



ORIGINAL ARTICLE

Implementation of Sub-Grid-Federation Model for Performance Improvement in Federated Data Grid

***Zarina Mohamad^a, Fadhilah Ahmad^a, Ahmad Nazari Mohd Rose^a, Fatma Susilawati Mohamad^a, and Mustafa Mat Deris^b**

^a Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Tembila Campus, 22200 Besut, Terengganu, Malaysia

^b Faculty of Information Technology and Multimedia, Universiti Tun Hussein Onn Malaysia, 86400 Batu Pahat, Johor, Malaysia

*Corresponding author: zarina@unisza.edu.my

Received: 26/12/2015, Accepted: 15/02/2016

Abstract

In this work, a new model for federation data grid system called Sub-Grid-Federation was designed to improve access latency by accessing data from the nearest possible sites. The strategy in optimising data access was based on the process of searching into the area identified as 'Network Core Area' (NCA). The performance of access latency in Sub-Grid-Federation was tested based on the mathematical proving and simulated using OptorSim simulator. Four case studies were carried out and tested in Optimal Downloading Replication Strategy (ODRS) and the Sub-Grid-Federation. The results show that Sub-Grid-Federation is 20% better in terms of access latency and 21% better in terms of reducing remotes sites access compared to ODRS. The results indicate that the Sub-Grid-Federation is a better alternative for the implementation of collaboration and data sharing in data grid system.

Keywords: Data grid, replication, scheduling, access latency

Introduction

A grid consists of a large number of heterogeneous resources with multiple domains of organizations for various applications implementation. These large scale applications require large amount of storages, high speed networking, good web technologies and high-end computing to facilitate collaboration and data-intensity in scientific research (Antunes and Helder, 2011; Hamrouni et al., 2015). Foster et al. (2001) state that distributed heterogeneous resources such as databases, scientific instruments and computers will be available for selection, discovery, exchange, sharing and aggregation on a grid platform. The main focus of Grid computing is the capability to provide computation for multiple organizational domains without any limit with regards to the number of organizations, departments, users and various applications (Sashi and Thanamini, 2010).

A data grid can access and manage huge amount of data sets up to terabytes and petabytes depending on the project requirements (Mansouri et al., 2013). The emerging trend in the scientific applications, demonstrate that the huge amount of data sets are processed

and produced by these applications. This gigantic size of data in applications such as in the security, user management, resource management, resource discovery, job scheduling, data replication, high speed network protocols, and data management requires the support and the functionality of data grid (Srikumar et al., 2006). Thus, it is of paramount significance for the use of data grid in analysis, storing, and sharing of data among scientific collaborative research around the globe.

Data sharing in data grid requires collaboration between different organizations and can be seen in e-Science (Hey and Trefethen, 2002; Katz and Zhang, 2014), as a scientific community for the collaborative environment. Likewise, e-Science relates to the sets of services, techniques, personnel and organizations, which have become a collaborative network (Antunes and Helder, 2011).

Grids can be organized in different ways. In particular, grids can be federated with each other. The federated model allows grids to fit into different institutions with independent administration and different locations that are interoperable with each other so that data can be shared.

Grid data management recommends six types of replication strategies for three different kinds of access patterns (Ranganathan and Foster, 2001a). Caching or No Replication, Plain Caching, Cascading Replication, Best Client, Fast Spread and Caching plus Cascading Replication are the six types of replication strategies proposed. The performance of the strategies can be evaluated based on the following three data patterns:

1. Random access pattern - this algorithm does not have pattern locality.
2. Temporal locality pattern – data have some forms of locality where files that have been retrieved, are still possible to be retrieved again.
3. Geographical and temporal locality pattern - Geographical locality signifies that the currently retrieved files by a user are expected to be retrieved by nearby users (Mckinley et al., 1996).

Various techniques have been suggested to effectively perform data replication across sites, and jobs to site assignment in data grid (Ranganathan and Foster, 2002b). Utilizing a simulator, a performance study on these various algorithms for scheduling have been carried out. For that purpose, a remote site has been selected by a scheduler to deliver a job based on one of the following algorithms:

1. JobLocally – Jobs that always locally run.
2. JobRandom – Randomly scheduling a job.
3. JobDataPresent - Scheduling a job to a site which has the least load of the required data.
4. JobLeastLoaded - Scheduling a job to a site with the least number of waiting jobs.

The above algorithms can be combined with any of the following replication strategies as follows:

1. DataDoNothing - No replication.
2. DataLeastLoad - When exceeding a file threshold level, a replication is performed at a site which has the least number of jobs in the waiting queue.
3. DataRandom- When exceeding a file threshold level, a replication is done at a random site.

The outcome of this research highlights the importance of data locality in job scheduling. Abawajy (2004) proposes a heuristic algorithm called Proportional Share Replica Policy as a solution to improve Cascading technique. This heuristic algorithm allocates a number of replicas of data to the best site. Firstly, the algorithm calculates the distribution ideal load. Subsequently, replicas are placed at a potential site that has the ability to serve the request for replica at a better rate or equal to the calculated ideal load. The ideal load is calculated using the following formula:

$$\text{Load} = \text{Totalrequest} / (\text{Originalcopy} + \text{Replicas})$$

Bandwidth Hierarchy Replication (BHR) by Park et al. (2004) is capable of reducing data access time which will help avoiding data grid network congestion. The BHR strategy will provide the opportunity of network level locality, when the required file resides in a place which has a large amount of bandwidth. The location of sites in a same data grid may be within the region where they are closely linked, also known as network region. A country, for example, can be considered as this network region. For multiple sites within a region, network bandwidth in a region will be larger than the network bandwidth between sites across the regions. Therefore, time for file fetching will be lessened if the requested file is available in the same region. The strategy of BHR is to decrease time for data access by means of increasing network level locality.

