BIOINFORMATICS APPLICATIONS NOTE

Vol. 19 no. 12 2003, pages 1585–1586
DOI: 10.1093/bioinformatics/btg192

ClustalW-MPI: ClustalW analysis using distributed and parallel computing

Kuo-Bin Li

Bioinformatics Institute, 30 Medical Drive, Singapore 117609, Republic of Singapore

Received on December 15, 2002; accepted on March 2, 2003

ABSTRACT

Summary: ClustalW is a tool for aligning multiple protein or nucleotide sequences. The alignment is achieved via three steps: pairwise alignment, guide-tree generation and progressive alignment. ClustalW-MPI is a distributed and parallel implementation of ClustalW. All three steps have been parallelized to reduce the execution time. The software uses a message-passing library called MPI (Message Passing Interface) and runs on distributed workstation clusters as well as on traditional parallel computers.

Availability: The source codes are written in ISO C and are available at http://www.bii.a-star.edu.sg/software/clustalw-mpi/. An open source implementations of MPI is available from http://www-unix.mcs.anl.gov/mpi/.

Contact: kuobin@bii.a-star.edu.sg

INTRODUCTION

In addition to the traditional massively parallel computers, distributed workstation clusters now play an important role in scientific computing perhaps due to the advent of commodity high-performance processors, low-latency/high-bandwidth networks and powerful development tools (Sterling, 2001). To fully utilize the relatively inexpensive CPU cycles available to today's scientists, we present a parallel implementation of the popular multiple sequence alignment tool, ClustalW.

ClustalW (Thompson *et al.*, 1994) can be classified as a bioinformatics application having semi-regular computational patterns (Trelles, 2001), which means the algorithms are composed of both synchronous and asynchronous steps. The first step of ClustalW involves calculating a distance matrix between each pair of sequences. This is an easy target for coarse-grained parallelization since all elements of the distance matrix are independent. The second step of ClustalW determines the topology of the progressive alignment. Finally the last step obtains the multiple alignment progressively. For the last two steps, there is no simple coarse-grained parallel solution because of the data dependency problem.

Existing versions of parallel ClustalW were all designed for shared-memory multiprocessor machines (Mikhailov *et al.*, 2001; Duzlevski, 2002). Mikhailov's version is widely used by Internet ClustalW servers. It runs only on SGI (Silicon Graphics Inc., Mountain View, CA, USA) computers. Duzlevski's version used Posix threads and can be run on symmetric multiprocessor computers. In addition, large bioinformatics centers (for example, Institut de Biologie et Chimie des Protéines, France and Institut de Génétique et de Biologie Moléculaire et Cellulaire, France) have their own parallel implementation.

ClustalW-MPI is targeted for workstation clusters with distributed memory architecture, which, compared to shared-memory machines, generally have smaller network bandwidth and longer message latency. Our implementation does not require proprietary hardware or software.

METHODS

The parallelization of the distance-matrix calculation is a problem of allocating time-independent tasks to parallel processors. We used a scheduling strategy called *fixed-size chunking* (FSC, (Hagerup, 1997)) where batches of tasks of one fixed size are to be allocated to available processors. Giving out large batches minimizes the communication overhead but may incur high processor idle time, whereas small batches reduce the idle time but may lead to high overhead.

The data of the speedup test comprises of 500 protein sequences with an average length of about 1100 amino acids. They were obtained from the BLASTP results with the query sequence (GI:21431742), a cystic fibrosis transmembrane conductance regulator. In the speedup test we allocated about 80 pairwise alignments to each processor at a time. In Figure 1, the data labeled with *pairalign* shows that efficient parallelization indeed were achieved on our test cluster. The cluster is made of eight dual-processor PCs (Pentium III, 800 MHz) and interconnected with the standard Fast Ethernet.

