PARALLEL AND DISTRIBUTED KNOWLEDGE DISCOVERY ON THE GRID: A REFERENCE ARCHITECTURE

M. CANNATARO AND D. TALIA
ISI-CNR, Via P. Bucci, cubo 41-c, 87036 RENDE (CS)
ITALY
E-mail:{cannataro, talia}@deis.unical.it

In an increasing number of scientific and commercial areas, tools and systems for the analysis of large data sets are emerging as important resources. In particular, when large data sets are coupled with geographic distribution of data, users and systems, it is necessary to combine different technologies for implementing high-performance distributed knowledge discovery systems. The discipline that study and use those tools is named Parallel and Distributed Knowledge Discovery (PDKD). In this paper we introduce a reference software architecture for PDKD systems that is built on top of computational grids that provide dependable, consistent, and pervasive access to high-end computational resources. The proposed architecture uses the grid services and defines a set of additional layers to implement the services of distributed knowledge discovery process on distributed computers where each node can be a sequential or a parallel machine.

1 Introduction

The information stored today in computers is very large and it is growing very rapidly. Whereas until some years ago the main problem was the lack of information, today and in the future the main problem is that we have too much information to deal with and it is hard to understand what in it is important. Just to give some examples, Wal-Mart handles about 20 million of transactions each day using a database of more that 4 terabytes, the NASA satellites system for earth observation generates 50 gigabytes of images each our, finally the Human Genoma project collected several terabytes of data on human genetic code that must be analyzed.

To solve these problems, computer scientists are working on software tools that can analyze data to find useful patterns in them. This operation is called *data mining* or *knowledge discovery in databases* (*KDD*). More properly, data mining is the basic component of the KDD process for the semi-automatic discovery of patterns, associations, changes, anomalies, and semantically significant structures and events in data. Typical examples of data mining tasks are data classification and clustering, event 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 size of data sources mean that we cannot do it unaided, but must use fast computers, applying sophisticated software tools from statistics to artificial intelligence. Today data mining is used in commerce, scientific data analysis, banking, medicine and economics with very interesting, and sometime surprising, outcomes. Industries, finance companies, public administrations, scientific laboratories and Web-based enterprises are benefiting from this technology and they are gaining positions over competitors by discovering knowledge in their own databases.

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 that must be able to handle and analyze multi-site data repositories. The combination of large data set size, geographic distribution of data, users and resources, and computationally intensive analysis demand for a parallel and distributed data management and analysis infrastructure for *parallel and distributed knowledge discovery* (PDKD).

This paper discusses and proposes a reference architecture for geographically distributed PDKD systems called *Knowledge Grid*. The proposed architecture is built on top of a computational grid that provide dependable, consistent, and pervasive access to high-end computational resources. The proposed architecture uses the grid services and defines a set of additional layers to implement the services of distributed knowledge discovery process 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 executive managers that must use a knowledge management system that operates on several data warehouses located in the different company establishments.

Some recent papers propose PDKD systems that address these issues, but there is no work in designing an integrating PDKD architecture that makes of use of the *grid computing* resources for data-mining petabyte-scale applications that both address high-performance and wide area operations.

This paper represents a first step in the design and implementation of a PDKD architecture that integrate data mining techniques and computational grid resources. Section 2 describes the existing approaches in the implementation of PDKD systems. Section 3 discusses the requirements of parallel and distributed data mining on grids and what basic services the grid offers as basic support for PDKD. Section 4 describes the Knowledge Grid architecture and define the features and services of each layer of the proposed architecture.

2 Existing Approaches for PDKD

In the literature are described some PDKD systems that support high-performance distributed data mining. Those systems operate on clusters of computers or over the Internet, but, none of those we know make use of the computational grid infrastructure for the implementation of the basic services of authentication, data access, communication and security.

Papyrus [7] is a distributed data mining system developed for clusters and super-clusters of workstations as composed four software layers: data management, data mining, predictive modeling, and agent or Bast. The data management layer is implemented as a global data warehouse that allows to move data from node to node. The Bast layer selects strategies and resources and move predictive models among the cluster nodes. Thus this system has the ability to combine data and models movement. For this Papyrus is based on mobile agents implemented using Java aglets.

