A Model for Dynamic Period in Data Grid Replication

Faouzi Ben Charrada, Habib Ounelli, and Hanène Chettaoui

Abstract—**Data grid provides scalable infrastructure for storage resource and data files management, which supports several scientific applications. Replication is a technique used in data grid to increase the files availability, to improve the access time and to reduce the bandwidth consumption. An important problem to be addressed is when replication should be trigged (called period). The period has an important effect on the effectiveness of the replication strategy. In this paper, we propose a model for managing the dynamic period. The proposed model automatically adapts to the grid behavior. We also evaluate the performance of the proposed period using the OptorSim simulator [1]. Our obtained results show that the dynamic period is more effective than the static period.**

Index Terms—**Data grid, dynamic period, optorsim, replication.**

I. Introduction

New scientific applications such as Particle Physics, High Energy Physics and Genetics, to cite a few, manage and generate large data files that can be shared by several researchers around the world. For this reason, we resorted more to data grids. Indeed, the data grids are large scale systems and provide scalable infrastructure for storage resource. Such data files represent a fundamental challenge and must be available to all applications with reasonable access time. Replication is a technique often used to solve this problem. Replication consists in storing several copies of the same file in several grid sites in order to increase the files availability, to improve the access time and to reduce the bandwidth consumption. Several works have been proposed in the literature to resolve the problem of replication in data grids [2]-[4]. An efficient replication strategy must answer three fundamental questions [2], [3]:

1) What are the files to replicate?
2) What are the sites to place the candidate files for replication?

How to select the best replica of a file among many replicas available in the grid? In addition to these three questions, it is important to determine the timing (period) to trigger the replication algorithm [5]?.

Recent works on grid replication considered a static period, fixed by the grid administrator, after which the replication algorithm is trigged. For example, Rahman *et al.* [4], [6] propose a period of 40 jobs submissions. Ranganthan *et al.* [5] propose an adjustable period where the number of existent replicas is compared to the required number of replicas periodically. If, during the last three periods there was no action needed, the period would be increased. On the other hand, if consecutive periods show that more replicas are needed, the period will be decreased. Ranganthan et al. give no indication on the choice and adjustment of this period.

In this paper, we are interested in strategies that use the period concept. We demonstrate that the choice of the period affects greatly the response time and hence balancing the overall load of the grid. In addition to this choice problem, we propose a model for managing a dynamic period and show with several experiments that a dynamic period is far more effective than a static period whatever the adopted replication strategy.

The paper is divided into five sections. Section 2 shows the importance of the period on the effectiveness of the replication strategy. Section 3 presents the proposed model to manage the dynamicity of the period. Section 4 validates our proposed model using OptorSim simulator. It shows the influence of the dynamic period in ENU (The Effective Network Usage) [1], [3] and response time parameters. In our experiments, we consider each time a static period and a dynamic period for different numbers of jobs. Section 5 provides an overview of this work and the future direction.

II. Importance of The Period

This section shows the importance of the period on the effectiveness of the replication strategy using the OptorSim simulator. We begin by presenting the used simulation environment. Next, we present the obtained results.

A. Simulation Environment

OptorSim is a simulator for data grid written in Java [1], [7]. Its objective is to study the effectiveness of replication algorithms in data grid [8]. A data grid, in OptorSim, consists of a set of sites. Each site may contain a Computing Element (CE) and/or Storage Elements (SE). We note that a site without CE and SE acts as a router [9]. The grid also contains a Resource Broker (RB) which is responsible for jobs scheduling.

Fig. 1 summarizes the OptorSim architecture by showing its various components.

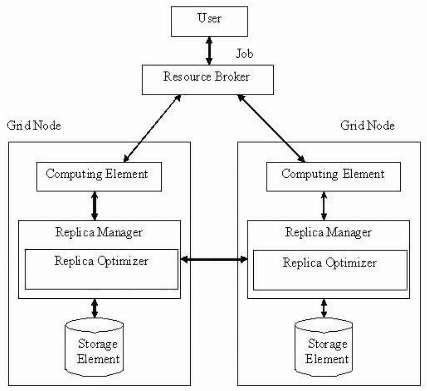

Fig. 1. OptorSim architecture.

