# Negotiation Mechanisms for Resolution Conflicts among Replicas in Data Grid

Ghalem Belalem
*Dept. of Computer Science, Faculty of Sciences,*
*University of Oran - Es Senia, Oran, Algeria*
*Ghalem1dz@gmail.com*
Belabbes Yagoubi
*Dept. of Computer Science, Faculty of Sciences,*
*University of Oran - Es Senia, Oran, Algeria*
*byagoubi@yahoo.fr*

**Abstract**

*Data Grid environment seek to harness geographically distributed resources that deal with data-intensive problems such as those encountered in high energy physics, bio-informatics, and other disciplines. In general, grids enable the efficient sharing and management of computing resources for the purpose of performing large complex tasks. To be able to sharing data, it is recommended to use the replication technique. This technique provides an improvement in performance, fault tolerance and load balancing. The replication management and its implementation are not simple tasks and produce other problems, like consistency management of replicas. One of the concerns major in the consistency management approaches called optimistic, it is the conflicts resolution among replicas. In this paper we present negotiation mechanisms based on the various negotiation forms between virtual consistency agents to be able to critical situation for sites and to reduce the number of conflicts among replicas to converge them more quickly in data grids.*

## 1. Introduction

Data replication is a fundamental mechanism for most distributed systems for a number of highly desirable properties. It enhances performance and scalability by enabling local data replicas to be accessed rather than a centralized physical data source, potentially located on a remote server. Replication improves fault tolerance if replicas are ensured to be kept consistent: should one replica be lost, the contents of any other consistent replica can be used to recover the lost data. Finally, it increases availability as applications may continue to access their local replicas even in case of failure or network inaccessibility of other replicas. In spite of these advantages, there are many problems which should be solved, like : placement of replicas [4, 8, 15], degree of replication [11], the selection of a replica for a request [9] and consistency management of replicas [3,7]. The main objective of a replica consistency approach is to avoid or even reduce the inconsistency between replicated data. Many current applications can barely tolerate a certain degree of contradiction between replicas where the strong consistency is not a condition [12]. Our principal aim, in this paper, is to propose a hybrid mechanism of negotiation for the decision-making to conflicts between the replicas. This mechanism of negotiation is integrated in the hybrid consistency approach [3] inspired from the two pessimistic and optimistic traditional approaches. The structure of our present paper will as follows: the next section will dedicate to the description of the model used in our adapted negotiation mechanisms. In section 3, we describe our negotiation mechanisms for the decision-making to the meeting of the divergences between replicas which cannot be solved while basing itself on a whole of metric interesting for the decision-making in the process of consistency management, then we present the algorithms of our negotiation process. Section 4 is reserved for some experiences to show the benefits of the proposed approach, finally section 5 will enclose this work by the presentation of some future tracks.

## 2. Bi-Levels Model for Consistency Approach

For the consistency management of replicas in large scales systems, we proposed a process of consistency management which profit as well as possible from between the traditional approaches pessimistic and optimistic [2]. This process uses a model of two levels: level 0 is physical and comprising the localization of replicas, for level 1 is logical one and represents the various agents, where each agent is responsible for part of level 0.

In our work, we consider a grid as a collection of distributed collections of Computing Elements (CE's) and Storage Elements (SE's). These elements are linked together through a network to form a Site or a Cluster. Sites are in turn linked together to form a grid. Replicas are stored on Storage Elements and are accessible from Computing Elements. Our model presented in Figure 1 is described as follows:

