Grid Federation: Number of Jobs and File Size Effects on Jobs Time

Musa Sule Argungu[1,2], Suki Arif[1] and Mohd. Hasbullah Omar[1]
[1]*InterNetWorks Research Laboratory, School of Computing, Universiti Utara Malaysia. 06010 Sintok, Kedah, Malaysia.*
[2]*Department of Computer Science, University of Science and Technology, Aliero. Kebbi State Nigeria.*
argungu@internetworks.my

***Abstract*—Grid federation is fast emerging as an alternative solution to the problems posed by the large data handling and computational needs of the existing numerous worldwide scientific projects. Efficient access to such extensively distributed data sets has become a fundamental challenge in grid computing. Creating and placing replicas to suitable sites, using data replication mechanisms can increase the system's performance. Data Replication reduces data access time, ensures load balancing as well as narrows bandwidth consumption. In this paper, an enhanced data replication mechanism called EDR is proposed. EDR applies the principle of exponential growth/decay to both file size and file access history, based on the Latest Access Largest Weight (LALW) mechanism. The mechanism selects a popular file and determines an appropriate number of replicas as well as suitable grid sites for replication. It establishes the popularity of each file by associating a different weight to each historical data access record. Typically, recent data access record has a larger weight, which signifies that the record is more relevant to the current situation of data access. By varying the number of jobs as well as file sizes, the proposed EDR mechanism was simulated using file size and job completion time as the variable metrics. Optorsim simulator was used to evaluate the proposed mechanism alongside the existing Least Recently Used (LRU), and Least Frequently Used (LFU) Mechanisms. The simulation results showed that job completion time increases by the growth in both file size and number of jobs. EDR shows improved performance on the mean job completion time, compared to LRU and LFU mechanisms.**

***Index Terms*—Data Grids; Data Grid Federation; Data Replication; EDR Optimizer**

I. INTRODUCTION

Data Grids (DG) are complex, heterogeneous, diverse systems that span several sites and organizations [1-3]. A Data Grid Federation (DGF) [4] could be formed by bringing together these individual complex and heterogeneous DG systems via a Peer-to-Peer Wide Area Network [5-7]. A DG system and its federation exhibit similar characteristics, except for the underlying topology, which is defined by the bandwidth connectivity amongst the various sites and cross registrations of numerous users connected to the DGF system [5].

A federation of multiple DG systems revolves around factors such as the cross-registration of multiple users, heterogeneous resources, enormous data files, and different context. Depending on the federation settings, the factors may be set to either no cross-registration, partial cross-registration, or complete cross-registration. The cross-registration of users automatically defines the type of connectivity between the sites, and hence forms the basis for the federation architecture [5]. By varying the type of cross-registration across sites, up to 1500 DGF approaches could be realized [5]. The researchers in [5] further outlined ten of the DGF approaches that are either actively put to use, or were proposed to be used within scientific projects. In a DGF setting, each Data Grid is referred to as a zone, hosting its metadata catalog for managing the logical namespaces of its users, resources, files, and context [5].

The DG project [8] by European Union (EU) is one of the building blocks for deploying robust computing infrastructure and middleware services for the management of large-scale data across widely distributed scientific communities. Within the DG project, a Replica Optimization Service (ROS) [9] is deployed to address the issues related to replica optimization. Before deploying the ROS, it is important to select one or more optimization mechanisms it will use. These mechanisms also work effectively in a DGF environment under a wide range of conditions [5], so they should be tested thoroughly before deployment [10]. One way to achieve a realistic evaluation of various strategies is to define a grid simulation environment that closely mimics a real Data Grid [11]. The simulation environment should be capable of simulating some grid jobs using a suitable optimization strategy, and to collect measurements based on which the strategy is evaluated [10-14].