The concept of BHR has been studied and compared to HRS, and it is found to be similar to the idea of "network locality" (Chang et al., 2007). The difference between HRS and BHR can be observed in two aspects. In HRS, using required replica within the same cluster is always the top-most priority, while BHR searches all sites to find the most popular replica and has no distinction between intra-cluster and inter-cluster. It could be anticipated that HRS will avoid inter-cluster-communication and be stable in hierarchical network architecture with variable bandwidth. Secondly, HRS considers the popularity of replicas at site level, while BHR is based on cluster level.

Sashi and Thanamini (2010) propose an improved BHR concept for a topology where sites in the same regional location network are clustered together and named as Modified BHR. The improved algorithm attempts file replication within a region and, for future usage, keep it in the site where it has regularly been accessed. The usage of network bandwidth and job execution time can be reduced compared to the original BHR. Better job accomplishment could be achieved if the requested replica is retrieved within the same region. In the beginning, data are all generated within the master site and circulated to the Region Header. Access frequency of all files is determined and replicas of popular files are stored in sites where they are accessed for the maximum time, with consideration for the geographical and temporal locality. Geographical locality means files recently accessed by a client are likely to be accessed by nearby clients. Temporal locality means files accessed recently are likely to be accessed again.

Khanli et al. (2011) propose a dynamic replication method named Predictive Hierarchical Fast Spread (PHFS) that can read intensive data grids. PHFS improves the dynamic replication strategy in the data grid especially when involved with spatial locality and predictive methods. In addition, data objects hierarchical replication, PHFS optimizes the usage of storage resources to gain more localities in accesses in dissimilar layers from the multi-tier data grid environment.

Lee et al. (2012) suggest Popular File Replicate First algorithm (PFRF) based on an adaptive data replication algorithm that utilizes limited storage space of data grid. Eventually, this algorithm allows the patterns of data access to adapt the changes of users' requirements.

The model of replication strategy called Optimized Downloading Replication Strategy (ODRS) proposed by Jiang and Yang (2007) has been formulated based on expected access latency in replication strategy for federated data grid environment. ODRS is often used in one of the following cases:

- Case 1: There is replica of file in the request site,
- Case 2: There is no replica of file in request site but the replica is in other site in the same cluster,
- Case 3: There is no replica of file in request site and other sites in the same cluster, but there is replica of file in other clusters.

The rest of this paper is organized as follows. Section 2 presents the Sub-Grid-Federation model. The simulation results and analysis for the proposed model are presented in section 3. Section 4 summarizes and concludes this study.

Sub-Grid-Federation Model and Problem Formulation

Sub-Grid-Federation

The main objectives of *Sub-Grid-Federation* system are for data sharing and collaboration in the environment of extremely large database. The major characteristic of *Sub-Grid-Federation* is that the system is made of several sub data grids, which is modeled according to the federated data model. The model can best be viewed in the example shown in Fig. 1, which is made of three sub data grids and nine clusters. The arrangement is such that any two clusters/sub grids are just peer to each other and part of a logical independent system. In any cluster, there will always be a site identified as the header site, while the rest of the clusters are known as the normal sites.

