

# Clusters and Grids for Distributed and Parallel Knowledge Discovery

Mario Cannataro

ISI-CNR, Via P. Bucci, 41/c,  
87036 Rende, Italy  
[cannataro@si.deis.unical.it](mailto:cannataro@si.deis.unical.it)

**Abstract.** Parallel and Distributed Knowledge Discovery (PDKD) is emerging as a possible killer application for clusters and grids of computers. The need to process large volumes of data and the availability of parallel data mining algorithms, makes it possible to exploit the increasing computational power of clusters at low costs. On the other side, grid computing is an emerging “standard” to develop and deploy distributed, high performance applications over geographic networks, in different domains, and in particular for data intensive applications. This paper proposes an approach to integrate cluster of computers within a grid infrastructure to use them, enriched by specific data mining services, as the deployment platform for high performance distributed data mining and knowledge discovery.

## 1 Introduction

Knowledge Discovery in Database (KDD) is a general term to indicate the semi-automatic analysis of large volumes of data to find “useful” knowledge, based on the use of techniques and algorithms collectively named data mining (DM) [3]. KDD is a complex process both in time and space, so it can benefit from the use of parallel algorithms that can both speedup the process and improve the accuracy of results [2].

The unprecedented rate at which data is being produced by many fields of human activity, such as retail, banking, e-commerce, sensor data collection, physics and nuclear simulation and experimentation, and the increasing need to analyze and use them in effective way, poses new challenges to the designers and users of KDD and DM systems. These systems should be able:

- to face with large, high dimensional data sets, that very often are geographically distributed;
- to analyze different type of data, where DBMS managed data could be only a short percentage of all data (e.g., mining of unstructured or semi-structured web contents);
- to be used in an easy way, on the basis of high level DM programming models and output specification languages;
- to offer intuitive, easy to use interfaces, including advanced immersive interfaces;
- to cope with security and privacy of data.

In the latest years some KDD systems have been designed implemented on traditional parallel computing platforms to speedup their execution time and scaleup performance with respect to the problem size. However, KDD systems should make use of emerging techniques and advances in parallel and distributed computational environments, such as *cluster computing* and *grid computing*, and high performance networks as well. Very often the task of parallelizing algorithms or their porting over parallel platforms, is not trivial and not cheap, so together with speedup, important goals (perhaps long term) are architecture independence and portability. This architectural abstraction objective could be taken into account in the design of data mining algorithms together with the goal of interoperability among data mining tools. Efforts are made in the field of distributed data mining, facing with data distribution but also with networks details [7,8,9]. A language-based approach in the design of DM and KDD systems, offering a set of data mining primitives and data types, could both simplify the design of algorithms and allows for architecture transparency.

The previous problems cannot be faced only from an architectural point of view, and in fact many efforts are devoted to discover new algorithms and to integrate different scientific and theoretical knowledge into existing ones (e.g. scaling data mining solution to linear algebra or global optimization) [10,11].

This paper, addresses the main requirements of new Parallel and Distributed Knowledge Discovery (PDKD) systems [5] from an architectural point of view, with regards to their implementation and use on cluster computers and grids of geographically distributed clusters. Section 2 gives the basic concepts of cluster computers, grid computing and PDKD. Section 3 presents major architectural requirements. Finally, Section 4 contains conclusions and outlines future work.

## 2 Background

### 2.1 Clusters and Grids

Cluster computing is a new area of distributed computing, based on the availability of commodity computers, networks and middleware software [17,18]. Cluster computers are composed from tens to thousands of commodity, stand-alone computers connected by a high performance network, working together as a single computing resource. Each node of a cluster is a complete computer, hosting its own operating system and software – moreover, nodes can be heterogeneous with regard to software, hardware, and configuration. Basic cluster aspects are:

- Single System Image, e.g., the possibility to manage the systems as it was composed by a single node.
- High level communication mechanisms.
- Process allocation, migration and load balancing.
- Support to parallel and distributed computing.
- High availability.
- Heterogeneity transparency (at various levels).
- Parallel I/O and File Systems.

