ARPN Journal of Engineering and Applied Sciences

SURVEY ON JOB SCHEDULING MECHANISMS IN GRID ENVIRONMENT

Argungu S. M.[1, 2], Suki Arif[1] and Mohd. Hasbullah Omar[1]
[1]Internetworks Research Laboratory, School of Computing, Universiti Utara Malaysia, UUM Sintok, Kedah, Malaysia
[2]Department of Computer Science, University of Science and Technology, Aliero, Kebbi State, Nigeria
E-Mail: sm279arg@gmail.com

ABSTRACT

Grid systems provide geographically distributed resources for both computational intensive and data-intensive applications. These applications generate large data sets. However, the high latency imposed by the underlying technologies; upon which the grid system is built (such as the Internet and WWW), induced impediment in the effective access to such huge and widely distributed data. To minimize this impediment, jobs need to be scheduled across grid environments to achieve efficient data access. Scheduling multiple data requests submitted by grid users onto the grid environment is NP-hard. Thus, there is no best scheduling algorithm that cuts across all grids computing environments. Job scheduling is one of the key research area in grid computing. In the recent past many researchers have proposed different mechanisms to help scheduling of user jobs in grid systems. Some characteristic features of the grid components; such as machines types and nature of jobs at hand means that a choice needs to be made for an appropriate scheduling algorithm to march a given grid environment. The aim of scheduling is to achieve maximum possible system throughput and to match the application needs with the available computing resources. This paper is motivated by the need to explore the various job scheduling techniques alongside their area of implementation. The paper will systematically analyze the strengths and weaknesses of some selected approaches in the area of grid jobs scheduling. This helps researchers better understand the concept of scheduling, and can contribute in developing more efficient and practical scheduling algorithms. This will also benefit interested researchers to carry out further work in this dynamic research area.

Keywords: data grids, job scheduling, data replication.

INTRODUCTION

Grid computing (GC) is a new trend in distributed computing system (DCS) that enables the management of heterogeneous, geographically distributed and dynamically available resources in an efficient way (Kant Soni, *et al*, 2010). It expands the boundaries of what we perceived as distributed computing and supercomputing. Resource management and job scheduling are the most fundamental concerns when deploying grid infrastructure. Jobs could be defined as packages that are executed using appropriate computing elements (CE) at a point on the grid. Jobs may evaluate an expression, run a single or multiple commands to perform a given task, analyze data, or control of scientific equipment. The terms such as transactions, work entities, or submissions are used in the grid industry to mean the same thing as jobs. In whatever form, these Jobs need to be scheduled onto the grid environment prior to execution. However, the scheduling mechanisms that are typically deployed along with proprietary Grid Management Software (GMS) have limitations when the number of grid jobs is large (Jacob, *et al*, 2005). In recent years, researchers have proposed several efficient scheduling algorithms that are used in grid computing to allocate grid resources with a special emphasis on job scheduling (Abba, *et al.*, 2012; Abdurrab and Xie, 2010; Anikode and Tang, 2011; Aparnaa and Kousalya, 2014; Chang, *et al.*, 2008; Chen, *et al*., 2009; Coutinho, et al., 2014; Dang, *et al*., 2007; Elghirani, *et al*, 2007; Maheshbhai, 2011; Chen, *et al*., 2009; Mansouri, *et al*., 2011). With further advancement in the field of grid computing, many universities and public institutions are likely to leverage on the grids for enhancement of their computing infrastructure. This paper is organized as follows: Section II provides an overview of the grid system resources and classifications. Section III presents a detailed study of some proposed job scheduling algorithms in Grid Computing. Section IV provides an analysis and implementation-wise discussion among all the surveyed papers. Section V concludes this study with a briefing on research focus and finally the references.