In this paper, an enhanced data replication mechanism for improving the performance of DGF environment is proposed. The write-up starts by highlighting the key elements of a realistic grid infrastructure, which forms the basis of the simulation environment used in this paper. The effect of file size and some jobs on job completion time for all jobs running within the DGF environment is evaluated. This write-up analyzes some important metrics used for evaluating DG systems. Jobs completion time is the metric used, with varying file sizes and a different number of jobs per unit time. The performance of the proposed replication mechanism, which uses the concept of file exponential growth/decay policy [3], based on the LALW mechanism [2] was evaluated in OptorSim DG Simulator [1-2]. The proposed EDR uses the existing file access patterns that came with OptorSim simulator. Henceforth in this write-up, the terms DG and DGF are used for Data Grid and Data Grid Federation, respectively unless otherwise stated.

The rest of this paper is organized as follows: Section II examines related literature. The features of a realistic simulation environment are discussed in Section III. Section IV describes the proposed EDR mechanism in detail, along with a set of performance measurements to evaluate the mechanism. The specific setup used for the simulation and

the results are presented in Section V. Conclusion is given in Section IV together with suggestions for further research.

II. Related Literature

In [6-7], the authors proposed and implemented a mechanism for job scheduling and dynamic data replication in DGF system, with the aim of improving data access latency. The mechanism, otherwise known as SGFRS, accesses data from an area identified as Network Core Area (NCA). The mechanism, which was evaluated in OptorSim simulator, successfully achieved its aim of obtaining data from the nearest node by allocating areas for the searched zone using the concept of NCA. By relegating the search to the nearest node, the need for greater bandwidth has been reduced by the SGFRS. It has also managed to minimize expected data access latency of file.

The authors in [15] proposed a scheduling and replication strategy aimed to study the issues associated with integrating job scheduling and data replication in data-intensive scientific applications. The mechanism seeks to minimize overall job execution time by placing data replicas and jobs onto appropriate sites. The approach was designed to guarantee overall grid performance by developing a set of efficient heuristics, and the results were validated using OptorSim simulator. Similarly, A dynamic replication strategy in a hierarchical structure that selects best file replica was proposed in [16] based on the EU Data Grid CMS testbed topology. The model was evaluated using access time, network usage and storage usage performance metrics.

In [2], the researchers proposed a dynamic data replication mechanism called LALW. The mechanism identifies a popular file for replication, then computes an appropriate number of copies and locations for replicating such files. Each historical data access record is associated with a different weight. The weights help identify the relevance of each file record to the users as well as the grid federation system. A larger weight indicates more recent data access, which also means that the record is more pertinent regarding data access within the federation system. The Grid simulator OptorSim is used to evaluate the performance of the proposed replication mechanism. The simulation results show that LALW mechanism successfully optimizes the effective network usage; indicating that the LALW strategy can find out a popular file and replicates it to a suitable site without putting constraints on the bandwidth. An enhanced version of LALW was proposed by same researchers and named it enhanced-LALW, known as ELALW mechanism [17]. In addition to determining file weight, the ELALW also considers network distance between nodes for replica placement.

In this write-up, the proposed EDR mechanism considers storage capacity in addition to the distance between nodes for effective replica placement. Considerations to storage capacity come in three folds, that is, *highly loaded, lightly loaded* and *moderately loaded sites*. The aim is to improve on job completion time by placing data replica closer to the clients without compromising storage usage. Thus, placing data replica on the lightly loaded and moderately loaded nodes eliminates bottlenecks, thereby improving job completion time for all jobs. The proposed mechanism finds the least value file, in addition to finding out the largest weight file, using the concept of file growth /decay policy [3]. Computation of network distance between nodes proceeds based on the work of researchers in [17], while file weight and value are computed based on the work of researchers in [2]. Thus, the strategy not only decides which file to be replicated, but it also determines which files need to be deleted to accommodate incoming file replicas, based on their access weights [2] and lower value [3], respectively.

