International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:4, 2008

A Consistency Protocol Multi-Layer for Replicas Management in Large Scale Systems

Ghalem Belalem and Yahya Slimani

***Abstract*—** Large scale systems such as computational Grid is a distributed computing infrastructure that can provide globally available network resources. The evolution of information processing systems in Data Grid is characterized by a strong decentralization of data in several fields whose objective is to ensure the availability and the reliability of the data in the reason to provide a fault tolerance and scalability, which cannot be possible only with the use of the techniques of replication. Unfortunately the use of these techniques has a height cost, because it is necessary to maintain consistency between the distributed data. Nevertheless, to agree to live with certain imperfections can improve the performance of the system by improving competition. In this paper, we propose a multi-layer protocol combining the pessimistic and optimistic approaches conceived for the data consistency maintenance in large scale systems. Our approach is based on a hierarchical representation model with tree layers, whose objective is with double vocation, because it initially makes it possible to reduce response times compared to completely pessimistic approach and it the second time to improve the quality of service compared to an optimistic approach.

***Keywords*—** Data Grid, replication, consistency, optimistic approach, pessimistic approach.

I. Introduction

UNLIKE distributed systems, Grid architectures introduce new challenges, such as high latencies and dynamic resource availability. The Data Grid [1], for example, connects a collection of thousands of geographically distributed computers and storage resources located in different parts of the world to share data and resources between users. Sharing data can be obtained by using replication technique to improve the access of these data. The replication has been widely used in traditional distributed systems for providing high availability, fault tolerance and good performance. But its implementation is very difficult [2], [3], like placement of replicas, degree of replicas, choosing replicas. Among the principal problems in the use of the techniques of replication, is that of the maintenance consistency between the various replicas distributed between several computers. The principal objective of consistency approach is to avoid or to reduce contradictions between replicas. In this article, we propose a three layers model in order to maintain consistency in large scale environments. The structure of our paper is presented as follows: section 2 will present the concept of consistency by detailing two approaches of pessimistic and optimistic consistency. Section 3 will describe our proposed multi-layer protocol for consistency management in large scale systems and the algorithms associated to our protocol. Section 4 will present the choice of metrics used and the preliminary results of simulation of our consistency protocol multi-layer, finally some directions for future work are proposed in section 5.

II. Consistency Management

The Consistency is a relation which defines the degree of similarity between copies of a distributed entity. In the ideal case, this relation characterizes copies which have identical behaviors. In the real cases, where the copies evolve in a different way, consistency defines the limits of divergence authorized between these copies.

Consistency is ensured by synchronization between the copies (replicas). To reach the copies, a protocol of management of coherence is necessary, which ensures the mutual consistency between the copies according to a behavior defined by a consistency model.

The consistency protocol gives an ideal view as if there is only one user and only one copy of the data in the system. The pessimistic approach and the optimistic approach are two strategies of maintenance of consistency. They represent the two edges of the dilemma coherence availability [4]. The pessimistic approach is interested in consistency more than availability, while the optimistic approach supports the availability more than the consistency.

A. Pessimistic Approach

It is a traditional strategy of management of consistency [5]. In this strategy, the users do not observe any contradiction between the copies of the same shared data. In terms of consistency, it appears for the users like if there is only one copy [6]. Conceptually, an update in a copy is propagated to all the other copies in a synchronous way, and no copy is accessible before it will be up to date (for this reason it is called pessimistic). When nodes or networks break down, the access to the data is refused to prevent the users from taking contradictory data [6]. For example, in the case of partition of the network, this means that the access to the data can be refused until the handing-over of the partition.

1) Advantages: Among the advantages of pessimistic consistency, we can cite:
   - Divergences between copies are not allowed and the consistency is strong.
   - Operations are carried out in a definitive way and provide reliable results.
2) Disadvantages: The pessimistic approach has many disadvantages. Most significant are:

G. Belalem is with the Dept. of Computer Science, Faculty of Sciences, University of Oran - Es Senia, Oran, Algeria; e-mail: ghalem1dz@gmail.com

- The Need for a process of synchronization between copies is too expensive for the environments on a large scale and not realizable in the environments with partitions.
- Response time is very height.
- Scalability is limited, the degree of availability decreases as the number of replicas of the system increases [7], [8].

B. Optimistic Approach

The optimistic strategy allows users to reach any copy for the reading or the writing operations, even when there are breakdowns of network or when some copies are unavailable.