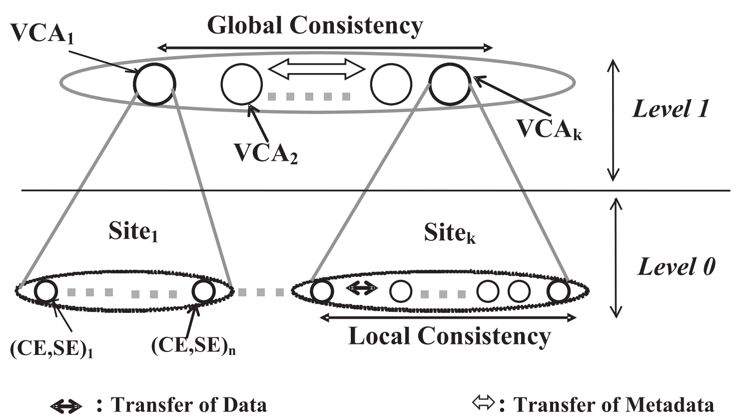



**Figure 1: Bi-Levels Model for Consistency Management with Virtual Consistency Agent**

- Level 0: in this level we find sites that compose a grid. Each site contains a set of Computing Elements (CE's) and Storage Elements (SE's). Replicated data are stored on SE's and accessed from CE's via reading or writing operations;
- Level 1: in this level we define *k* Virtual Consistency Agents (VCA's) each one corresponding to each site of a grid. A virtual consistency agent $VCA_i$ is responsible to manage replica consistency within a site $S_i$, that we call intra-site consistency. Then, each $VCA_i$ cooperates with others $VCA_j$ to ensure a replica consistency for the whole grid. It detects a conflict situation (significant conflicts's factor of replicas), the process of cooperation with a basis for negotiation has begun between agents candidates with the conflicts resolution met.

## 3. Consistency Management and Negotiation Process

We will describe, in what follows, the core of the consistency process based on the negotiation between *VCA* (see Figure 2).

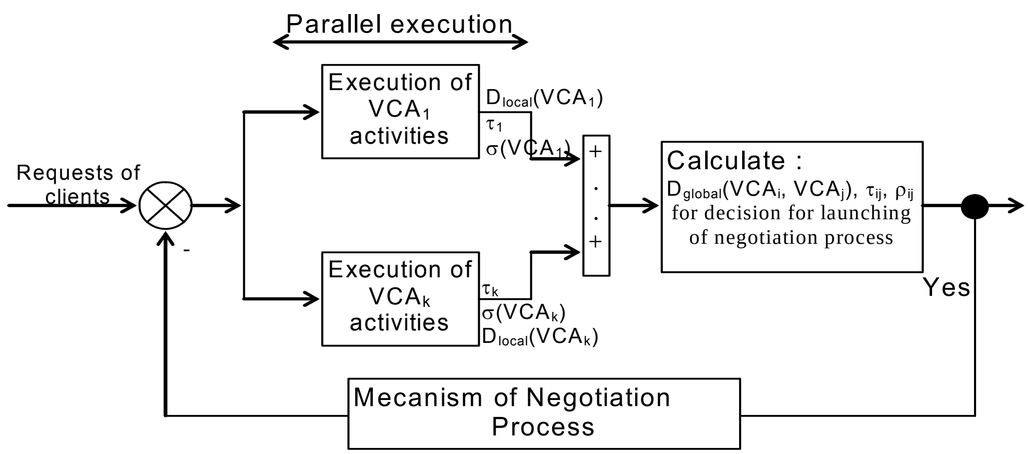



**Figure 2: General process of our approach for reduction of divergent replicas in Data Grid**

We will start by giving definitions of some basic elements, then we will present the two levels of service of consistency management and finally the mechanism of negotiation used for the conflicts resolution between the replicas.

### 3.1. Process Consistency Management Elements

The main concepts used in our process for consistency management of replicas are:

a. Agent: An agent is a program that operates autonomously and accomplishes unique tasks without direct human supervision. It cooperates or competes with others agents to perform some set of tasks or satisfy some set of goals [1].
b. Replication Strategy: Among the strategies most commonly used, we can cite: Single-Master or Multi-Masters [5,13];
c. Divergence: two replicas ($r_i$, $r_j$), of the same data, are known as divergent, if $Metadata(r_i) \neq Metadata(r_j)$ and $version(r_i) \neq version(r_j)$, we speak about weak divergence.
d. Conflict: two replicas ($r_i$, $r_j$), of the same data, are known as in conflict, if $Metadata(r_i) \neq Metadata(r_j)$ and $version(r_i) = version\ (r_j)$, we speak about strong divergence. We associate this definition of conflict, *Conflicts_Nbr* metric to indicate the number of conflicts of *VCA*.