Grid system resources

Grid systems in general, leverage on efficient utilization of a pool of heterogeneous systems with the aim of optimizing workload management. It takes advantage of the computational facilities within an organization such as the servers, network nodes, and storage elements or data backups. These resources collectively come together to produce robust computing environment (Jacob, *et al.*, 2005). Grid computing enables individual users and organizations to utilize the untapped CPU cycles, databases, scientific instruments and storage elements (SE) from millions of computer systems situated across the global network, at a minimal access cost. The end users pay little or no attention to the locations or sources of these services. Grid computing is synonymous to the way

ARPN Journal of Engineering and Applied Sciences

electrical grids are shared by various power companies. It presents a model for data and computing resource sharing, irrespective of their location or origin. Also, Grid technology has emanated from existing technologies such as the distributed computing, World Wide Web (WWW), various cryptography applications and the Internet (Magoules, F. *et al*., 2010). Grid users usually submit their jobs to the grid operating system via an interface. Then, the grid system will be responsible for locating suitable/ available computing resources that can serve the needs of the users.

Grid system classifications

Grids systems are typically classified according to the following groups (Jacob, *et al*., 2005) namely; Computational Grids, Data Grids and Services Grids.

Computational grids are systems of computing that bring together various machines from different organizational setup in a cycle-stealing mode. The power of this conglomeration of machines will be higher than a typical mainframe or minicomputer. The focus is to provide robust computing environment.

Data grids systems are developed to provide solutions for large-scale data management problems. The infrastructure comprises both hardware and software facilities that make use of available data stores to synthesize new information for sharing via a LAN or a wide area network.

Services grids refer to systems of computing in which different services are available for the users that cannot be provided by an individual local machine.

Also, the grid generally has been explained in (Qureshi, *et al*., 2014) to comprise two broad resource types, namely, *hardware* and *software* resources as shown on Figure-1. In the same write up, grid systems have been classified under four distinctive groups types thus; (i)-Computational Grids, (ii) Access Grids, (iii) Data Grids and (iv) Data-centric Grids. Another classification by (Magoules, F. *et al*), placed grids into four additional groups as follows: (v) Application Service Grids, (vi) Interaction Grids (vii) Knowledge Grids and (viii) Utility Grids.

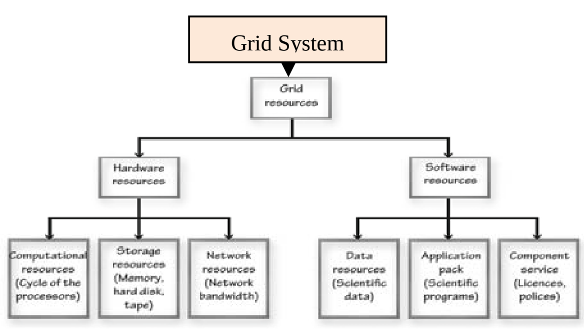

Figure-1. Grid resources types (Qureshi, *et al*., 2014).

Job scheduling techniques

Job scheduling has been described by (Kant Soni, *et al*., 2010) as the process of mapping jobs onto specific available computing resources, aimed at minimizing costs on the side of the end users. Need for job scheduling and effective computation is rapidly becoming one of the main challenges in GC and equally as vital for its success. According to (Jacob, *et al*., 2005), although various kinds of resources on the grid may be available for sharing and usage, they are usually accessed via an interfacing application or *job.* The term *application* is leveled as the highest piece of task on the grid. However, sometimes the term job is used equivalently. Applications may be broken down into any number of individual jobs, as illustrated in Figure-2. In a workflow environment, some jobs may require the output from other jobs and cannot be executed until the prerequisite jobs have been completed.

In the simplest of grid systems implementation, users may directly submit their jobs to a suitable machine for execution. More advanced grid systems would usually include a robust job scheduler for mapping various jobs onto the grid environment. In grid system, the term resource brokering is sometimes used interchangeably with scheduler.

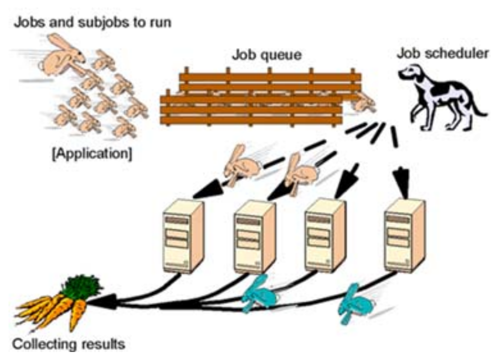

Figure-2. Example of job scheduling on grid environment (Jacob, *et al*., 2005).

