KNOWLEDGE GRID
An Architecture for Distributed Knowledge Discovery

Mario Cannataro[1] and Domenico Talia[2]

[1]ICAR-CNR
Via P. Bucci, Cubo 41-C
87036 Rende (CS)
Italy

[2]DEIS
University of Calabria
Via P. Bucci, Cubo 41-C
87036 Rende (CS)

{cannataro,taliad}@acm.org

Today there is a huge amount of information stored in digital data repositories. Often it is hard to understand what is the important and useful information in those massive data sets. To sift large data sources, computer scientists designed software techniques and tools that can analyze data to find useful patterns in them. These techniques contribute to define the so called *knowledge discovery in databases* (*KDD*) process. In particular, *data mining* (*DM*) is the basic component of the KDD process for the semi-automatic discovery of patterns, associations, changes, anomalies, events and semantically significant structures in data. Typical examples of data mining tasks are data classification and clustering, events and values prediction, association rules discovery, and episodes detection [3].

Attempts to automate the process of knowledge extraction date from at least the early 1980s, with the work on statistical expert systems. But today new techniques, mainly in the artificial intelligence field, such as rule induction, neural networks, Bayesian networks and genetic algorithms are used. The huge size of data sources means that we cannot do detailed analysis unaided, but must use fast computers applying sophisticated software tools from statistics to artificial intelligence.

Recently, several KDD systems have been implemented on parallel computing platforms to achieve high performance in the analysis of large data sets that are stored in a single site. However, KDD systems must be able to handle and analyze multi-site and multi-owner data repositories. The combination of large data set size, geographic distribution of data, users and resources, and computationally intensive analysis demands for new *parallel and distributed knowledge discovery* (PDKD) platforms. In this setting *computational grids* are an emerging infrastructure that enables the integrated use of remote high-end computers, databases, scientific instruments, networks, and other resources. Grid applications often involve large amounts of computing and/or data. For these reasons, we think grids can offer an effective support for the implementation and use of PDKD systems.

This article introduces and discusses a reference software architecture for geographically distributed PDKD systems called *Knowledge Grid*. The architecture is built on top of a computational grid that provides dependable, consistent, and pervasive access to high-end computational resources. The *Knowledge Grid* uses the basic grid services and defines a set of additional layers to implement the services of distributed knowledge discovery on world wide connected computers where each node can be a sequential or a parallel machine. The *Knowledge Grid* enables the collaboration of scientists that must mine data that are stored in different research centers as well as analysts that must use a knowledge management system that operates on several data warehouses located in the different company establishments.

Parallel and Distributed Data Mining on Grids

Parallel and distributed knowledge discovery is based on the use of high-bandwidth communication networks and high-performance parallel computers for the mining of data in a distributed and parallel fashion. This technology is particularly useful for large organizations, environments and enterprises that manage and analyze data that are geographically distributed in different data repositories or warehouses [6].

The *Grid* has recently emerged as an integrated infrastructure for coordinate resource sharing and problem solving in distributed environments. Grid applications often involve large amounts of data and/or computing, and are not easily handled by today's Internet and Web infrastructures. Grid middleware targets technical challenges in such areas as communication, scheduling, security, information, data access, and fault detection [4]. However, mainly because of the recent availability of grid middleware, till today a very few efforts has been devoted to the development of PDKD tools and services onto the computational grid. Because of the importance of data mining and grid technologies, it is very useful to develop data mining environments on grid platforms by deploying grid services for the extraction of knowledge from large distributed data repositories.

Motivated by these considerations, we designed a reference software architecture, the *Knowledge Grid*, for the implementation of PDKD systems on top of grid systems such as the Globus Toolkit and Legion. We attempt to overcome the difficulties of wide area, multi-site operation by exploiting the underlying grid infrastructure that provides basic services such as communication, authentication, resource management, and

information. To this end, we organized the *Knowledge Grid* architecture so that more specialized data mining tools are compatible with lower-level grid mechanisms.
The basic principles that motivate the architecture design of a grid-aware PDKD system, such as the *Knowledge Grid* we designed are

- *Data heterogeneity and large data sets handling,*
- *Algorithm integration and independence,*
- *Compatibility with grid infrastructure and grid awareness,*
- *Openness,*
- *Scalability,* and
- *Security and data privacy*.

