Framework for Evaluating Update Propagation Techniques In Large Scale Data Grid

Mohammed Radi[1], Ali Mamat[1], M.Mat Deris [2], Hamidah Ibrahim[1], Subramaniam Shamala[1]
[1] Faculty of Computer Science and Information Technology, Universiti Putra Malaysia,
43400 Serdang, Selangor, Malaysia

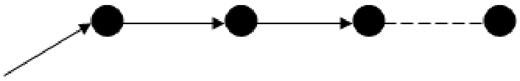
[2] Faculty of Information Technology and Multimedia, University of Tun Hussein Onn,
86400 Parit Raja, Batu Pahat, Johor, Malaysia

Abstract
This paper introduces a framework for evaluating update propagation techniques in large scale Data Grid. The framework qualitatively compares the update propagation techniques using an analytical model based network queuing model. To achieve accuracy, the framework takes into account, the computing element, storage element capacity in additional to the arrival rate, and the computational complexity moreover it consider the heterogeneity of the grid environment and the heterogeneity of the jobs.. The framework can predict the utilization, response time, update propagation response time of the update propagation techniques.

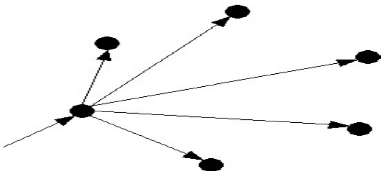
I. Introduction

Replication introduces the problem of maintaining consistency among the replicas when the files are allowed to be updated. The update information should be propagated to all replicas to guarantee correct read of the remote replicas. The design of replica consistency techniques is one of the foremost problems in the Data Grid [1,2]. Replica consistency techniques try to achieve efficient and scalable consistency maintenance among data grid sites. A number of replica consistency techniques have been proposed for data grid [2,3,4,5,6,7]. The update propagation with a single master scenario is a common technique for asynchronous replication. It can be done by a line technique as in Figure 1-a or in a radial way as in Figure 1-b [2,3,4,5,6,7]

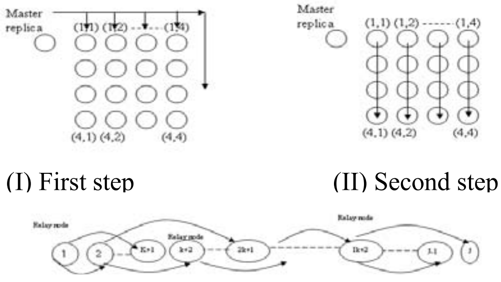
In [8] we propose an update propagation technique called UPG in which Updates reach other replicas using a propagation technique based on nodes organized into a logical structure network as shown in Figure 1-c.



(a)Propagation of update message using a line technique



(b)Propagation of update message using a radial technique



(I) First step (II) Second step

(III) Detail of the second step

(c)Propagation of update message using UPG technique

Figure1: the update propagation techniques

Until now there has been no systematic way to classify and compare the update propagation techniques. In this paper, we propose a framework for qualitative evaluation and comparing the update propagation technique.

The rest of this paper is organized as follow; section 2 described the assumed model. In section 3 the network queuing model framework components are described. Experiment results and discussions are given in section 4. We conclude the paper in section 5.

II. System Model and Goal

The system considered in this paper is a large scale data grid consist of a set of replica sites. The goal of update propagation technique is to propagate the update information carried put at the master site to all replica

Manuscript received September 5, 2008.
Manuscript revised September 20, 2008.

sites in order to maintain the replica consistency in the data grid environment.

The proposed evaluation framework is based on the following assumptions that were made in order to simplify the analysis of the update propagation techniques. These assumptions are widely used in similar studies [1,4,9,10,11]:

- The Grid consists a large number N of grid sites and there are R data sets in the system.
- Each site has a computing element and storage element.
- There is a communication system that allows any node to communicate to each other.
- We assume a full replication, that's means each grid site replicate all the R datasets at its local storage element.
- The job arrives to the grid follow a poison process with a mean rate λ job peer time unit.
- Two types of requests are allowed in the grid system job operation (J) and update operation (U), and the request is with probability P and 1-P for J and U respectively and the entire request only access one data set.
- The destination for each request is randomly chosen from the grid sites, and no priority is applied.