III. Simulation Environment

The OptorSim is designed to test various replication optimization strategies in a simulated grid environment before they are deployed in the real grid platform, which has been used by many researchers [1-3, 5-7, 12-15]. It emphasizes more on simulating data access optimization algorithms. The simulator uses discrete event simulation and is managed under the open source license. It was developed in the framework of the European Data Grid (EDG) project [8] as a joint effort of University of Glasgow and CERN [18], amongst others. The simulation environment is based on the EDG) project [8]. Other simulators exist for DG systems [19-24]. However, this write-up used OptorSim based on its emphasis on simulating data access optimizations algorithms and frequent uses in the related research domain.

In this setup, a data file is characterized by its name, size, and an index number, which represents its similarity to other files. The set of data files specifies a job it needs to analyze. As we are studying the performance of different optimization strategies, files are considered homogeneous. Thus only the movement of files records caused by replication is simulated in this study.

IV. Enhanced Data Replication Mechanism

The proposed EDR mechanism helps to improve data locality by increasing number of replicas within the federation system. It accesses the Current Disk Space (CDS), to ascertain the available storage capacity of a node. CDS is calculated as follows [25]:

$$CDS = S_{\text{reg}} - S_{\text{usage}} \quad (1)$$

where S_{reg} is maximum storage capacity of a node and S_{usage} is storage space occupied by resources on the node. If the CDS is higher than the size of the file to be replicated, the replica is placed on the node, and the job is scheduled on that node. If, however , the node has insufficient space, other nodes are contacted within the cluster. If all the nodes within the cluster returned a CDS less than the file size, then the mechanism will delete old files or replicas from the node. For removing a file from the node, the access frequencies of all the files stored on the node are compared. The file with the least access frequency are deleted and replaced with the incoming replica. The algorithm for implementing the Enhanced Data Replication Mechanism is shown in Figure 1.

This study uses access history for stored files to find the least value file [2]. The header nodes manage file information in each of the clusters of the DG federation. A given file record in a cluster header is stored according to the following format<File_ID, Cluster_ID, Number> [2-3]. The "Number" indicates how many times a given file (File_ID) was accessed by the cluster (Cluster_ID). The concept of file weight and file value has a direct bearing on both file size and file access history based on LALW [2], buttressed in [3] and [17]. In this write-up, the authors are

interested in creating space for an incoming file replica. Each file weight increases by the increase in access rate, and decreases by the decline in access rate. Similarly, file value changes with a corresponding increase or decrease in access rate. Thus, the principle was applied to file access history to help determine the files with less popularity.

//Procedure for replica placement

i- Access source /remote data & compute storage requirement (SR) for replication

ii- Input list of Storage Elements (SEs) within the current cluster

iii- Compute Current Disk Space (CDS) for all the SEs within the cluster

iv- Locate lightly loaded SEs for replication

IF NONE

v- Locate moderately loaded nodes

vi- Compare storage requirement (SR) with Current Disk Space for the SEs

Is SR >= CDS?

IF Yes

vii- While network distance <= average network distance for all sites: Locate and delete least value file(s) from the appropriate moderately /highly loaded nodes to make them lightly loaded,

viii- Place file replica for the popular file within the lightly loaded SE or moderately loaded SEs

END

Figure 1: Enhanced Data Replication Mechanism

A. Evaluation Metrics for Grid Optimization Strategies

There are several measures [26], which can be considered in the evaluation of Grid optimization strategies. Since the effectiveness of a strategy could depend on the adopted measure, it is typical to select a suitable metric for evaluation. In this write-up, mean job execution time (MJET) is used to evaluate the performance of the proposed mechanism. MJET is defined as the total time to execute all the jobs, divided by the number of jobs completed [2]. The average job execution time is calculated as time to execute a file (T_i), plus the time consumed by a job in waiting queue (W_i), divided by the number (n) of jobs completed. It is expressed mathematically as [2]:

$$MJET = \sum_{i=1}^{n} \frac{(T_i + W_i)}{n} \quad (2)$$

where n = number of jobs processed by the system, T_i = time to execute the ith job, and W_i = waiting time of ith job that has been spent in the queue.