1) Advantages: Optimistic strategy present several advantages compared to pessimistic coherence [6], among them, we have:
   - Availability: accesses to the data are never blocked.
   - Flexibility of networks's management: the networks do not need to be entirely connected so that they will be entirely accessible.
   - Scalability: a great number of elements can be supported by the grid because the synchronous communication is not necessary to accept updates. It is applied in the environments on a large scale like Globus [9] or Legion [10].
2) Disadvantages: In spite of these advantages, optimistic consistency suffers from:
   - The states of copies can be temporarily mutually contradictory.
   - An update can be applied to one copy without being synchronically applied to other copies, and there will can be even a substantial time since the application of an update in a copy until the propagation of the update to other copies. The concurrent updates with the various copies can present conflicts, for example, in a distributed system of air line reservation which uses the optimistic strategy of consistency [4], two copies can accept a reservation for the same seat.

III. Multi-layer Model for Consistency Management

The pessimistic approach ensures a strong consistency for clients'response, but this approach is an impracticable solution for the large scale systems such as for example the Grid environment. This paper presents a hybrid approach for replicas consistency management. According to [11], designers of replicated systems for large scale systems had to choose between pessimistic consistency, with its associated performance overheads, and optimistic consistency, with no guarantees regarding the probability of conflicting writes or stale reads.

Between the two extremes *(Fig.1)*, we present a hybrid approach for replicas consistency management, where application designers can bound a reduced response time and an acceptable quality of service. Our approach is based on a hierarchical representation model with three layers, which facilitates the replica consistency management in the Data Grid.

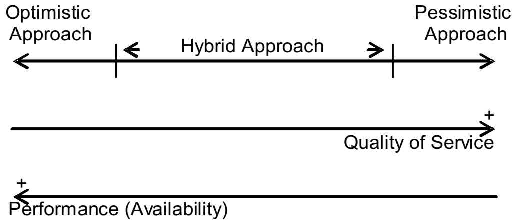

Fig. 1. Position Hybrid Approach

A. Presentation of the Model

The proposed approach uses a hierarchical model of a grid where the replicas of a data are located. This hierarchical model is tree-based and it is composed by only three layers. In our work, we consider a grid as a collection of distributed collection of Computing Elements (CEs) and Storage Elements (SEs). Replica are stored on Storage Elements and are accessible from Computing Elements. Each replica attached to additional information is called metadata [12] (TimeStamp, indices, versions, catalogues, …). The hierarchical model gives a tree-based view of a grid and defines the communication flows needed to ensure replica consistency (*Fig.2*). Our proposed model is composed of three layers, which collaborate between them to ensure a maintenance of coherence of the system.

1) Layer 0 of the grid is composed of the elementary entities of types elements of storage (SE) and computing of elements (CE). 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. In this study, we are interested in elements of storage which represents physical supports of storages of the replicas;
2) Layer 1 of the grid is composed of the sites. Each site gathers a whole of elementary elements of layer 0. Each site is responsible for the management of consistency of its group (consistency in intra-site) and cooperates with the other sites to ensure the total consistency of the grid (consistency inter-site);
3) Layer 2 is composed of an intelligent module, its principal function is the decision-making for the resolution of the conflicts which cannot be solved within layer 1.

By its simplicity, the multi-layer model proposed can be easily adapted to the large scale systems thanks to its main features that we can summarize as follows: (i) Simplicity, (ii) Transparency, (iii) Hybrid consistency management, (iv) Incremental consistency management, (v) Diversity of strategies, (vi) Passage to the scale.

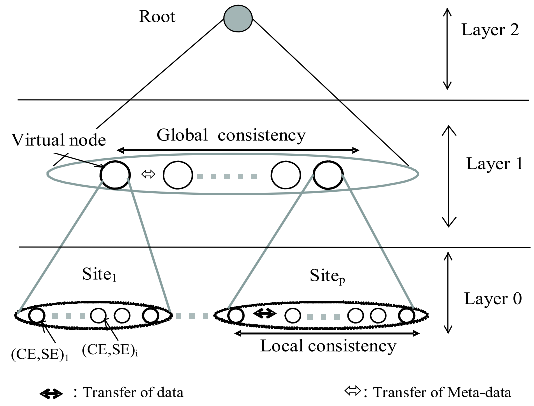

Fig. 2. Multi Layer Model Architecture

B. Process of the Hybrid Approach

To achieve the replica consistency, we define a service of consistency management which represents the core of hybrid model proposed. This service proceeds in three steps: the first step is carried out inside a site (we call it the intra-site) called local consistency and the second involves the different sites of a grid (we call it the inter-sites) called global consistency. The third stage is intended for the resolution of the conflicts inter-sites not yet solved by layer 1, called conflict manager.