Integrating scheduling and replication in data grids

The scheduling and replication method (Zarina, et al., 2013) was formulated to study the problem of integrating job scheduling and data replication in data intensive scientific applications. The objective is to minimize total job execution time by placing data replicas and jobs onto appropriate sites. This was designed to guarantee overall grid performance by developing a set of efficient heuristics and the results were validated using simulations and analysis. Strengths include: (i) reduction in total job execution time; (ii) Scheduling independent jobs; (iii) Load balancing. Weakness: Did not consider job dependencies in workflow environment.

ARPN Journal of Engineering and Applied Sciences

Scheduling based on reliability, time and cost constraints

The reliability, time and cost constraints based-scheduling problem (Maheshbhai, 2011) involves a workflow application. Different Grid Service Providers (GSPs) take advantage of this approach by placing higher price for short-make-span and lower price for long make-span for the same type of service. The scheduling problem allocates services to GSPs in such a way that workflow could be achieved within the users' deadline requirements; thereby minimizing associated cost. The motive is to find a schedule that enhances the user-preferred Quality of Service (QoS) parameter and satisfies all QoS constraints. This approach is summarized on Figure-3.

$$\mathrm{OS} = \begin{cases} \dfrac{\mathrm{min\_Reliability}}{\mathrm{max\_Reliability}}, \ \textit{Reliability Optimization} \\ \dfrac{\mathrm{min\_Makespan}}{\mathrm{max\_Makespan}}, \ \textit{Makespan Optimization} \\ \dfrac{\mathrm{min\_Cost}}{\mathrm{max\_Cost}}, \ \textit{Cost Optimization} \end{cases} = \quad (1)$$

Figure-3. Reliability_time_cost_scheduling.

Where min_Reliability = minimum reliability of all services, max_Reliability = 100%, min_Makespan = estimated minimum makespan of the workflow application and max_Makespan = the estimated maximum makespan.

The approach has the advantage of reducing jobs processing time, although it fails to address bandwidth consumption issues.

Job scheduling for dynamic data replication strategy in grid systems

A model for federation data grid system called Sub-grid-federation had been proposed by (Zarina, *et al.*, 2013), to improve data access latency on the grid. This approach performs by accessing data from an area recognized as 'Network Core Area' (NCA). Access time is optimized by accessing data from the nearest node, thereby norrowing the searched zone areas. By relegating the search to the nearest node, the need for a greater bandwidth has been reduced. Thus, it merits include: Reduced access time and bandwidth consumption, while its weaknesses include (i) failure to address storage utilization, (ii) Relegating the research to a network core area may constitute bottlenecks.

Enhanced adaptive scoring job scheduling algorithm

Enhanced Adaptive Scoring Job Scheduling algorithm (Aparnaa and Kousalya, 2014) partitions jobs according to data intensive or computational intensive and based on that jobs are scheduled. It does so by computing Job Score (JS) along with the memory requirement of each cluster. Due to the dynamic nature of grid environment, each time the status of the resources changes the job score (JS) is also computed and jobs are allocated to the most appropriate resources. The approach has the advantage of (i) scheduling both data intensive and compute intensive jobs (ii) effectively reduced job failure rate and makespan time. However, it failed to consider independent jobs in a workflow environment.

A job schedule model based on grid environment

This approach was proposed by (Chen, *et al*., 2009) to optimize CPU utilization and throughput at the same time and minimizes turnaround time. Grid nodes are divided into supervisor node (S_0), supervisor backup node (B_1), and execute nodes (X_1). The approach uses to its advantage the backup node in the event of superior node failure. Also, it has the strength of maximizing CPU utilization as well as job throughput. Its drawbacks include high communication overhead, and failure to address jobs/ resources constraints.