Two types of requests are allowed in the grid system, which are job request and update request. The job and Update requests served at a selected site; by accessing the computing element and the storage element. Such system can be modeled as an open queuing network model with multiple job classes.

III. Network Queuing Model

Network Queuing Mode (NQM) provides a framework to defined, parameterization and evaluation of a system. NQM can be evaluated using mean value analysis framework as in Figure 2. We model the Data Grid as open queuing network model with multiple job classes, the solution adapt for this model follows the operational analysis approach described in[12,13]. The model considers Separable Queuing Networks which are a subset of networks of queues distinguished by the fact that the equilibrium distribution of the states of the models has a particularly simple analytical model.

A. Definition

Defining a network queuing model of a particular system is made relatively straightforward by the close correspondences between the attributes of the queuing network model and the attributes of the system.

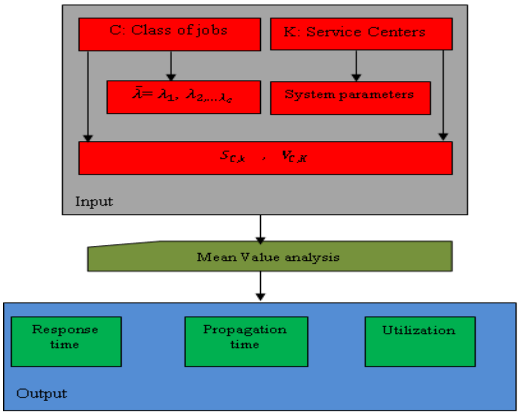


Figure2: mean value analysis framework

The service centers, customer classes, are the attributes of the network queuing model. The relationship between those two components is defined by defining the number of visit the service time and the arrival rate. Figure 3 shows the interaction between the components of the developed queuing network model.

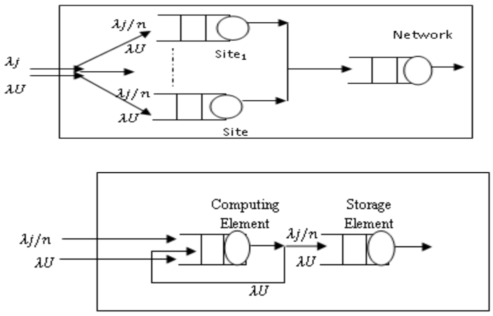


Figure3: Network queuing model

- **Service centers**

Computing element and storage element are the service centers in the grid system. The service centers are defined as follows:

Computing Element:

Notation as $Com\_Element_i$: for each Grid site *i*, there is one computing element to provide the grid user with the job execution cycles. The numbers of computing elements in the system are defined as:

$$Com\_Element_i \quad i=1...N$$

A Storage element:

Notation as $Storg\_Element_i$: for each Grid site I, there is one storage element to provide the grid user with the storage capacity; it may be a mass storage system or disk pool. The number of storage element in the system is defined as:

$$Storg\_Element_i \quad i=1...N$$

- **Customer Classes**

Customers represent users or transactions (requests) of the system. Two types of requests are allowed in the grid system which are job request-J and update request-U. The job request involves processing the request at the selected site, including accessing the computing element and the storage element. The propagation algorithm overhead is divided into two parts: default part and conditional part. The default part accounts for the time to check the FIFO order, deciding the message sender and the target file, checking the relay list, and to whom it needs to send a message. The conditional part counts for the time to maintain timestamp and handles acknowledgements. The communication overhead includes message processing at the computing element for sending and receiving the message and running the network protocol, propagation process and the operating system. Also the message encounters network delay involving transmission delays, propagation delays, and media access time. The classes of jobs are defined in Table 1.

Table 1: The Class of Jobs

Class	Description
J	Job operation in the processing phase
U	Update operation in the processing phase
P_Default	Processing the default part of propagation algorithm
P_Conditional	Processing the conditional part of the propagation overhead
P_C	Processing the message in the communication phase

- **Service time**

The service time for each class of jobs at the service centers is defined as the time needed by the service center to process such customer class.