1) Local Consistency: Local consistency is also called intra-site consistency. Its principal objective is to ensure consistency in a continuous way between the various replicas of the same data inside a site, which corresponds to make converge the replicas towards a relative replica for a site and it is founded on the optimistic approach of replication. It is started in an alternative way with global consistency. Its principal stapes are:

a- Replication strategy: This task defines the replication policy to use at the layer of a given site. The policy can be different from one site to another. Thus, it will be possible to apply customized policies based on techniques such as single master, multi-masters, quorum, etc. For more details about the existing techniques, see [13], [14].

b- Treatment of the request: With the reception of a request, subjected by a client towards a given site, it is immediately treated by the CE according to the strategy of replication of the site receiving and sent to the customer. For a request of writing, information of the metadata of this replica will be updated (version, timestamp,. . .).

c- Update propagation of the : In the event of a request of the writing type, a propagation of the updates is started for sleeping period of the site and it is carried out if and only if the replica is dominant by its number of version compared to the target. If it isn't the case the propagation is refused. In practice, the replica source diffuses its updates with the other replicas of the same site.

d- Detection and resolution of the conflicts intra-site: If two versions of two metadata different are identical then a conflict is detected between two replicas. Therefore, a resolution of conflict is as follows:

i. Choose the replica with the most recent date of the last updating;

ii. If the number of version does not permit to select one single representative, then choose the replica in the storage element with the highest reliability coefficient;

iii. If the problem of selection persists, then choose the most popular replica;

iv. In the worst case, use a random function to choose the replica (equality on all the other criteria).

2) Global Consistency: Global consistency is also called inter-sites consistency. Its principal objective is to ensure consistency between the various replicas of the same data of the Grid, which corresponds to make converge the replicas towards a reference replica for a Grid and it is founded on the pessimistic approach of replication. This reference replica will transmit its information towards the other representatives of the nodes. The Moment of release of global consistency can be launched according to several situations:

a- If the account of conflicts of a site exceeds a certain threshold, i.e. a rate of inconsistencies is very high, the site becomes unable to correctly serve the requests of the clients, we will speak about a divergence of the replicas according to a local view;

b- If the average of account of conflicts of the whole sites exceeds a certain threshold, that corresponds to a divergence of the copies according to a global view;

c- If the distance between two copies of intra-site or inter-sites reaches a breaking value, this corresponds to the margin between two replicas of the same data;

d- If the rate of writing reaches a given value;

e- After each past period (periodically).

Each site is represented by a node on the layer 1, which collaborates with the other nodes whose objective is to ensure a total coherence for the Grid. This mechanism of collaboration is based on the principle of negotiation between the various representatives of the sites (nodes) to find with a common solution converge towards a reference replica.

3) Conflicts Manager: The conflicts manager represented by the root has like tries principal coordination for the resolution of the conflicts between the nodes. The nodes coordinate in a collaborative way between them by means of the manager.

If the step of global consistency cannot manage, the collaboration to converge towards a replica of reference, the manager of conflicts intervenes to decide on a replica reference according to global consistency release moment of the situation : *a* , *b-c* and *d-e* (*Fig.3*). The decisions taken by the manager for the resolution of the conflicts inter-sites can be classified in three scenarios:

i. First scenario of decision relates to the first situation, the initiating site of the release of the total coherence transmitted the most popular metadata of a replica means to the conflicts manager by the intermediary of the node representing. The manager of conflicts

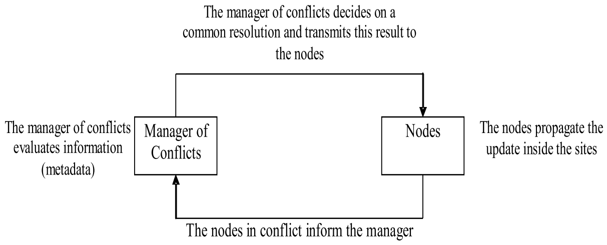
The manager of conflicts decides on a common resolution and transmits this result to the nodes
The manager of conflicts evaluates information (metadata)
Manager of Conflicts
Nodes
The nodes propagate the update inside the sites
The nodes in conflict inform the manager