In some sense, clusters comprise functionality of both parallel and distributed systems. Currently, the use of cluster computers is increasing in many application domains, and efforts are made at different level, from the operating system to the middleware, to enhance their functionality and manageability. From a programmer perspective, cluster computing is a cheap and so affordable way to conduct parallel computation. In the PDKD domain, clusters are often used “only” to execute parallel DM algorithms. Recent researches address the problem of integrate clusters within grids [12,13] to obtain worldwide high performance computing environments.

In the last few years, the Web is becoming a powerful infrastructure for analyzing distributed data. More and more application deployment and communication platforms (Java, CORBA) have been developed on it, allowing new transactional and distributed services. To scale-up this trend to massive data analysis and distributed problem solving over geographical wide networks, “the grid” architecture, allowing secure and effective sharing of computing resources and networks, is emerging.

The grid term refers to middleware software allowing transparent remote access to distributed instrumentation and data, yet providing security, resource management, access management, accounting, and other services necessary for applications, users, and resource managers to operate effectively (e.g., Globus, Legion).

Grids have successfully been used especially in computational science to solve large scale experiments and simulations. Currently, grids are used to share computer, supercomputer, dedicated hardware, databases, and clusters of computers, in their various flavors. Today, a need for a *standard* grid architecture framework is emerging as a necessary step to extend the scope of utilization to different scientific domains, comprising data intensive applications. The Integrated Grid Architecture [12] is a promising effort in this directions. It comprises the following layers:

- A set of resources (possibly empty) that can be used to *grid-enable* basic computing or network resources. They can allow the implementation of basic communication functionality (e.g. Quality of Service), resource reservation, allocation and monitoring, and so on.
- A set of *grid services* (middleware), i.e. instrumentation-independent and application-independent services. Among those, authentication, authorization and accounting, remote data location, service level agreement.
- A set of *application toolkits*, providing more specific services and resources for the particular domain, e.g., parallel and distributed computing, remote data access and selection, algorithm selection (problem solving).
- At the top we find the *grid-aware* applications, i.e., applications designed on the top of grid services.

Nowadays, many international projects are developing components for the grid architecture [15, 16] and these efforts will play a key role in the future of high performance computing.

## 2.2 Parallel and Distributed Knowledge Discovery and Data Mining

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. As mentioned before, 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. Significant examples where PDKD environments could be used are:

- large organizations that need to mine their distributed data;
- mining of data owned by different organization;
- mining of distributed, geographic wide, sensor originated data;
- mining of data using different techniques in different sites to get and compare output in parallel.

The basic aspect of a PDKD systems is the distributed mining of the data, that can be performed according to the following schemes:

- *Move Data*. The data residing over remote computing nodes is selected and then transmitted to a central node (possibly a cluster) for processing.
- *Move Models*. Each node processes the data locally, and send the predictive model to another node for further processing.
- *Move Results*: Each node processes the data locally until a result is obtained, and send it to another node for further processing .

Some distributed data mining models and systems have been proposed [19, 20, 21] supporting a combination of these schemes. However, the architecture and communication infrastructure aspects are often approached from an application point of view. In the simplest cases, the data are owned by the same organization, and their formats and availability is known. In a more complex case, concerning multi-owned, heterogeneous data, some of the distributed data mining requirements could be filled by the combination of grid services and cluster computers resources.

Combining the results and models emerging in these apparently far areas, such as cluster computing, grid computing and distributed data mining, 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 efforts that are in progress to define standards for the data mining process such as the CRISP-DM process model that aims to provide a standard process structure for carrying out data mining and the predictive modeling markup language (PMML) specification for dealing with different input and output data formats [1].

### **3 Basic Architecture to Support PDKD Systems**

In this Section we give some basic directions for the integration of KDD in the framework of grid of clusters. The definition of an architecture for PDKD can be approached from different points of view. Here, using a practical approach, the goal is the definition of basic mechanisms and approaches to allow the data mining of geographically distributed data, using cluster of computers interconnected by grids, leveraging general purpose and specific grid services.