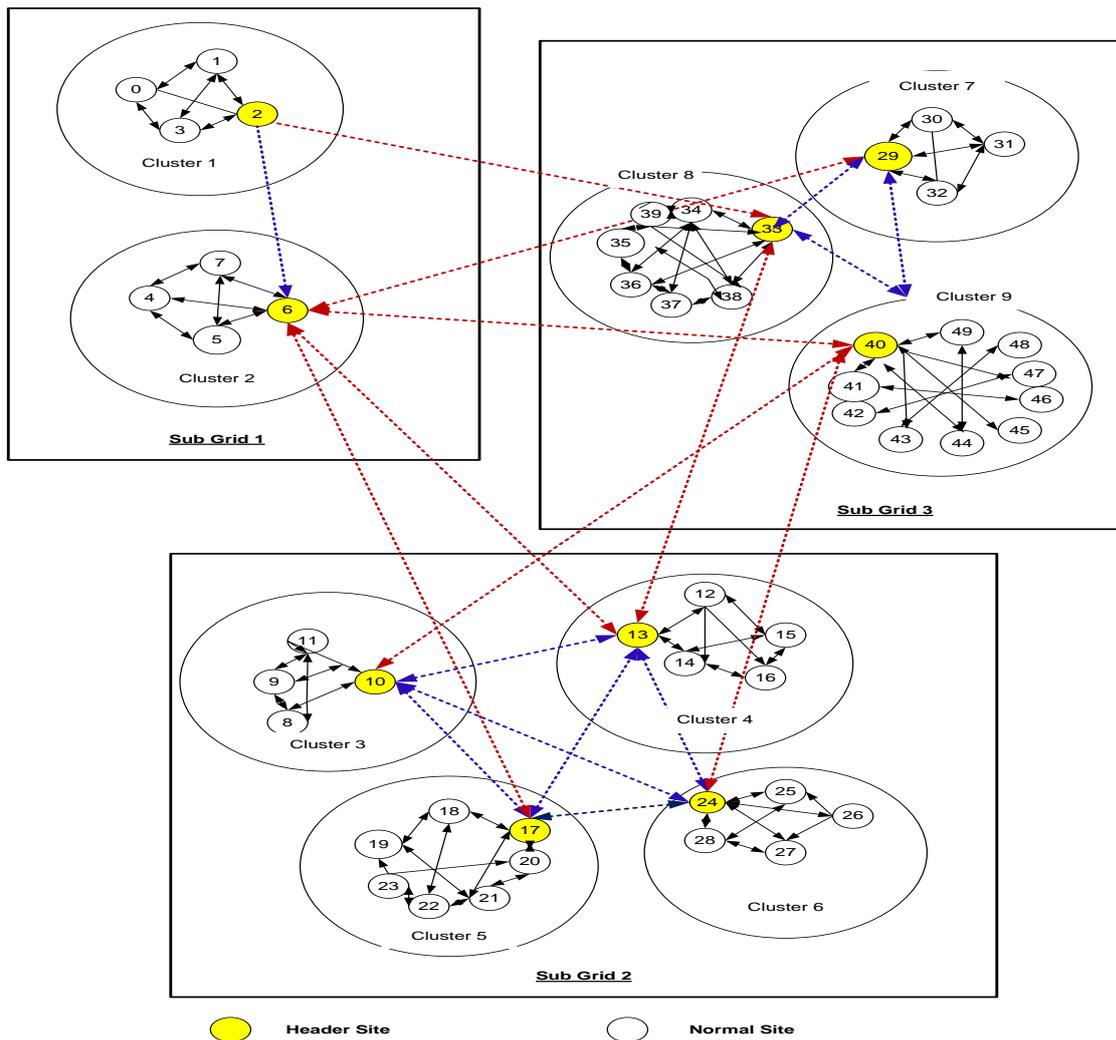


Figure 1. An instance of grouping in *Sub-Grid-Federation*

The normal site will be afforded finite local storage space for the sake of storing data replicas whilst the header site is responsible for storing index information of all the sites in their cluster. Besides that, the header site must also maintain index information of other interconnected clusters or sub grids in the data grid. One of the basic roles of the header site is to respond to request messages from any normal sites that of the same cluster. The header site will also be responsible to liaise with other header sites and response to its requests. The header site will use its index information to locate the normal sites that hold the

requested file. This mechanism guarantees that the file searched can be located if available. In *Sub-Grid-Federation*, data search will be constantly confined to the searched zone identified as 'Network Core Area' (NCA). Primarily, the core area of the federation grid is the inner most core which is defined as the NCA. It is within the NCA, the initial search is focussed on.

Terms and Definitions of Sub-Grid-Federation

There are M sites, Q clusters and G sub grids in the system. A site s_k belongs to a cluster c_i only and cluster c_i belongs to sub grid g_t only, with $k = 1, 2, \dots, M$ and $i = 1, 2, \dots, Q$ and $t = 1, 2, \dots, G$. For a cluster c_i , its size is m_i , with $m_i \geq 1$ and $\sum m_i = M$. While the size of sub grid g_t is q_t with $q_t \geq 1$ and $\sum q_t = Q$.

Assume that the size of cluster $m_1 \geq m_2 \geq \dots \geq m_Q$ and the size of sub grid is $q_1 \geq q_2 \geq \dots \geq q_Q$. Then assume that the sites belong to cluster c_i are $s_{M_{i-1}+1}, s_{M_{i-1}+2}, \dots, s_{M_i}$, where $M_i = \sum_{a=1}^i m_a$, $i = 1, 2, \dots, Q$ and $M_0 = 0$. While the clusters belong to sub grid g_t are $c_{Q_{t-1}+1}, c_{Q_{t-1}+2}, \dots, c_{Q_t}$ where $Q_t = \sum_{b=1}^t Q_b$, $Q_t = \sum_{b=1}^t Q_b$, $t = 1, 2, \dots, G$ and $Q_0 = 0$. On the other hand, a site s_k belongs to cluster $c_i = c(k)$ and a cluster $c(k)$ belongs to sub grid $g_t = g(k)$, where $c(k)$ mapping from site s_k to cluster c_i and $c(k)$ mapping from cluster $c(k)$ to cluster to sub grid g_t .

There are N unique files f_j in the data grid system, where $j = 1, 2, \dots, N$. For every site, the storage space is of the same size and it can store K replica files, so the system can store up to $M * K$ replica files. The data set stored in site n_k is D_k . Each file is associated with a normalized request rate of λ_j for file f_j per site, which is a fraction of all requests that are issued for the j th file. The normalized cumulative request rate of a site for all files in the system is;

$$\lambda = \sum_{i=1}^N \lambda_j = 1$$

For file f_j , there are r_j replicas uniformly distributed in the system, and assume $r_j \geq 1$. For a site n_k , there is at most one replica of f_j in its storage space. For file f_j , the probability of having a replica in site n_k is p_j . In Optimized Downloading Replication Strategy (ODRS) proposed by Jiang J. and Yang G. [1], there are three elements of hit ratio that have been considered which are as follows: $P(\text{local-hit})$, $P(\text{Intra-Grid-hit})$, $P(\text{Inter-Grid-hit})$.