Basic grid services

As mentioned before, grid infrastructure tools, such as Globus Toolkit [4] and Legion, provide basic services that can be effectively used in the development of the *Knowledge Grid* handling distributed, heterogeneous computing resources as a single virtual parallel computer. Just to outline the type of services, in figure 1 we list the Globus generic services and Data Grid services [2]. These services address several PDKD requirements discussed before and are helpful for the implementation of the *Knowledge Grid* architecture.

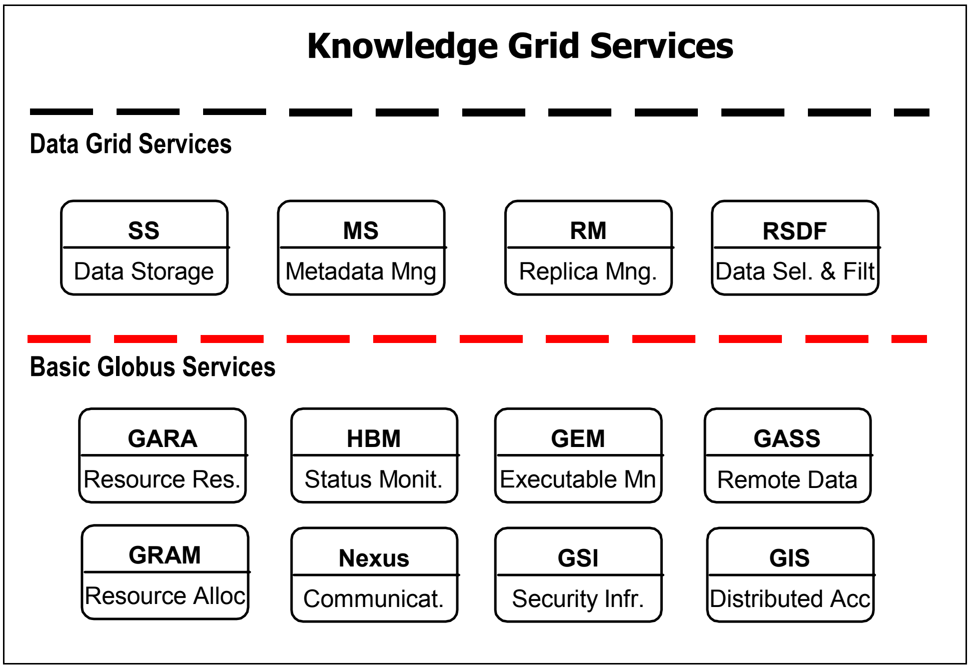

Fig 1. Globus Generic services and Data Grid services.

The Knowledge Grid Architecture

The *Knowledge Grid* architecture is defined on top of grid toolkits and services, i.e. it uses basic grid services to build specific knowledge extraction services [1]. Following the Integrated Grid Architecture approach [4], these services can be developed in different ways using the available grid toolkits and services. Here we discuss an architecture based on the Globus Toolkit 2.0.

Knowledge Grid services

The *Knowledge Grid* services are organized in two hierarchic levels: *Core K-grid layer* and *High level K-grid layer*. The former refers to services directly implemented on the top of generic grid services, the latter refers to services used to describe, develop, and execute PDKD computations over the *Knowledge Grid*. The *Knowledge Grid* layers are depicted in figure 2.