Manuscript received October 20, 2012; revised March 2, 2013.

The authors are with Department of Computer Sciences at the Faculty of Sciences of Tunis, Tunisia (e-mail: f.charrada@gnet.tn, habib.ounelli@fst.rnu.tn, hanene1ch@gmail.com).

DOI: 10.7763/IJMO.2013.V3.259

The grid topology (see Fig. 2) used in our experiments comprises 20 sites in Europe and the USA. Every site has a CE and initially empty storage of capacity 50 GB. In Fig. 2, the labels over network links sketch the available bandwidth in Gb/sec. The master files are produced in CERN.

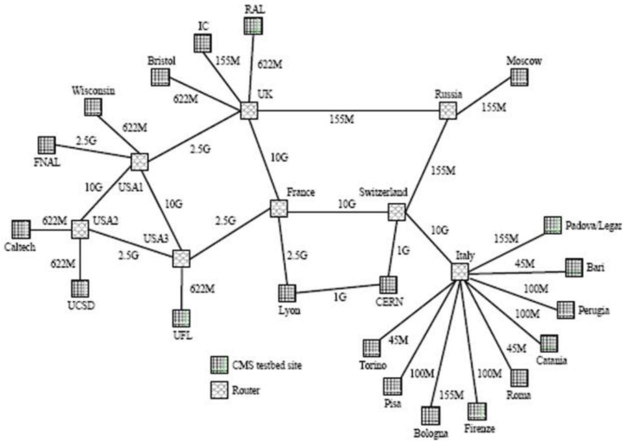

Fig. 2. Grid Topology

Table I show the parameters used in simulation experiment.

TABLE I: GRID AND JOB CONFIGURATION

Parameter	Value
Size of Single File	1GB
Number of Files	97
Access	Pattern Access
Scheduling	Access cost for current job + all queued jobs

B. Experiments and Analysis of Result

To mount the importance of choosing the period, we evaluate the three following replication strategies by varying the period:

- Best Client [2][10]
- DR2 [11]
- Periodic Optimiser [12]

We note that we choose these strategies because they use the period concept. The experiments consist in submitting a variable number of jobs and after each period (a number of submitted jobs), the replication algorithm is triggered. At the end of each simulation, we get the response time and the ENU parameter defined by (1).

$$ENU = \frac{N_{remote\ \ file\ accesses} + N_{file\ replications}}{N_{remote\ \ file\ accesses} + N_{local\ \ file\ accesses}} \quad (1)$$

To achieve our goal, the period variation depends on the number of jobs to be submitted within the simulation. Indeed, the period, noted *T*, varies according to the following formula:

$$T = \frac{Number\ of\ jobs}{100} \times x \quad (2)$$

With *x*=2, 10, 20, 50 and 70.

Formula (2) expresses that the period is equal to a percentage of the number of jobs to submit in the simulation.

Table II shows the response time in ms for each strategy and for different periods. The number of jobs is equal to 5000.

TABLE II: RESPONSE TIME FOR DIFFERENT PERIODS

T	Periodic Optimiser	Best Client	DR2
100 (2%)	48986	59645	52451
500 (10%)	51039	54637	46241
1000 (20%)	53067	133713	47234
2500 (50%)	150348	169267	95332
3500 (70%)	164479	179532	180608

We note, from this table, that the period has an effect on response time. A small period is much better for the response time parameter than a great period. Therefore, the replication strategy is more effective as the period is small.

The results of Table II are represented as curves in Fig. 3 to illustrate better the influence of period on the behaviour of each replication strategy. We note that the response time increases when the period increases. Thus, it is preferable to use a small period with a replication strategy using the period concept.