Heuristic algorithm for scheduling on grid computing environment (Wang, *et al.*, 2012)

This was designed based on the heuristics harder-first-faster-prior (HFFP). This approach was developed based on the premise that the whole grid resource scheduling problem involved three basic entities such as the job, the scheduler and the resources. These entities are shown on Figure-4. In this approach, jobs are scheduled based on their complexities and distance between release time and delivery time. Job complexity is determined by summing up the release time, job length and delivery time; and comparing the value with the job. If the sum is larger than the job, the problem at hand is said to be complex or hard. The most complex job in the queue will be assigned to a faster resource, and the job with a higher difference (distance) between the delivery time and the release time is processed first. These heuristics are denoted by the following formulae:

$$H(J_i) \quad = \quad (R_i+L_i+D_i) \ > \ J_i \qquad (1)$$

$$G(J_i) \quad = \quad (D_i\text{-}R_i) \qquad (2)$$

Where J_i is the i^{th} job on the grid, R_i is the release time, L_i is the length of the i^{th} job and D_i = delivery time, and $G(J_i)$ = distance value.

VOL. 10, NO. 15, AUGUST 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

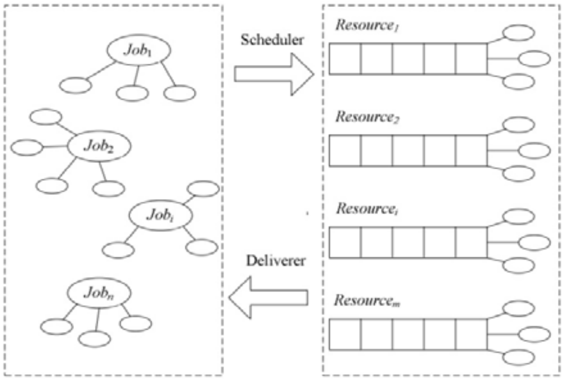

Figure-4. Key entities in Grid Job Scheduling (Wang, *et al*., 2012).

The approach strengths include (i-) ability to schedule grid jobs when the number of jobs is larger than the number of resources or when a single resource is laden with lots of jobs, (ii-) also, when job lengths are similar, it attains better make-span than other classical algorithms such as Min-min, Max-min and Max-min-L. It *weaknesses* include i- failure to address network utilization factor & memory usage; ii- when jobs are dissimilar in lengths, the algorithm may not perform as expected.

Novel job scheduling algorithm for improving data grid's performance (Mansouri, *et al.*, 2011)

The novel Combined Scheduling Strategy (CSS) was proposed to consider number of jobs waiting in queue, the location of required data for the job and the computing capacity of sites. The strategy executes jobs if requested files are available within local storage, otherwise; files transfer and replication is invoked.

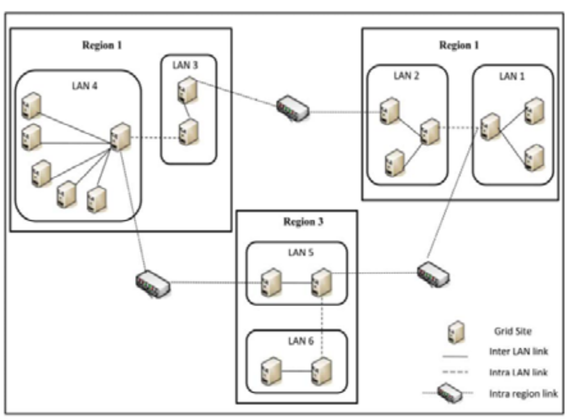

Figure-5. The CSS model (Mansouri, *et al.*, 2011).

CSS considers the number of jobs waiting in the queue, size of requested data files and size of computing elements (CE) of the site. It takes less job execution time than other strategies especially when number of jobs or size of the files or both increases. This model is shown on Figure-5.

The Random scheduler selects a computing node to execute specific job randomly. The Shortest Queue scheduler calculates all of the queue lengths of computing nodes and selects one that has the least number of jobs waiting in the queue. The Access Cost scheduler assigns jobs to CE with lowest file access cost. Strengths include (i-) reducing network and storage utilization, (ii-) mean job execution time and iii- access cost. However, load balancing is not considered by this approach.