### 3.2. Intra-Agent Level

The principal aim of the *VCA* is the control of process of local consistency. Its fundamental mission is to make converge the replicas towards the same local reference within the site. This process can be announced according the following phases:

- Phase of reception and treatment: according to the replication strategy of site, the *VCA* directs the customer's request towards the node of the free master so that it is treated, if not, it deposits it in the queue of its site.
- Phase of control of degree of divergence tolerated of *VCA*: The objective of this phase consists in following the evolution of degree of divergence of replicas within a site by the *VCA*. To study the evolution of the divergence of replicas inside a site, we put forward the three following measures:

  1. Measure rate of the number of conflicts per site ($\tau_i$): this measurement makes it possible using *VCA* to know the rate of conflicts by the total number of replicas of the same data inside a site;
  2. Measure distance within a site ($D_{local}$): we define $D_{local}$ measurement between the versions maximum and minimal of replicas of the same object inside site of *VCA*;
  3. Measure dispersion of versions ($\sigma_i$): this measurement makes it possible to inform us on the manner of dispersion of the versions of replicas around an average (see formula 1) of the same data inside a site (see formula 2), where $n_i$ is size of vector versions ($V_{it}$) of data $t$ in site $i$.

$$\overline{V}_i = \frac{1}{n_i}\sum_{t=1}^{n_i} V_{it} \qquad (1)$$

$$\sigma(VCA_i) = \sqrt{1/n_i \sum_{t=1}^{n_i} (V_{it} - \overline{V}_i)^2} \qquad (2)$$

We detect critical situations of one *VCA* to the meeting of one of the following cases:

a. $\tau_i$ > Rate of conflicts number tolerated;

b. $D_{local}(VCA_i)$ > Distance tolerated;

c. $\sigma(VCA_i) > \sigma_m$; where $\sigma_m$: Tolerance rate for dispersion of versions.

The algorithm to check the existence or not of critical situation of $VCA_i$, is given by:

**Algorithm1** *Local Critical_Situation*

```
Require: VCAi;  τm, Dm–L ; σm
{σm : threshold dispersion tolerated
  τm: threshold of conflicts number tolerated
  Dm–L : threshold distance local tolerated }
 Calculate : τi, σi; Dlocal  for VCAi
 If(τi>τm) or (σi> σm) or (Dlocal> Dm–L) Then
   return (True)    // existence of critical situation of VCAi
Else
return (False)
EndIf
```

If *VCA* is in critical situation, then it starts the local process of negotiation (Algorithm Intra-Agent Level).

**Algorithm2** *Intra-Agent Level*

```
{  Is a request arriving at the site controlled by VCAI  }
If Strategy_Replication = Single_Master then
If Master = free Then
                      treatment of the request of the customer
             Else
                  to deposit the request in the queue of the site
EndIf
  Else
  If Strategy_Replication = Multi Master Then
      If ∃Master = free Then
                           treatment of the request of the customer
             Else
             To deposit the request in the queue of the site
       EndIf
    EndIf
EndIf
If Local Critical_situation(VCAi)  Then
                     Process of local negotiation
                                Else
                                  Propagate of updates.
EndIf
```

### 3.3. Inter-Agent Level

In Inter-agent level two situations can be treated. The first situation corresponds to the competitive negotiation and the second one represents cooperation negotiation.

1. The process of the competitive negotiation is started following the meeting of critical situation between agents. From a total point of view to follow the evolution of divergences of replicas between various *VCA*, we put forward the following measures:
   - Measure rate of the number of conflicts ($\tau_{ij}$): this measurement makes it possible to estimate the rate of conflicts between agents. For example, the rate of conflicts between $VCA_i$ and $VCA_j$ is given by formula (3).

$$\tau_{ij} = \frac{\tau_i + \tau_j}{n_i + n_j} \qquad (3)$$

- Measure global distance inter-agents ($D_{global}$): to study the distance between two *VCA*, we use the Euclidean distance. This measurement makes it possible to propose an estimate to us on the distance of the two *VCA* compared to the versions of the replicas of the same object. It is given by:

$$D_{global} = \sqrt{\sum_{t=1}^{n_i} \left| V_{it} - V_{jt} \right|^2} \qquad (4)$$

- Measure coefficient of correlation ($\rho_{ij}$): this measurement describes the degree of correlation between two *VCA* and it is represented according to covariance $Cov(VCA_i, VCA_j)$.
  The coefficient of correlation makes it possible to form groups of the *VCA* where $VCA_i$ inside the group are very attached the ones with others. The coefficient of correlation between $VCA_i$ and $VCA_j$ is presented under the formula (5):

$$\rho_{ij} = \frac{Cov(VCA_i, VCA_j)}{\sigma(VCA_i)\sigma(VCA_j)} \qquad (5)$$

We associate $\delta_{ij}$ a degree of confidence at coefficient of correlation $\rho_{ij}$. $\delta_{ij}$ is an actual value to check the assumption of correlation $\rho_{ij}$ between two *VCA*. A value $\delta_{ij}$ near to zero means that the assumption is false and that there is thus no correlation between $VCA_i$ and $VCA_j$. A value $\delta_{ij}$ distant from zero (positive or negative) means that the assumption is checked and that there is thus correlation between $VCA_i$ and $VCA_j$ (see formula 6).

