A New Data-Intensive Task Scheduling in OptorSim, an Open Source Grid Simulator

Mahshid Helali Moghadam
Department of Computer Engineering
University of Kashan
Kashan, Iran
mhelali@grad.kashanu.ac.ir

Seyyed Morteza Babamir
Department of Computer Engineering
University of Kashan
Kashan, Iran
babamir@kashanu.ac.ir

***Abstract*— Scheduling is one of the most important issues in executing tasks in grid systems. A data grid mainly deals with sharing and managing large amounts of distributed data in executing data-intensive applications. It is primarily a solution to satisfy the requirements of data-intensive tasks processing. OptorSim is a useful open source simulation tool for data grids. In this paper a new two-step data-intensive task scheduling called DSAS was proposed, implemented and, incorporated as a new scheduling package into OptorSim. As stated by the simulation results, the proposed scheduling strategy which was added to OptorSim, improves the mean response time and the mean waiting time of tasks compared to other common tasks scheduling strategies. The modified OptorSim benefits from a new scheduling package which improves the performance of task scheduling.**

***Index Terms*— OptorSim, data grid, task scheduling, data-intensive tasks.**

I. INTRODUCTION

Grid computing refers to a virtualized distributed computing system which provides the integration of distributed heterogeneous resources owned by different organizations [1, 2]. It mainly deals with large amounts of distributed data or large scale computation, and it enables a secure sharing of resources across the network [1]. Data grid is a grid system which provides managing and sharing of distributed data to execute data-intensive applications. Many scientific and engineering applications require access to large amounts of distributed data [1, 3].

OptorSim is an appropriate open source simulation tool to simulate the different characteristics and aspects of the data grid. It is a java-based data grid simulator which was developed under the Work Package 2 in the European Data Grid Project. It provides a suitable environment to study and simulate the different aspects of data grids including task scheduling and data replication [4].

Scheduling is an important issue in the grids. Scheduling defines how to appropriately assign resources to execute the tasks. Improving response time and increasing the throughput are the main objectives of scheduling algorithms in grid systems. In order to improve performance, a scheduling decision should take into account the characteristics of the grid, resources, and requirements of the tasks.

In this paper a two-step data- and size- aware data-intensive scheduling called DSAS was proposed, implemented and, incorporated as a new scheduling feature into OptorSim. The new implemented scheduling strategy considers both data related parameters, such as data access cost, and the task size parameter in a two-step decision mechanism. Reducing mean response time of the tasks and the mean waiting time are the main achievements of the proposed scheduling strategy.

Data access cost is one of the main factors for reducing the completion time of the tasks in data grids. Completion time of the tasks depends on the sites that tasks are assigned to and the retrieval of the required data [5]. In data grids, accessing distributed data depends on available bandwidth in different parts of the network. Waiting time the tasks spend in queues before being processed and processing time are other important factors of completion time of the tasks. One of the primary issues in reducing waiting time is to prevent the small tasks from having to wait too much for the completion of large tasks. The proposed scheduling strategy takes into account the task size distribution along with minimizing data access cost. It showed better performance than other common scheduling strategies in OptorSim.

The proposed added scheduling feature is suitable for application environments where a variety of tasks with different sizes submitted to the grid.

The rest of this paper is organized as follows. Section II presents a brief review of open source simulation tools for grids and also describes the related works proposed by different authors using OptorSim. Section III describes the proposed scheduling strategy implemented as a new feature in OptorSim. Section IV, describes the open source simulation environment, experiment parameters, implementation details, simulation results and performance evaluation of the proposed strategy in comparison to other common task scheduling strategies in OptorSim. Finally, Section V concludes the paper and presents the future works.

II. RELATED WORK

Simulation tools are widely used in many scientific areas of computer science including grid computing. They facilitate the study of behavior and performance of systems without need to have the actual one. Several open source simulation tools have

978-1-5090-4580-8/16/$31.00

been built for simulating grid systems. Some of the available open source grid simulators are as follows:

GridSim: is a simulator for modeling and simulating heterogeneous resources in grid systems. It provides various features for modeling computational grids [6].