Hybrid batch job scheduling algorithm for grid environment

A hybrid batch job scheduling method based on generic algorithm (GA) and discrete particle swarm optimization (PSO) was proposed by (Dehghani, *et al.*, 2014). The technique promised to reduce the makespan and flow time using the algorithm shown on Figure-6.

```
Initialize population with random solution & (min-min) (max-min) heuristic
repeat
{       //execute all the GA operators
        selection; crossover; mutation;
        //execute all the PSO operator offspring
        find pbest; find gbest;
        repeat
        { for each particle
        { evaluate fitness; update pbest, gbest, velocity & position}
        }   until termination = true
}   until termination = true
```

Figure-6. Algorithm for the HGA-PSO approach (Dehghani, *et al.*, 2014)

When compared to min-min, max-min, and discrete PSO the experimental results showed reduction in make span for every 7 out of 12 instances of Braun workload. The technique is limited by Braun benchmark, which has expected time to compute (ETC) matrix for executing 512 jobs on 16 nodes, even though it achieved better make-span than the existing heuristics.

Data replication approach with consistency guarantee for data grid (Abawajy and Deris, 2014)

In this approach, a data duplication problem on grid (DDG) was proposed aimed at minimizing data update cost, providing high availability and data consistency. The strategy firstly identifies the source of the original data to be duplicated and places copies of the data onto subsets of the grid sites with the exception of the source site. The proposed approach was compared with two existing approaches using response time, data consistency, data availability, and communication costs. The results showed that the proposed approach performs significantly better than the traditional approaches. The DDG assumes that all the site subsets have the facility to

ARPN Journal of Engineering and Applied Sciences

hold data copies. In case some sites do not possess SE for holding data, the approach needs to be modified

Hybrid approach for monitoring and scheduling grid jobs (Rajkumar, *et al*., 2014)

This approach was developed in parts including i) Monitoring Infrastructure, which comprises of a sensor that measures the characteristics of a system resource, an actuator and event service, ii-) Task Scheduler which is responsible for locating the appropriate computing nodes to execute particular types of applications. The Parallel Hybrid Particle Swarm Optimization (PH-PSO) algorithm combined the powers of discrete particle swarm optimization (DPSO) and continuous PSO, resulting into a combined optimization prediction model. The PH-PSO model is shown on Figure-7.

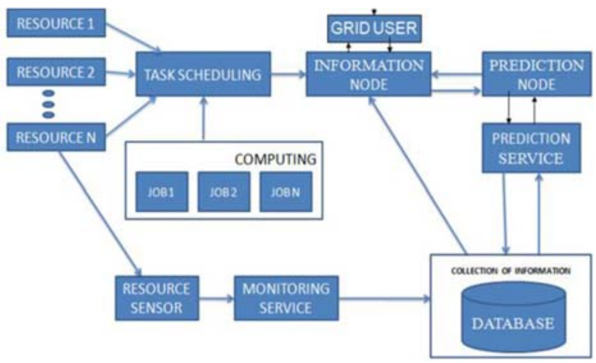

Figure-7. The PH-PSO model (Rajkumar, *et al*., 2014).

Evaluation results indicated that the new model of (PH-PSO) combined with the Nu-Support Vector Regression (NSVR) satisfy optimization requirements of online systems. The approach was compared with Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Simulations indicated that the later (PSO) recorded minimal error and cost-less optimization time, compared with the former (GA). The strengths include combinatorial optimization and resource monitoring sensors. However, the optimization time is proportional to the number of machine sets. This results in higher costs of optimization time as the number of machine set becomes large.

Comparison and analysis

Job Scheduling for Dynamic Data Replication Strategy in Heterogeneous Federation Data Grid Systems has potential when dealing with small number of jobs, but not suitable for large number of jobs. Enhanced, adaptive scoring sob scheduling algorithm has an added advantage of computing job scores and memory requirements prior to making schedules. Dynamic replication strategy plus dataset scheduler is capable of scheduling both data intensive and compute intensive jobs by computing job score on independent jobs. It effectively reduced Job failure rate and make-span time, as compared to other techniques. However it does not consider independent jobs in a workflow environment. The novel job scheduling algorithm is adaptive in that the random scheduler was able to select a computing node to execute specific job randomly; the shortest queue scheduler was able to calculate the queue lengths of computing nodes and selects one that has the least number of jobs waiting; the access cost scheduler was able to assign job to computing element with lowest file access cost. In a way it excels in reducing the network and storage utilization, mean job execution time as well as access costs at the expense of load balancing. The heuristic algorithm schedules jobs when the number of jobs outweighed the number of resources. The algorithm attains better makespan than the classical algorithms such as min-min and max-min. However, it does not consider network utilization and memory usage, and the algorithm may not perform well when jobs are dissimilar in lengths. The parameter-wise analysis of the survey papers is summarized in Table-1.