Once we have the distance matrix, a guide tree needs to be produced to serve as the topology of the final progressive alignment. The algorithm for generating the guide tree is the neighbor-joining method (Saitou and Nei, 1987). We

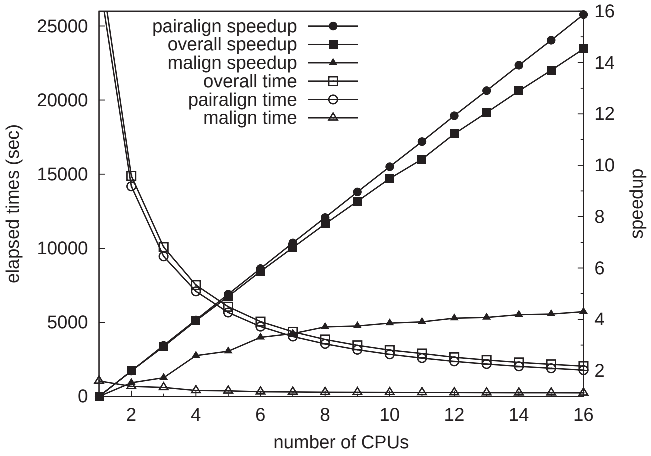

Fig. 1. Elapsed times and speedups for the ClustalW-MPI results of the 500-sequence data as a function of the number of processors. *Pairalign* is the CPU time for the calculation of pairwise distance, *malign* is the CPU time for progressive alignment.

have made slight modifications of the ClustalW codes so that the neighbor-joining tree can be done in $O(N^2)$ time while still retain the same results as the original ClustalW. For the 500-sequence test data the tree generation takes about 0.04% of the overall CPU time. In most cases the CPU time spent on this stage is less than 1% even for data containing 1000 sequences. We have a MPI implementation that parallelizes the searching of sequences having the highest divergence from all other sequences.

A mixed fine- and coarse-grained approach is used for the final progressive alignment stage. It is coarse grained in that all external nodes in the guide tree are to be aligned in parallel. The efficiency obviously depends on the topology of the tree. For well balanced guide tree, the ideal speedup can be estimated as $N/\log N$, where N is the number of nodes in the tree. We also applied the recursive parallelism paradigm (Andrews, 2000) to the linear space profile–profile alignment algorithm (Myers and Miller, 1988). Finally, the calculations of the forward and backward passes of the dynamic programming are also parallelized.

Figure 1 shows the elapsed times and the speedups obtained with ClustalW-MPI on the 500-sequence test data. The calculations of pairwise distances scale up as expected, up to 15.8 using 16 processors. For the essentially not parallelizable progressive alignment, our data shows that the speedup of 4.3 can be achieved with our mixed fine- and coarse-grained approach using 16 processors.

With the features of ClustalW-MPI, we demonstrate that it is possible to speedup lengthy multiple alignments with relatively inexpensive PC clusters.

ACKNOWLEDGEMENTS

The author would like to thank Choon Peng Chng and Atif Shahab for their suggestions in computer programming. This work was supported by the Bioinformatics Institute.

REFERENCES

Andrews,G.R. (2000) *Foundations of multithreaded, parallel, and distributed programming*. Addison-Wesley, Reading, MA.

Duzlevski,O. (2002) SMP version of ClustalW 1.82, unpublished, available from http://bioinfo.pbi.nrc.ca/clustalw-smp/.

Mikhailov,D., Cofer,H. and Gomperts,R. (2001) *Performance optimization of Clustal W: parallel Clustal W, HT Clustal, and MULTICLUSTAL*, White papers, Silicon Graphics, Mountain View, CA

Myers,E.W. and Miller,W. (1988) Optimal alignments in linear space. *Comput. Appl. Biosci.*, **4**, 11–17.

Hagerup,T. (1997) Allocating independent tasks to parallel processors: an experimental study. *J. Parallel Distrib. Comput.*, **47**, 185–197.

Saitou,N. and Nei,M. (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, **4**, 406–425.

Sterling,T. (2001) An introduction to PC clusters for high performance computing. *Int. J. High. Perform. C.*, **15**, 92–101.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.

Trelles,O. (2001) On the parallelisation of bioinformatics applications. *Brief. Bioinform.*, **2**, 181–194.