SimGrid: is a C language based simulator for simulation of applications in heterogeneous distributed environments. It is an event-based toolkit which provides many functions and features to build different application environments [7, 8].

MicroGrid: is a simulation tool which provides a virtualized grid environment using virtual clustered resources. It supports the simulation of various grid topologies and performance evaluation of distributed applications [9].

Another grid simulator which may not be available as the aforementioned tools is as follows:

GangSim: is a simulation tool for scheduling algorithms in large scale environments consisting of many processing and storage nodes. It is useful for simulating different types of workloads [10].

In this study, an open source data grid simulator called OptorSim was used. OptorSim is a simulation tool which was developed under the Data Grid Project of the Europe. It supports the various immediate independent task scheduling and data replication mechanisms.

With the growing application of data grids in scientific communities to support the needs of accessing and managing large amounts of data, a large number of research studies have been conducted on different aspects of data grids including task scheduling and replication mechanisms. In this context, several studies have been done using OptorSim. Some of them are as follows:

In [11] a task scheduling strategy called HCS based on data transfer time was proposed. It was simulated using OptorSim. The authors of [11] also proposed a hierarchical dynamic data replication strategy called HRS based on the proposed concept of [12] to improve data access in data grids. In HCS all tasks access a fixed number of different files with the same size.

In [12] a hierarchical dynamic replication strategy called BHR based on reginal locality was proposed.

In [13] an efficient algorithm to generate the network topology in OptorSim and a replica placement strategy called RC were proposed and implemented in OptorSim. In [14] a consistency management service was proposed and incorporated into OptorSim. In [15] a data replication strategy was proposed which used data properties to make some semantic data categories and replicate files based on these categories.

III. The Proposed Scheduling Strategy

In the proposed strategy, the submitted task was assigned to the proper node as soon as it arrived at the system. For selecting the best node among the available nodes, the first step was to select the best cluster with the lowest data access cost. In other words, the cluster of the nodes with the lowest data access cost is selected. Data access cost is an essential constituent of the completion time of the tasks. Thus, reducing the time to access required data causes a reduction in the completion time of the tasks; and consequently, improves the performance of the scheduling process. The model of data access cost in the cluster-based data grid is defined as follows:

Assume that task i is assigned to node j of the gird in order to be processed. CT_j^i is the completion time of task i on node j and DC_j^i is the data access time to access the required data from the processing node j. R_i is the list of required data replicas of task i, which are stored in various nodes, and B_{ij} is the network bandwidth between node i and node j. If task i is assigned to node j in cluster $c1$, then the data access cost is defined as data communication cost which is given by

$$DC_j^i = \sum_{\text{For all } F_K \text{ in } R_i} |F_k| / B_{jK} \quad (1)$$

Where R_i is the set of the required replicas, F_K is the k^{th} data replica in R_i , $|F_K|$ is the size of the replica, B_{jK} is the network bandwidth between node j and the source node of the k^{th} replica.

The main factor of the data access cost measure is data communication cost resulted from data communications among different clusters. In the next step, the proposed DSAS used a size-aware task assignment policy to assign the submitted task to the best node of the selected cluster. The tasks submitted to the grid have a wide range of sizes. Studies show that variability of task sizes is an important factor for choosing the proper task assignment policy in clusters or server farms [16]. In many real application environments, there are many small tasks and a few large tasks. In most applications, a heavy-tailed distribution is a proper accurate distribution to model the task sizes distribution.

In DSAS, it was assumed that the number of required files for a task shows the size of the task. In the host nodes, tasks were served in first come first serve (FCFS) order. In the proposed approach, along with data-aware policy, a size-aware task assignment policy called Size Interval Task Assignment with Equal Load (SITA-E) was used to assign the submitted tasks to the proper node in the selected cluster. In the SITA-E algorithm, a specific size range is assigned to each host node and each task is assigned to the proper node according to its size. These size ranges are specified in such a way that the total load assigned to each node of the cluster is the same.