Computing Element

The computing element is invoked by J, U, P_default, P_conditional, and P_C customer classes. Service time for each class of customer I at a computing element k for each visit can be defined as $S_{I,Com\_Element_K}$, where I can be J, U, P_default, P_conditional, or P_C.

The service time for J and U to be processed at the computing element k is modeled with an exponentially distribution, with mean $S_C^{J\_proc\_K}$, $S_C^{U\_proc\_K}$ respectively. The value for the service time is dependent on the power of the computing element of the site k $Com\_Element\_Power_K$ and the number of instructions needed to process the request Instructions_Need. $Com\_Element\_Power_K$ is modeled as a uniform distribution. The Instructions_Need is modeled as exponential distribution with mean $Instructions\_Need_C^{J\_Proc}$ or the Job request and $Instructions\_need_C^{U\_Proc}$ for the update request. The service time for J request at a site k is defined as

$$S_{J,Com\_Element_K} = S_C^{J\_proc\_K} = \frac{Instructions\_Need_C^{J\_Proc}}{Com\_Element\_Power\_K} \qquad (1)$$

And the service time for U request at a site k is defined as:

$$S_{U,Com\_Element_K} = S_C^{U\_proc\_K} = \frac{Instructions\_Need_C^{U\_Proc}}{Com\_Element\_Power\_K} \qquad (2)$$

Service time for P_C to be processed at the computing element k is modeled as exponential distribution with the mean $S_C^{P\_Proc\_com\_Element\_K}$, the value of the service time depends on the power of the computing element at site $Com\_Element\_Power_K$ and the number of instruction needed to process the message, which depends on the message size. As mentioned the power of the computing element is a uniform distribution and the number of instruction needed is an exponential distribution with the mean $Com\_Message_C^{Proc\_Message\_I}$

$$S_{P\_C,Com\_Element_K} = S_C^{P\_C\_Com\_Element\_K} = \frac{Com\_message}{Com\_Elemen} \qquad (3)$$

The service time for $P\_Default$ to be processed at the computing element k is $S_{P\_Default,Com\_Elemnt_K}$, which depends on the number of instructions needed to process the default part $Default\_Instruction\_need$ and the power of the computing element.

$$S_{P_{Default},Com_{Elemnt_K}} = \frac{Default\_Instruction\_need}{Com\_Element\_Power\_K} \qquad (4)$$

The service time for P_Conditional to be processed at the computing element k is $S_{P\_Conditional,Com\_Elemnt_K}$ which depends on the number of instructions needed to process the conditional part $Conditional\_Instruction\_need$ and the power of the computing element.

$$S_{P_{Conditional},Com_{Elemnt_K}} = \frac{Conditional\_Instruction\_need}{Com\_Element\_Power\_K} \qquad (5)$$

Storage Element

The storage element is invoked by job request and the update request.

The service time for J and U to be processed at the storage element of the site k is $S_{J,Stor\_Element_K}$ and $S_{U,Stor\_Element_K}$ respectively. Which are modeled to be an exponential distribution with mean $S_C^{J\_Stor\_Element\_K}$ $S_C^{U\_Stor\_Element\_K}$. The value of the service time at a site k is dependent on the number of access time needed to access the storage element Access_Need. Storage element power at the site k is Stor_Element_Power_k. The power of the storage element at a site k is modeled to be a uniform distribution, and the access time needed $Access\_Need_{C}^{J\_Acc\_Stor\_K}$ and $Access\_Need_{C}^{U\_Acc\_Stor\_K}$ as exponential distribution with mean $S_{C}^{J\_Storage\_K}$ and mean $S_{C}^{U\_Stor\_K}$. The service time for J request at a site k is defined as:

$$S_{J,Stor_{Element_K}} = S_{C}^{JStorage_K} = \frac{Access\_Need_{C}^{J\_Acc\_stor\_K}}{Stor\_Element\_Power\_k} \qquad (6)$$

The service time is Update request at storage element of site K defined as follow:

$$S_{J,Stor_{Element_K}} = S_{C}^{U\_Stor\_K} = \frac{S_{C}^{U\_Acc\_Stor\_K}}{Storg\_Element\_Power\_k} \qquad (7)$$