Sub-Grid-Federation enhances the ODRS model by adding sub grids so that the hit ratio that is considered into account are: $P(\text{local-hit})$, $P(\text{Cluster-hit})$, $P(\text{Intra-Grid-hit})$ and $P(\text{Inter-Grid-hit})$. Consequently, when there is a request for a file; the request may be served in the following sequence: local site, local cluster, local sub grid or other sub grids. The cumulative (average) hit ratio of the local site is $P(\text{local-hit})$, indicating the probability of a file requests served by a local site from any sites in the system. Similarly, the cumulative (average) hit ratio of a local cluster, a local sub grid and of other sub grids are defined as $P(\text{Cluster-hit})$, $P(\text{Intra-Grid-hit})$ and $P(\text{Inter-Grid-hit})$ respectively. The sites in each cluster are connected with LAN topology.

Expected Access Latency of Sub-Grid-Federation Replication Strategy

Assume that a site n_k is requesting the file f_j . The expected access latency can be calculated by taking into concern the following four events.

Case 1: Event EI_{kj} , which means there is a replica of file f_j in n_k .

$$P(EI_{kj}) = P(f_j \in D_k) = \frac{r_j}{M} = \rho_j \quad (1)$$

Case 2: Event Ec_{kj} , which means there is no replica of file f_j in n_k , but file f_j hits in the other sites of cluster $c(k)$.

Case 3: Event Eg_{kj} , which means there is no replica of file f_j in local cluster $c(k)$ but file f_j hit in other clusters $c-kj$ of the same sub grid $g(k)$.

Case 4: Event Eg_{-kj} , which means there is no replica of file f_j in sub grid $g(k)$ therefore, file f_j must be hit in other sub grids. Then we have

$$P(Eg_{-kj}) = \left(1 - \frac{r_j}{M}\right)^{q_{g(k)}m_{c(k)}} = (1 - \rho_j)^{q_{g(k)}m_{c(k)}} \quad (2)$$

In order to get the formula for Case 3, the following calculations are constructed:

$P(Ec_{-kj}) - P(Ec_{-kj})$ where $P(Ec_{-kj})$ means there is no replica of f_j in local cluster $c(k)$, so the replica of f_j must be in other cluster $c-kj$ and $P(Eg_{-kj})$ is Case 4, therefore

$$P(Eg_{kj}) = P(Ec_{-kj}) - P(Eg_{-kj}) = (1 - \rho_j)^{m_{c(k)}} - (1 - \rho_j)^{q_{g(k)}m_{c(k)}} \quad (3)$$

Since there is at least one replica of f_j in data grid, the event in Case 2 is;

$$\begin{aligned} P(EI_{kj}) + P(Ec_{kj}) + P(Eg_{kj}) + P(Eg_{-kj}) &= 1 \\ P(Ec_{kj}) &= 1 - P(EI_{kj}) - P(Eg_{kj}) - P(Eg_{-kj}) = 1 - \rho_j - (1 - \rho_j)^{q_{c(k)}m_{g(k)}} \end{aligned} \quad (4)$$

Having the above probabilities, the expected cumulative hit ratios can be calculated as;

$$P(\text{local} - \text{hit}) = \sum_{j=1}^N \frac{\lambda_j}{\lambda} P(EI_{kj}) = \sum_{j=1}^N \lambda_j \rho_j \quad (5)$$

$$PP(\text{Cluster} - \text{hit}) = \sum_{j=1}^N \frac{\lambda_j}{\lambda} P(Ec_{kj}) = \sum_{j=1}^N \lambda_j \left\{ 1 - \rho_j - (1 - \rho_j)^{m_{c(k)}} \right\} \quad (6)$$

$$PP(\text{Intra} - \text{Grid} - \text{hit}) = \sum_{j=1}^N \frac{\lambda_j}{\lambda} P(Eg_{kj}) = \sum_{j=1}^N \lambda_j \left[(1 - \rho_j)^{m_{c(k)}} - (1 - \rho_j)^{q_{g(k)}m_{c(k)}} \right] \quad (7)$$

$$P(\text{Inter} - \text{Grid} - \text{hit}) = \sum_{j=1}^N \frac{\lambda_j}{\lambda} P(Eg_{-kj}) = \sum_{j=1}^N \lambda_j (1 - \rho_j)^{q_{c(k)}m_{g(k)}} \quad (8)$$

The relation between t_l , t_o , t_g and t_G . λ_j is the normalized request rate for file f_j at this site. In this model, the costs of accessing a replica file are as follows;

- t_l = costs when accessing a replica file from a site's local storage space
 - t_o = costs when accessing a replica file from a remote site of the same cluster
 - t_g = costs when accessing a replica file from remote site of other cluster but within same sub grids.
 - t_G = costs when accessing a replica file from remote site of other sub grids
- where $t_l \leq t_o \leq t_g \leq t_G$.

$t(n_k, f_j)$ denotes the access latency of the site n_k when requesting file f_j . To get the expected access latency for a requested file, applying the Equation (1) to Equation (4). The expected access latency of site n_k for requesting file f_j is $E(t(n_k, f_j))$

$$\begin{aligned} &= P(EI_{kj}) \times t_l + P(Ed_{kj}) \times t_o + P(Eg_{kj}) \times t_g + P(Eg_{-kj}) t_G \\ &= t_o + (t_l - t_o) \rho_j + (t_g - t_o) (1 - \rho_j)^{m_{c(k)}} + (t_G - t_g) (1 - \rho_j)^{q_{g(k)} m_{c(k)}} \end{aligned}$$

For simplicity, let $(t_l - t_o) = t_{lo}$, $(t_g - t_o) = t_{gc}$, and $(t_G - t_g) = t_{Gg}$.

