sorted list then each task is added on a separate list on First in first out (FIFO) basis. When a place is available is available in sorted list, the first task in FIFO list is moved to the sorted list and assigned to the available processor. This reduces the time intricacy as each task is moved in and out from sorted list only once.

- Task Duplication based Scheduling (TDS)

Darbha & Agrawal 1998, proposed this algorithm. This algorithm is based on the presumption of unlimited availability of the memory and a fixed cost among two processors. This algorithm is based on following three steps (Yajman 1999):

1. In the first step, *est (earliest start),* ect *(earliest completion), fpred (favorite predecessor)* parameters are computed.

2. In this step, each task is traversed in a bottom up fashion to compute *lact (latest allowable completion), level (level of any node)* Parameters.
3. In the last step, all tasks are stored in a queue are assigned to *fpred (favorite predecessor)* until all the tasks in queue are executed.

Other notable dependent task scheduling algorithms are TANH *(Task duplication-based scheduling Algorithm for Network of Heterogeneous systems.* Ranaweera & Agrawal 2000), DSC (*Dominant Sequence Clustering,* Yang & A. Gerasoulis 1994*),* TANH + HSPA1 & HSPA2 (*Posterior Task Scheduling Algorithms for Heterogeneous Computing Systems.* Shen &Young 2007*).*

Independent Task Scheduling

In this type of scheduling, a set of independent tasks appear into the system. These tasks are assigned to the system on the basis of a common strategy to achieve a higher operational throughput (Dong & Akl 2006). Some of the important independent task scheduling algorithms are given below:

- *Minimum Execution Time (MET)*

In this algorithm, each task is assigned to the resource which is most likely to complete it in a minimum possible time. This algorithm doesn't take into account the availability of the resource i.e. whether the resource is available at that particular moment of time and it goes ahead with assigning the task to that resource. The idea behind this algorithm is to assign a best resource to the task. The problem with this algorithm is that if a resource is expected to execute a task faster than all other tasks will be assigned to it. This may cause a serious uneven distribution of load (Dong & Akl 2006; Gharehchopogh et.al. 2013).

- Minimum Completion Time (MCT)

MCT chooses a task which is expected to complete execution in a least amount of time from a group of tasks i.e. a task having minimum time for its completion. In this algorithm, a task is assigned to a resource which is expected to execute the task sooner than other resources (Gharehchopogh et.al. 2013). This aim of this algorithm is to benefit from the advantages of load balancing along with MET (Dong & Akl 2006).

- Min-Min Algorithm

This algorithm aims to execute larger number of tasks in a less amount of time. This algorithm begins by making a list of all the unassigned task in a set U. Next, a set minimum completion time (*M)* for each task in *U* is computed. Now, the minimum completion time of a task in set *M* is found and that task is assigned to a resource for its completion. The task assigned to the resource is then removed from the set of unassigned task *U.* This process is repeated until all the tasks are assigned to the resources and set *U* is empty. (Dong & Akl 2006).

- Max-Min Algorithm

This algorithm resembles Min-Min algorithm except the way tasks are assigned to the resources. This algorithm also starts by making a list of all the unassigned task in a set U and computing a set minimum completion time (*M)* for each task. In the next step, it differs from Min-Min by choosing a task with maximum expected completion time and assigning that task to resources until the set of all tasks *U* is empty (Dong & Akl 2006). This algorithm allows a task to be assigned to a resource which is seen as the best resource to execute that task (Gharehchopogh et.al. 2013).

- XSuffrage

Suffrage is a popular independent task scheduling algorithm. The suffrage algorithm works on the concept that a particular task must always be authorized for its intended resource. If the task fails to reach its intended resource then it is bound to suffer. The suffrage value of each task is calculated as the difference between MCT and second best MCT. Higher suffrage value means higher priority for the task execution (Dong & Akl 2006). The issue with this algorithm arises when there is input/output data for tasks which results in clustering of resources. This results in reducing the priority of the tasks and decreases the performance of suffrage algorithm (Gharehchopogh et.al. 2013). To overcome this problem Casanova et.al 2000, proposed *XSuffrage. XSuffrage* works by calculating suffrage value of each task in cluster. This value is called *cluster level suffrage* and is calculated as the difference between best and second best MCT of the cluster. Priority among the tasks is determined by the highest value and cluster level suffrage and task are assigned to the resources based on this value (Casanova et.al 2000).

Conclusion & Future Work

Grids are a heterogeneous organization of resources which can be used to achieve a common goal. Grids can be used to solve high to solve high resource oriented problems in a fast and efficient way. To achieve full potential of any grid, a best task scheduling strategy must be applied. Scheduling algorithms are in charge of administering the resources and distributing task among each resource efficiently.

Despite being a highly researched area, task scheduling in grids is still a highly attractive and compelling topic. In this study, we first briefly discussed the architecture of an enterprise grid. Next we studied the types of scheduling algorithms in grids. At last, we studied and discussed different task scheduling algorithms. A particular task to be scheduled on grid can be either dependent on other tasks called dependent task scheduling or it can't be affected by other tasks called independent task scheduling. The most important and commonly discussed algorithms from both dependent and independent task scheduling were discussed in this paper. In the future course of study, we would like to evaluate and compare the performance of each algorithm against other algorithms. Further we would also like to propose a new and efficient task scheduling algorithm for grids.