- **Number of visit**

The total at each service center is defined as a model input. It counts how many times each customer of a class visit any service center. Each job or Update request is processed at a grid site by visiting the site computing element and storage element one time. When the site process the message, it access the computing element one time, also for the default part and the conditional part access the computing element one time to process the propagation algorithm to send a message through internet. The number of visits can be obtained as follows:

$V_{J,\ com\_element} = V_{j,\ storage\ element} = 1$
$V_{U,\ com\_element} = V_{U,\ storage\ element} = 1$
$V_{P,\ com\_element} = 1$
$V_{P,Default,\ com\_element} = 1$
$V_{P,\ Conditional,\ com\_element} = 1$

- **The Arrival Rates**

The request arrival rate at the Data Grid is modeled by a poison process with parameter λ. As stated when the customer classes are defined as J and U, the requests are allowed in the Data Grid. The requests are with probability P and 1-P for J and U respectively. The J rate in the system is with parameter λ (P) and the U rate is with parameter λ (1 -P). During the execution of request a Storage Broker (SB) receives a sequence of file requests. It is assumed that the jobs are dispatched for execution to different sites on the grid by some scheduling system. For simplicity, the presented model will start when the scheduling decision has already been made and the job is about to start execution on a particular site. The target site of the job is randomly chosen from all the N grid sites. In this system the degree of replication is N=r, where N is the number of grid sites and r is the number of replicas for any of the dataset R. Only one of the sites is the master replica of the data set F_i and the others are the secondary replicas. In Grid system the job operation can be execute at any grid site that contain a replica of the data set while the update operation can only be executed at the site that contains a master replica of that data set.

In a full replication environment the probability that the site k will receive a J request is 1/N, this implies that the job rate for any site is:

$$\lambda J = \lambda P/N \qquad (8)$$

Any update request to the system is processed by all the sites. Any site should process the entire update request, which implies that the U rate of the site k will be obtained as follows:

$$\lambda U = (1- P) \qquad (9)$$

Rate of processing the default part of the propagation algorithm in the UPG and Line is the same as the update processing rate and it can be obtained as:

$$\lambda P\_Default = \lambda (1- P) \qquad (10)$$

And for the radial technique it's equal to 1 at the master site and 0 for all other sites.

In UPG the probability that the site will be a relay site is 1/k for each update request, since having λ (1- P) update requests, and the site going to process the conditional part k times for each relay process the rate of processing the conditional part of the update propagation algorithm is,

$$\lambda P\_Conditional = k \lambda (1-P) 1/k \qquad (11)$$

for the radial technique its equal to λ (1- P) (N-1). And for the UPG and line the arrival rate of the conditional part of the update propagation overhead is:

$$\lambda P\_Conditional = \lambda (1-P) \qquad (12)$$

The computing element at a site k processes the class P_C when it sends and receives a message. The site receives update propagation message with rate λ (1- P), and sends the update propagation message with rate (λ (1-P). In total the site will send and receive an update propagation message with rate:

$$\lambda P\_C = 2 \lambda (1-P) \qquad (13)$$

B. Evaluation

Response time and Utilization are general performance matrices to evaluate the network queuing model. And in order to evaluate the speed of the propagation technique, an update propagation response time is introduce which is the time in which the updates reach all replicas.

- **Resource utilization**

The utilization of each class of request at each service center computes as follows:

$$U_{C,K} = \lambda_{C} \times V_{C,K} \times S_{C,K} \qquad (14)$$

The utilization of the resource can be computed using:

$$U_K = \sum_{i=1}^{C} U_{C,K} \qquad (15)$$

The computing element utilization, storage element utilization at any site of the Grid can be obtained by the equations:

$$U_{Comp\_Element_k} = U_{J,Comp\_Element_k} + \qquad (16)$$

$$U_{U,Comp\_Element_K} + U_{P\_C,Comp\_Element_K} + U_{P\_Default,Comp\_Element_K} + U_{P\_Conditional,Comp\_Element_K}$$

$$U_{Stor\_Element_K} = U_{J,Stor\_Element_K} + U_{U,Stor\_Element_K} \qquad (17)$$