Another interesting distributed data mining suite based on Java is PaDDMAS [11], a component-based tool set that integrates pre-developed or custom packages (that can be sequential or parallel) using a dataflow approach. PaDDMAS provides three types of components: data managemnt components, data analysis components and data visualization components. Each component is wrapped as a Java or CORBA object with its interface specified in XML. Components can be located at distributed nodes and connectivity to databases is provided thorough JDBC bridges.

Kensington Enterprise data mining [1] is a PDKD system based on a three-tier client/server architecture in which the three tiers include: client, application server and third-tier servers (RDBMS and parallel data mining service). The data models (knowledge) built on a remote server are moved, under the control of the application server, to the client node for evaluation. The Kensington system has been implemented in Java and uses the Enterprise JavaBeans component architecture. Via a CORBA interface C+MPI data mining packages can be integrated in the Java framework and databases located on the Internet can be accessed via a JDBC connection.

JAM [12] is an agent-based distributed data mining system that has been developed to mine data stored in different sites for building so called meta-models as a combination of several models learned at the different sites where data are stored. JAM uses Java applets to move data mining agents to remote sites. In this way, the knowledge discovery is speeded up by executing in parallel a number of data mining processes on different data subsets and then combining the results through meta-learning. A sort of meta-learning, called *collective data mining*, is implemented also in the BODHI system [8]. BODHI is another agent-based distributed data mining system implemented in Java as a hierarchy of four main components: the individual agents, the agent stations on each processing node, the facilitator for coordination of agent stations and the user interface for system configuration and control. Like JAM and the Kensington system, BODHI support the migration of mining agents over the nodes where data are stored and collect the produced models on a central node where models are combined and analyzed.

Besides the systems we discussed here, other distributed data mining systems that have recently been developed or are in development are WoRLD, MASSON [9], PADMA and DMA.

Concurrently with these research work on distributed data mining, several research groups are working in the computational grid area developing algorithms, components, and services that can be exploited in the implementation of distributed data mining systems. Thus, this work could be useful integrated with work on parallel and distributed data mining to obtain world-wide grid based PDKD systems for the analysis of large data collections in scientific and commercial areas.

3 Parallel and Distributed Data Mining on Grids

Traditional sequential KDD typically requires local (central) storage of data, which may not always be stored in a single repository. Their collection in some cases is not feasible because of their large size, limited network bandwidth, security concerns, scalability problems, or just because they are have different owners or are geographically distributed. PDKD is the application of the KDD techniques to distributed, large, possibly heterogeneous, volumes of data that are residing over computing nodes distributed on a geographic area. Several parallel algorithms for single data mining tasks such as classification, clustering and association have been designed in the past years. However, it lacks a proposal for integrated environments that use novel computing platforms to PDKD environments that integrate different sources, models, and tools. 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 [5].

The Grid has emerged recently as an integrated infrastructure for high-performance distributed computation. 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 [10]. However, till today no efforts are devoted to the development of PDKD tools and services on 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. The only effort that has been done in the direction of data intensive applications on the grid is the *Data Grid* project that aims to implement a data management architecture based on two main services: storage system and metadata management [2]. This project is not concerned with data mining issues, but its basic services could be used to implement higher-level grid services such as the ones we intend to develop.

Motivated by these considerations, we are working in the definition of a reference software architecture, which we call the *Knowledge Grid*, for the implementation of PDKD systems on top of grid toolkits such as Globus, Legion, and NASA IPG. 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 organize the knowledge grid architecture so that more specialized data mining tools are compatible with lower-level Grid mechanisms and also with the Data Grid services. This approach benefits from "standard" grid services that are more and more utilized and offers an open PDKD architecture that can be configured on top of grid middleware in a simple way.

3.1 Requirements

Here we identify the basic principles that motivate the architecture design of the grid-aware PDKD system we propose.

Data heterogeneity and large data size. The system must be able to cope with very large and high dimensional data sets that are geographically distributed and stored in different types of repositories as structured data in DBMS, text in files or semi-structured data in Web sites.

Algorithm integration and independence. The architecture must allow the integration of different data mining algorithms and suites and must be as independent as possible from the data mining algorithms used for knowledge extraction. A data interface orthogonal to the data mining tools so that data access will be uniform is to be defined.