In SITA-E, it is assumed that the task size distribution has a mean value M and min is the lowest possible value and max is the largest possible value of tasks sizes. Let $F(x) = Pr\{X \leq x\}$ is the cumulative distribution function of the task sizes. Then with h host nodes in a cluster, the task size ranges for nodes can be computed as follows [16]:

$$\int_{x_0=min}^{x_1} x.dF(x) = \int_{x_1}^{x_2} x.dF(x) \ldots = \int_{x_{h-1}}^{x_h=max} x.dF(x) = \frac{M}{h} \quad (2)$$

Where tasks with sizes in the range of x_{i-1} to x_i are assigned to node i.

IV. Modification of the Open Source Simulator and Implementation Details

OptorSim, an open source grid simulator, was used to implement the proposed scheduling strategy and evaluate its performance. OptorSim is a java-based open source simulation tool for simulating data grid structure. It allows the evaluation of scheduling and replication algorithms in data grids. It includes

978-1-5090-4580-8/16/$31.00 ©2016 IEEE

essential components to simulate data grids with various data access and replication mechanisms [17, 18]. In this research, we implemented DSAS and incorporated it as a new scheduling strategy in OptorSim. We evaluated the performance of DSAS by comparing its performance with five other scheduling strategies with three replication mechanisms. Four scheduling strategies were embedded in OptorSim, the last one was a scheduling approach in a related work which we implemented and added it to OptorSim to provide the possibility of performance evaluation for the proposed scheduling approach. In the following Sections, the simulation environment of OptorSim and its properties, implementation details, simulation parameters and simulation results will be described.

A. Simulation Tool Environment

OptorSim simulates the data grid as a distributed system consisting of several nodes, each of which may have computational and data storage resources. In OptorSim each node may have zero or several computing elements and storage elements. The computing component provides computational resource to run submitted jobs and uses the required data replicas stored on storage elements of the nodes. If the required replicas were not found in the storage element of the processing node, it is fetched from other nodes. If there is not enough disk space in the storage element to store the new replica, a replacement mechanism, such as deleting the least frequently used (LFU) or least recently used (LRU), is used to get enough free space. In the simulated data grid there is a component called Resource Broker which accepts the submitted tasks and assigns them to suitable nodes based on the selected scheduling strategy [18]. In order to simplify the simulation of data grids, it is reasonable to assume data replicas are read only. Consequently, data can be replicated without any concern about data consistency. In OptorSim, in order to avoid the risk of deleting all copies of one file, there is a master copy for each file which cannot be deleted by replication mechanisms. The location of the master copies is specified in the configuration files of OptorSim [18, 11].

The Behavior of the simulation tool and input parameters of the system are defined in configuration files including the grid configuration, job configuration and simulation parameters files. The topology of the simulated grid, the content of each node, including the status of the resources, and the communication network between nodes are specified in the grid configuration file. The job configuration file contains information about distributed data files in grid, types of tasks and processing policies of each node (the tasks each node will accept and will run). The simulation parameters file contains the specification of the general simulation parameters [18].

B. Implementation Details for Modification of OptorSim

In this study, in order to implement the proposed DSAS as a new scheduling feature in OptorSim and also evaluate its performance in comparison with other scheduling strategies, a Java class of type ResourceBroker called DSASResourceBroker was implemented and added to the source of the simulation tool. It had a method called FindCE, which found the best computing element (host node) to process the submitted task according to DSAS strategy. In order to evaluate the performance of DSAS strategy, the HCS scheduling strategy proposed in [11] was also implemented and its ResourceBroker class, called HCSResourceBroker, was incorporated into OptorSim as well as common scheduling strategies.