Fig. 3. Process of resolution of conflicts

layers the various representatives nodes of the grid of layer 1, by the principle of the most frequent metadata. These representatives propagate new information at interior of the sites by diffusion with all the SE containing the replica concerned.

ii. Second scenario of decision relates to situations *b* and *c*. The conflicts manager calculates a total average starting from the local averages of the divergences of the sites, seeks the replica nearest in number to versions to this total average. This selected replica represents the reference for these conflicts;

iii. Third scenario of decision, relates to situations *d* and *e*. The conflicts manager seeks the most popular replica for the whole sites. This found replica will represent the global reference of replica.

C. Algorithms of the Proposed Approach

The approach proposed is a hybrid process, based on the use of the pessimistic and optimistic strategies [11], and described by three algorithms: hybrid, optimistic and pessimistic in the reason of maintain the consistency in data grids. This section describes the two main algorithms our model uses to implement the consistency management services discussed previously. The fist algorithm secures the intra-site consistency, while the second one the inter-sites consistency.

Algorithm 1 CONSISTENCY MAIN DRIVER

```
repeat
  while Not(Passage Consistency Inter-sites) do
    Consistency Intra-site();
  end while
  Consistency Inter-sites();
until End time of Simulation = True
```

The algorithm for the intra-site consistency management is inspired from the optimistic approach. It starts by identifying the type of request it receives as well as the replication strategy to apply in given a site (i.e, single-Master, multi-Master, etc.). It allows the propagation of the update, detects and resolves the conflicts, then executes customer's request.

The inter-sites consistency algorithm is derived from pessimistic approach. The algorithm starts by selecting a representative replica for each site. In case of a multi-Master strategy, the representative replica is chosen by election (Steps similar to the election of super-Master). Then, the metadata of replica is propagated towards the representatives nodes of layer1, afterwards the phase of election of the super Master is started. After that, a procedure is called to check whether or not there are conflicts between super-Master and representatives nodes. In the presence of conflict, another procedure will be launched to repair it.

The mechanism of global consistency can be described as follows: each node tries to publish information of its metadata on a common space between the nodes called *blackboard* by the principle of domination of the contents of the versions vector (*Algorithm 4*).

Algorithm 2 CONSISTENCY INTRA-SITE

```
while (List-Request= {}) Or Not(Consistency- Inter-sites) do
  if Not(Conflict) then
    Propagate the update intra-site
  else
    Resolution Conflict
    Propagate the update intra-site
  end if
end while
if Type (request) = Reading then
  Execute Request;
else
  if Master=Free then
    Execute Request;
  else
    Insert request in List-Request
  end if
end if
```

Algorithm 3 CONSISTENCY INTER-SITES

```
for all sites do
  if Strategy=Single-Master then
    representative ← Master
  else
    Election one representative of a site
  end if
end for
Negotiation($\bigcup$representatives)
if $Result$ then
  Service of manager of conflicts
  Diffusion of the contents of the Blackboard towards the nodes
  Propagate the update Intra-site
end if
```

Algorithm 4 NEGOTIATION

```
Result ← True
for all Nodes do
  if Blackboard(Metadata(Version)) < Node_i(Metadata(Version)) then
    Blackboard(Metadata) ← Node_i(Metadata)
  else
    if (Blackboard(Metadata(Version)) = Node_i(Metadata(Version))) then

      if Blackboard(Metadata) ≠ Node_i(Metadata) then
        Result ← False {Conflicts exist}
        return Result
      end if
    end if
  end if
end for
return Result {Conflicts No exist}
```

IV. SIMULATION AND RESULTS

In the absence of real trace data, we have implemented a simulation tool to study and analyze the protocol of coherence discussed in the preceding sections. The tool is based in the *OptorSim* Grid simulator [15], [16], extended with our proposed protocol. We have compared our protocol with the two traditional approaches (pessimistic and optimistic), we have chosen two categories of metrics:

- The first category of metrics called also measurements of performance, allows to study the behavior of our approach with that of the pessimistic approach. This category can comprise several measurements, such as: the response time of a request, latency, propagate time of the updates, or capacity of queue by site per unit time.
- The second category of metric called also measurements of qualities of service (QoS), allows to study the quality of the rendered service of our approach compared to the optimistic one. Among measurements used, we find the number of conflicts per site, the distance between the numbers of versions of the replicas by site, or the average distance from these versions for the whole sites per unit time.