$$\delta_{ij} = \rho_{ij} \sqrt{(n_i - 2)(1 - \rho_{ij}^2)} \qquad (6)$$

We define a critical situation between $VCA_i$ and $VCA_j$ if one of the following conditions is checked:

a. $\tau_{ij}$ > Rate of conflicts number tolerated;
b. $D_{global}(VCA_i, VCA_j) > D^{m\_G}$ tolerated;
c. $| \rho_{ij} | < \varepsilon$ ; where $0 < \varepsilon << 1$

The main steps of algorithm, to check the existence or not of critical situation between $VCA_i$ and $VCA_j$, are described by the algorithm 3.

```
Algorithm3  Global Critical_Situation
Require:  VCA_i, VCA_j; D^{m_G} ; τ^m
{ τ^m : threshold of conflicts number tolerated
D^{m_G} : Distance tolerated
ρ^m : threshold of correlation tolerated }
Calculate : τ_ij, ρ_ij; D_global  for VCA_i, VCA_j
 If(τ_ij>τ^m) or (ρ_ij>ρ^m) or (D_global(VCA_i, VCA_j)> D^{m_G})  Then
      return (True)
    {existence of critical situation between VCA_i and VCA_j}
    Else
        return (False)
EndIf
```

If the event of the critical situation between *VCA* is detected, we carry out then the construction *VCA* groups in conflict and start of the process of negotiation intermediate by the following algorithm (Algorithm 4):

```
Algorithm4  Competitive Group
Indice ← {1,…,k}; Gr_comp ← ∅; r ← 1;
{ k : total number of VCA of the system
Gr_comp : group of agents in competition }
   While (Indice≠ ∅) and (r < k) Do
        Gr_inter ← Indice - {r}
        Gr_comp ← {VCA_r};
        While (Gr_inter ≠ ∅) Do
        Choose  i  of Gr_inter set such as  the value i is smallest possible
           Gr_inter ← Gr_inter – {i};
           If Global critical situation (VCA_r, VCA_i) Then
                      Gr_comp ← Gr_comp ∪ {VCA_j}
                      Indice ← Indice - {i}
          EndIf
        EndWhile;
        If || Gr_comp) > 1 || Then
             Negotiation process of intermediate situation for Gr_comp ;
        EndIf
   Repeat
    r ← r +1
   Until (r ∈ Indice) or (r > k);
EndWhile
```

2. In the process of cooperative negotiation, the *VCA* try to reach the total common utility of the system, which corresponds to consistency global by cooperation mechanism between *VCA*. This stage of inter-agent level is described by negotiation process of global situation.

### 3.4. Negotiation Process

As announced before, the mechanism of negotiation combines several cooperation forms for conflicts resolution according to the various situations met.

**3.4.1. Negotiation of Local Situation:** In local situation, the *VCA* of the site in critical situation, acts like an initiator of negotiation, it announced its conflict situation (crisis plan) for the group of *VCA* on network by diffusion. The agents receive and evaluate this situation. The *VCA* which have the capacity to solve this crisis plan send to the initiator bids by indicating their capacities to carry out this announced crisis and whole information associated of correlation degree to this situation. At this time the initiator has a role of arbitration, it evaluates the bids and grants its crisis plan to the *VCA* the best suitable one also called the contractor. Lastly, the initiator and the contractor exchange information necessary during achieving of this critical situation.

**3.4.2. Negotiation of Intermediate Situation:** In this stage of process, the *VCA* in conflicts situation gather to find agreement consensus to their critical situation, by the following algorithm.

***Algorithm5*** *Intermediate Situation*

```
For all elements of the Group do
research the most stable VCA
EndFor
VCAs ← most stable VCA of Group
σ(VCAs) ← Min {σ(VCAi)}    ∀ i ∈ Group
Propagate informations of VCAs toward the other VCA of Group
EndFor
```

**3.4.3. Negotiation Process of Global Situation:** In this situation, global consistency is put in priority, the whole of *VCA* negotiate their information to arrive at consensus of agreement. The negotiation process used aims at converging the various replicas, of the same data in the complete system, towards a global reference replica according to one of the two following alternatives:

A. One of the aims of *QoS* (Quality of Service) in the consistency management is obtained the most recent information. Generally this information is associated at replicas versions. The principle of negotiation of downward bidding is used to arrive at consensus of agreement (see Dutch_auction Algorithm).