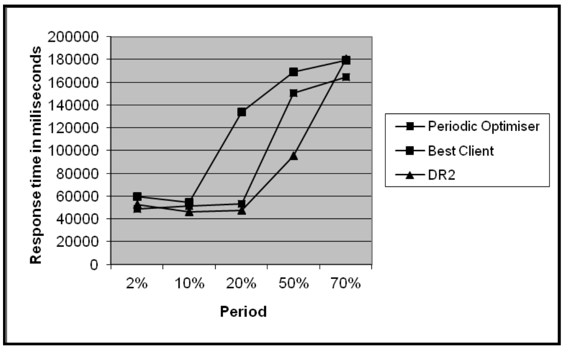

Fig. 3. Response time of ”Best Client”, ”DR2” and ”Periodic Optimiser” strategies

Table III and Fig. 4 sketch the ENU evaluation for the three replication strategies ”Best Client”, ”DR2” and ”Periodic Optimiser” in the same conditions as the response time.

TABLE III: ENU FOR DIFFERENT PERIODS

T	Periodic Optimiser	Best Client	DR2
100 (2%)	0,03	0,05	0,04
500 (10%)	0,14	0,18	0,07
1000 (20%)	0,21	0,6	0,09
2500 (50%)	0,7	0,86	0,34
3500 (70%)	0,84	0,94	0,99

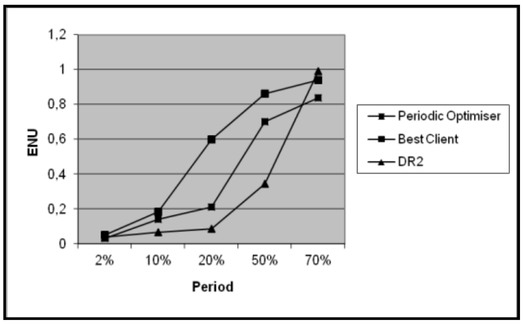

Fig. 4. ENU of ”Best Client”, ”DR2” and ”Periodic Optimiser” strategies

Again, a small period gives ENU significantly better than a great period and this whatever the adopted replication strategy.

We note that we have conducted a series of experiments by varying the number of jobs and for different periods and we have obtained the same gaits curves representing the response time and ENU parameters.

Thus, we show that the period affects the quality results. Indeed, over the period is smaller, more the replication strategy is effective. This may be due to the grid dynamicity. Indeed, when the grid is dynamic in terms of requested files, it is preferable to trigger the replication algorithm frequently. In contrast, it is not desirable to trigger the replication algorithm frequently when the number of requests is low. To achieve this, we propose a model for a dynamic period adapting to the grid behavior. The proposed model is presented in the next section.

III. Model of the Proposed Dynamic Period

This section provides a model for managing the dynamic period in replication. We consider that a period corresponds to a number of submitted jobs in the grid. To take into account the replicas placement, the (n+1)th period (T_{n+1}) is defined by the replications number (#Replica$_{\mathrm{Tn}}$) made during the previous period (T_n).

The period T_{n+1} is given by the following formula:

$$T_{\mathrm{n+1}} = \frac{T_n}{\frac{\#\mathrm{Replica}_{T_n}}{T_n} + 1} \qquad (3)$$

As this formula, if the number of replications made in Tn increases then the new period T_n+1 decreases and vice versa. Thus, the more the grid becomes dynamic the more the period decreases. In this way, the period regularly and automatically adapts to the grid behavior. This model of dynamic period is validated in the next section.

IV. Validation of the Proposed Model

In this section, we evaluate and compare the dynamic period, defined by the formula (3), to a static period in the case of the ”Best Client”, ”DR2” and ”Periodic Optimiser” strategies.

To show the importance of the dynamic period compared to the static period, we measure the ENU and response time parameters for a number of jobs equal to 500 and 1000.

Table IV shows the response time in ms for each strategy and for a number of jobs equal to 500. The last column shows the percentage gain, defined by the formula (4), of the dynamic period compared to the static period.

$$Gain\ in\ \% = \frac{Value_{Static\ \ period} - Value_{Dynamic\ \ period}}{Value_{Static\ \ period}} \times 100 \qquad (4)$$

TABLE IV: Response time for 500 Jobs

Strategies	Static period	Dynamic period	Gain in %
Best Client	7475	4456	40,39%
Periodic Optimiser	8721	4936	43,40%
DR2	9023	8007	11,26%

We note that the response time has improved significantly with the dynamic period. Indeed, when the number of replications is small, it is not preferable to trigger the replication algorithm frequently (long period) and consequently the response time decreases. This confirms the effectiveness of the dynamic period compared to the static period. The results of Table IV are represented as a histogram in Fig. 5 to illustrate better the successful use of a dynamic period instead of a static period.