ARPN Journal of Engineering and Applied Sciences

Table-1. Summary of job scheduling techniques in grid environment.

Approach	Topology	Scheduling/ Replication policy	Performance metrics	Simulator/ Platform	Scheduling	Replication
Novel Job Scheduling for Improving Data Grid's Performance (Mansouri, et al., 2011)]	Hierarchy	Shortest Queue, Access Cost &Random scheduling	Access time, Job throughput, Network utilization, Storage utilization	Optorsim/ Datagrid	✓	**x**
Scheduling, Dynamic Data Replication Strategy (Zarina, et al., 2013)	Federated Data Grid	NCA (Network Core Area)	Access Latency, Network Utilization	Optorsim/ Data Grid	✓	✓
Enhanced, Adaptive Scoring Job Scheduling Algorithm (Aparnaa, et al., 2014)	distributed	Job Score, Memory Requirement, SE Capacity and Processing Power	Storage Utilization, Transmission Power, Processing Power Job completion, Time and Job Failure Rate.	Gridsim/ Compute + Data Grid	✓	x
Dynamic Replication Strategy (Dang, et al., 2007)	Based on EU Data Grid	Property-based organization of data:	Network Utilization, Storage Usage, Job Throughput	Optorsim/ Data grid	x	✓
Scheduling Reliability, Time, Cost and Constraints (Maheshbhai, 2011)	Hierarchy	Distributed/ Workflow environment	Mean Job Time, Total Job Time, Size of SE	OptorSim	✓	x
Integrating Scheduling and Replication with Performance Guarantee (Anikode and Tang, 2011)	Hierarchy	Distributed	Access time	Optorsim/Data Grid	✓	✓
Heuristic algorithm for scheduling (Wang, et al., 2012)	distributed	Complex job first, larger distance first	Mean job time, Resource utilization	GridSim/ Compute Grid	✓	x
Hybrid batch job scheduling (Dehghani, et al., 2014)	distributed	Generic and discrete (PSO)	Mean job time, Resource utilization	Gridsim/ Compute Grid	✓	x
Hybrid Approach for Monitoring and Scheduling (Rajkumar, et al., 2014)	distributed	Discrete and continuous PSO	Mean job time, Resource utilization	Gridsim/ Compute Grid	✓	x

CONCLUSIONS AND RESEARCH OUTLOOK

In this paper, various job scheduling algorithms in Grid Computing have been analyzed. The discussion on the various papers reviewed with respect to parameters such as topology, scheduling policy, simulation environment, and performance was done to get feedback on different types of job scheduling. These facts can be adopted by researchers to develop better and robust scheduling algorithms. It was evident that the gap between job scheduling and data replication is narrowing down as studies revealed some researchers focused their attention to developing combined replication and scheduling algorithms. As the primary aim of data replication is to create load balance amongst the various nodes in a site, it could also mean a positive direction towards developing algorithms that are capable of job scheduling and load balancing simultaneously. Also, the fact that some jobs are typically resource-hungry, computing jobs requirement in terms of memory and storage capacity is vital and will have the tendency of making sure jobs are completed within time and at lesser costs. Thus, the study explored job scheduling techniques in terms of topology, performance and scheduling policies that are used to schedule jobs in grid environments. The main contribution here is that job scheduling combined with data replication promised to improve the performance of grid system in terms of access latency and job throughput, particularly when access to large data items is involved. In future, research on job scheduling can be geared towards load-balanced schedulers that can efficiently and dynamically schedule jobs in a balanced passion, with little or no biasing to a particular grid topology. Also, issues such as achieving load balancing at local site and minimizing complexity of the scheduling algorithms can be considered as a future direction while developing new scheduling algorithms.

ACKNOWLEDGEMENTS

The authors wish to thank the Ministry of Education, Malaysia for funding this study under the Long Term Research Grant Scheme (LRGS/bu/2012/UUM/Teknologi Komunikasi dan Infomasi).