***Algorithm6*** *Dutch_auction*

```
AGT_Max ← Choose  VCA with largest vesrion
{ k : total number of VCA }
i ← 1
Repeat
If Majority_vote(AGT_Max) Then
        AGT_Max propagates data and metadata to other VCA
        {The bid is accepted by majority of  VCA }
        Exist()
 EndIf
i ← i+1 ;
Choose VCAr having the ith greatest version
 {It proposes its offer to the other VCA of Grid }
AGT_Max ( VCAr
Until i>=k  { No VCA could be accepted by majority }
English_auction. (If Algorithm Dutch_auction leads with failure)
```

B. The second alternative is based on the principle of stability of the *VCA*. We use in this alternative the principle of the English auction (see English_auction Algorithm).

***Algorithm7*** *English_auction*

AGT ← Choose VCA with largest vesrion
{ Degree of stability from the maximum version;
k : total number of VCA }
**For** i = 1 **to** k **do**
**If** ($\sigma_i < \sigma_{AGT}$ ) **and** ($\tau_i < \tau_{AGT}$) **Then**
    AGT ← $VCA_i$
    {Favor the stability of VCA }
**EndIf**
**EndFor**
AGT propages metadata and data to other VCA

The algorithm (see Algorithm8) to study the degree of the correlation between $VCA_i$ and $VCA_j$ is given by:

***Algorithm8*** *CorrelationDegree*

**Require:** $VCA_i$, $VCA_j$
Calculate : $\rho_{ij}$, $\delta_{ij}$
**If** ($||\rho_{ij}|-1|<\varepsilon$) **or** ($\delta_{ij} > \varepsilon$) **Then** {Where $0< \varepsilon << 1$}
    **return** (False)
**Else**
  **return** (True)
**EndIf**

## 4. Experimental Study

In order to validate and to evaluate our approach of consistency management of replicas compared to the traditional approaches (pessimistic and optimistic), we carried out series of experiments whose results and interpretations are covered in this section. In order to analyze the results relating to the experimentation of our approach, we used four metrics to know the response time, the divergences number and conflicts number between replicas. To carry out the various experiments of our approach, we fixed certain number of parameters of simulation whose values are defined in Table 1. These parameters are common to the whole of simulations which carried out the simulator OptorSim [2,14].

**Table 1. Simulation parameters**

| Simulation parameters | Interval |
|---|---|
| Number of Sites | [5..50] |
| Number of Nodes | [10..500] per site |
| Bandwidth inter-sites | [100..1000] Mb/s |
| Size of data (file) | [100..10000] MB |
| Number of Requests | [20..500] |
| Number of Replicas | [1..100] per data |

To validate our approach compared to pessimistic and optimistic approaches, and in order to analyze and interpret the results obtained of experiments by various approaches, we used three measurements: response time, numbers divergences and conflicts among replicas. For the first experiments, we chose the following parameters: 2 sites, 50 nodes, 10 files, 10 replicas per file, and we varied the number of requests.\\

a. ***Effects on response time :*** Figure 3 shows the curves of the variation in response time of our approach and the pessimistic approach. We can note that from 60 requests, our approach gives better results than the pessimistic and that difference is very significant from 80 requests.

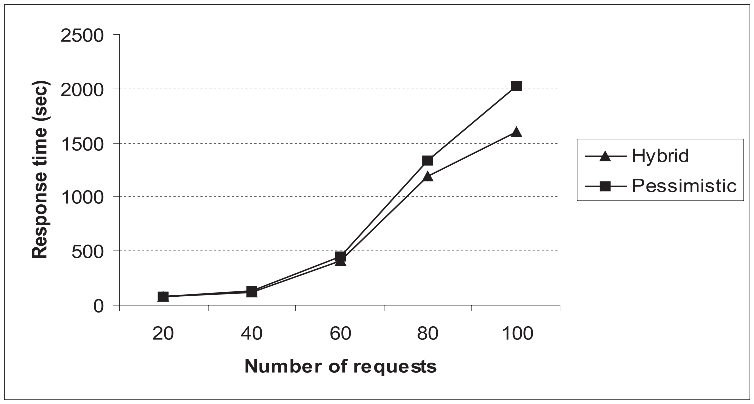



**Figure 3: Average response time / Number of requests**

b. ***Effects on number of divergences :*** Figure 4 illustrates the evolution of number of divergences according to the number of requests for optimistic and hybrid approaches. We notice here a very clear reduction in number of divergences for hybrid approach compared to the optimistic approach.

