A Parallel Algorithm for Lattice Construction

Jean François Djoufak Kengue[1], Petko Valtchev[2], and
Clémentin Tayou Djamegni[3]

[1] Département d'informatique, Faculté de Sciences, Université de Yaoundé 1, Cameroun
[2] DIRO, Université de Montréal, Canada
[3] Laboratoire d'informatique, Faculté de Sciences, Université de Dschang, Cameroun

Abstract. The construction of the concept lattice of a context is a time consuming process. However, in many practical cases where FCA has proven to provide theoretical strength, e.g., in data mining, the volume of data to analyze is huge. This fact emphasizes the need for efficient lattice manipulations. The processing of large datasets has often been approached with parallel algorithms and some preliminary studies on parallel lattice construction exist in the literature. We propose here a novel divide-and-conquer (D&C) approach that operates by data slicing. In this paper, we present a new parallel algorithm, called DAC-ParaLaX, which borrows its main operating primitives from an existing sequential procedure and integrates them into a multi-process architecture. The algorithm has been implemented using a parallel dialect of the *C++* language and its practical performances have been compared to those of a homologue sequential algorithm.

1 Introduction

FCA [19] has already found a wide range of applications in various domains, in particular in data mining and information retrieval where the volume of data to analyze is usually huge. However, the construction of the concept lattice or even the extraction of the concept set can be a time consuming task because of the potentially exponential growth of the lattice size in the number of data items. Therefore, there is a room for the design of efficient manipulation methods for concept lattices and derived structures such as iceberg lattices and implication bases.

Utilization of parallel processing is a typical approach for dealing with large datasets [2]. It allows the work load to be divided among a set of computing units which communicate in the process of constructing the solution of the initial sequentially defined problem. To that end, these units may establish various modes of collaboration such as data sharing, remote procedure calls, message sending, etc.

Unlike previous studies of parallel lattice construction, we follow a data-centered approach. The approach amounts to slicing the input context into disjoint fragments and assigning each fragment to a different processing unit. Once the processing of a particular unit is finished, an assembly of the results for two neighbor fragments, i.e., the concept lattices of the respective subcontexts, takes place. The assembly task is repeated until a single global lattice is obtained. The approach represents a parallel homologue of an existing sequential algorithm for lattice construction of divide-and-conquer (D&C) type [17].

B. Ganter and R. Godin (Eds.): ICFCA 2005, LNAI 3403, pp. 249–264, 2005.

In this paper, we present a concrete parallel algorithm, called DAC-ParaLaX, which implements the D&C approach. DAC-ParaLaX relies on a multi-task architecture made of three different sorts of processes: concept servers, shared data servers, and concept assemblers. Each sort plays specific role in the global collaboration: while servers provide access to data and partial results, assemblers use those information chunks to create new concepts and link them in the factor lattice under construction. The entire set of processes is divided into blocks: Each block is assigned a specific fragment of the initial table whereby the aim is to construct the lattice of the fragment. Moreover, at the end of a parallel assembly round, blocks assigned to neighbor fragments are merged.

The algorithm has been implemented in a parallel dialect of the *C++* language using the *STL* and the *MPI* standard libraries. Experiments has been carried out on a cluster of 16 CPUs running Linux and related by a Myrinet-type network. A comparison of the sequential and the parallel versions of the algorithm along the performance axis is provided together with a discussion of the observed strengths and weaknesses of our D&C approach.

The paper is organized as follows. In section 2 the basic principles of the sequential lattice assembly are recalled. Section 3 describes the transition from the sequential to a parallel design of the lattice assembly approach. The current realization of that design, the DAC-ParaLaX algorithm, is presented in section 4. Finally, section 5 summarizes the results of an experimental study on the performances of the algorithm.

2 The Sequential D&C Algorithm

In its sequential version, the D&C lattice construction [17] is composed of a series of assembly tasks performed along a recursive binary split of the initial context.

2.1 Global Construction Process

The global construction process has three steps:

1. The initial context $C = (O, A, I)$ is recursively split into two parts until contexts of singleton attribute sets are obtained. The result is a (strictly) binary tree of contexts, further termed D&C-tree, where the leafs correspond to single-column tables while the context at each inner node is the *apposition*[1] of the contexts at the children nodes.
2. The lattices for each leaf context are constructed in a direct manner.
3. The lattices of all the inner contexts are constructed by assembling the lattices corresponding to children contexts. The process is itself a multi-step one: At each step, the nodes of a particular depth in the D&C-tree are processed. The lattices for inner nodes of depth $i - 1$ are obtained from the lattices for nodes at depth i. The final result is provided by the lattice of the root node.

The lattice constructing tasks involved in the above process, i.e., in steps two and three, are described with further details in the following paragraphs.

[1] Apposition is the horizontal concatenation of contexts sharing the same set of objects [5].