The various parameters to be taken into account for simulation can be presented in the following table.

Notation	Definition
k	Number of Sites
$Site_j$	Number of (CE,SE)
NQ	Total number of requests
Ta_i	Arrival time of $request_i$
F_{ji}	Reliability $(CE,SE)_i$ of $site_j$
Bd_j	Network Bandwidth of $site_j$
$\sum_1^k \|site_j\|$	Total number of replicas

The objective of these first experiments is to prove that quality of service is better than optimistic approach.

- For metric called *number of conflicts*, we evaluate the number of conflicts by an interval of selected time. *Figure4* shows that the hybrid approach contains a number of conflicts much lower than the optimistic approach. Conflicts count by sites number is a conflict count for different number of sites, we observe in the figure *Figure5* that when we increase the sites number, conflicts number increase for the two approaches (*Optimistic, Hybrid*) until the end of simulation, but the result of the hybrid approach proposed are better than the optimistic approach.
- The metric *distance from conflicts* is defined by the difference between the maximum number of version and the minimal number of version of replicas. *Figure 6* shows merely that this distance is very significant in the optimistic approach compared to our approach. We observe, in figures 4, 5 and 6, that the divergence is very fast in the optimistic approach.

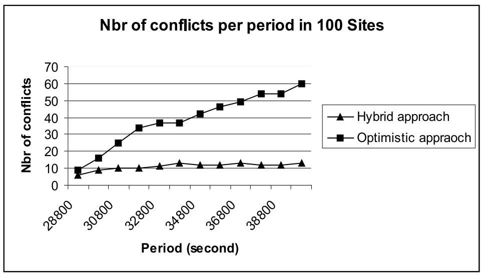

Fig. 4. Count conflicts by period

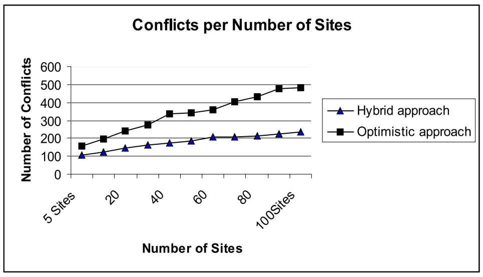

Fig. 5. Conflicts count by sites number

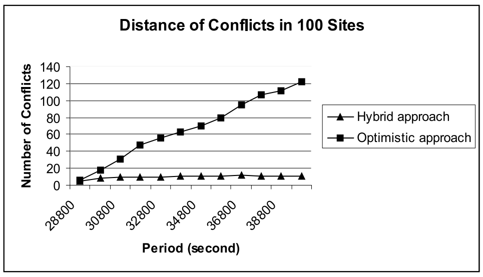

Fig. 6. Distance by Period

In order to study the performance and the availability, we chose to compare our approach with the two protocols of pessimistic consistency: *ROWA* (Read One Write All) and *majority Quorum* [14]. The results of simulation shown in *Figures 7 and 8* prove that the protocol suggested gives better results compared to the two pessimistic protocols. We notice that protocol *ROWA* very quickly becomes impracticable when the number of sites increases, the consequence is that it is an unsuitable protocol to large scale systems. From *Figure8*, we propose the average table of the profits to gain by the use of our multi-layer protocol compared to the pessimistic protocols. For example, we can see that when the number of sites reaches 100 for an average of 10 replicas by site, the profit to be gained compared to the *Quorum* protocol is 77% and of 91% for that of *ROWA*.

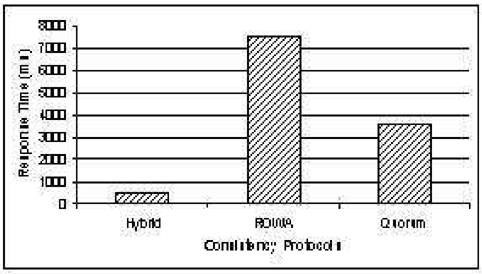

Fig. 7. Average response time of 20 Sites

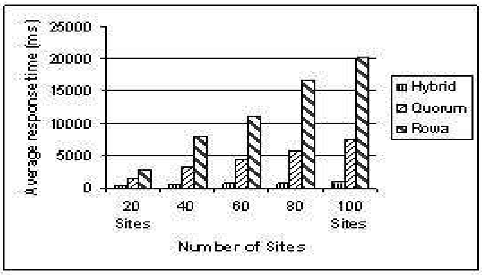

Fig. 8. Average response time per number of sites

Number of Sites	Profit Ours/Quorum	Profit Ours/ROWA
20	54%	72%
40	70%	87%
60	71%	87%
80	74%	90%
100	77%	91%

