The Knowledge Grid: Towards an Architecture for Knowledge Discovery on the Grid

Mario Cannataro, Domenico Talia, and Paolo Trunfio

ISI-CNR
Via P. Bucci, Cubo 41C
87036 Rende (CS), Italy
{cannataro,talia,trunfio}@si.deis.unical.it

Knowledge Discovery in Databases (*KDD*) employs a variety of software systems and tools, collectively called *data mining*, to find useful patterns, models and trends in large volumes of data. In many scientific and commercial applications, it is necessary to perform the analysis of large data sets, maintained over geographically distributed sites, by using the computational power of distributed and parallel systems. These techniques are investigated in the domain of *Parallel and Distributed Knowledge Discovery* (*PDKD*). In this area grid technologies may play a significant role in providing an effective computational support for knowledge discovery applications. Here we propose a software architecture for geographically distributed PDKD applications called *Knowledge Grid*, which is designed on top of computational grid mechanisms, provided by grid environments such as Globus. The Knowledge Grid uses the basic grid services such as communication, authentication, information, and resource management to build more specific PDKD tools and services. The Knowledge Grid services are organized into two layers: *core K-grid layer*, which is built on top of generic grid services, and *high level K-grid layer*, which is implemented over the core layer.

The core K-grid layer comprises two basic services: *Knowledge Directory Service* (*KDS*) and *Resources Allocation and Execution Management* (*RAEM*). The KDS manages the metadata describing characteristics of relevant objects for PDKD applications, such as data sources, data mining software, results of computations, data and results manipulation tools, execution plans, etc. The information managed by the KDS is stored into three ad hoc repositories: the metadata describing features of data, software and tools are stored in a *Knowledge Metadata Repository* (*KMR*), the information about the knowledge discovered after a PDKD computation is stored in a *Knowledge Base Repository* (*KBR*), whereas the *Knowledge Execution Plan Repository* (*KEPR*) stores the execution plans describing PDKD applications over the grid. The goal of RAEM services is to find a mapping between an execution plan and available resources on the grid, satisfying user, data and algorithms requirements and constraints.

The high level K-grid layer comprises the services used to build and execute PDKD computations over the grid. The *Data Access* (*DA*) services are used for the search, selection, extraction, transformation and delivery of data to be mined. The *Tools and Algorithms Access* (*TAAS*) services are responsible for the search, selection, downloading of data mining tools and algorithms. The *Execution Plan Management* (*EPM*) is a semi-automatic tool that takes the data and programs selected by a user, and generates a set of different possible execution plans. Execution plans are stored in the KEPR to allow for the implementation of iterative knowledge discovery processes, e.g., periodical analysis of the same data sources that vary during time. The *Results Presentation Service* (*RPS*) specifies how to generate, present and visualize the PDKD results (rules, associations, models, classification, etc.), and offers methods to store in different formats these results in the KBR.

We are implementing a prototype of the system on top of Globus. In particular, we implemented the main data structures and services of KDS and DA. The metadata describing relevant objects for PDKD computations, such as data sources and data mining software, are represented by XML documents into a local repository (KMR), and their availability is published by entries into the *Directory Information Tree* maintained by a LDAP server, which is provided by the *Grid Information Service* (*GIS*) of the Globus Toolkit. The main attributes of the LDAP entries specify the location of the repositories containing the XML metadata, whereas the XML documents maintain more specific information for the effective use of resources. We implemented the basic tools of the DA service, allowing to find, retrieve and select metadata about PDKD objects on the grid, on the basis of different search parameters and selection filters. Moreover, we are modeling the representation of execution plans as graphs, where nodes represents computational elements (data sources, software programs, results, etc.) and arcs represents basic operations (data movements, data filtering, program execution, etc.). We plan to consider different network parameters, such as topology, bandwidth and latency, for PDKD program execution optimization.