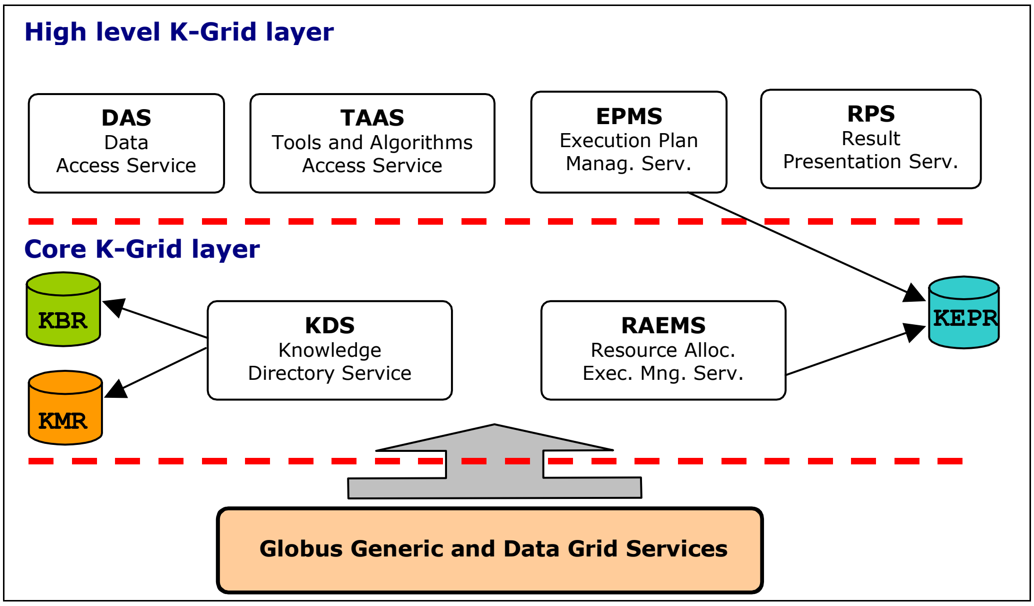

Fig. 2. Knowledge Grid architecture layers.

The figure shows layers as implemented on the top of Globus services; moreover, the Knowledge Grid data and metadata repositories are also shown. In our architecture, a generic *grid node* implements the Globus middleware, whereas a *K-grid node* node implements also the *Knowledge Grid* services.

Core K-grid layer

The Core K-grid layer has to support the definition, composition and execution of a PDKD computation over the Grid. Its main goals are the management of all metadata describing characteristics of data sources, third-party data mining, data management, and data visualization tools and algorithms. Moreover, this layer coordinates the PDKD computation execution, attempting to match the application requirements and the available grid resources. This layer comprises the following basic services:

Knowledge Directory Service (KDS). This service extends the Globus Monitoring and Discovery Service (MDS) and is responsible for maintaining a description of all the data and tools used in the *Knowledge Grid*. The metadata managed by the KDS regard the following kind of objects:

- data sources providing the data to be mined, such as databases, plain files, XML documents and other structured or unstructured data. Usually data to be mined are extracted from their sources only when needed;
- tools and algorithms used to search/find, extract, filter and manipulate data (data management tools);
- tools and algorithms used to analyze (mine) data (data analysis tools);
- tools and algorithms used to visualize, store and manipulate PDKD computation results (data visualization tools);
- PDKD execution plans, which are graphs describing the interactions and data flows among data sources, DM tools, visualization tools, and result storing. In fact, an execution plan is an abstract description of a PDKD grid application;
- PDKD results, i.e. the "knowledge" discovered after a PDKD computation.

The metadata information are represented by XML documents and are stored in a *Knowledge Metadata Repository (KMR)*. For example, they describe features of different data sources that can be mined, such as location, format, availability, available views and level of aggregation of data.

Whereas it could be infeasible to maintain the data to be mined in an ad hoc repository, it could be useful to maintain a repository of the "knowledge" discovered after a PDKD computation. This is useful not only for the analysis of results, but also because the output of a computation could be used as input of another computation. These information (see below) are stored in a *Knowledge Base Repository (KBR)*, but the metadata describing them are managed by the KDS. The KDS is so used not only to search and access raw data, but also to find pre-discovered knowledge that can be used to compare the output of a given PDKD computation when varying data, or to apply data mining tools in an incremental way.

Data management, analysis and visualization tools usually are pre-existent to the *Knowledge Grid*, so they should not be stored in any specialized repository (i.e. they resides over file systems or code libraries). However, to make them available to PDKD computations, relevant metadata have to be stored in the KMR. In a similar way, metadata are to be stored to allow the use of data sources. Finally, the *Knowledge Execution Plan Repository (KEPR)* is used for storing the execution plans of PDKD computations.