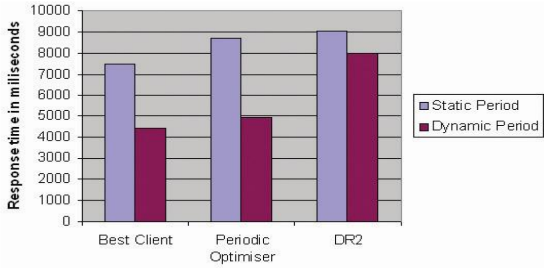

Fig. 5. Response time for 500 jobs

Table V and Fig. 6 show the ENU evaluation of 500 jobs for the three strategies.

TABLE V: ENU for 500 Jobs

Strategies	Static period	Dynamic period	Gain in %
Best Client	0,11	0,08	31,58%
Periodic Optimiser	0,36	0,19	45,72%
DR2	0,48	0,46	3,82%

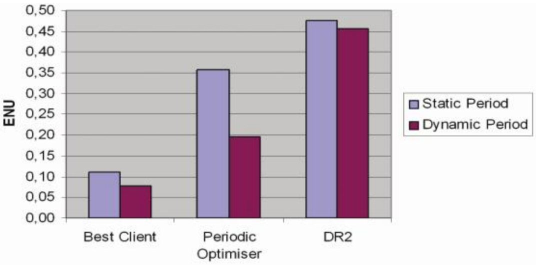

Fig. 6. ENU for 500 jobs

Again, the dynamic period gives an ENU significantly better than the static period and this whatever the adopted replication strategy. This also confirms the effectiveness of the dynamic period compared to the static period.

Table VI and Fig. 7 show the response time in ms for each strategy and for a number of jobs equal to 1000.

TABLE VI: Response Time for 1000 Jobs

Strategies	Static period	Dynamic period	Gain in %
Best Client	15118	10028	33,67%
Periodic Optimiser	10173	6475	36,35%
DR2	14146	11682	17,42%

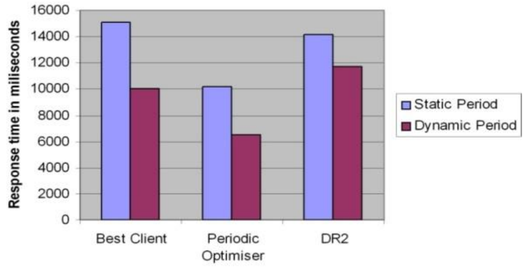

Fig. 7. Response time for 1000 jobs

We note that the response time using a static period is higher than that using a dynamic period. Thus, the dynamic period is more effective than static period regardless of the used replication strategy.

Table VII and Fig. 8 show the ENU evaluation for 1000 jobs and for the three strategies.

TABLE VII: ENU FOR 1000 JOBS

Strategies	Static period	Dynamic period	Gain in %
Best Client	0,38	0,32	15,32%
Periodic Optimiser	0,18	0,16	12,79%
DR2	0,39	0,3	23,05%

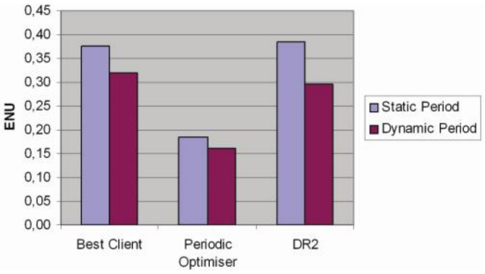

Fig. 8. ENU for 1000 jobs

Again, it is clear that the dynamic period is more effective. The parameter ENU decreases using the dynamic period for the three strategies ”Best Client”, ”Periodic Optimization” and ”DR2”.

This series of experiments shows that it is advantageous to use a dynamic period with any replication strategy. Indeed, the dynamic period automatically adapts to the grid behavior and allows thus to optimize the response time and the ENU.

