Load Balancing to Increase the Consistency of Replicas in Data Grids

Ghalem Belalem, University of Oran, Es Senia, Algeria
Naima Belayachi, University of Oran, Es Senia, Algeria
Radjaa Behidji, University of Oran, Es Senia, Algeria
Belabbes Yagoubi, University of Oran, Es Senia, Algeria

ABSTRACT

Data grids are current solutions to the needs of large scale systems and provide a set of different geographically distributed resources. Their goal is to offer an important capacity of parallel calculation, ensure a data effective and rapid access, improve the availability, and tolerate the breakdowns. In such systems, however, these advantages are possible only by using the replication technique. The use of this technique raises the problem of maintaining consistency of replicas of the same data set. In order to guarantee replica set reliability, it is necessary to have high coherence. This fact, however, penalizes performance. In this paper, the authors propose studying balancing influence on replica quality. For this reason, a service of hybrid consistency management is developed, which combines the pessimistic and optimistic approaches and is extended by a load balancing service to improve service quality. This service is articulated on a hierarchical model with two levels.

Keyword: Consistency, Data Grid, Hierarchical Model, Load Balancing, Replication

INTRODUCTION

Both computer system and network development has enabled large data set creation.

Most often, these data sets are replicated in multiple copies. This replication can meet different objectives: availability, performance, fault tolerance, etc. Unfortunately the more you reply, the more differences emerge and the more one is subject to inappropriate behaviour, and poor quality consequently. Broadly speaking, we can say that despite all its advantages, replication raises some problems (Gray et al., 1996) such as: (i) The choice of the entity to be replicated, (ii) Degree of replication, (iii) Replica placement (Belalem & Slimani, 2007; Xu et al., 2002), (iv) Replica choice which aims to select the best replica in terms of consistency (Ranganathan & Foster, 2001), (v) The problem of replica consistency, which deals with synchronization of multiple data copies updates in order to have a completely coherent view (Saito...
However, ensuring replica set reliability requires high consistency, a fact which can degrade performance. By contrast, good performance necessitates consistency loosening, which penalizes reliability (Belalem & Slimani, 2007).

In this paper, our proposal is to extend service management consistency by a load balancing sub-service to improve the service quality (QoS) in Data Grid (Foster & Kesselmann, 2004). The proposed approach is based on a hierarchical model with two levels.

The main purpose is maintaining the system coherence on a large scale system while maintaining acceptable performance.

**CONSISTENCY MANAGEMENT APPROACHES**

The Consistency is a relation which defines the degree of similarity between copies of a distributed entities. In the ideal case, this relation characterizes copies which have identical behaviors. Although in the real cases, even when the copies evolve in a different way, consistency defines the threshold of dissimilarity authorized between these copies. We hope of a consistency protocol which ensures the execution of the operations of users, the mutual consistency of copies in accordance with a behavior defined by a model of coherence. The consistency protocol gives an ideal view as if there is only one user and only one copy of the data in the system. Replica consistency management can be achieved, either synchronously, using the so-called pessimistic algorithms, or asynchronously, deploying optimistic ones (Belalem & Slimani, 2007; Saito & Shapiro, 2005). Fundamental tussles between pessimistic and optimistic approach are those related to scalability and security. The execution of pessimistic consistency assures that any change in one replica is atomically notified to all other replicas. Therefore, there is an inherent guarantee that all replicas will have the same data all the time, making this approach indispensable in the mission of critical and sensitive applications like the distributed banking application. On the other hand, the optimistic approach is employed for applications (large scale systems, mobile environments and system weakly coupled), which evolves rapidly in terms of response time for example. So that we can say that, the pessimistic approach is interested in consistency more than availability, while the optimistic approach supports the availability more than the consistency (Belalem & Slimani, 2007; Saito & Shapiro, 2005).

**Techniques of Pessimistic Consistency**

The technique of pessimistic consistency is interesting, since it guarantees a data consistency all the time. This approach gives users an illusion of having a single, highly available copy of data. However, the guarantee of the total maintenance of consistency involves a high communication cost (Pacitti et al., 1999; Yu & Vahdat 2001). The pessimistic algorithms (Saito & Shapiro 2005) prohibit the access to a replica unless it is updating. The advantage of the pessimistic approach is to avoid the problems involved in the reconciliation. A basic protocol, called RAWA (Read Any Write All) (Zhoun et al., 2004) consists in obtaining an exclusive bolt on all the copies before to effect a writing (respectively reading) on one of the copies. The availability of the readings is improved with protocol ROWA (Read Once Write All) (Goel et al., 2005). The readings lock and access only one copy, while the write access mode continues to lock and modify all the copies. Nevertheless, this protocol is blocking in the event of breakdowns. An alternative ROWAA (Read Once Write All Available) (Kemme, 2000; Zhoun et al., 2004) adapts this protocol to the cases of crashes by locking only the available copies. When a copy covers its availability, it must initially synchronize itself to execute the remaining updates. Another strategy of replication is proposed by the vote protocol family by Quorum (Amir & Wool, 1998; Rodrigues & Raynal, 2003). The transactions are sent to a