Resource Allocation and Execution Management Services (RAEMS). These services are used to find a mapping between an execution plan and available resources, with the goal of satisfying requirements (e.g., performance, response time, and I/O bandwidth) and constraints (e.g., available computing power, storage, network bandwidth and latency). The mapping has to be effectively obtained (co-) allocating resources. After the execution plan has been started, this layer has to manage and coordinate the application execution. Other than using the KDS and the MDS services, this layer is directly based on the GRAM services. Resource requests of single data mining programs are expressed using the Resource Specification Language (RSL). The analysis and processing of the execution plan will generate global resource requests that in turn are translated into local RSL requests for local GRAMs and communication requirements for Nexus or other high level communication services.

High level K-grid layer

The high-level K-grid layer comprises the services used to compose, validate, and execute a PDKD computation. Moreover, the layer offers services to store and analyze the discovered knowledge. Main services are:

Data Access Services (DAS). The Data Access Services are responsible for the search, selection (*Data search services*), extraction, transformation and delivery (*Data extraction services*) of data to be mined. The search and selection services are based on the core *KDS* services. On the basis of the user requirements and constraints, the DAS automates the searching and finding of data sources to be analyzed by the DM tools.

The extraction, transformation and delivery of data to be mined (*Data extraction*) are based on the Globus GASS services and use the KDS. After useful data have been found, the data mining tools can require some transformation, whereas the user requirements or security constraints may need some data filtering before extraction. These operations can usually be done after the DM tools are chosen.

Tools and Algorithms Access Services (TAAS). These services are responsible for search, selection, downloading of data mining tools and algorithms. As before, metadata regarding their availability, location, configuration, etc., are stored in the KMR and managed by the KDS, whereas the tools and algorithms are stored in the local storage facility of each K-grid node. A node wishing to "export" data mining tools to other users has to "publish" them using the KDS services, which store the metadata in the local portion of the KMR. Some relevant metadata are parameters, format of input/output data, kind of data mining algorithm implemented, resource requirements and constraints, and so on.

Execution Plan Management Services (EPMS). An execution plan is an abstract description of a PDKD grid application**.** It is a graph describing the interaction and data flows between data sources, extraction tools, DM tools, visualization tools, and storing of knowledge results in the KBR. In simplest cases the user directly describes the execution plan, using a visual composition tool where the programs are connected to the data sources. However, due to the variety of results produced by the DAS and TAAS layers, different execution plans can be produced, in terms of data and tools location, strategies to move or stage intermediate results and so on. Thus, the EPMS is a semi-automatic tool that takes the data and programs selected by the user, and generates a set of different execution plans that satisfy user, data and algorithms requirements and constraints.

Execution plans are stored in the *Knowledge Execution Plan Repository* to allow the implementation of iterative knowledge discovery processes, e.g. periodical analysis of the same data sources that vary during time. More simply, the same execution plan can be used to analyze different sets of data. Moreover, different execution plans can be used to analyze in parallel the same set of data, and to compare the results using different point of views (e.g., performance, accuracy).

Results Presentation Services (RPS). This layer specifies how to generate, present and visualize the PDKD results (rules, associations, models, classification, etc.). Moreover, it offers the functions to store these results in different formats in the Knowledge Base Repository (e.g. graphics, animations, texts, etc.). The result metadata are stored in the KMR to be managed by the KDS. Pre-discovered knowledge can be used as input for a new discovery process. Thus the KDS is so used also to find available pre-discovered knowledge or to apply data mining tools in an incremental way.

Related work

Whereas some PDKD systems supporting high-performance distributed data mining recently appeared [5] (for a short review see [1]), there are really few projects attempting to build knowledge Grids on the top of computational Grids. More specifically, many PDKD systems operate on clusters of computers or over the Internet, but, none of those, to the best of our knowledge, makes use of the computational grid basic services (e.g. authentication, data access, communication and security services). On the other hand, emerging knowledge Grids can be roughly classified as domain-specific (e.g., TeraGrid, Discovery Net), and domain-independent knowledge Grids. The *Knowledge Grid* we designed is one of the first attempts to build a domain-independent knowledge discovery environment on the Grid.