ARPN Journal of Engineering and Applied Sciences

REFERENCES

Abawajy, J. H., and Deris, M. M. 2014. Data Replication Approach with Consistency Guarantee for Data Grid. Computers, IEEE Transactions on. 63(12), 2975-2987.

Abba, H. A., Shah, S. N. M., Zakaria, N. B., and Pal, A. J. 2012. Deadline Based Performance Evaluation of Job Scheduling Algorithms. In Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 Intern. Conf. pp. 106-110.

Abdullah, M., Othman, M., Ibrahim, H., and Subramaniam, S. 2008. A new load balancing scheduling model in data grid application. 2008 Intern. Sympoium on Information Technology

Abdurrab, A. R., and Xie, T. 2010. FIRE: A File Reunion Based Data Replication Strategy for Data Grids. In Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on. pp. 215-223.

Anikode, L. R., and Tang, B. 2011. Integrating scheduling and replication in data grids with performance guarantee. GLOBECOM - IEEE Global Telecomms Conference.

Aparnaa, S. K., and Kousalya, K. 2014. An Enhanced Adaptive Scoring Job Scheduling algorithm for minimizing job failure in heterogeneous grid network. In Recent Trends in Information Technology (ICRTIT), 2014 International Conference on. pp. 1-6.

Ben Charrada, F., Ounelli, H., and Chettaoui, H. 2010. An Efficient Replication Strategy for Dynamic Data Grids. In P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010 International Conference on. pp. 50–54.

Cameron, D. G., Carvajal-Schiaffino, R., Millar, a. P., Nicholson, C., Stockinger, K., and Zini, F. 2003. Evaluating scheduling and replica optimisation strategies in OptorSim. Proceedings. First Latin American Web Congress.

Carman, M., Zini, F., Serafini, L., and Stockinger K. 2002. Towards an Economy-Based Optimization of File Access and Replication on a Data Grid.

Chang, H.-C., Li, K.-C., Lin, Y.-L., Yang, C.-T., Wang, H.-H., and Lee, L.-T. 2005. Performance issues of grid computing based on different architecture cluster computing platforms. In Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on. Vol. 2, pp. 321-324.

Chang, R.-S., Chang, H.-P., and Wang, Y.-T. 2008. A dynamic weighted data replication strategy in data grids. In Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on. pp. 414–421.

Chen, D. Y., Wu, J. J., and Liu, P. 2009. Data-bandwidth-aware job scheduling in grid and cluster environments. Proceedings of the International Conference on Parallel and Distributed Systems – ICPADS. pp. 414-421.

Chen, Y., Guan, Z., and Shao, X. 2011. A comparative analysis of job scheduling algorithm. In Management Science and Industrial Engineering (MSIE), 2011 International Conference on. pp. 1091–1095.

Chettaoui, H., and Charrada, F. B. 2012. A Decentralized Periodic Replication Strategy Based on Knapsack Problem. In Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on. pp. 3-11.

Coutinho, F., Verdino, E., Ossian, J., & Santana, R. (2014). GGreen: A Greedy Energy-Aware Scheduling Algorithm on Grid Systems. 2014 IEEE 17th International Conference on Computational Science and Engineering. pp. 285–290.

Dang, N. N., Hwang, S., and Lim, S. B. 2007. Improvement of data grid's performance by combining job scheduling with dynamic replication strategy. Proceedings of the 6th International Conf. on Grid and Cooperative Computing, GCC 2007, (Gcc), pp. 513-520.

Dehghani Zahedani, S., and Dastghaibyfard, G. 2014. A hybrid batch job scheduling algorithm for grid environment In Computer and Knowledge Engineering (ICCKE), 2014 4th International conference on. pp. 763-768.

Djebbara, M. R., and Belbachir, H. 2010. Dynamic threshold for replicas placement strategy In Machine and Web Intelligence (ICMWI), 2010 Intern. Conf. pp. 394–401.

Elghirani, A., Subrata, R., and Zomaya, A. Y. 2007. Intelligent Scheduling and Replication in Datagrids: a Synergistic Approach. In Cluster Computing and the Grid, 2007. CCGRID 2007. Seventh IEEE International Symposium on. pp. 179-182.