V. Conclusions and Future Works

In this paper, we proposed a tree-based approach dealing with replica consistency in large scale systems. This approach combines two approaches to replica consistency, namely optimistic and pessimistic one. Thanks to this combination, we have defined two complementary consistencies. A local consistency within the sites of a given grid, hence we can execute p local consistency operations simultaneously using only local information of a site. The main advantage of these parallel local consistency operations is to avoid totality communication between sites to reach consistency between replicas. In the case where local consistency fails to obtain consistency, we perform a second consistency using the nodes of layer 1 in the tree. We have presented the model based approach focusing on each level representation and algorithm. The results of simulation obtained are very satisfactory, and show that the approach is very promising especially when requiring quality of service and acceptable response time. Some works can be led to the future as:

- Experimentation of the proposed approach on a real grid;
- Supply the layer1 a multi-agents system to decide on the choice of the Global reference replica;

References

[1] I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kauffman Publishers Inc., San Francisco, 1999.

[2] K. Ranganathan and I. Foster. Identifying dynamic replication strategies for a high-performance data grid. In Springer Berlin, editor, *Grid: Second International Workshop*, volume 2242, pages 75–86, Denver,CO, USA, 12 November 2001.

[3] 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):1383–1398, 2002.

[4] H. Yu and A. Vahdat. Design and evaluation of a conit-based continuous consistency model for replicated services. *ACM Transactions on Computer Systems*, 20(3):239–282, Aug 2002.

[5] P. A. Bernstein and N. Goodman. The failure and recovery problem for replicated databases. In *PODC '83: Proceedings of the Second Annual ACM symposium on Principles of Distributed Computing*, pages 114–122, New York, NY, USA, 1983. ACM Press.

[6] Y. Saito and M. Shapiro. Optimistic replication. *ACM Comput. Surv.*, 37(1):42–81, 2005.

[7] H. Yu and A. Vahdat. The costs and limits of availability for replicated services. In *SOSP '01: Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, pages 29–42, New York, NY, USA, 2001. ACM Press.

[8] H. Yu and A. Vahdat. Minimal replication cost for availability. In *PODC '02: Proceedings of the Twenty-first Annual Symposium on Principles of Distributed Computing*, pages 98–107, New York, NY, USA, 2002. ACM Press.

[9] M. Ripeanu and I. Foster. A decentralized, adaptive replica location mechanism. In IEEE Computer Society, editor, *HPDC-11 02*, volume 0, page 24, Los Alamitos, CA, USA, 23-26 July 2002.

[10] B. S. White, M. Walker, M. Humphrey, and A. S. Grimshaw. Legionfs: a secure and scalable file system supporting cross-domain high-performance applications. In *Supercomputing '01: Proceedings of the 2001 ACM/IEEE Conference on Supercomputing (CDROM)*, pages 59–59, New York, NY, USA, 2001. ACM Press.

[11] G. Belalem and Y. Slimani. A hybrid approach for consistency management in large scale systems. In IEEE Computer Society, editor, *ICNS 06*, volume 0, page 71, Silicon Valley, USA, 16-19 July 2006.

[12] A. Domenici, F. Donno, G. Pucciani, H. Stockinger, and K. Stockinger. Replica consistency in a data grid. *Nuclear Instruments and Methods in Physics Research A*, 534, 2004.

[13] Y. Amir and A. Wool. Optimal availability quorum systems: Theory and practice. *Information Processing Letters*, 65(5):223–228, 1998.

[14] S. Goel, H. Sharda, and D. Taniar. Replica synchronisation in grid databases. *Int. J. Web and Grid Services*, 1(1):87–112, 2005.

[15] W. H. Bell, G. D. Cameron, L. Capozza, A. P. Millar, K. Stockinger, and F. Zini. Optorsim : A grid simulator for studying dynamic data replication strategies. *Int. Journal of High Performance Computing Applications*, 17(4):403–416, 2003.

[16] W. Bell, D. Cameron, R. Carvajal-Schiaffino, P. Millar, C.Nicholson, K. Stockinger, and F. Zini. *OptorSim v1.0 In-stallation and User Guide*, February 2004.