Then, $E(t(n_k, f_j)) = t_o + t_{lo} \rho_j + t_{go} (1 - \rho_j)^{m_{c(k)}} + t_{Gg} (1 - \rho_j)^{q_{g(k)} m_{c(k)}}$

By considering the request rate of every file, the expected access latency for site n_k requesting any file in data grid can be computed as follows;

$$\begin{aligned} t_k &= \sum_{j=1}^N \frac{\lambda_j}{\lambda} \times E(t(n_k, f_j)) \\ &= \sum_{j=1}^N \lambda_j \left[t_o + t_{lo} \rho_j + t_{go} (1 - \rho_j)^{q_{g(k)} m_{c(k)}} + t_{Gg} (1 - \rho_j)^{q_{g(k)} m_{c(k)}} \right] \end{aligned} \tag{9}$$

Minimizing t is the objective of this research, with the above constraints;

- 1) The number of all replicas in the system is less than the total storage space of a data grid;

$$\sum_{j=1}^N r_j \leq KM, \text{ or } \sum_{j=1}^N \rho_j \leq K$$

- 2) The number of the replicas of a file is at least one, and the most is the number of system size, i.e.

$$1 \leq r_j \leq M, \text{ or } \frac{1}{M} \leq \rho_j \leq 1, \quad j = 1, 2, 3, \dots, N$$

Then, the constrained optimization problem is;

$$\begin{aligned} \text{Min} & \left\{ t = \sum_{j=1}^N \lambda_j \left[t_o + t_{lo} \rho_j + t_{go} (1 - \rho_j)^{m_{c(k)}} + t_{Gg} (1 - \rho_j)^{q_{g(k)} m_{c(k)}} \right] \right\} \\ \text{s.t.} & \sum_{j=1}^N \rho_j \leq K, \frac{1}{M} \leq \rho_j \leq 1, \quad j = 1, 2, 3, \dots, N \end{aligned} \tag{10}$$

The Implementation of Sub-Grid-Federation and Discussion

OptorSim provides simulations of file replication strategies such as replica placement, replication scheduling and replica consistency maintenance. This simulation framework uses Java programs that customize simulation scenario, defines the network topology and jobs, and other relevant items with a set of configuration files. Fig. 2 shows the architecture for a conceptual model used in *OptorSim* simulator. A *Resource Broker (RB)* controls the scheduling of job to Grid Sites. Each site handles its file content with a *Replica Manager (RM)*, with which a *Replica Optimizer (RO)* contains the replication algorithm which drives automatic creation and deletion of replicas.

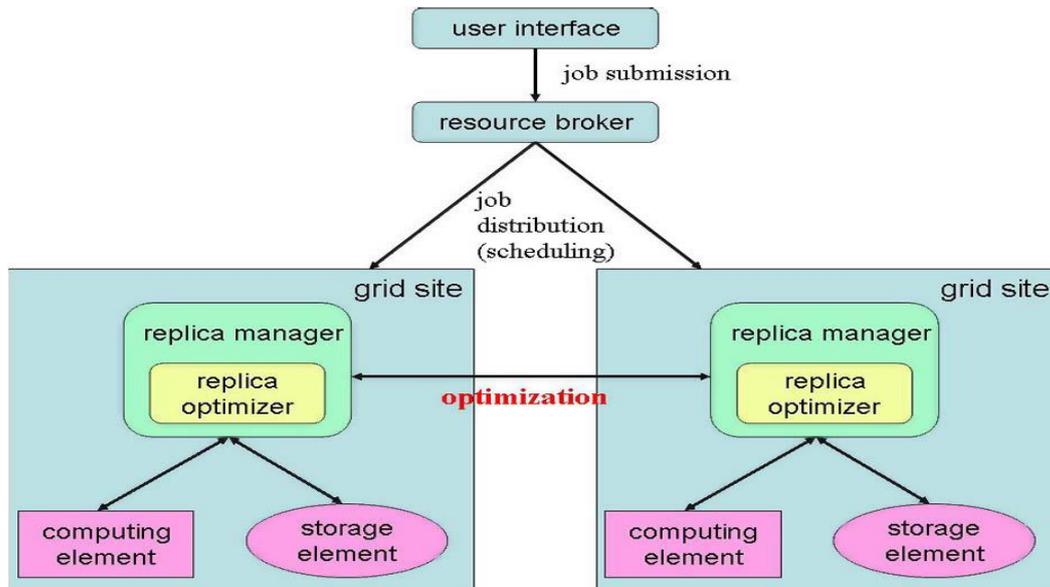


Figure 2. Grid architecture for *OptorSim*

The Grid Configuration File

In the experiments conducted by Lee et al. (2012) and Mansouri et al. (2011), the value range of resource capacity and its processing speed, together with network bandwidth are used as the basis of the research. However, the sub grids components that consist of the number of clusters and sites are the basis for the study on the *Sub-Grid-Federation* in this paper. The parameters of grid configuration file are shown in Table 1. The network bandwidth between sites in a cluster is set to 1000 Mbps for the network coverage of sites from different clusters (intra sub grid/ intra region). Nevertheless, as for the same sub grid the network bandwidth is set to 100 Mbps and as for the network bandwidth between sites of a different sub grid (inter-region) is set at 10 Mbps.