Elghirani, A., Subrata, R., and Zomaya, A. Y. 2009. Intelligent scheduling and replication: A synergistic approach. Concurrency Computation Practice and Experience. 21(3), 357-376.

Gai-Mei, G., and Shang-Wang, B. 2012. Design and Simulation of Dynamic Replication Strategy for the Data

Grid. In Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on. pp. 901-903.

Haghighat, A. T. (n.d.). The Impact of Bandwidth and Storage Space on Job Scheduling and data replication strategies in Data Grids. pp. 283-288.

Hao, Y., and Liu, G. 2015. Evaluation of nine heuristic algorithms with data-intensive jobs and computing-intensive jobs in a dynamic environment. IET Software. 9(1), 7-16.

Jacob B. Brown, M. K. Fukui, N. Trivedi. 2005. Introduction to Grid Computing. USA: IBM Red Books International Technical Support Organization. pp. 6-42

Liu, W. L. W., and Du, W. D. W. 2009. Reputation-Aware Scheduling for Storage Systems in Data Grids. 2009 Fourth International Conference on Frontier of Computer Science and Technology.

Magoules, F. *et al*. Introduction to Grid Computing. London UK CRC Press, Taylor and Francis Group, 2010, pp.10-14.

Paul, D., and Aggarwal, S. K. 2014. Multi-objective Evolution Based Dynamic Job Scheduler in Grid. 2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems. pp. 359-366.

Qureshi, M. B. *et al*. 2004. Survey on Grid Resource Allocation Mechanisms,'' Springer Science Business+Media Dordrecht.

Ranganathan, K., and Foster, I. 2002. Decoupling computation and data scheduling in distributed data-intensive applications. Proceedings 11th IEEE International Symposium on High Performance Distributed Computing. 2002.

Scholar, P. G. 2014. An Enhanced Adaptive Scoring Job Scheduling Algorithm for Minimizing Job Failure in Heterogeneous Grid Network.

Soni, V. K., Sharma, R., and Mishra, M. K. 2010. An analysis of various job scheduling strategies in grid computing. Signal Processing Systems (ICSPS), 2010 2nd International Conference on. 2, 162-166.

Suri, P. K., and Singh, M. 2009. JS2DR2: An effective two-level job scheduling algorithm and two-phase dynamic replication strategy for data grid. ACT 2009 - Intern. Conf. on Advances in Computing, Control and Telecomms Technologies. 232-236.

Zarina, M., Ahmad, F., bin Mohd Rose, A. N., Nordin, M., and Deris, M. M. (2013). Job scheduling for dynamic data replication strategy in heterogeneous federation data grid systems. In Informatics and Applications (ICIA), 2013 2nd Intern Conf. pp. 203–206.

Maheshbhai, L.A., "Job scheduling based on reliability, time and cost constraints under Grid environment," Engineering (NUiCONE), 2011 Nirma University International Conference on. vol., no., pp.1, 5, 8-10 December.

Quan Liu; Yeqing Liao. 2009. Grouping-Based Fine-Grained Job Scheduling in Grid Computing. Education Technology and Computer Science, 2009. ETCS '09. First International Workshop on. vol.1, no., pp. 556,559, 7-8 March 2009.

Wuu-Yee Chen; Tsang-Yean Lee; Nai-Wen Kuo; Yuan-Hwa Mei; Shan-Kai Chen; Yu-Chen Yan. 2009. Job Schedule Model Based on Grid Environment," INC, IMS and IDC, 2009. NCM '09. Fifth Intern. Joint Conf. vol., no., pp.538, 543, 25-27 August.

Wang, J., Wu, G., Zhang, B., and Hu, X. 2012. A Heuristic Algorithm for Scheduling on Grid Computing Environment. 2012 Seventh China Grid Annual Conference. 36-42.

Mansouri, N., Dastghaibyfard, G., and Horri, a. 2011. A Novel Job Scheduling Algorithm for Improving Data Grid's Performance. 2011 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. 142-147.

Rajkumar, S.; Rajkumar, N.; Suresh, V.G., "Hybrid approach for monitoring and scheduling the job in heterogeneous system," Information Communication and Embedded Systems (ICICES), 2014 International Conference on. vol., no., pp. 1, 5, 27-28 February. 2014.