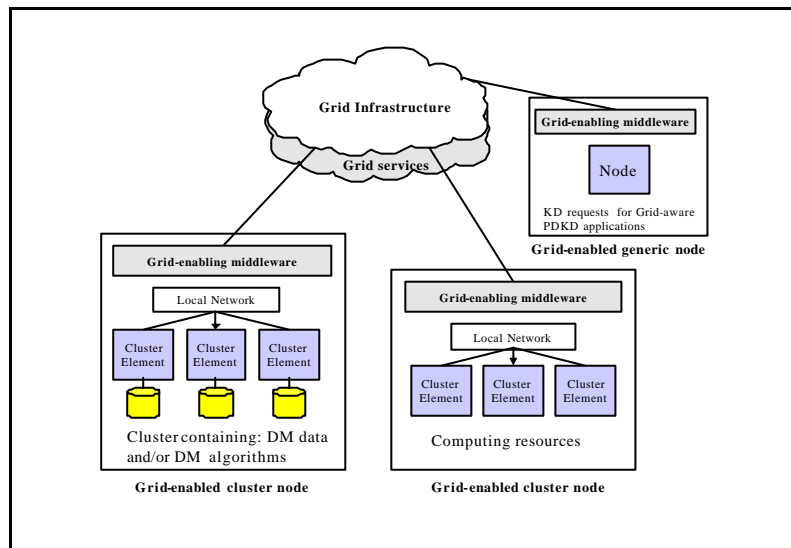
The main problems that should be faced are:

- the definition of the middleware to *grid-enable* cluster computers, yet preserving the cluster autonomy and hiding as much as possible its internal organization and management;
- the integration of specific distributed knowledge discovery requirements in the grid architecture, and finally
- the design of *grid-aware* distributed knowledge discovery systems, using basic and domain specific grid services.

The overall architecture, shown in figure 1, is composed by two main components:

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

Here the hypothesis is that the grid continues to serve pre-existing connected nodes. Each node can be a generic computer or a cluster of processors capable of parallel data mining computation. Moreover, a generic node can be involved in the PDKD environment in different ways. It can start a PDKD process, or it just owns data available for distributed data mining, or it can be charged of the execution of a data mining algorithm.



**Fig. 1.** Grid of cluster computers to solve PDKD problems

### 3.1 Grid-Enabling Clusters

Grid-enabling a cluster of computers means that:

1. the grid is aware of cluster resources (computing power, data and application) availability;
2. the cluster is able to access grid-services (i.e. to search the grid, to receive, negotiate, accept/reject and serve *grid requests*);

The mutual recognition between cluster and grid is based on a grid-enabling middleware, that should be present on the cluster computer. The goal of this software is to:

- Announce the cluster presence, such as configuration details, and its availability to share resources (data for DM analysis, algorithms for DM execution, simply computing power) and to accept external program code (e.g. Java DM agents). The degree of resources availability (type of cluster, kind of nodes, library, etc) and their security requirements, are care of the cluster, but should adhere to minimal requirements. This task can be accomplished by using directory services as LDAP or by using grid information service, as in Globus [15].
- Authenticate grid-requests, and, on the basis of service level agreement policy, to accept them - a specific PDKD requirement involves the protection of data to be mined, based on a hierarchy of authorization levels. As previously said, basic security services can be directly offered by the grid. A possible schema, using Public Key Technology, could be based on interoperable Trusted Third Parties, which implement “roaming of security services”, as in GSM wireless systems.
- Accept and serve grid-requests. To maintain cluster autonomy, its public description should suffice a requester to choose the cluster as a participant to PDKD, but should prevent access and management of internal resources (scheduling, resource reservation and so on). A resource allocation plan could be negotiated and agreed between parts (e.g., min-max service level). We want underline the importance of cluster autonomy in a heterogeneous multi-owned grid of clusters, where the cluster should be transparent when it shows data and resource availability, but should be *opaque* with respect to the way a task is internally processed.

### 3.2 Knowledge Discovery Grid-Services

The main services that the grid computing environment must offer to specifically support the knowledge discovery process are:

1. Search of input data for a given KDD goal; different level of details should be allowed (e.g., data warehouse and OLAP aggregation could be useful for). For this task a specialized search engine based on mobile agents could be provided.
2. Selection of useful data (the owner should provide some detailed data description and eventually it can provide data conversion routines).
3. Selection of Data Strategy, i.e. how the DM can choose to access and manage relevant data (major strategies are Move Data, Move Results and Move Models, to which correspond different processing strategies). Moreover, to cope with congested networks, priority-based transmission protocols could be used guaranteeing the real-time delivery of (approximate) data [22].
4. Search of useful discovery tools and algorithms among those that are available through the grid.
5. Selection of Task strategy, i.e. how the DM can choose to coordinate or not data mining algorithms over several nodes (*independent learning* vs *coordinated learning*), enabling multiple and collaborative use of different data mining models that can run on different nodes (clusters or processors).

