BIOINFORMATICS APPLICATIONS NOTE Vol. 26 no. 15 2010, pages 1903–1904
doi:10.1093/bioinformatics/btq304

Sequence analysis

Cloud-Coffee: implementation of a parallel consistency-based multiple alignment algorithm in the T-Coffee package and its benchmarking on the Amazon Elastic-Cloud

Paolo Di Tommaso[1], Miquel Orobitg[2], Fernando Guirado[2], Fernado Cores[2], Toni Espinosa[3] and Cedric Notredame[1,*]

[1]Centre For Genomic Regulation (Pompeu Fabra University), Carrer del Doctor Aiguader 88, 08003 Barcelona, [2]Department of Computer Science and Industrial Engineering, University of Lleida, Campus de Cappont, C. de Jaume II 69, E-25001 Lleida and [3]Department of Computer Architecture and Operating Systems, Universitad Autonoma de Barcelona, Campus UAB, Edifici Q, 08193 Bellaterra (Barcelona), Spain

Associate Editor: Burkhard Rost

ABSTRACT

Summary: We present the first parallel implementation of the T-Coffee consistency-based multiple aligner. We benchmark it on the Amazon Elastic Cloud (EC2) and show that the parallelization procedure is reasonably effective. We also conclude that for a web server with moderate usage (10K hits/month) the cloud provides a cost-effective alternative to in-house deployment.

Availability: T-Coffee is a freeware open source package available from http://www.tcoffee.org/homepage.html

Contact: cedric.notredame@crg.es

Received on March 30, 2010; revised on May 18, 2010; accepted on June 2, 2010

1 INTRODUCTION

T-Coffee was the first package featuring an implementation of the consistency-based progressive alignment algorithm (for Reviews see Edgar and Batzoglou, 2006). While the use of consistency is widely recognized as an important development in the field of multiple sequence alignment, this improvement comes at a cost. Consistency-based methods have increased CPU and memory consumption proportional to the number of sequences being processed. This overhead is incompatible with the size of the datasets being generated by next generation sequencing technologies. Heuristic solutions have been proposed (Pei *et al*., 2003) that rely on a pre-clustering of the sequences, but no effort had yet been reported to extensively parallelize consistency based algorithm and benefit from the widely available multi-core CPUs. To our knowledge this report is the first one describing a parallelization of all steps involving consistency-based alignments. The new implementation is available from version 8.00 and higher of T-Coffee. In practice, its usage is transparent to the user, with the package automatically and explicitly switching to parallel mode whenever multi-core processors are available. The procedure supports all modes of T-Coffee for aligning protein (O'Sullivan *et al*., 2004) or RNA (Wilm *et al*., 2008) sequences. It runs on all the platforms where the standard T-Coffee can be compiled and installed (Windows, MacOSX, Linux, UNIX, etc.).

*To whom correspondence should be addressed.

We benchmarked this new implementation on the Amazon Elastic Cloud (EC2) in order to estimate its efficiency and also measure the cost associated with running this type of application on a cloud-environment. Cloud computation is providing the community with an increasingly popular alternative to in-house computational resources. It is therefore becoming an important question to determine the relative cost of these new environments when comparing them with more traditional infrastructures.

2 METHOD

Parallelization is controlled with the flag *-multi_core=templates_jobs_relax_msa*, where the flag value is composed by four different keywords, each controlling a specific alignment step. These are *template*s: template selection; *jobs*: library computation; *relax*: library relaxation; *msa*: progressive alignment. Parallelization can be switched off (e.g. *-multi_core=no*) or customized (e.g. *-multi_core=jobs*).

(1) *Template selection.* In T-Coffee most advanced modes [R-Coffee, 3D-Coffee and PSI-Coffee (Kemena and Notredame, 2009)], structural templates are automatically associated with user-provided sequences. Templates can either be identified by running a BLAST against an appropriate database or produced by modeling (i.e. RNA secondary structure prediction). This simple process, iterated on the list of user provided sequences can be computationally intensive and constitutes a natural and trivial target for parallelization.

(2) *Library computation.* Library computation involves computing pairwise (or multiple) alignments using some pre-defined method. Release 8.00 of T-Coffee can use 21 external sequence and structure aligners. Parallelization is achieved by grouping the alignment tasks into a number of individual jobs equal to the number of available processors. Jobs are then submitted simultaneously, and their output merged into a single library by the master process.

(3) *Library extension.* The library extension [named relaxation in ProbCons (Do *et al*., 2005)] involves re-evaluating the score for aligning every residue pair. This operation is achieved by considering in turn every pair of sequences and subsequently updating the library. It can therefore be parallelized using the library computation strategy.

(4) *Progressive alignment.* The progressive alignment stage involves aligning sequences (or profiles) two by two, while following the order indicated by a binary guide tree. In practice, going from leaf to tip, one uses dynamic programming (Needleman and Wunsch, 1970) to align the two child sequences/profiles associated with each node. Parallelization is achieved by separately processing all the independent nodes, with optimal

Table 1. Summary of the CPU/Cost for running RV11 onto the Amazon EC2.