- **Response time**

This section shows how the response time for each class of requests is computed. The response time for a class C at a service center K can be found from the equation:

$$R_C^K = \frac{V_{C,K} \times S_{C,K}}{1 - U_{,Com\_Element_i}(\lambda)} \qquad (18)$$

Response time for job request: The job request needs to be processed at one the site K before returning the result to the user or the grid system. The response time can be obtained by the summation of the time spent in the computing element and the time spent at the storage element.

$$R_J^K = \frac{S_{J,Com\_Element_K}}{1 - U_{Com\_Element\_K}(\lambda)} + \frac{S_{J,Stor\_Element_K}}{1 - U_{Stor\_Element\_K}(\lambda)} \qquad (19)$$

Response time for update request: The update request will process at the originating site (master site) before returning control to the user. Then the response time of the update request will be obtained by summing the total time in the computing element and the time spend at the storage element.

$$R_U^K = \frac{S_{U,Com\_Element_K}}{1 - U_{Com\_Element\_K}(\lambda)} + \frac{S_{U,Stor\_Element_K}}{1 - U_{Stor\_Element\_K}(\lambda)} \qquad (21)$$

Update propagation response time

One of the most important parameter to evaluate the propagation technique is the Update propagation time. Update propagation time is defined as the time at which an update originating at a site j has reached and has been executed at all replicas. To be able to compute the update propagation response time, first compute the time needed for the updates to reach the last site which includes all the time needed for the default part, the time need for the conditional part and the time need for the message processing at all the intermediate sites. Then compute the time need to process the updates locally.

IV. EXPERIMENTAL RESULTS AND DISRUPTIONS

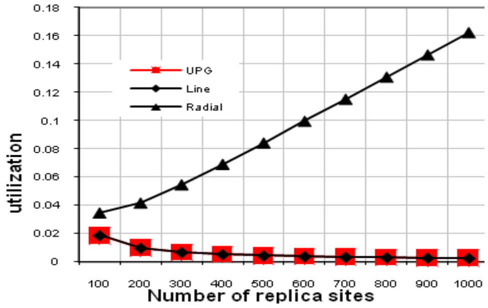
In this section we compare the three techniques in term of the resource utilization, the job response time and the update propagation response time using the propose network queuing model. We consider a system for which the parameters are given in Table 2. For the proposed protocol K is chosen to be equal to 5 and the number of column is set to be equal to 5. The experiment is run many times and varying the number of N, where N= 100 × h (h=1,

2…10). For the proposed protocol K is chosen to be equal to 5.

Figure 4 depicts the master site computing element utilization, Figure 5 master site job response time. The horizontal axis in Figures 4 and 5 indicates the number of sites in the networks. The vertical axis indicates the master site utilization Figure 4, and master site job response time in Figure 5.

Figure 5, shows the computing resource utilization at the master site, the resource utilization can show how much the master site is loaded by process the three techniques, less load at the master site give high efficiency. As we seen in Figure 4 UPG has less load on the master site same as the line technique, while the radial technique have a very bad effect on the master site utilization.

In UPG technique and the line technique, the master site utilization is decreases with the number of replica site while it linearly increase using the radial technique, this result also can show that the propose technique scale well be increasing the number of sites. As mention before increasing the load at the master site badly affect the master site efficiency as shown in Figure 5, the master site job response time is given when using the three techniques. The job response time using UPG and the line techniques is less than using the radial technique.



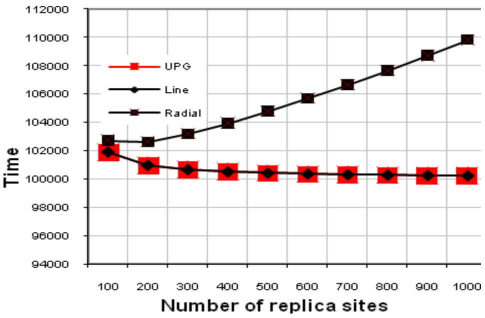


Figure 5: Master Site Job Response Time and Number of Sites.