Compatibility with grid infrastructure and grid awareness. The higher levels of the architecture use the basic grid services for implementing wide area communication, cooperation and resource management of a PDKD system. Thus the data mining services are interfaced with the lower levels of the grid infrastructure. The interface is aware of the grid services and accesses them for supporting the data mining components in the distributed execution of knowledge discovery tasks.

Openness. The system architecture must be open to be integrated with new data mining tools and knowledge discovery packages. Sequential and parallel analysis models will be added to extend the knowledge discovery services without affecting the lower levels of the Knowledge Grid architecture.

Scalability. The architecture must be scalable both in terms of number of nodes used for performing the distributed knowledge discovery tasks and in terms of performance achieved by using parallel computers to speed up the mining task.

Security and data privacy. Security and privacy issues are vital features in wide area distributed systems. The grid services offer a valid support to the PDKD system to cope with user authentication, security and privacy of data. Basic grid functionality (e.g., Globus security infrastructure - GSI) are able to support secure client-server interactions without impacting on the usability of the grid infrastructure and services.

3.2 *What the Grid offers*

As mentioned before, grid infrastructure tools, such as Globus [4] and Legion [6], 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 tables 1 and 2 we list the core Globus services [4] and the Data Grid services [10]. These services address several PDKD requirements discussed before and need to be included in the Knowledge Grid architecture.

Table 1. Basic Globus Services.

Name	Description
GRAM	Resource allocation and process management
Nexus	Unicast and multicast communication services
GSI	Authentication and related security services
MDS (GIS)	Distributed access to structure and state information
HBM	Monitoring of status of system components
GASS	Remote access to data via seq. and parallel interfaces
GEM	Construction, caching, and location of executables
GARA	Advanced resource reservation and allocation

Table 2. Data Grid Services.

Name	Description
RSDF	Replica selection and data filtering services
RM	Replica management services
MS	Metadata information management services
SS	Storage system and data access services

For each of the listed services, C and/or Java application programming interface (API) is defined for use by developers, thus the higher-level components of a PDKD system will use these services by calling the corresponding APIs.

4 A Reference 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 Grid services. Following the Integrated Grid Architecture approach, these services can be developed in different ways using the available grid toolkits and services, such as Globus, Legion, SNIPE, etc. However, in this paper an architecture based on the Globus toolkits will be discussed.

As in Globus, the Knowledge Grid offers some global services based on the cooperation and combination of local services. For example, on each node of Globus the MDS (recently named GIS) service provides information about system components available on that node (local service), and it is implemented by a LDAP server. So the Global MDS services are obtained on the basis of the local services through the cooperation of those LDAP servers. In the Knowledge Grid, the discovery of data or algorithms is based on a Knowledge Directory Service (KDS), developed on and extending the basic Globus MDS service.

The Knowledge Grid, shown in figure 1, is composed by two main components:

- a set of knowledge grid-enabled (K-grid) nodes declaring their availability to participate to some PDKD computation, that are connected by,
- a grid infrastructure, offering basic grid-services (authentication, data location, service level negotiation, etc.) and implementing the Knowledge Grid services.

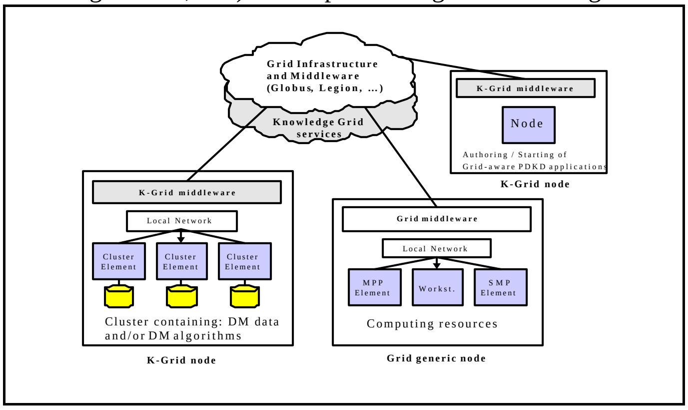

Figure 1. Knowledge Grid components interconnection with generic nodes and K-grid nodes.