Here we very shortly mention the most significant Grid-based projects/systems discussing differences and common aspects with respect to our *Knowledge Grid* system.

The TeraGrid project is building a powerful grid infrastructure, connecting four main sites in USA (San Diego Supercomputer Center, National Center for Supercomputing Applications, Caltech and Argonne National Lab) that will provide access to tera-scale amounts of data. The most challenging application on the TeraGrid will be the synthesis of knowledge from very large scientific data sets. The use of the *Knowledge Grid* services can be potentially effective in those applications.

The ADaM (Algorithm Development and Mining) system is an agent-based data mining framework developed at the University of Alabama in Huntsville used to mine hydrology data in parallel from four sites. This system uses a design approach similar to the *Knowledge Grid* principles but the system architecture is simpler and the system purpose is limited in the application area for which the system has been designed to.

The Discovery Net is a newly announced EPSRC's project (Engineering and Physical Sciences Research Council), at Imperial College. This system aims to develop High Throughput Sensing (HTS) applications by using the Kensington Discovery Platform on the top of the Globus services. In this case the rationale is to port a Javabased distributed data mining system to grid platforms using the Globus toolkit. The main question mark here is how the pre-existent system can adapt to grid mechanisms and policies.

Finally, the National Center for Data Mining (NCDM) at the University of Illinois at Chicago (UIC) is developing some significant testbeds on knowledge discovery over Grids.

In summary, these emerging knowledge discovery-oriented Grids are almost all facing specific application domains. Our system, other being independent by the application domain, adopts specifically designed tools for the management of knowledge discovery processes that allow a user to evaluate and compare different knowledge models and for the transparent integration of parallel and sequential data mining tools and algorithms.

Current work and conclusion

A prototype of the *Knowledge Grid* has been implemented on top of Globus. We implemented the main components of the system in a toolset called Visual Environment for Grid Applications (*VEGA*) that offers the functionalities of the DAS and TAAS services (search and selection of data source and tools), the EPMS services (design of a PDKD application), the RAEMS services (optimization and translation of the execution plan on the Globus code), and the RPS services (result collection and presentation) through a graphical interface that a user can utilize to compose and execute a knowledge discovery computation in a simple way.

Grid computing is the most promising framework for future implementations of high performance data intensive distributed applications. Although today the Grid is mainly used for scientific applications in a near future it will be used for industrial and commercial applications. In these areas knowledge discovery is very important and critical. Furthermore, the Internet is shifting from an information and communication infrastructure to a *knowledge delivery infrastructure*. The discovery and extraction of knowledge from geographically distributed sources will be more and more important in many everyday life activities. The *Knowledge Grid* represents a significant step in the process of studying the unification of knowledge discovery and Grid technologies and defining an integrating architecture for distributed data mining and knowledge discovery based on Grid services. Such an architecture will accelerate progress on very large-scale geographically distributed data mining by enabling the integration of currently disjoint approaches and revealing technology gaps that require further research and development.

References

1. M. Cannataro, D. Talia, P. Trunfio, KNOWLEDGE GRID: High Performance Knowledge Discovery Services on the Grid. *Proc. GRID 2001*, LNCS, pp. 38-50, Springer-Verlag, 2001.
2. Chervenak A., Foster I., Kesselman C., Salisbury C. and Tuecke S., The Data Grid: towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of Network and Computer Applications*, **23**, pp.187-200, 2001.
3. Fayyad U.M. and Uthurusamy R. (eds.), Data mining and knowledge discovery in databases. *Communications of the ACM* **39**, 1997.
4. Foster I. and Kesselman C. (eds.) *The Grid: Blueprint for a Future Computing Inf.*, Morgan Kaufmann Publishers, 1999, pp. 105-129.
5. Kargupta H. and Chan P. (eds.), *Advances in Distributed and Parallel Knowledge Discovery, AAAI/MIT Press*, 2000.
6. Zaki M. J. and Ho C.-T. (eds.), *Large-scale Parallel Data Mining*, Lecture Notes in Artificial Intelligence, Vol. 1759, Springer-Verlag, 2000.