Figure 1 illustrates the structure of classes and packages in OptorSim. It consists of 6 packages. The dependencies between them are shown in Fig. 1. Package org.edg.data.replication.optorsim includes the main classes of OptorSim and also uses the classes of other packages. Figure 1 depicts a portion of the classes of this package. Classes of DSASResourceBroker and HCSResourceBroker were added to the main package of OptorSim, i.e., org.edg.data.replication.optorsim. DSAS and HCS were imported and known as new scheduling strategies in the simulation tool. By selecting each of these new items in the configuration files, the scheduling of tasks submitted to the grid were done accordingly. Therefore, a modified OptorSim was developed with new scheduling features to implement various simulation scenarios and evaluate the performance of the scheduling strategies

C. Experiment Environment for Performance Evaluation of the Proposed Scheduling

The topology of the simulated cluster grid to evaluate the performance of the proposed scheduling strategy, is given in Fig. 2. This grid contains 3 clusters with 27 nodes in which some nodes (13 nodes) have both a computing and a storage element. *Node S* is a master storage node which contains the master copy of all files. Other nodes which do not have computing or storage elements are network nodes and are used for network connections. As shown in Fig. 2 connecting links between various nodes have differing bandwidths.

The topology of the grid shown in Fig. 2 with the status of the nodes and communication network were modeled by a numerical matrix. The numerical matrix of the simulated grid is given in Fig. 3. Each row of the matrix shows the information of one node. In this matrix the first column shows the number of worker nodes of the computing component. It is assumed that there is one or zero computing element in each node. The second and third column present the number of processors in each worker node and the number of storage elements in the node respectively.