There are three sub grids and every sub grid has more than two clusters. Each cluster has more than two sites and every site has CE with linked SE. In Sub Grid 1, there are 2 clusters, Site 2, is the master files for Cluster 1 and Site 4 is the master files for Cluster 2. The next Sub Grid is Sub Grid 2, which Site 8 is the master files for Cluster 3, Site 12 as the master files for Cluster 4, Site 18 is the master files for Cluster 5 and Site 25 is the master files for Cluster 6. Sub Grid 3 consists of 3 clusters, with Site 20 as the master files for Cluster 7, Site 34 is the master files for Cluster 8 and Site 41 is the master files for Cluster 9. Site 2, Site 6, Site 10, Site13, Site 17, Site 24, Site 29, Site 35 and Site 40 are the header sites on each cluster. Each blue dotted line between two header sites demonstrates the

communication of inter-cluster, and the red dotted line demonstrates the communication of inter-sub grid (inter region).

Table 1. Simulation parameters for grid file

Topology Parameters	Value
Number of sub grids (region)	3
Number of clusters in each region	More than 1
Number of sites in each cluster	More than 1
Storage space in each site	10GB
Intra-site connectivity bandwidth	1000Mbps
Inter-cluster connectivity bandwidth	100Mbps
Inter-sub grid connectivity bandwidth	10Mbps

Results and Discussion

This section elaborates the comparison between the proposed algorithms, *Sub-Grid-Federation* replication strategy (SGFRS) against ODRS. The ODRS (Jiang and Yang, 2007) is designed basically for replication algorithm in basic model of federated data grid system. In general, the main thrust of ODRS is to assign jobs for the process of replication of data either within cluster or other cluster of the federated system. The main concern of this algorithm is to reduce access time, and it is achievable by distributing the jobs to many available sites.

Total Access Time

The comparison between SGFRS and ODRS are based on the measured access time. Fig. 3 displays the runtimes against varying number of jobs for the two algorithms. The total execution time using ODRS is longer, and is about 20% more than the execution time taken when simulated with SGFRS. Therefore SGFRS fared better than ODRS, in terms of performance. This can be seen when simulated with 400 numbers of jobs. In that simulation, the total job execution time taken by SGFRS is 1191,000ms, while it took around 1498,060ms for ODRS.

Inter-Communication Bandwidth

Fig. 4 shows the average job processing time for 500 jobs. In the experiment, the bandwidth of sites within inter-cluster has been set constantly at 100Mbps, while the bandwidth between sites, within the inter-sub-grid has been set in the range of 10 Mbps to 900Mbps. When comparing SGFRS with ODRS with varying inter-communication bandwidth, SGFRS has shown faster execution time. Therefore, it can be concluded that SGFRS can be effectively utilized when inter-cluster bandwidth (inter-cluster bandwidth but within same sub grid) is larger than inter-sub-grid bandwidth because ODRS has no distinction between intra-sub-grid and inter-sub-grid.

