Using Parallel Algorithms for Searching Molecular Sequence Databases

Carla Correa Tavares dos Reis
Oswaldo Cruz Foundation
ctavares@cict.fiocruz.br

Rubem Mondaini
Federal University of Rio de Janeiro

Abstract

This work presents the development of algorithms for approximate string matching using parallel methods. It intends to do the maximum of molecular sequences comparisons per unity of time. The parallel program implementation has carried out in C on an available twenty processing nodes clustering architecture using a model of parallel programming systems, the MPI (Message-Passing Interface), which is as library of subroutines. We use one of the possible approaches to reduce the time spent on comparisons of molecular database sequences by distributing the data among processors, which achieves a linear speedup (time) and requires constant space memory per processor [6].

1. Introduction

The current sequences comparisons algorithms have been used as much to identify differences and similarities in DNA and in protein pairs alignments as to reveal molecular functions and sequences abnormalities.

The algorithms used in the comparison of molecular sequences are based on the search of optimum approximate string matching. Considering a string (pattern) for consultation, one or more database strings, and an integer variable, the algorithm seeks all the occurrences of the consultation string in the database, considering the minimum number of k differences. Edit operations (inserts, exclusions and substitutions) correct those differences, converting one string into the other, by looking for approximate maximum alignment among the strings. And as larger the number of similarities among these strings, minor is the number of existent differences among both (edit distance). Those algorithms allow efficient researches in proteins and DNA databases, focusing identical groups between the two sequences and, due to that, demanding a few comparisons per consultation. The additional methods of tables for positions comparison and the diagonal method.

The current versions of comparison sequences algorithms are extremely dependent of the processor speed and of great space of memory, so no longer correspond to the public's future expectations, due mainly to the expressive growth of database and the recent rates of 10^{25} comparisons among the residues of the involved sequences. That amount of operations demands a long processing time, considering the use of top serial processors.

The aims of this work consist on the usage of parallel methods to develop approximate string matching algorithms, attending the needs of a larger number of operations in molecular sequences analysis per unit of time. The strings alignment problems are among the most important faced by researchers in the area of computational biology. We present a parallel algorithm for this problem, reporting its speedup in comparison to its sequential version. Moreover, we are interested in showing that parallel methods can play an important role in solving molecular sequences alignments with a better running time than the others.

2. Materials and methods

To generate the parallel version of a serial approximate string matching algorithm, we have considered the possible forms of parallel programming by using a parallel architecture with twenty processing nodes Intel Xeon DP, dual processed cluster. This work was based on the distribution of the sequence databases and on the communication operations among the processors. The evaluation of algorithms performance intended to verify the spent time in tasks distribution, like the overhead in communication among the nodes and in data parallel processing. The parallel programming model employed on this work was the MPI (Message-Passing Interface), which has been widely used on parallel distributed systems.

3. Approximate string matching algorithms

The comparison of sequences is an important tool for researches in molecular biology area, which aims to determinate the molecular structure and the function of the underlying sequences.

The search of coincidences between two strings consists on the variation of the minimum difference occurrences between the pattern string characters and text string characters. Each difference is due to the occurrence of dissimilarities as much the superfluous characters (gap) presents in any of them. Therefore, these algorithms [1] [2] [3] [4] [5] were developed to solve problems of comparison of strings that consider important the conversion of a string into the other with the minimum operations cost (differences for insertions, deletions or characters substitutions).

4. The parallel algorithm version

The variations of serial algorithms are among the faster *O(mn)time* These algorithms *O(mn)* were selected to evaluate the methods of parallelism due to its simplicity and efficiency in the comparison of sequences with *k-differences*.

The strategy used in this work takes into consideration the simplest and efficient alternatives of the parallel programming systems, the parallelism of data. Where a structure of data is distributed among the processors, and the individual ones execute the same instructions for its respective portions of data. The most attractive aspects of the paradigm of parallel programming are the data distribution. MPI parallel programming model will divide the data among the processors. In case of great volumes of data, the parallelization of the data become is the most attractive solution because the easy implementation. The sequence database used in this work was the current protein sequence database SWISS-PROT (Release 46.3 of 15-03-05), which contains 176469 sequence entries, comprising 63878124 amino acids.

The only factor that could influence negatively would be the data mapping of non regular and non static structures. This issue would be a problem for any parallel program (including programs that obtain parallelism in division of program control and instructions), but mainly for programs of parallelized data because the data mapping occurs during compilation time. Other aspects that should be considered for a great data mapping are: the load balancing (the same workload for both processors, a uniform distribution of the data) and the principle of data locality (to minimize the communication problems among the processors).

5. Conclusions

For the MPI parallel models of dynamic sequences comparison algorithms O(mn) used in this work, a good performance was verified in all experiments. These experiments have considered variations in the amount and the length of the molecular sequences.

The parallel version had presented a linear Speedup, besides the execution time and the Efficiency measures were a little influenced by the overhead communication generated by the processing nodes. This behavior is explained by the fact of great amounts of data have been processed, what facilitated to hide any cost, regarding the traffic of information. The initial supposition was that, in MPI ambient, the delay generated by the change of messages among the processing nodes would be appropriate just for parallel processing of great quantities of data, however, the MPI program presented an optimum performance in any test.

It is also important to emphasize that, in the comparison between the serial processing and the parallel processing (under the operation conditions offered by MPI ambient), the parallel version always gave the best results (running and data distribution times).

6. References

[1] JOKINEN, P., TARHIO, J., UKKONEN E., A Comparison of Approximate String Matching Algorithms, *Software-Practice and Experience*, 26(12), 1439-1458, 1996.

[2] LANDAU, Gad M., VISHKIN, Uzi. Fast parallel and serial approximate string matching. *Journal of Algorithms*, v. 10, p. 157-169, 1989.

[3] LECROQ, T., Experiments on string matching in memory structures, *Software-Practice and Experience*, 28(5), 561-568, 1998.

[4] LIU, Z., DU, X., and ISHII, N., An improved adaptive string searching algorithm, *Software-Practice and Experience*, 28(2), 191-198, 1998.

[5] MICHAILIDIS P., MARGARITIS K., String Matching Algorithms, *Technical Report*, Dept. of Applied Informatics, University of Macedonia, Greek, 1999.

[6] REIS, Carla C. T. Utilização de métodos de paralelização em algoritmos de comparação de seqüências moleculares. Brasil: Federal University of Rio de Janeiro/NCE, mai. 2001.