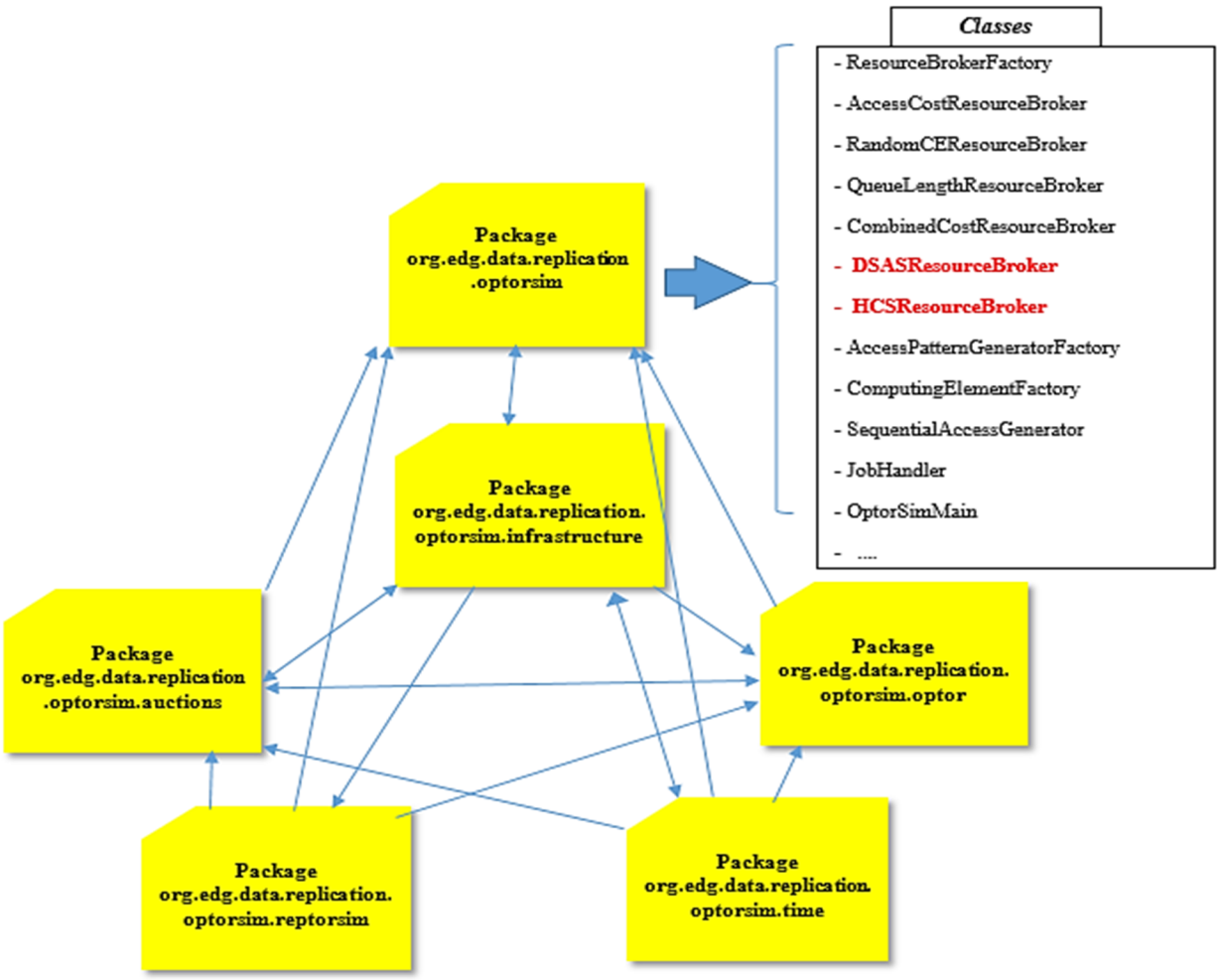

Fig. 1 Structure of OptorSim and the modification details

The fourth column indicates the size of the storage element in MB. Other columns illustrate the connections between nodes and the bandwidth of these connections. Table 1 illustrates the simulation parameters used in the experiment.

In our experiment, there were 30 types of submitted tasks, each of which required a different number of files. To make the experiment environment similar to real conditions, many of the task types were those that required only a few numbers of files, i.e. small size, and the remaining few task types were large size. While running, tasks were randomly selected and submitted to the resource broker based on the CMS DC04 pattern [18] in which the submission of the tasks is based on a Gaussian distribution. Tasks were scheduled on the suitable nodes according to the proposed scheduling strategy as soon as they arrive at the system. In DSAS strategy, the cluster of the nodes with lowest data access cost is selected as the best cluster in the first step. In the second step, according to the size ranges defined for server nodes using SITA-E policy, the submitted task was assigned to the proper node in the selected cluster. Tasks accessed the required data files based on Sequential Access pattern i.e. the order specified in the job configuration file. Different replacement mechanisms such as no replication, LRU, LFU were used to manage the storage space for storing new replicas in the nodes.

In this study, the performance of DSAS and five common scheduling algorithms of OptorSim including Random, QueueLength (ShortestQueue), Access Cost, Queue Access Cost (QAC) and HCS with three replication strategies including No Replication, LRU, LFU were evaluated based on the mean response time. As shown in Fig. 4 DSAS has the lowest mean response time for any replication mechanisms in comparison with other scheduling strategies and gives better performance.