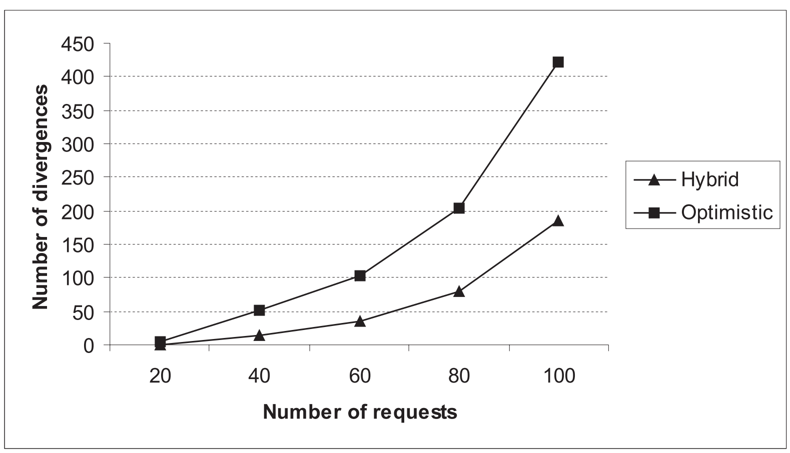



**Figure 4: Number of divergences / Number of requests**

c. ***Effects on number of conflicts*** : In figure 5, we were interested in evolution of number of conflicts compared to number of requests. The analysis of the curves shows a significant reduction in number of conflicts with the hybrid approach.

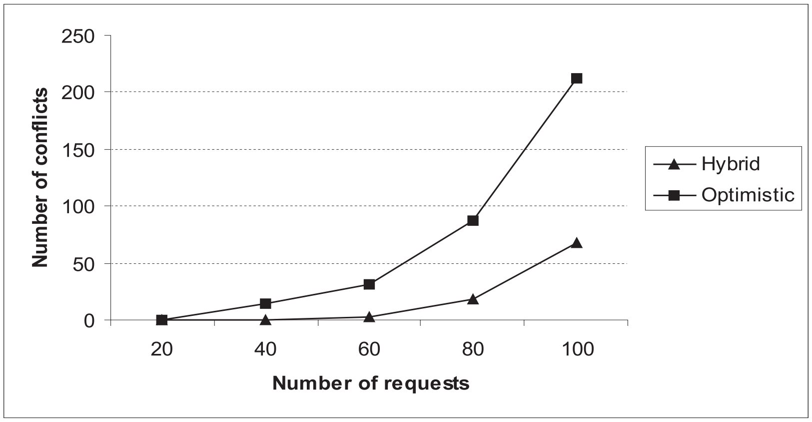



**Figure 5: Number of conflicts / Number of requests**

## 5. Conclusion and future works

The main problem introduced by replication techniques is maintaining consistent replicas. In Data Grid environment, strong consistency is not adapted due to their prohibitive cost. Weak consistency approaches can be used in these systems by tolerating divergences among replicas for at least some time period. In these divergence situations, reconciliation poses many problems and in particular in mechanisms of conflicts resolution among replicas. In this paper, we presented negotiation mechanisms for resolution conflicts among replicas in data grids. There are number of directions which we think are interesting and are worth further investigation. We can mention:

- Development of Web service for consistency management of replicas: We propose to integrate our approach in the form of Web service in the Globus environment by using technology WSDL [6];
- In the current version of our approach, we placed replicas randomly. It is worthwhile to explore the possibility of making a static or dynamic placement to improve QoS in the data grid [8];
- Load balancing: From this point of view, and for improving even more performance and the quality of service of our approach, we propose to extend it by a service of load balancing [10,16], which allows to balance the requests on the various sites of Data Grid;
- To take into account the factor time during the phase of negotiation, i.e., to define intervals of time during which we must make to us decisions;
- Propose incorporation of weight in the suggested plans by *VCA* candidates with the resolution of critical situations, in function of their stability viewpoint of consistency degree and the size of membership group.