B. Simulation Setup

In this section, the write-up describes the simulation setup, to depict a realistic grid environment, starting with the configuration parameters. Table 1 contains some of the important parameters used for the simulation.

Table 1
Configuration parameters used for the simulation

Parameter	Value
Number of jobs	50 to 1000
Single file size (GB)	2.5 to 10
Job Scheduler	Access Cost for current jobs + all queued jobs
Optimization Mechanism	EDR-Occasionally Replicate, Delete least access lowest weight file
Access Pattern	Zipf Distribution

The simulation was run using jobs numbers ranging from 50 to 1000 on the increment of 100, 200, 400, 600, 800 and 1000. Files sizes were varied from 2.5GB, 5Gb, and 10GB.

C. Grid Topology Simulated.

The OptorSim Simulator could simulate any Grid topology [12]. The simulator accepts a Grid topology in the form of *M x N* square matrix of bandwidth configuration file. In this paper, the bandwidth connectivity is configured to match the DGF setup in Figure 2. Every site in this setup has a computing element (CE) and initial unfilled storage of 80 GB capacity. The site containing original master files has a capacity of 100 GB. The storage capacity values used are arbitrary scaled down representatives from the actual resources at the CMS sites [1]. The topology is shown in Figure 2.

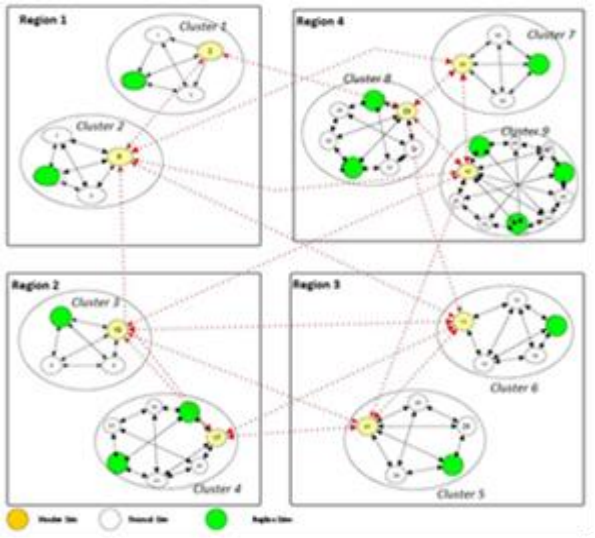

Figure 2: Data Grid Federation Architecture

The simulated workloads for the grid federation model were based on a sample data extract from the high-energy physics experiment (HEP) [27] test-bed data. There are six (6) types of jobs, and 1,000 jobs were processed in this study. Each file has a size of 10 GB, totaling to 10,000 GB for processing 1000 jobs. Each CE takes approximately one second to process each file. Similarly, a total of 5000GB and 2500GB for file sizes 5GB and 2.5GB, respectively for processing 1000 jobs.

V. RESULTS AND DISCUSSION

In this section, the authours present simulation results, using OptorSim to evaluate storage usage by varying number of jobs and size of the files used for the simulation. The simulation was run several times, by controlling the simulation environment. The simulation was run by keeping one parameter constant as well as re-starting the simulator after each run. After simulating for the 20th time, the root-mean-square value of the total jobs times was compared against the absolute values, resulting in a negligible variation of about 0.5%. This is to be expected because, in certain situations, the mechanism has to select between two equally valid options randomly. For instance, if the mechanism seeks to get a certain file in the shortest time possible, and there are two sites, which could deliver the file in the same time, one of the sites is chosen randomly. This is also due to the way

Java implements certain objects, such as the hash tables used for holding data items. All the mechanisms were tested along with the existing heuristic (Queued Access Cost + All Queued Jobs) scheduling mechanism. The results from the simulation for varying number of submitted jobs and file sizes ranging from 2.5GB to 10GB are shown in Figure 3(a)-(c).

