

Design and experimentations on an efficient data persistency service for ASP architecture

Bruno Del-Fabbro, David Laiymani, Jean-Marc Nicod, and Laurent Philippe

Laboratoire d'Informatique de l'université de Franche-Comté
16 route de Gray, 25030 Besancon Cedex - France
[delfabbro, laiymani, nicod, philippe]@lifc.univ-fcomte.fr

Abstract. A lot of grid computing researches has focused on providing reliable access to data produced by computations. In ASP (Application Service Provider) environments, the problematic of data management have led to only few works. In ASP platforms, managing data differs from others techniques used in grid environments. The issue of an ASP platform is to optimize computation times by decreasing data transfers between clients and the platform. In this way, not all data need to be managed by the platform and a few number of them can be considered persistent. In order to achieve such a goal, we have designed and implemented a data management service based on data persistence called DTM (Data Tree Manager). This service is based on a full data identification and on a hierarchical structure that provides data location and data movement between computational servers. In this paper, we present the architecture of the DTM service and a set of experimental results that exhibit the feasibility and the efficiency of our approach.

1 Introduction

Scientific applications present data requirements that increase every year. As grid environments [?,?] give answers to computational power requirements, they must also address the paradigm of data management. A lot of projects, like European Data Grid project have led to the definition of an unified interface for data access. The principle is that data produced by computations and stored on heterogeneous resources have to be accessed in a reliable and efficient way by users for their later analysis. As these users do not always know the name and the location of data they want to access, data must be referenced by their characteristics such as their production information (which application, what tests criteria) or their creation date. These works [?,?] have resulted in the notion of data virtualization which consists in the separation of the physical location of the data from its logical view. This concept has led to the paradigm of metadata[?].

In the approach previously exposed, data management consists in providing access to data for users that have not specific information about data location, name or attributes. An other approach is to place data close to computational servers in order to increase performances by decreasing network traffic. This is typically the case of GridRPC environments as defined in [?] (also named ASP platforms) in which sequences of computations are submitted to the platform. Data sent or produced can then be used for several computational requests. Note that the two orientations are not antagonistic. Indeed data produced by ASP platforms can then be virtualized in order to