V. Conclusion and Future Work

In this paper, we proposed a model for managing a dynamic period for the problem of replication in data grids. The proposed model reflects perfectly the behavior of the grid, by taking into account the replicas placement conducted during the previous period. We perform also several experiments through OptorSim simulator. The obtained results show that the proposed dynamic period gives a better performance than a static period and this whatever the used replication strategy. In future work, we plan to generalize the model with the addition of a weighting for the formula (3) so that the period increases if the number of replications goes to zero i.e. the grid reached a steady state. Additionally, we suggest other models to manage the dynamicity of period in data grid replication.

References

[1] D. G. Cameron, R. Carvajal-Schiaffino, J. Ferguson, A. P. Millar, C. Nicholson, K. Stockinger, and F. Zini, *OptorSim v2.1 Installation and User Guide*, 2006.

[2] K. Ranganathan and I. Foster, “Identifying Dynamic Replication Strategies for a High Performance Data Grid,” in *Proc. Springer-Verlag. Workshop on Grid Computing*, 2001, pp. 75-86

[3] H. H. E.AL-Mistarihi and C. H. Yong, “Replica Management in Data Grid,” *International Journal of Computer Science and Network Security*, vol. 8, pp. 22-32, 2008

[4] R. M. Rahman, K. Barker, and R. Alhajj, “Replica Placement Strategies in Data Grid,” *Journal of Grid Computing*, vol. 6, pp. 103-123, 2008

[5] K. Ranganathan, A. Iamnitchi, and I. Foster, “Improving Data Availability through Dynamic Model-Driven Replication in Large Peer-to-Peer Communities,” in *Proc. (IEEE) IEEE/ACM Symp. Cluster Computing and the Grid*, 2002

[6] R. M. Rahman, K. Barker, and R. Alhajj, “Performance evaluation of different replica placement algorithms,” *International Journal of Grid and Utility Computing*, vol. 1, pp. 121-133, 2009

[7] W. H. Bell, D. G. Cameron, L. Capozza, A. P. Millar, 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

[8] W. H. Bell, D. G. Cameron, L. Capozza, A. P. Millar, K. Stockinger, and F. Zini, “Simulation of Dynamic Grid Replication Strategies in OptorSim,” in *Proc. International workshop on grid computing,* 2002.

[9] D. G. Cameron, A. P. Millar, C. Nicholson, R. Carvajal-Schiaffino, F. Zini, and K. Stockinger, “Analysis of Scheduling and Replica Optimisation Strategies for Data Grids using OptorSim,” *Journal of Grid Computing*, vol. 2, pp. 57-69, 2004

[10] K. Ranganathan and I. Foster, “Design and Evaluation of Dynamic Replication Strategies for a High-Performance Data Grid,” in *Proc. International Conference on Computing in High Energy and Nuclear Physics*, 2001.

[11] P. K. Suri and M. Singh, “DR2: A Two-Stage Dynamic Replication Strategy for Data Grid,” *International Journal of Recent Trends in Engineering*, vol. 2, pp. 201-203, 2009

[12] F. B. Charrada, H. Ounelli, and H. Chettaoui, “An Efficient Replication Strategy for Dynamic Data Grids,” in *Proc. Fifth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2010, pp. 50-54

Faouzi Ben Charrada is working as an Assistant Professor in the Department of Computer Sciences at the Faculty of Sciences of Tunis, Tunisia. His current research interests focus on grid, data replication and scheduling.

Habib Ounelli is a professor in the Department of Computer Sciences at the Faculty of Sciences of Tunis, Tunisia. His current research interests focus on fuzzy logic, flexible querying, databases and their applications, FCA, cooperative systems, grid, data replication and scheduling.

Hanène Chettaoui is working as an assistant in the Department of Computer Sciences at the Higher Institute of Multimedia Arts, Tunisia. She is a PhD candidate in computer science at the Faculty of Sciences of Tunis. She received her MS degree from Faculty of Sciences of Tunis, Tunisia, in 2008. Her current researcher interests include data grid, data replication, computing grid and scheduling.