Because of the grid infrastructure continues to serve pre-existing grid applications, generic grid nodes could not participate in the Knowledge grid. A K-grid node can start a PDKD application, or it just provide data for distributed data mining, or it can execute data mining packages. The users of the Knowledge Grid can be:

- Domain users, i.e. scientists that are mainly interested in executing available PDKD applications, analyzing and comparing their results with the knowledge available on the grid or that can develop their own PDKD applications composing existing DM tools and algorithms, using the Knowledge Grid services.
- Domain developers, that own or develop DM components, and allow their use to third party users by publishing them using the Knowledge Grid services.

- Data managers, that owns or produce data sources useful to conduct PDKD computations and allow their use to third party users by publishing them using the Knowledge Grid services.

The publishing of data sources availability requires the specification of a security access policy, based on the security services offered both by the core grid and by the Knowledge grid (e.g., access is allowed to some Globus users, but the data are to be aggregated or filtered before moving to the users). In the rest of the section the basic Knowledge Grid services will be presented.

4.1 Knowledge Grid services

The Knowledge Grid services (layers) 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 are used to describe, develop and execute PDKD computations over the Knowledge Grid. The Knowledge Grid layers are depicted in figure 2. The figure shows layers as implemented on the top of Globus services, moreover, the knowledge grid data and metadata repositories are also shown.

4.1.1 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 tools, data management, and data visualization tools and algorithms. Moreover, this layer has to coordinate the PDKD computation executions, attempting to match the application requirements and the available grid resources. This layer comprises the following basic services:

Knowledge Directory Service (KDS). This service extending the basic Globus MDS service, 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 source 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 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 computations results (data visualization tools);
- PDKD execution plans, they are graphs describing the interaction and data flow between data sources, DM tools, visualization tools, and result storing. 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 (eXtensible Markup Language) documents and are stored in a *Knowledge Metadata Repository (KMR)*. For example, they describe the different sources of data that can be mined, 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. 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.

Because of the data management, analysis and visualization tools are usually pre-existent to the Knowledge Grid 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. Another important repository is the *Knowledge Execution Plan (KEP)* storing the execution plans over the grid of PDKD computations.

Resource allocation and execution management. These services are used to find a mapping between an execution plan and available resources, with the goal of satisfying requirements (computing power, storage, memory, database, network bandwidth and latency) and constraints. The mapping has to be effectively obtained (co)- allocating resources. After the execution plan has been started, this layer had 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.

4.1.2 High level K-grid layer

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

Data Access. 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* service. On the basis of the user requirements and constraints, the Data access service automates (or assists the user in) 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 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 can require some data filtering before extraction. These operations can usually be done after the DM tools are chosen. The extraction functions can be embedded in the data mining programs or, more usefully, can be coded and stored in a utility repository, accessible by the KDS.

Tools and algorithms access. This layer is responsible for the search, selection, downloading of data mining tools and algorithms. As before, the 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 m,ining algorithm implemented, resource requirements and constraints, and so on.

Execution plan management. 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, storing of knowledge results in the KBR. In simplest cases the user can directly describe 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 Data access and Tool access 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 Execution Plan management is a semi-automatic tool that takes the data and programs selected by the user, and generates a set of different, possible execution plans that satisfy user, data an algorithms requirements and constraints.

Execution plans are stored in a *Knowledge Execution Plan (KEP)* 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 set 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. This layer specifies how to generate, present and visualize the PDKD results (rules, associations, models, classification, etc.). Moreover, it offers the API to store in different formats these results in the Knowledge Base Repository. The result metadata, are stored in the KMR to be 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.