Instance type	Features	ECU	Scale	Speedup (%)	Total time	Total cost ($)
Small	1 CPU – 1.7 GB – 32 bit	1	1	100	2272	0.054
Extra large	4 CPU – 15 GB – 64 bit	8	6.5	81	359	0.068
Double Extra large	4 CPU – 32 GB – 64 bit	13	9.0	88	257	0.086
Quad. Extra large	8 CPU – 64 GB – 64 bit	26	12.4	47	200	0.133
Medium	2 CPU – 1.7 GB – 32 bit	5	3.8	76	607	0.029
Extra large	8 CPU – 7 GB – 64 bit	20	9.2	46	266	0.050

Instance is the name given by Amazon to a CPU configuration, an ECU is a virtual core. Scale: speedup relative to the 1 ECU unit expressed as the fraction of expected speedup achieved. Total time: the total amount of time and cost is the amount charged by Amazon for a corresponding job.

speedup achieved when dealing with perfectly balanced trees. Whenever this is not the case, recent results suggest that unbalanced trees can be re-balanced (as a pre-processing) without any significant loss in accuracy (Orbitg *et al*., 2009). The computation of the guide tree itself has been left out from the parallelization, since in practice it is not a limiting factor, given the size of the instances T-Coffee can deal with and given the reliance on guide trees obtained through fast k-tup comparisons (Edgar, 2004).

3 RESULTS

The parallelization procedure presented here should only be considered an initial framework in which more sophisticated schemes could be implemented. It relies entirely on the UNIX *fork()* function, with child and parent processes communicating via temporary files. For benchmarking purpose, this new implementation has been deployed onto the Amazon Elastic Compute Cloud (EC2) service (http://aws.amazon.com/ec2/). The EC2 is a resource provided by Amazon that makes it possible to run software on virtual machines of various sizes. The Amazon machine image (AMI) used to perform this benchmark is available on: http://s3.amazonaws.com/tcoffee-machine-x32-1.0.1/ (32 bit version) and http://s3.amazonaws.com/tcoffee-machine-x64-1.0.1/ (64 bit version). We also used this benchmark as an opportunity to assess Amazon billing procedure with respect to MSA computation. The way Amazon charges for CPU is complex, with the final bill depending on a combination of I/O, CPU consumption and virtual machines specifications. Benchmarking was done on the RV11 subset of BaliBase version 3 (Thompson *et al*., 2005). Results (Table 1) indicate that the new implementation is effectively taking advantage of the multi-core implementation. For instance, the medium instance (5 EC2 Computation Units, ECU) runs 3.7 times faster than its small counterpart (1 ECU). This speedup is observed consistently across all instances, although the relative speedup decreases with the largest instances, probably owing to the increased I/O limiting the speedup. For instance, with 26 ECU, one only obtains 47% of the expected speedup.

In practice, the most widely used public T-Coffee server is the one running on the Vital-IT platforms (http://tcoffee.vital-it.ch). It delivers roughly 7000 jobs a month, with an average size close to that of the benchmarks used here (maximum number of sequences 50). If replicated on the EC2, using a dedicated Hi-CPU extra large instance, the cost would be roughly 1000 US$ per year, a figure reasonably competitive with the cost of locally installed/maintained hardware. This suggests that EC2 based deployment of bioinformatics services is a cost effective alternative to in-house deployment of similar services, especially in the case of CPU intensive analysis such as MSA computation.

ACKNOWLEDGEMENTS

The authors would like to thank Carsten Kemena, Ionas Erb, Chang Jia Ming, Jean-François Taly.

Funding: Centro de Regulación Genómica (CRG to C.N., P.T., C.K.); Plan Nacional (BFU2008-00419); Lleida University, the Spanish ministry of education (TIN2008-05913 to M.O., F.G., F.C.); the Consolider Project (CSD 2007-00050); Super-computacion y e-Cienia (SYEC).

Conflict of Interest: none declared.

REFERENCES

Do,C.B. *et al*. (2005) ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Res.*, **15**, 330–340.

Edgar,R.C. (2004) Local homology recognition and distance measures in linear time using compressed amino acid alphabets. *Nucleic Acids Res.*, **32**, 380–385.

Edgar,R.C. and Batzoglou, S. (2006) Multiple sequence alignment. *Curr. Opin. Struct. Biol.*, **16**, 368–373.

Kemena,C. and Notredame,C. (2009) Upcoming challenges for multiple sequence alignment methods in the high-throughput era. *Bioinformatics*, **25**, 2455–2465.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

O'Sullivan,O. *et al*. (2004) 3DCoffee: combining protein sequences and structures within multiple sequence alignments. *J. Mol. Biol.*, **340**, 385–395.

Orbitg,M. *et al*. (2009) Exploiting parallelism on progressive alignment methods. *J. Supercomp.*, **1**, 1–9.

Pei,J. *et al*. (2003) PCMA: fast and accurate multiple sequence alignment based on profile consistency. *Bioinformatics*, **19**, 427–428.

Thompson,J.D. *et al*. (2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins*, **61**, 127–136.

Wilm,A. *et al*. (2008) R-Coffee: a method for multiple alignment of non-coding RNA. *Nucleic Acids Res.*, **36**, e52.