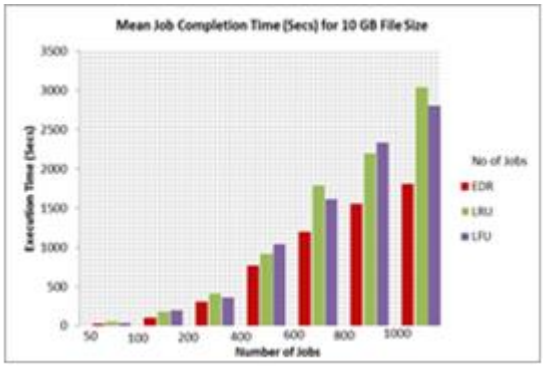

Figure 3(a): Comparing EDR, LRU, and LFU for 10 GB File Size

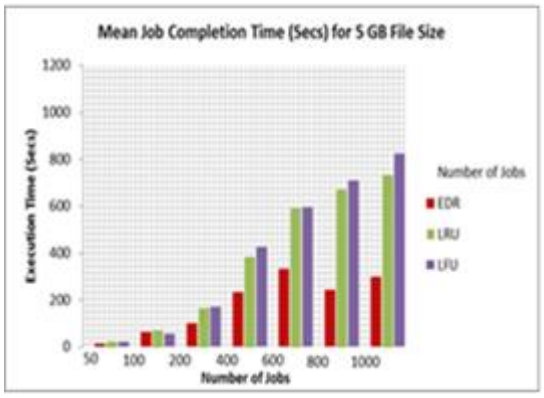

Figure 3(b): Comparing EDR, LRU, and LFU for 5 GB File Size

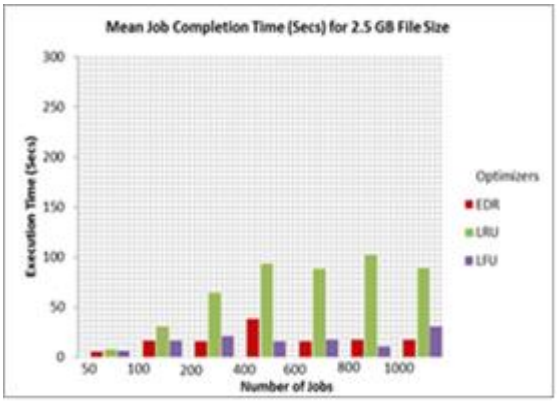

Figure 3(c): Comparing EDR, LRU, and LFU for 2.5 GB File Size

The impact of number of jobs and file size for EDR mechanism was evaluated alongside LRU and LFU mechanism for some jobs ranging from 50 to 1000, and file. The EDR, LRU and LFU optimizers showed a similar pattern of poison distribution along the number of jobs and job completion time. In this paper, mean job time, no of jobs, and file sizes are of concern. It was observed that job completion time decreases with the corresponding decrease in file size, for all the mechanisms under review. Also, jobs time increases steadily as number of jobs rises from 50 to 1000, with irregular pattern between 400 to 1000 for 2.5 GB file size. This shows that a large file could run faster and complete in a shorter time when split into smaller chunks and made to run on different machines. This is one of the advantages of DG federation, whereby idle machines could be used for concurrent job executions to the benefits of the users.

Figure 3(a) compares Mean Job Time for EDR, LRU and LFU for 1000 Jobs, and 10 GB File Size. Figure 3(b) compares Mean Job Time for the three mechanisms, using 5 GB file size for jobs run of 1000. Lastly, Figure 3(c) plot for 2.5 GB file size, 1000 jobs for the three mechanisms under review. It is evident from the results that file size as well number of jobs both have an incremental effect on the total completion time for grid jobs.