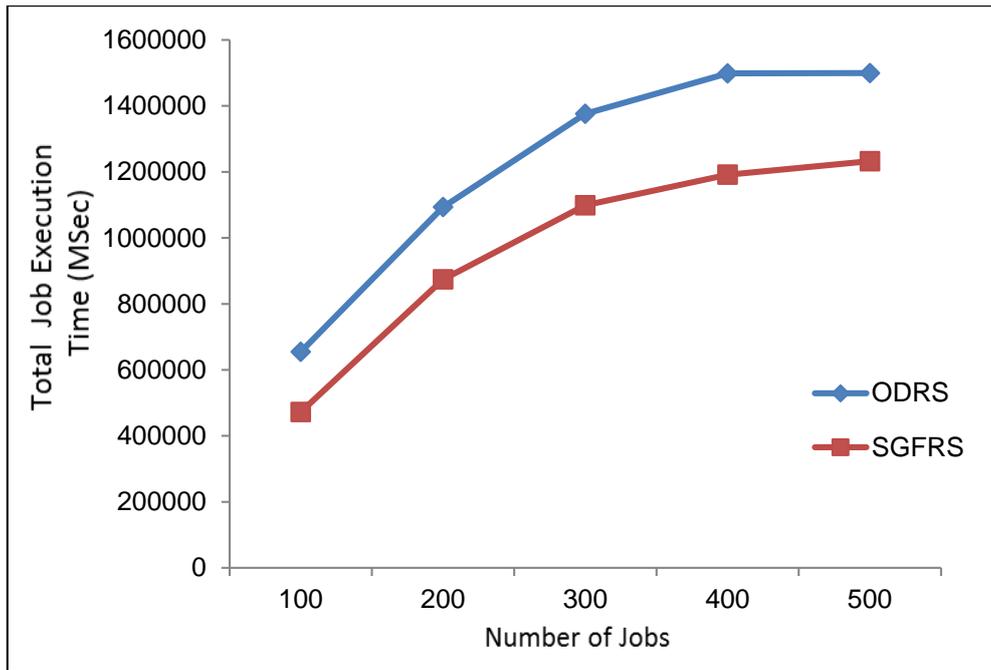


Figure 3. Average job time based on varying number of jobs

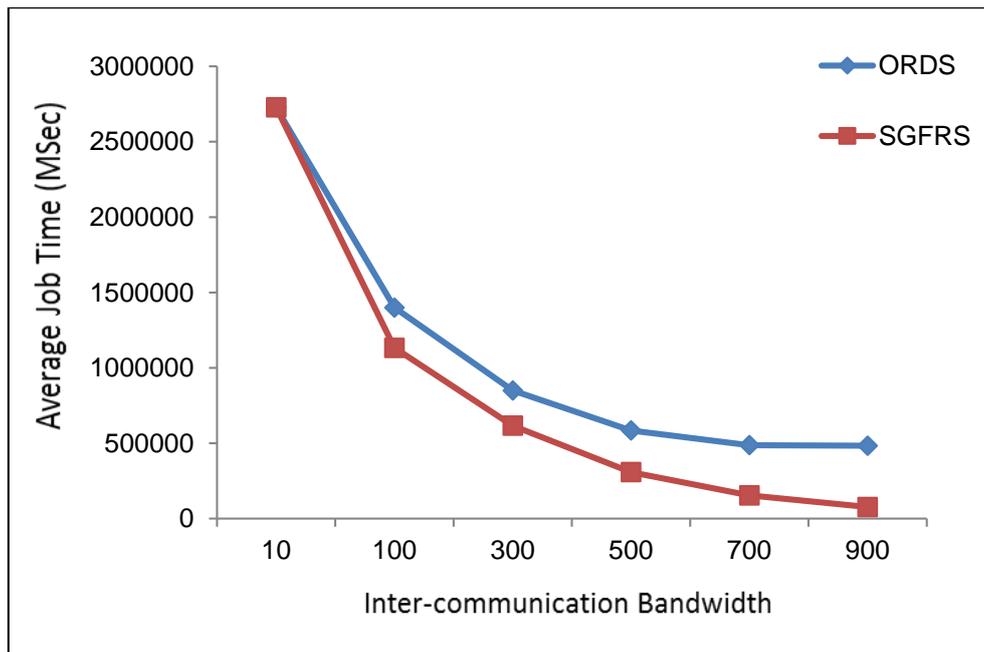


Figure 4. Average job time based on varying inter-communication bandwidth for 500 jobs

The Number of Inter-Communication

The job execution time is the summation of file transmission time, queuing time and job processing time. Since the most vital factor in influencing time for job execution of data-intensive job in a data grid is the time for file transmission, the proposed scheduling algorithm SGFS can effectively reduce the time for file transmission by means of valid scheduling and proper data replication as demonstrated by the experiments. Selecting the best site according to position of data by the job, the SGFS can effectively decrease the number of

inter sub grid (region) communication. As shown in Fig. 5, inter cluster or inter sub grid (region) communications in SGFS are strictly curtailed in the local sub grid only when replica of file is non-existent in the local site and local cluster. But as for the same scenario, ODRS will expand its access to other regions as well. As found during the simulation for ODRS, 21% of files are accessed from other regions although the file exists in the same region. Therefore the simulation is a testament that SGFS performs less inter-communications when there is no file in the local site and local cluster.