References

R. Buyya and D. Abramson and J. Giddy and H. Stockinger, *Economic Models for Resource Management and Scheduling in Grid Computing*, in J. of Concurrency and Computation: Practice and Experience, Volume 14, Issue.13-15, pp. 1507-1542, Wiley Press, December 2002

Wu, Ming; Xian-He Sun, 2003 "A general self-adaptive task scheduling system for non-dedicated heterogeneous computing," Cluster Computing, 2003. Proceedings. 2003 IEEE International Conference on , vol., no., pp.354,361, 1-4 Dec. 2003 doi: 10.1109/CLUSTR.2003.1253334

Domenici A., Donno F., Pucciani G., Stockinger H., Stockinger K., 2004. *Replica consistency in a Data Grid*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 534, http://dx.doi.org/10.1016/j.nima.2004.07.052.

Sha Fan, "*Session Scheduling Algorithm of Grid Computing,"* Knowledge Discovery and Data Mining, 2010. WKDD '10. Third International Conference on , vol., no., pp.3,6, 9-10 Jan. 2010. doi: 10.1109/WKDD.2010.87

Dong, F., & Akl, S. G. (2006). *Scheduling algorithms for grid computing: State of the art and open problems*. School of Computing, Queen's University, Kingston, Ontario.

Legrand, A.; Marchal, L.; Casanova, H., "*Scheduling distributed applications: the SimGrid simulation framework*," Cluster Computing and the Grid, 2003. Proceedings. CCGrid 2003. 3rd IEEE/ACM International Symposium on , vol., no., pp.138,145, 12-15 May 2003. doi: 10.1109/CCGRID.2003.1199362

Schopf J. M. "*A General Architecture for Scheduling on the Grid*", special issue of JPDC on Grid Computing, April, 2002

X. He, X. Sun, G. Laszewski, *A QoS guided scheduling algorithm for grid computing, in: Int. Workshop on Grid and Cooperative Computing*, GCC02, Hainan, China, 2002

T. Casavant, and J. Kuhl, *A Taxonomy of Scheduling in General-purpose Distributed Computing Systems*, in IEEE Trans. on Software Engineering Vol. 14, No.2, pp. 141--154, February 1988

Y. Zhu, *A Survey on Grid Scheduling Systems*, Department of Computer Science, Hong Kong University of science and Technology, 2003.

R. Braun, H. Siegel, N. Beck, L. Boloni, M. Maheswaran, A. Reuther, J. Robertson,M. Theys, B. Yao, D. Hensgen and R. Freund, *A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems*, in J. of Parallel and Distributed Computing, vol.61, No. 6, pp. 810-837, 2001

R. Buyya, D. Abramson, and S. Venugopal, 2005. *The Grid Economy*, in Proc. of the IEEE, Vol. 93, No. 3, pp. 698-714, IEEE Press, New York, USA, March 2005.

Forti A. (2006) *DAG Scheduling for Grid Computing systems*. Ph.D. Thesis, University of Udine - Italy,

Singh K. and Sonia 2013. *Optimized Performance Task Scheduling algorithm for Grid Computing. American International* Journal of Research in Science, Technology, Engineering & Mathematics. 2013 issue 2 volume 2

Pop, F.; Dobre, C.; Cristea, V. 2008, "*Performance Analysis of Grid DAG Scheduling Algorithms using MONARC Simulation Tool*," Parallel and Distributed Computing, 2008. ISPDC '08. International Symposium on , vol., no., pp.131,138, 1-5 July 2008 doi: 10.1109/ISPDC.2008.15

Sharma M., Singh G. and Kaur H. 2012 *A study of bnp parallel Task scheduling algorithms Metric's for Distributed database system* International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.1, January 2012 DOI : 10.5121/ijdps.2012.3112 157

A. Radulescu and A. J. C. van Gemund, *On the Complexity of List Scheduling Algorithms for Distributed Memory Systems*, in Proc. of 13th International Conference on Supercomputing, pp. 68-75, Portland, Oregon, USA, November 1999.

S. Darbha and D.P. Agrawal 1998, *Optimal Scheduling Algorithm for Distributed Memory Machines*, in IEEE Trans. on Parallel and Distributed Systems, vol. 9, no. 1, pp. 87-95,January 1998

Yajman P.1999. A *Study of Scheduling Algorithms for Multiprocessor Systems and distributed Memory Machines*. Online Available: *www.uta.edu/cse/levine/aos/reports/yajm_aos-paper-praveen*

S. Ranaweera and D. P. Agrawal 2000, *A Task Duplication Based Scheduling Algorithm for Heterogeneous Systems*, in Proc. of 14th International Parallel and Distributed Processing Symposium (IPDPS'00), pp. 445-450, Cancun, Mexico, May 2000.

T. Yang and A. Gerasoulis 1994, *DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors*, in IEEE Trans. on Parallel and Distributed Systems, vol. 5, no.9, pp.951--967, 1994

Shen L., Young T., 2007. *Choe Posterior Task Scheduling Algorithms for Heterogeneous Computing Systems.* High Performance Computing for Computational Science - VECPAR 2006. Lecture Notes in Computer Science Volume 4395, 2007, pp 172-183

Gharehchopogh F.S., Ahadi M., Maleki I., Habibpour R., Kamalinia A, 2013 *Analysis of Scheduling Algorithms in Grid Computing Environment* International Journal of Innovation and Applied Studies Vol. 4 No. 3 Nov. 2013, pp. 560-567

H. Casanova, A. Legrand, D. Zagorodnov, and F. Berman 2000. *"Heuristics for Scheduling Parameter Sweep Applications in Grid Environments*", Ninth Heterogeneous Computing Workshop, IEEE Computer Society Press, pp. 349-363, 2000