Table 2: system parameters

Parameter	Description	Value
N	Number of grid site	100..1000
R	Average number of files	10
R	Percentage of data replication	r=N
Bandwidth	Network bandwidth	500Mb/s
λ	Requests arrival rate	1/25000 ms
P	Probability of job request	0.9
Instruction_needed_J	Number of instructions to process the job request	50,000,000
Instruction_needed_U	Number of instructions to process the Update request	100,000
Access_needed_J	Mean storage time of the job request	50,000,000
Access_needed_U	Mean processing time of the Update request	100,000
P_Default	Number of instructions to process the Default part of the propagation algorithm	4000
P_Conditional	Number of instructions to process the Conditional part of the propagation overhead per destination	2000
Size	Message size to be send in the propagation process	5,000
Message_Instruction_required	Number of instructions needed to send, receive a message	20,000
Com_Element_Power	Power of the processing element	1000
Storg_Element_Power	Power of the storage element	1000

Figure 6 depicts the update propagation response times; the horizontal axis in Figures 6 indicates the number of sites in the networks and the vertical axis indicates update propagation response time.

As shown in Figure 6 the UPG can propagate the update information to all sites faster than Radial and Line techniques

In the three techniques update propagation response time increase with the number of replica site, but using the propose technique the increasing fraction is too small, which impels that the propose technique can scale well with high number of replica sites.

Figure 7 depicts the resource utilization of the three techniques under high load where lamda= 0.0004 and P=0.1; the horizontal axis in Figures 7 indicates the number of sites in the networks and the vertical axis indicates the computing element utilization. The curve for the radial technique is not appearing in the figure since the utilization of the commuting element using the radial technique is more than one.

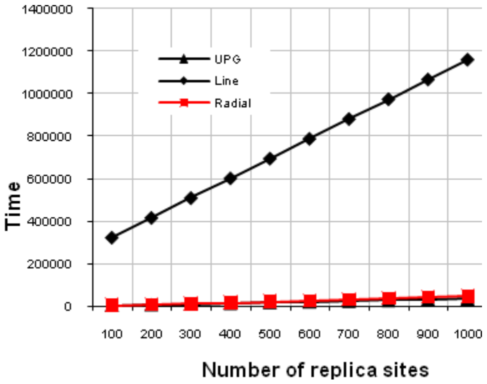


Figure 7: Update Propagation Response Time And Number Of Replica Sites.

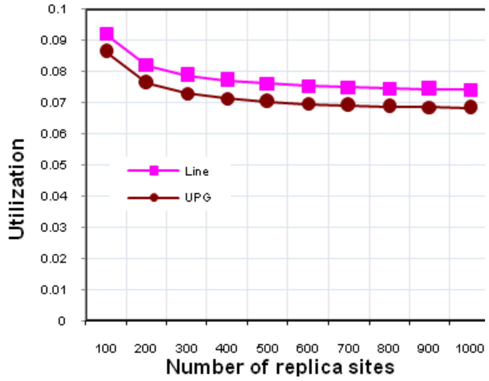


Figure 7: Master Site Utilization And Number Of Replica Sites With High Arrival Rate.

It clearly seen that the utilization of the master site will never reach to 1, and will not be saturated even with high write to read ratios and high number of replica sites. We can conclude that UPG technique scales well with high number of replica sites and high arrival rates and, the system will never be saturated

V. CONCLUSION

This paper introduces a framework for the evaluation of the existing and new update propagation techniques in large scale data grid. The framework takes into account, the computing element, storage element capacity in additional to the arrival rate, and the computational complexity moreover it considers the heterogeneity of resources and the heterogeneity of the jobs. The framework was based in the network queuing model and it's able to compute the resource utilization, the response time. And can evaluate the update propagation technique over the grid environment with different number of grid sites and different user requests arrival rates. We evaluate and compare three update propagation techniques using the proposed framework. In the future work we are going to evaluate the update propagation techniques under different parameter setting.

REFERENCES

[1] Andrea, Domenici, Donno Flavia, Pucciani Gianni, Stockinger Heinz, and Stockinger Kurt. "Replica Cpnsistency in Data Grid." *Nyclear Istruments and Methods in Physics Reserch Sechtion*, 2004: 24-28.

[2] Houda, Lamehamedi, and Szymanski Boleslaw. "Decentralized Data Management Framework for Data Grids." Future Generation Computer System, 2007: 109-115.