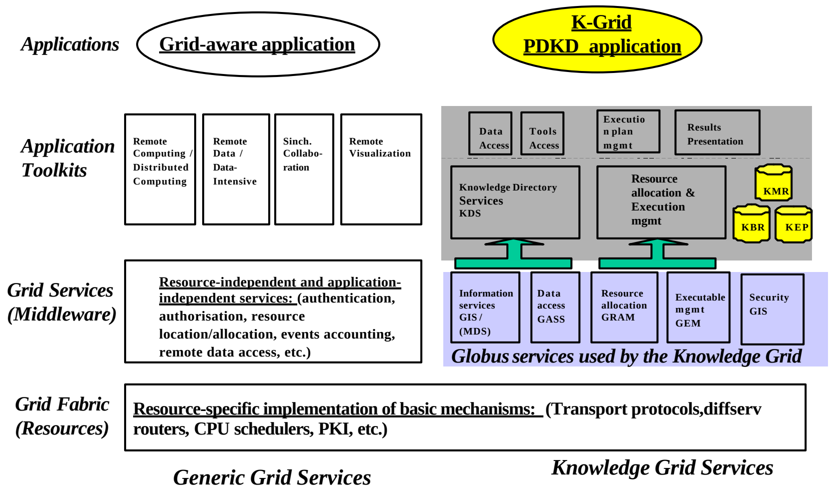

Figure 2. Knowledge Grid architecture layers.

Results presentation. This layer specifies how to generate, present and visualize the PDKD results (rules, associations, models, classification, etc.). Moreover, it offers the API to store in different formats these results in the Knowledge Base Repository. The result metadata, are stored in the KMR to be managed by the KDS.

5 Conclusion

Parallel and distributed data mining suites and computational grid technology are two critical elements of future high-performance computing environments that will enable entirely new classes of advanced applications. Their integration is a challenge whose achievements could produce many benefits in several strategic application areas. Combining the results and models emerging in these apparently far areas, it could empower PDKD and allow a new class of applications to be developed. Each of these areas could leverage on the counterpart efforts reducing costs, and domain scientists could be free to concentrate over specific problems, benefiting by this architectural independence. Furthermore, PDKD could benefits from the research efforts that are in progress in the computational grid area. This work represents the first steps in the process of studying the unification of PDKD and computational grid technologies and defining an integrating architecture for distributed data mining and knowledge discovery based on grid services. We hope that the definition of 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. Finally, we plan to construct a first implementation for this architecture so as to enable large-scale experimentation.

References

1. Chattratichat J., Darlington J., Guo Y., Hedvall S., Köler M. and Syed J., An architecture for distributed enterprise data mining. *HPCN Europe 1999*, *Lecture Notes in Computer Science*, **1593** (*Springer*, 1999) pp. 573-582.
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 Appls* (2000).
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., Globus : a metacomputing infrastructure toolkit. *International Journal of Supercomputing Applications* **11** (1997) pp. 115-128.
5. Freitas A.A. and Lavington S.H., *Mining Very Large Databases with Parallel Processing* (Kluwer Academic Publishers, 1998).
6. Grimshaw A.S., Ferrari A., Knabe F., and Humphrey M., Wide-area computing: resource sharing on a large scale. *Computer* **32** (1999) pp. 29-37.
7. Grossman R., Bailey S., Kasif S., Mon D., Ramu A. and Malhi B., The preliminary design of papyrus: a system for high performance, distributed data mining over clusters, meta-clusters and super-clusters. *International KDD'98 Conference* (1998) pp. 37-43.
8. Kargupta H., Park B., Hershberger, D. and Johnson, E., Collective data mining: a new perspective toward distributed data mining. In HILLOL KARGUPTA AND PHILIP CHAN (eds.) *Advances in Distributed and Parallel Knowledge Discovery* (AAAI Press 1999).
9. Kimm H. and Ryu T.-W., A framework for distributed knowledge discovery system over heterogeneous networks using CORBA. *KDD2000 Workshop on Distributed and Parallel Knowledge Discovery* (2000).
10. Moore R., Baru C., Marciano R., Rajasekar A. and Wan M., Data-intensive computing. In I. FOSTER AND C. KESSELMAN (eds.) *The Grid: Blueprint for a Future Computing Inf.* (Morgan Kaufmann Publishers, 1999) pp. 105-129.
11. Rana O.F., Walker D.W., Li M., Lynden S. and Ward M., PaDDMAS: parallel and distributed data mining application suite. *Proc. International Parallel and Distributed Processing Symposium (IPDPS/SPDP)*, (IEEE Computer Society Press, 2000) pp. 387-392.
12. Stolfo S.J., Prodromidis A.L., Tselepis S., Lee W., Fan D.W., Chan P.K., JAM: Java agents for meta-learning over distributed databases. *International KDD'97 Conference* (1997) pp. 74-81.