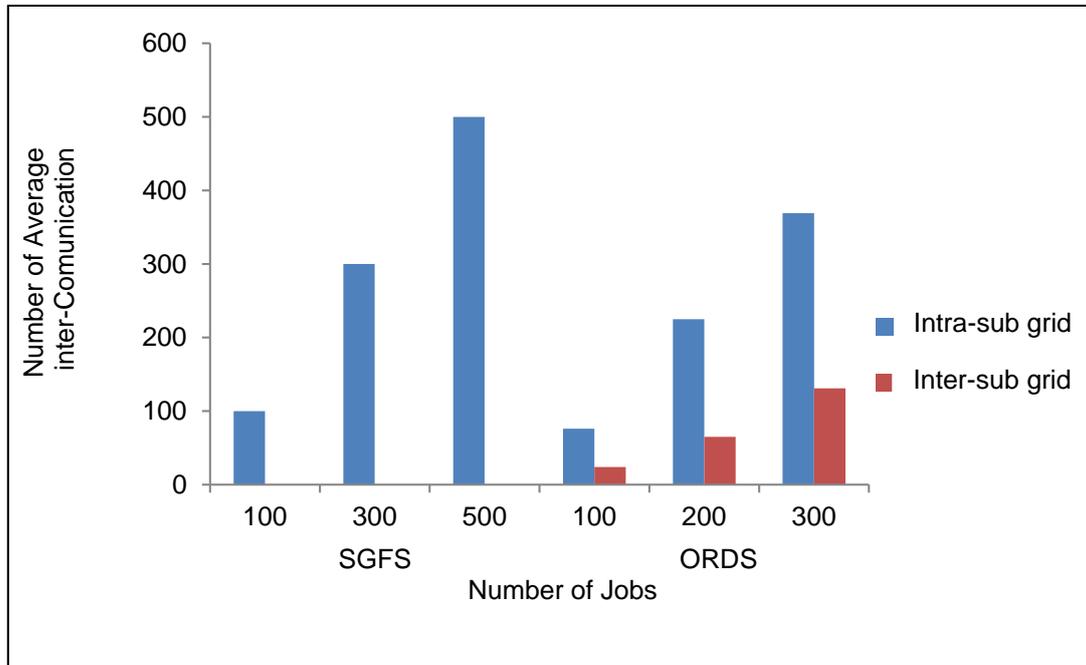


Figure 5. Average number of inter-communication

Conclusion

The *Sub-Grid-Federation* is designed operating in the environment of reduced access latency, when the local data may not be sufficient enough to fulfill their users' requirements. *Sub-Grid-Federation* permits the use of optimized data transparently from the federation data grid. The algorithm will always attempt to map to the nearest possible data by accessing the data defined by 'Network Core Area' (NCA). When the replicas of data are distributed in the federated system, the user can access the nearest possible data; this will reduce the data access latency. There are four possible expected access latencies in Sub-Grid-Federation: (i) Case 1: There are replicas of data in request site; (ii) Case 2: There are no replicas of data in request site but replicas of data in other sites in the request cluster; (iii) Case 3: There are no replicas of data in the request cluster but replicas of data in other clusters in same sub grid; (iv) Case 4: There are no replicas of data in request sub grid but replicas of data in other sub grid. It has been demonstrated that Sub-Grid-Federation has been successfully being addressed i) *Sub-Grid-Federation* is 20% better in terms of access latency, and ii) *Sub-Grid-Federation* is 21% better in terms of reducing remotes sites access compared to ODRS. The *Sub-Grid-Federation* model can be further enhanced by considering its implementation in federated cloud environment.

Acknowledgement

The authors gratefully acknowledge financial support from the Ministry of Higher Education, under the project FRGS-2-2013-ICT07-UniSZA-0201.

References

- Abawajy, J.H. (2004). Placement of File Replicas in Data Grid environments. Heidelberg: Springer.
- Antunes, G. & Pina, H. 2011. Using Grids Federations for Digital Preservation. In the 8th International Conference on Preservation of Digital Objects (IPRES 2011). November 1 - 4, 2011, Singapore.
- Chang, R. S., Chang, J. S., & Lin, S. Y. (2007). Job scheduling and data replication on data grids. *Future Generation Computer Systems*, 23(7), 846-860.
- Foster, I., Kesselman, C., & Tuecke, S. (2001). The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, 15(3), 200–222.
- Hamrouni, T., Hamdeni, C., & Ben Charrada, F. (2015). Impact of the distribution quality of file replicas on replication strategies. *Journal of Network and Computer Applications*, 56, 60–76
- Hey, T., & Trefethen, A. E. (2002). The UK e-Science Core Programme and the Grid. *Journal of Future Generation Computer Systems*, 18(8), 1017–1031.
- Jiang, J., & Yang, G. (2007). An Optimal Replication Strategy for data grid system. *Front Computer Science China*, 1(3), 338-348.
- Katz, D. S., & Zhang, Z. (2014). Special issue on eScience infrastructure and applications. *Future Generation Computer Systems*, 36, 335-337.
- Khanli, L. M., Isazadeh, A., & Shishavanc, T. N. (2011). PHFS: A dynamic replication method, to decrease access latency in multi-tier data grid. *Future Generation Computer Systems*, 27(3), pp. 233-243.
- Lee, M. C., Leu, F. Y., & Chen, Y-P. (2012). PFRF: An adaptive data replication algorithm based on star-topology data grids. *Future Generation Computer Systems*, 28(7), 1045-1057.
- Mansouri, N., Dastghaibyfar, G.H., & Horri, A. (2011). A novel job scheduling algorithm for improving data grid's performance. 2011, P2P, Parallel, Grid, Cloud, and Internet Computing. International Conference on, P2P, Parallel, Grid, Cloud, and Internet Computing, International Conference on 2011, pp. 142-147, doi:10.1109/3PGCIC.2011.30
- Mckinley, K.S., Carr, S., & Tseng, C-W. (1996). Improving data locality with loop transformations. In *ACM Trans. Program. Lang. Syst.* Vol. 18. (pp. 424–453) New York: ACM Press.
- Park, S-M., Kim, J-H, Go, Y-B., & Yoon, W-S. (2003). Dynamic grid replication strategy based on internet hierarchy, in: International Workshop on Grid and Cooperative Computing, in: Lecture Notes in Computer Science, Vol. 3033, pp. 838-846.
- Raganathan, K., Lamnitchi, A., & Foster, I (2002). Improving Data Availability through Model-Driven Replication for Large Peer-to-Peer Communities. In: Proceedings of Global and Peer-to-Peer Computing on Large-Scale Distributed Systems Workshop, Berlin, Germany.
- Ranganathan, K., & Foster, I. (2001). Design and evaluation of dynamic replication strategies for a high-performance data grid. In: International conference on computing in high energy and nuclear physics. Beijing, China, 2001.

Sashi, K., & Thanamini, A. S. (2010). Dynamic replication in a data grid using a Modified BHR Region Based Algorithm. *Future Generation Computer Systems*, 27(2011), 202-210.

Venugopal, S., Buyya, R., & Ramamohanarao, K. (2009). A Taxonomy of Data Grids for Distributed Data Sharing, Management, and Processing. *ACM Computing Surveys*, 38(1), 1-53.

How to cite this paper:

Mohamad, Z., Ahmad, F., Rose, A.N.M., Mohamad, F.S. & Deris, M.M. (2016). Implementation of sub-grid-federation model for performance improvement in federated data grid. *Malaysian Journal of Applied Sciences*, 1(1), 55-67.