[3] Jaechun, No, Park Chang, and Park Sung-Soon. "Data Replication Techniques for Intensive Applications." International Conference on Computational Science. Reading: Springer Berlin / Heidelberg, 2006. 1063-170.

[4] Ruay-Shiung, Chang, and Chang Jih-Sheng. "Adaptable Replica Consistency service for Data Grid." third International Conference on Information Technology: New Generation(ITNG'06). Las Vegas: IEEE Computer Society, 2006. 646-651.

[5] Yuzhong, Sun, and Xu Zhiwei. "Grid Replication Coherent Protocol." 18th International Parallel and Distributed Processing Symposium (IPDPS'04). Beijing: IEEE, 2004. 232-239

[6] Andrea, Domenic, Donno Flavia, Pucciani Gianni, and Stockinger Heinz. "Relaxed Data Consistency with CONStanza." Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06). IEEE Computer Society, 2006. 425-429.

[7] Ghalem, Belalemi, and Slimani Yahya. "Consistency Management for Data Grid in OptorSim Simulator." International Journal of Multimedia and Ubiquitous Engineering, 2007: 103-118.

[8] Mohammed, Radi, Mamat Ali, Deris M. Mat, Ibrahim Hamidah, and Shamala1 Subramaniam. "Update Propagation Technique for Data Grid." ICCSA 2007 . Kuala Lumpur, Malaysia: Springer Berlin / Heidelberg, 2007. 115-127.

[9] Mark, Carman, Zini Floriano, and Serafini Luciano. "Towards an Economy-Based Optimisation of File Access and Replication on a Data Grid." 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02). IEEE Computer Society , 202. 340 -345.

[10] Ruay -Shiung, Chang, Chang Jih-Sheng, and Lin Shin-Yi. "Job and data replication on Data Grids." Future Generation Computer Systems, 2007: 846-860.

[11] Ruay- Shiung, Chang, and Chen Po-Hung. "Complete and fragmented replica selection and retrieval in Data Grids." Future Generation Computer Systems, 2007: 536–546.

[12] Edward, D. Lazowska, Zahorjan John, G. Scott Graham, and C. Sevcik Kenneth. Quantitative System Performance - Computer System Analysis Using Queueing Network Models . Prentice-Hall, Inc, 1984.

[13] Jain, raj. The Art of Computer Systems Performance Analysis. New York: Wiley- Interscience, 1991.

Mohammed Radi received his Bachelor degree of Computer Science from Alazhar University-Gaza, palestine in 2001, the Master degree of Computer Science from University of Jordan- Amman, Jordan in 2003. After he had graduated, he worked as a lecturer at Computer Science department Faculty of Applied Science, Al Aqsa University, Palestine. Currently he is a PhD candidate at faculty of computer science and Information Technology, UPM. His research interests are Data grid, Grid computing, replication and performance modeling.

Ali Mamat is an associate professor in computer science at University Putra Malaysia Serdang. He obtained his Ph.D. in Computer Science from University of Bradford, U.K. in 1992. He has published more than 50 papers in international journals and proceedings. His research interests include databases, XML storage and web semantics.

Mustafa Mat Deris received the B.Sc. from University Putra Malaysia, M.Sc. from University of Bradford, England and Ph.D. from University Putra Malaysia. He is a professor of computer science in the Faculty of Information Technology and Multimedia, UTHM, Malaysia. His research interests include distributed databases, data grid, database performance issues and data mining. He has published more than 80 papers in journals and conference proceedings. Currently he is the Dean of the Faculty of Information Technology and Multimedia, UTHM, Batu Pahat.

Hamidah Ibrahim is currently an associate professor at the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. She obtained her PhD in computer science from the University of Wales Cardiff, UK in 1998. Her current research interests include databases, transaction processing, and knowledge-based systems.

Shamala K. Subramaniam received the B.S. degree in Computer Science from University Putra Malaysia (UPM), in 1996, M.S. (UPM), in 1999, PhD. (UPM) in 2002. Her research interests are Computer Networks, Simulation and Modeling, Scheduling and Real Time System. She already published several journal papers.