6. Map a data mining computation to one or more nodes that autonomously schedule and start the computation. Moreover, it should be possible both to search and access distributed DM algorithms libraries, and to send portable object code to clusters not owning proper DM algorithms, using recognized platforms as CORBA, Java and enhanced systems [14].
7. Collect and present the output of the data mining process that has been executed in a parallel and distributed way. Moreover, the user should be able to analyze results using advanced visual interface, such as immersive virtual environments [8]

These basic PDKD services must be coupled and integrated with more general services that support user authorization, authentication, accounting, logging and monitoring.

### 3.3 Grid-aware PDKD systems

Grid-aware PDKD systems could be described using a language-based approach, combining DM models and algorithms specified using PMML or CRISP-DM [1], with rewriting techniques as used in CORBA. A “grid PDKD program” should:

- specify data set names and search for possible useful data available over the grid;
- describe PDKD computation (data need, application description, process structure, data access pattern, parallelism degree, etc);
- use the grid-services to dynamically match program requirements with the available resources.

At some extent, grid-aware PDKD systems are *not cluster-aware*, in the sense that they negotiate services, maintaining cluster autonomy. The use of the basic and specific grid services, allows architecture independence, but requires the development of middleware software. However, the Internet architecture is an important reference for various aspects.

An important aspect to be considered is the impact over performance of high latency and (often) low bandwidth of grids that involves both general grid applications and PDKD computations. Latency and propagation delay make synchronous communications inefficient and impractical. Other than using asynchronous communications or sending multiple messages in parallel, latency can be hidden by overlapping computation and communication as much as possible. It could be obtained designing applications able to advance computation as long as data are received (data driven).

Although the most intuitive (not easy) approach to this problem is to develop communication layers able to face with these constraints [23, 15], a more comprehensive approach could cope with them at the programming level [24]. A grid application should use communication techniques which are enough expressive to describe it and allow the grid support to dynamically select the best communication and transmission strategy, on the basis of the current situation (available resources and bandwidth). Moreover, the system should trade-off parameters such as degree of parallelism and process granularity, that affect the communication patterns. In other words, the constraints of the grid environments, other than faced at the various levels of the grid architecture, could be accounted in a more integrated way allowing the coexistence of different programming models and execution strategies. These could

be dynamically chosen at run time by the grid support. For example, a portion of an application which is allocated over a SMP cluster could use shared-memory communication techniques, whereas another portion, remotely started, could use enhanced communication techniques, improve computation and communication overlapping, or use topology-aware communications.

## 4 Conclusions and future work

Cluster computing, grid infrastructure, and data mining are three key areas of computer science and they will play a paramount role in the next years. A big challenge for the future is to integrate them for the implementation of high performance knowledge discovery systems that will improve industry and business processes. In fact, distributed mining of large, geographically dispersed, multi-owned, heterogeneous data is not more an option, but a need to solve large scale, inter-country problems.

In this paper we proposed a basic framework to integrate cluster of computers within a grid infrastructure to use them as the deployment platform for high performance distributed data mining and knowledge discovery. The paper, addressed the main architectural requirements of PDKD systems with regards to their implementation and use on cluster computers and grids of geographically distributed clusters. To give a more complete and detailed specification of a PDKD framework will be necessary to further study how each layer that composes the architecture will be implemented, which current and novel tools can be utilized for, and which detailed services should be offered. These issues constitute our future work along the direction of effective exploitation of the computing power of novel architectures for high performance knowledge discovery.

## Acknowledgements

I wish to thank the anonymous referees for the useful comments and Domenico Talia for his help on an early version of the manuscript.