VI. Conclusions and Future Works

The graphs of Mean Job Time Vs. Number of Jobs for 1000 Jobs and File Sizes of 10 GB, 5 GB and 2.5 GB as shown in Figures 3.1(a-c) exhibit poison distribution patterns over different time's intervals for both EDR, LRU and LFU mechanisms. The jobs were run at various time intervals, starting with 50, 100 jobs, 200 jobs, 400 jobs, 600 jobs, 800, and 1000 jobs in that order. By comparison, EDR outperforms the LRU and LFU mechanism regarding jobs completion time for 1000 jobs.

The advantage EDR has over LRU and LFU is that the former performs advanced replication of files that may be required by next job based on their access weights, which reduces job completion time subsequently. In contrast, LRU and LFU always replicate, which may affect the expected completion time. In the future, we intend to add other performance indicators that are significant such as different types of end users and effect of replications on SE and CE Usage. Also, the study will be extended to simulate 1,000,000 jobs. New job scheduler will also be introduced, along with our EDR optimizer, for further enhancement of the DGF performance. The new mechanisms will be tested against the effect of data replication on SE usage in the near future. It is expected that the mechanisms can optimise SE usage and access time for a high number of jobs. Also, the mechanism will be compared with ELALW mechanism against job completion time.

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).

References

[1] D. G. Cameron, R. Carvajal-Schiaffino, A. P. Millar, C. Nicholson, K. Stockinger, and F. Zini, "Evaluating scheduling and replica optimisation strategies in OptorSim", *Journal of Grid Computing*, pp. 57-69, March 2004.

[2] R. S. Chang and H. P. Chang, "A Dynamic Data Replication Strategy using Access-Weights in Data Grids*", Future Generation Computer System*, 22, pp. 254-268, 2008.

[3] M. K. Madi and S. Hassan, "Dynamic replication algorithm in Data Grid: a survey", In *International conference on network applications, protocols, and services*, November 2008.

[4] M. A. Salehi, B. Javadi, and R. Buyya, "Preemption-aware admission control in a virtualized grid federation," in *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, 2012, pp. 854–861.

[5] A. Jagatheesan and R. W. Moore, "Data grid and grid-flow management systems", in *Proceedings of IEEE International*

Conference on Web Services, 2004, pp. xxix-xxix.

[6] Z. Mohamad, F. Ahmad, A. N. M. Rose, F. S. Mohamad and M. M. Deris, "Job scheduling for dynamic data replication strategy in heterogeneous federation data grid systems", In *2013 2nd* IEEE *International Conference on Informatics and Applications (ICIA)*, September 2013, pp. 203-206.

[7] Z. Mohamad, F. Ahmad, A. N. M. Rose, F. S. Mohamad and M. M. Deris, "Implementation of Sub-Grid-Federation Model for Performance Improvement in Federated Data Grid", *Malaysian Journal of Applied Sciences*, vol. 1, no. 1, pp. 55-67, 2016.

[8] P. Kunszt, "European DataGrid project: Status and plans", *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 502, no. 2, pp. 376-381, 2003.

[9] A. Sulistio, C. S. Yeo and R. Buyya, "A taxonomy of computer-based simulations and its mapping to parallel and distributed systems simulation tools", *Software-Practice and Experience*, vol. *34 no.* 7, pp. 653-674, 2004.

[10] K. Jain, A. V. Vidhate, V. Wangikar, and S. Shah, "Design of file size and type of access based replication algorithm for data grid", *in Proc. ACM International Conference & Workshop on Emerging Trends in Technology* (ICWET '11), New York, NY, USA, 2011, pp. 315-319.

[11] W. Zhao, X. Xu, N. Xiong, and Z. Wang, "Dynamic replica replacement strategy in data grid", in *Proc. 8th IEEE International Conference on Computing Technology and Information Management (ICCM)*, 2012, vol. 2, pp. 578-584.

