A CUSTOMIZABLE MULTI-AGENT SYSTEM FOR DISTRIBUTED DATA MINING

Giuseppe Di Fatta[a] Giancarlo Fortino[b]

[a] *School of Systems Engineering, University of Reading, Whiteknights, Reading RG6 6AY, U.K., G.DiFatta@reading.ac.uk,*
[b] *DEIS – Università della Calabria, Via P. Bucci cubo 41c, 87036 Rende (CS), Italy, g.fortino@unical.it*

Abstract

We present a general Multi-Agent System framework for distributed data mining based on a Peer-to-Peer model. The framework adopts message-based asynchronous communication and a dynamic load balancing policy that is particularly suitable for irregular search algorithms. A modular design allows a separation of the general-purpose system protocols and software components from the specific data mining algorithm. While the general architecture has been implemented and successfully tested on a parallel frequent subgraph mining algorithm, several interesting issues still have to be explored. The present work will discuss them and will introduce the ongoing research efforts aimed at exploiting and leveraging the MAS for distributed data mining applications.

1 P2P-BASED MAS for Distributed Data Mining

The last decade has seen an ever increasing availability of large amounts of data in many fields of science and in many IT applications. Data mining techniques have become popular techniques, which can reveal the valuable knowledge hidden in the rough data. However, these techniques often require high performance approaches in order to cope with the overwhelming amount of data and the complexity of algorithms. An effective approach is based on distributed and parallel computing. Clusters of Workstations, Grid computing infrastructures, massively parallel systems and multi-core technology make distributed and parallel data mining a very appealing solution in many application fields. In this context, we believe that a general Distributed Data Mining (DDM) framework can enable and accelerate the deployment of practical solutions. The architecture should particularly pay attention to scalability and heterogeneity of the computational infrastructure and adopt dynamic load balancing policies.

Among the emergent paradigms for distributed computing, the Agent paradigm has demonstrated to be particularly suitable for supporting the construction of flexible and effective frameworks for distributed computation. According to the agent paradigm a distributed computation framework is designed in terms of a Multi-Agent System (MAS), i.e. a system composed of several agents, capable of reaching goals that are difficult to achieve by an individual system. In addition, MASs can manifest self-organization and complex behaviours, even when the individual strategies of all their agents are simple.

Several efforts have been devoted to enable DDM through MASs. In [3] the authors present a MAS for context-based distributed data mining. The proposed MAS architecture is client/server and is basically composed of a Miner Agent at the server side which distributes mining tasks to Local Agents which, after task completion, send the results back to the Miner Agent. Due to this simple scheme, the Miner Agent represents a performance and scalability bottleneck. In [2] the authors review four well established agent-based DDM systems (BODHI, PADMA, JAM, Papyrus). The first three systems are based on a centralized architecture in which a facilitator (or coordinator) agent interacts with the mining agents in a way similar to the one proposed in [2]. Papyrus is conversely Peer-to-Peer (P2P) oriented.

The work in [1] proposes the architecture of a customizable MAS for general-purpose distributed data mining, which exploits P2P concepts to improve performance and scalability. In particular, the MAS

architecture is organized as a flat P2P network of nodes, each of which is a MAS. Such an organization supports an efficient dynamic load balancing particularly suitable for irregular search algorithms. The MAS has been customized for the frequent subgraph mining problem for the discovery of discriminative molecular fragments. Experimental tests in [1] on real molecular compounds confirmed its effectiveness.

Further applications will allow the evaluation of the generality of the architecture. In order to customize the framework for a specific domain, the problem and the sequential application should have some general characteristics, i.e. it should be possible to partition the original problem in independent sub-problems. Typical examples are data parallel problems and problems based on a search strategy. Parallel applications with more complex communication patterns can also be adopted, but they require a greater effort to customize the framework. The simple parallel computing model at which the architecture has been applied already includes important data mining problems with broad applications, e.g. classification trees, association rule mining and in general all the frequent pattern mining problems (itemsets, sequences, trees, subgraphs). Even though the parallel algorithm is often quite straightforward (e.g. parallel backtracking), the parallel efficiency is severely limited by the irregularity of the search space. Static partitioning and load balancing cannot be applied to data mining problems and simple dynamic load balancing policies may also be ineffective for highly irregular search problems. For this reason, we believe that there is the need of a general approach for this category of problems. Nevertheless, interesting research activities could look at the generalization of the architecture for different and more complex parallel computing models.

The sequential algorithm has to be modified in order to be embedded in the system according to the interface of the *Worker* agent (see [1] for a general description of the MAS architecture). Three methods have to be implemented to adapt the sequential code.

A partitioning method (*Work Splitting*) divides a sequential task into two independent subtasks. The *Work Splitting* mechanism depends on the particular data mining applications. In applications based on a search tree, the search nodes are typically stored in a stack and the work splitting mechanism corresponds to the selection of one or more elements of the local stack to generate and donate a subtask to an idle *Worker*. A second method has to merge two partial results. The two methods, partitioning and merging, will be invoked asynchronously at any peer of the distributed system according to the agent protocols and the selected dynamic load balancing policy. These two methods are the only real effort required to the programmer in order to embed a sequential algorithm in the DDM environment. A third method will be used to invoke the execution of a sequential task. This simply corresponds to a wrapper of the sequential algorithm to comply with the *Worker* interface.

Future research efforts will focus on a more accurate formalization of the model and its simulation on very large-scale systems. Other research directions include the adoption of the approach in other application domains to verify and extend its general applicability and the introduction of advanced and intelligent services based on the MAS potentiality. For example, ongoing research efforts are looking at the dynamic and autonomous management of overlay networks of peers. Agents can dynamically organize themselves in clusters in order to aggregate computing resources for computing-intensive subtasks. This will be particularly useful for those problems whose complexity is not known in advance or it is difficult to be estimated with enough precision. At run time further resources will be dynamically aggregated, whenever necessary, for specific subtasks. This dynamic computing resource aggregation process will play orthogonally with the search space partitioning. It can take into account updated network and systems loads and configurations for an efficient partitioning of the resources. Resource partitioning is also a practical mechanism to provide greater scalability to the architecture.

References

[1] G. Di Fatta, G. Fortino. *A Customizable Multi-Agent System for Distributed Data Mining*. Proceedings of the 22nd ACM Symposium on Applied Computing (SAC 2007), Special Track on Agents, Interactions, Mobility, and Systems (AIMS), March 11 - 15, 2007, Seoul, Korea. (in press)

[2] M. Klusch, S. Lodi, G. Moro. Agent-based Distributed Data Mining: The KDEC Scheme. Intelligent Information Agents - The AgentLink Perspective. Lecture Notes in Computer Science 2586 Springer 2003.

[3] Y. Xing, M.G. Madden, J. Duggan, G. Lyons. A Multi-Agent System for Context-based Distributed Data Mining. Technical Report Number NUIG-IT-170503, Department of Information Technology, NUI, Galway, 2003.