## References

1. G. Piatetsky-Shapiro, The data mining Industry coming of age, *IEEE Intelligent Systems*, pp. 32-34, november/december 1999
2. A. Freitas, S. Levington, *Mining Very Large Databases with Parallel Processing*, Kluwer, 1998.
3. M.J.A. Michael, J.A. Berry, *Data Mining Techniques*, John Wiley & Sons, 1997.
4. D. Abramson, From PC Clusters to a Global Computational Grid, 1<sup>st</sup> IEEE Workshop on Cluster Computing (IWCC99), Melbourne, 1999.
5. R. Moore, Collection-Based Data Management, Workshop on Large-Scale Parallel, KDD Systems (KDD99), San Diego, CA, 1999.



6. S. Bailey, E. Creel, R. Grossman, S. Gutti, H. Sivakumar , A high performance implementation of the data space transfer protocol (DSTP), Workshop on Large-Scale Parallel, KDD Systems (KDD99), San Diego, CA, 1999.
7. U. Dayal, Large-Scale Data Mining Applications: Requirements and Architectures, Workshop on Large-Scale Parallel KDD Systems (KDD99), San Diego, CA, 1999.
8. G. Williams, Integrated Delivery of Large-Scale Data Mining Systems, Workshop on Large-Scale Parallel KDD Systems (KDD99), San Diego, CA, 1999.
9. R. Grossman, S. Kasif, R. Moore, D. Rocke, J. Ullman, Data Mining Research: Opportunities and Challenges, A report on three NFS Workshops on Mining Large, Massive and Distributed Data, available at <http://www.ncdm.uic.edu/m3d-finalreport.htm>
10. B. Grossman and Yike Guo, Communicating Data Mining: Issues and Challenges in Wide Area Distributed Data Mining, Workshop on Large-Scale Parallel KDD Systems (KDD99), San Diego, CA, 1999.
11. V. Kumar, Large-Scale Data Mining: Where is it Headed? , Workshop on Large-Scale Parallel KDD Systems (KDD99), San Diego, CA, 1999.
12. Building the Grid: An Integrated Services and Toolkit Architecture for Next-Generation Networked Applications, Working Draft, [http://www.gridforum.org/building\\_the\\_grid.htm](http://www.gridforum.org/building_the_grid.htm).
13. Foster and C. Kesselman, editors, The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, 1999.
14. Foster, G. H. Thiruvathukal, S. Tuecke, Technologies for Ubiquitous Supercomputing: A Java Interface to the Nexus Communication System, *Concurrency: Practice and Experience*, special issue edited by G. C. Fox, June 1997.
15. The Globus project, available at <http://www.globus.org>.
16. The Nimrod project, available at <http://www.dgs.monah.edu/~davida/nimrod.html>.
17. Rajkumar Buyya (editor), High Performance Cluster Computing: Architectures and Systems, Prentice Hall PTR, NJ, USA, 1999.
18. M. Baker, editor, Cluster Computing White Paper, <http://www.dcs.port.ac.uk/~mab/tfcc/WhitePaper/>.
19. R. L. Grossman, S. Kasif, D. Mon, A. Ramu and B. Malhi, The Preliminary Design of Papyrus: A System for High Performance, Distributed Data Mining over Clusters, Meta-Clusters and Super-Clusters, Proceedings of the KDD-98 Workshop on Distributed Data Mining, AAAI, 1999.
20. S. Stolfo, A. L. Prodromis, P.K. Chan, JAM: Java Agents for Meta-Learning over Distributed Databases, Proc. of the 3<sup>rd</sup> Int. Conf. On Knowledge Discovery and data Mining, AAAI Press, CA, 1997.
21. Y. Guo et al., Meta Learning for parallel Data Mining, in Proc. of the 7<sup>th</sup> Parallel Computing Workshop, 1997.
22. Albanese, M. Cannataro, P. Rullo, D. Saccà, Transmitting Datacubes over Congested Networks, Proc. of the IEEE International Conference on Coding and Transmission (ITCC2000), Las Vegas, 2000 (to appear).
23. Foster, I., A Grid-Enabled MPI: Message Passing in Heterogeneous Distributed Computing Systems, Proc. of the SC98 Conference, Orlando, USA, Nov. 7-13, 1998.
24. DiNucci, D. "The Role and Requirements of a Grid Programming Model", available at <http://www.elepar.com/GPMWG/gpm.1.ps>