[12] H. Stockinger, F. Donno, E. Laure, S. Muzaffar, P. Kunszt, G. Andronico and P. Millar, "Grid Data Management in action: Experience in running and supporting data management services in the EU Datagrid Project", *arXiv preprint cs/0306011*, June 2003.

[13] B. H. William, D. G. Cameron, L. Capozza, A. P. Millar, K. Stockinger, and F. Zini, "Evaluation of an economy-based file replication strategy for a data grid", in *Proceedings of 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid), 2003*, pp. 661-668.

[14] L. Guy, P. Kunszt, E. Laure, H. Stockinger and K. Stockinger "Replica management in data grids", in *Global Grid Forum*, July 2002, vol. 5, pp. 278-280.

[15] R. L. Anikode and B. Tang, "Integrating scheduling and replication in data grids with performance guarantee", in *conf. 2011 IEEE Global Telecommunications Conference (GLOBECOM 2011)*, pp. 1-6.

[16] P. Vashisht, R. Kumar, and A. Sharma, "Efficient dynamic replication algorithm using agent for data grid", *The Scientific World Journal*, pp. 767016-767016, 2014.

[17] N. Mansouri and A. Asad, "Weighted data replication strategy for data grid considering economic approach", *Int. J. Comput. Elect. Auto. Control Inf. Eng*, vol. 8, pp. 1336-1345, July 2014.

[18] B. H. William, D. G. Cameron, L. Capozza, A. P. Millar, K. Stockinger and F. Zini, "Simulation of Dynamic Grid Replication Strategies in OptorSim", in *International Workshop on Grid Computing*, Springer, Berlin, Heidelberg, 2002, pp. 46-57.

[19] M. R. K. Grace, S. S. Priya and S. Surya, "A survey on grid simulators", *Int. J. Comput. Sci. Inf. Technol. Secur*, *2*(6), 1224-1230, 2012.

[20] S. A. Monsalve, F. G. Carballeira and A. C. Mateos, "Analyzing the performance of volunteer computing for data-intensive applications", *2016 International Conference on High-Performance Computing & Simulation (HPCS)*, Innsbruck, 2016.

[21] H. Casanova, "Simgrid: A toolkit for the simulation of application scheduling", in *Proc. First IEEE/ACM international symposium on Cluster computing and the grid, 200*, pp. 430-437

[22] C. L. Dumitrescu and I. Foster, "GangSim: a simulator for grid scheduling studies", In *Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05), Volume 02* (CCGRID '05), IEEE Computer Society, Washington, DC, USA, 1151-1158, 2005.

[23] H. J. Song, X. Liu, D. Jakobsen, R. Bhagwan, X. Zhang, K.Taura, and A. Chien, "The microgrid: a scientific tool for modeling computational grids", in *Proceedings of the 2000 ACM/IEEE Conference on Supercomputing (SC '00)*, 2000, pp. 53-53.

[24] R. Buyya and M. Murshed, "Gridsim: A toolkit for the modeling and simulation of distributed resource mgt and scheduling for grid computing", *Concurrency and computation: practice and experience*, vol. 14, no. 13-15 pp. 1175-1220, 2002.

[25] M. Lei, and S. Vrbsky, "A Data Replication Strategy to Increase Data Availability in Data Grids", in *Proc. 2006 International Conference on Grid Computing and Applications*, Las Vegas, NV, June 2006, pp. 221-227.

[26] R. Jain, "Art of Computer Systems Performance Analysis: Techniques for Experimental Design Measurements Simulation and Modeling", *John Wiley & Sons, Inc*., 1991.

[27] F. Jolfaei and A. T. Haghighat, "The impact of bandwidth and storage space on job scheduling and data replication strategies in data grids", in *Proc. 8th IEEE International Conference on Computing Technology and Information Management (ICCM)*, 2012, vol. 1, pp. 283-288.