978-1-5090-4580-8/16/$31.00 ©2016 IEEE

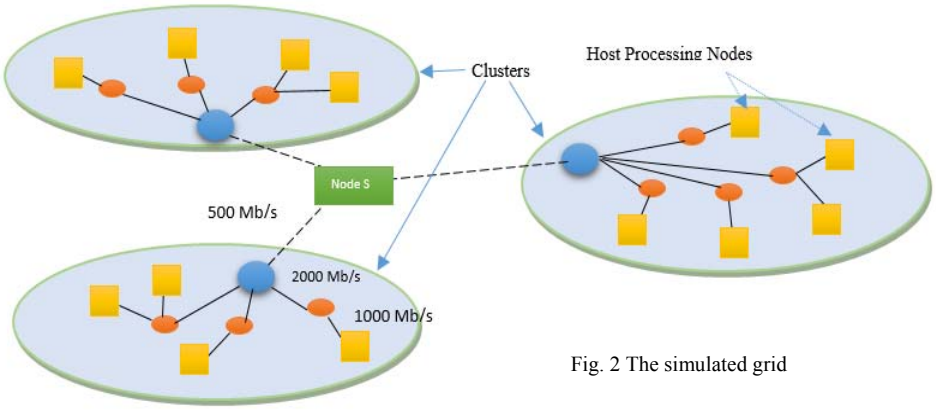

Fig. 2 The simulated grid

TABLE 1 Grid-job parameters

Grid- Job Parameters
Number of Task Types = 30
Number of Initial Files = 300
Size of Single File= 1 GB
Storage of Node S= 300 GB
Storage at Other Sites = 10 GB

```
1 1.0 1 10000 0. 0. 1000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1 1.0 1 10000 0. 0. 1000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 1000. 1000. 0. 0. 0. 0. 0. 2000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1 1.0 1 10000 0. 0. 0. 0. 1000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 1000. 0. 0. 0. 2000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1 1.0 1 10000 0. 0. 0. 0. 0. 0. 1000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 0. 0. 1000. 0. 2000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 2000. 0. 2000. 0. 2000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 500. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1 1.0 1 10000 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1 1.0 1 10000 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 0. 0. 0. 0. 0. 1000. 1000. 0. 0. 0. 0. 0. 2000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1 1.0 1 10000 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1000. 0. 0. 0. 2000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1 1.0 1 10000 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1000. 0. 2000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 2000. 0. 2000. 0. 2000. 0. 500. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 0. 0. 0. 0. 0. 500. 0. 0. 0. 0. 0. 0. 0. 500. 0. 2000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 500.
0 0.0 1 300000 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 2000. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1 1.0 1 10000 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1000. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1000. 0. 0. 0. 0. 0. 2000.
1 1.0 1 10000 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1000. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1 1.0 1 10000 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1 1.0 1 10000 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1 1.0 1 10000 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0.0 1 0 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
```

Fig. 3. Numerical matrix of the simulated grid

V. Conclusion

In this paper a new two-step scheduling strategy was proposed, implemented and incorporated into OptorSim. OptorSim is an open source data grid simulator which supports various scheduling and data replication strategies. The proposed data- and size- aware scheduling, was a proper strategy for scheduling data-intensive tasks in data grids. According to the simulation results, the combined data- and size- aware strategy not only optimized data access cost but also reduced the mean response time and mean waiting time of tasks compared with other common scheduling strategies of OptorSim. It prevented small tasks from waiting a long time in the queues for the large tasks to be completed. Hence, it improved the mean response time of the tasks in the grid system. Therefore, the modified OptorSim benefits from a new scheduling strategy which gives better performance compared to other common scheduling strategies.

978-1-5090-4580-8/16/$31.00 ©2016 IEEE

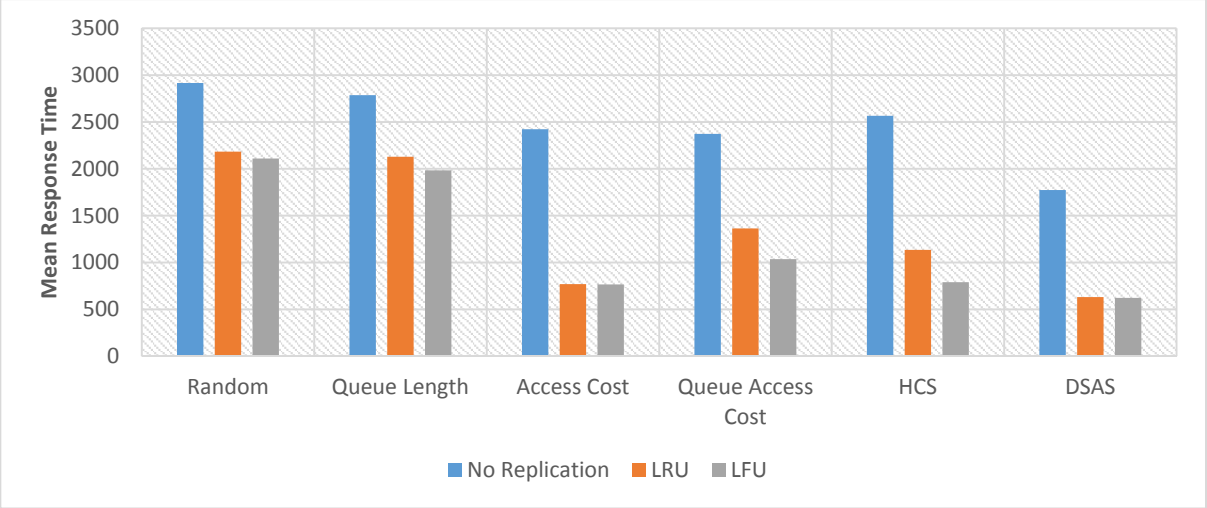

Fig. 4 Mean response time of 1000 submitted tasks with CMS DC04 pattern for various scheduling strategies and replication algorithms

REFERENCES

[1] F. Xhafa and A. Abraham, "Computational models and heuristic methods for Grid scheduling problems," Future generation computer systems, vol. 26, pp. 608-621, 2010.

