IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB8924

Chapter V

Parallel and Distributed Data Mining through Parallel Skeletons and Distributed Objects

Massimo Coppola
University of Pisa, Italy

Marco Vanneschi
University of Pisa, Italy

ABSTRACT

We consider the application of parallel programming environments to develop portable and efficient high performance data mining (DM) tools. We first assess the need of parallel and distributed DM applications, by pointing out the problems of scalability of some mining techniques and the need to mine large, eventually geographically distributed databases. We discuss the main issues of exploiting parallel and distributed computation for DM algorithms. A high-level programming language enhances the software engineering aspects of parallel DM, and it simplifies the problems of integration with existing sequential and parallel data management systems, thus leading to programming-efficient and high-performance implementations of applications. We describe a programming environment we have implemented that is based on the parallel skeleton model, and we examine the addition of object-like interfaces toward external libraries and system software layers. This kind of abstractions will be included in the forthcoming programming environment ASSIST. In the main part of the chapter, as a proof-of-concept we describe three well-known DM algorithms, Apriori, C4.5, and DBSCAN. For each problem, we explain the sequential algorithm and a structured parallel version, which is discussed and compared to parallel solutions found in the

This chapter appears in the book, *Data Mining: Opportunities and Challenges,* edited by John Wang.

literature. We also discuss the potential gain in performance and expressiveness from the addition of external objects on the basis of the experiments we performed so far. We evaluate the approach with respect to performance results, design, and implementation considerations.

INTRODUCTION

The field of knowledge discovery in databases, or *Data Mining* (DM), has evolved in the recent past to address the problem of automatic analysis and interpretation of larger and larger amounts of data. Different methods from fields such as machine learning, statistics, and databases, just to name a few, have been applied to extract knowledge from databases of unprecedented size, resulting in severe performance and scalability issues. As a consequence, a whole new branch of research is developing that aims to exploit parallel and distributed computation in the computationally hard part of the mining task. The parallelization of DM algorithms, in order to find patterns in terabyte datasets in real-time, has to confront many combined problems and constraints, e.g., the irregular, speculative nature of most DM algorithms, data physical management, and issues typical of parallel and distributed programming, like load balancing and algorithm decomposition. Fast DM of large or distributed data sets is needed for practical applications, so the quest is not simply for parallel algorithms of theoretical interest. To efficiently support the whole knowledge discovery process, we need high-performance applications that are easy to develop, easy to migrate to different architectures, and easy to integrate with other software. We foster the use of high-level parallel programming environments to develop portable and efficient high-performance DM tools. An essential aspect of our work is the use of structured parallelism, which requires the definition of the parallel aspects of programs by means of a fixed, formal definition language. High-level parallel languages of this kind shelter the application programmer from the low-level details of parallelism exploitation, in the same way that structured sequential programming separates the complexity of hardware and firmware programming models from sequential algorithm design. Structured parallel languages are a tool to simplify and streamline the design and implementation of parallel programs.

A common issue in DM and in high-performance computing is the need to efficiently deal with a huge amount of data in complex memory hierarchies. Managing huge input data and intermediate results in the former case, and avoiding excessive amounts of communications in the latter case, highly complicate the algorithms and their implementation. Even if current research trends aim at pushing more and more of the mining task into the database management support (DBMS), and at developing massively parallel DBMS support, the problem of scaling up such support beyond the limits of shared-memory multiprocessors is yet to be solved. In our view, high-level programming environments can also provide a higher degree of encapsulation for complex data management routines, which at run-time exploit the best in-core and out-of-core techniques, or interface to existing, specialized software support for the task. The enhanced interoperability with existing software is definitely a great advantage in developing high-performance, integrated DM applications. We will sustain our perspective by showing how to apply a structured parallel programming methodology based on skeletons to DM