## 6. References

[1] S. Alda, A. B. Cramers, J. Bilek, and D. Hartmann. Support of collaborative structural design processes through the integration of peer-to-peer and multiagent architectures. In *Proceedings of the 10th International Conference on Computing in Civil and Building Engineering (ICCCBE-X)*,Weimar, Germany, 02-04 June 2004.

[2] G. Belalem and Y. Slimani. Consistency management for data grid in optorsim simulator. *In International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, pages 554-560, Korean Bible University, Seoul, Korea, 26-28 April 2007.

[3] G. Belalem and Y. Slimani. A hybrid approach to replica management in data grids. *International Journal of Web and Grid Services (IJGWS)*, 3(1), pp:2-18, 2007.

[4] G. Belalem and F. Bouhraoua. Dynamic strategy of placement of the replicas in data grid. In Victor Malyshkin, editor, *Parallel Computing Technologies: 9th International Conference, Pact2007*, volume 4671 of LNCS, pages 496-506, Pereslavl-Zalessky, Russia, September 2007. Springer.

[5] R-S. Chang and J-S. Chang. Adaptable replica consistency service for data grids. *In Third International Conference on Information Technology: New Generations (ITNG'06)*, pages 646-651, Las Vegas, USA, 2006.

[6] I. Foster and C. Kesselmann. The Grid 2: Blueprint for a new computing infrastructure. Morgan Kaufmann Publishers, 2004.

[7] J. Gray, P. Helland, P.E. O Neil, and D. Shasha. The dangers of replication and a solution. *In ACM SIGMOD International Conference on Management of Data*, pages 173-182, Montreal, Quebec, Canada, 4-5 June 1996.

[8] C. Haddad and Y. Slimani. Economic model for replicated database placement in grid. In *Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGRID'07)*, pages 283-292, May 2007.

[9] B. Wang L. Xu and B. Ai. A strategy for data replication in data grids. In Roland GlowinskiWu Zhang, Zhangxin Chen andWeiqin Tong, editors, *Current Trends in High Performance Computing and Its Applications: Proceedings of the International Conference on High Performance Computing and Applications*, LNCS, pages 557-562, Shanghai, P.R. China, 8-10 August 2004. Springer.

[10] Y. Li and Z. Lan. A survey of load balancing in grid computing. In Ji-Huan He Jun Zhang and Yuxi Fu, editors, *Computational and Information Science, First International Symposium (CIS'04)*, volume 3314 of *Lecture Notes in Computer Science (LNCS)*, pages 280-285, Shanghai, China, December 16-18, 2004. Springer.

[11] Y. Mansouri and R. Monsefi. Optimal number of replicas with qos assurance in data grid environment. In *Second Asia International Conference on Modelling Simulation (AMS'08)*, pages 168-173, Kuala Lumpur, Malaysia, 13-15 May 2008.

[12] C. Olston and J. Widom. Ef_cient monitoring and quering of distributed, dynamic data via approximate replication. *IEEE Data Eng. Bull.*, 28(1), pp:11-18, 2005.

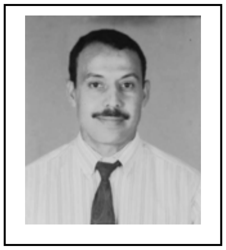
[13] Y. Saito and M. Shapiro. Optimistic replication. *ACM Comput. Surv.*, 37(1), pp: 42.81, 2005.

[14] A. P. Millar L. Capozza K. Stockinger W. H. Bell, D. G. Cameron and F. Zini. Optorsim - a grid simulator for studying dynamic data replication strategies. *High Performance Computing Applications*, 17(4), pp:52-59, 2003.

[15] J. Xu, B. Li, and D. J. Lee. Placement problems for transparent data replication proxy services. *IEEE Journal on Selected Areas in Communications*, 7(20), pp:1383-1398, 2002.

[16] B. Yagoubi and Y. Slimani. Task load balancing strategy for grid computing. *Journal of Computer Science*, 3(3), pp:186.194, 2007.

# Authors



**Ghalem Belalem**

Graduated from university of Oran, Algeria, where he received PhD degree in computer science in 2007. He is now a research fellow of management of replicas in data grid. His current research interests are: distributed systems, grid computing and data grid, placement of replicas and consistency in large scale systems.

**Belabbes Yagoubi**

Graduated from university of Oran, Algeria, where he received PhD degree in computer science in 2007. He is currently leading a research team on the management of resources in grids. His main research interests are in the grid computing, distributed systems, load balancing and resource management challenges in Grids.