[2] I. Foster and C. Kesselman, The Grid 2: Blueprint for a new computing infrastructure: Elsevier, 2003.

[3] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke, "The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets," Journal of network and computer applications, vol. 23, pp. 187-200, 2000.

[4] D. Cameron, R. Carvajal-Schiaffino, A. Millar, C. Nicholson, K. Stockinger, and F. Zini, "Optorsim: a simulation tool for scheduling and replica optimisation, in Data Grids," in proc. of CHEP, 2004.

[5] J. Kołodziej and S. U. Khan, "Data scheduling in data grids and data centers: a short taxonomy of problems and intelligent resolution techniques," in Transactions on Computational Collective Intelligence X, ed: Springer, 2013, pp. 103-119.

[6] R. Buyya and M. Murshed, "Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing," Concurrency and computation: practice and experience, vol. 14, pp. 1175-1220, 2002.

[7] A. Legrand, L. Marchal, and H. Casanova, "Scheduling distributed applications: the simgrid simulation framework," in 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGrid 2003, 2003, pp. 138-145.

[8] H. Casanova, A. Legrand, and M. Quinson, "Simgrid: A generic framework for large-scale distributed experiments," in Tenth International Conference on Computer Modeling and Simulation, 2008, pp. 126-131.

[9] H. J. Song, X. Liu, D. Jakobsen, R. Bhagwan, X. Zhang, K. Taura, et al., "The microgrid: a scientific tool for modeling computational grids," in ACM/IEEE Conference on Supercomputing, 2000, pp. 53-53.

[10] C. L. Dumitrescu and I. Foster, "GangSim: a simulator for grid scheduling studies," in IEEE International Symposium on Cluster Computing and the Grid, 2005, pp. 1151-1158.

[11] R.-S. Chang, J.-S. Chang, and S.-Y. Lin, "Job scheduling and data replication on data grids," Future Generation Computer Systems, vol. 23, pp. 846-860, 2007.

[12] S.-M. Park, J.-H. Kim, Y.-B. Ko, and W.-S. Yoon, "Dynamic data grid replication strategy based on Internet hierarchy," in International Conference on Grid and Cooperative Computing, 2003, pp. 838-846.

[13] X.-y. REN, R.-c. WANG, and K. Qiang, "Using optorsim to efficiently simulate replica placement strategies," The Journal of China Universities of Posts and Telecommunications, vol. 17, pp. 111-119, 2010.

[14] G. Belalem, "Economic Model for Consistency Management of Replicas in Data Grids with OptorSim Simulator," in International Conference on Networks for Grid Applications, 2008, pp. 121-129.

[15] N. N. Dang, S. B. Lim, and C. Yeo, "Combination of replication and scheduling in data grids," International Journal of Computer Science and Network Security, vol. 7, pp. 304-308, 2007.

[16] M. Harchol-Balter, M. E. Crovella, and C. D. Murta, "On choosing a task assignment policy for a distributed server system," Journal of Parallel and Distributed Computing, vol. 59, pp. 204-228, 1999.

[17] W. H. Bell, D. G. Cameron, A. P. Millar, L. Capozza, K. Stockinger, and F. Zini, "Optorsim: A grid simulator for studying dynamic data replication strategies," International Journal of High Performance Computing Applications, vol. 17, pp. 403-416, 2003.

[18] D. G. Cameron, R. Schiaffino, J. Ferguson, P. Millar, C. Nicholson, K. Stockinger, et al., "OptorSim v2. 0 installation and user guide," ed, 2004.

978-1-5090-4580-8/16/$31.00 ©2016 IEEE