Wrapping up BLAST and other applications for use on Unix clusters

Karsten Hokamp, Denis C. Shields, Kenneth H. Wolfe and Daniel R. Caffrey

Department of Genetics, University of Dublin, Trinity College, Dublin 2, Ireland and Department of Clinical Pharmacology, Royal College of Surgeons in Ireland, 123 Stephen’s Green, Dublin 2, Ireland

Received on May 28, 2002; revised on July 18, 2002; accepted on June 6, 2002

ABSTRACT

Summary: We have developed two programs that speed up common bioinformatic applications by spreading them across a UNIX cluster. (1) BLAST.pm, a new module for the ‘MOLLUSC’ package. (2) WRAPID, a simple tool for parallelizing large numbers of small instances of programs such as BLAST, FASTA, and CLUSTALW.

Availability: The packages were developed in Perl on a 20-node Linux cluster and are provided together with a configuration script and documentation. They can be freely downloaded from http://wolfe.gen.tcd.ie/wrapper.

Contact: daniel_rcaffrey@cambridge.pfizer.com; karsten@oscar.gen.tcd.ie

INTRODUCTION

Inexpensive systems such as Beowulf clusters, have become increasingly popular in both the commercial and academic sectors of the bioinformatics community. Clusters typically consist of a master machine/node that distributes the bioinformatic application amongst the other nodes (slaves/clients). These often require installation of special software on each node or modification of the bioinformatics programs. A simpler solution consists of so-called wrappers like MOLLUSC (Jongeneel et al., 1997), that improve search times for SSEARCH (Pearson, 2000), pfscan, and pfsearch (Luthy et al., 1994). This involves the databases being split up into smaller portions that are distributed amongst the clients. On each node, a portion of the database is searched and the master merges the results into a single file. MOLLUSC allows incorporation of modules for other search programs. Here we describe: (1) BLAST.pm, a module that allows MOLLUSC to run BLAST (Altschul et al., 1997) on a UNIX cluster. (2) WRAPID, an independent tool designed for processing large numbers of small applications.

SYSTEMS AND METHODS

BLAST.pm

The success of the BLAST package is partially due to its short search times, relative to programs such as FASTA or SSEARCH (Brenner et al., 1998). However, under heavy load, parallelization would further improve its performance. Also, memory should be big enough to store the entire database or be split into smaller volumes. Thus, dividing a large database into smaller portions that can be individually searched with BLAST on multiple nodes would be advantageous. However, BLAST E-values are dependent on database size, query size, and letter composition. Specifically, the E-values are calculated from an effective search space. This is the product of the effective lengths of the query and the database. Figure 1 shows that the effective search space has a near-linear relationship with the search space. Using these linear regression equations, the BLAST.pm package estimates the effective search space and forces this value using the ‘-Y’ option of BLAST for each client. The use of the estimated effective search space yields similar results. Using a standard set of sequences, we have found that E-values will be almost identical to those produced when searching against the entire, unpartitioned database (see documentation). The BLAST.pm Perl module is called by the MOLLUSC package through a command-line that closely resembles BLAST

mollusc blastall -p blastp -a 4 -d nr.aa -i /fullpath/query -o /fullpath/result -T

The majority of command line options are allowed, but certain options are either forced or removed to facilitate formatting or generation of statistics (see documentation).

WRAPID

Where the whole process of a bioinformatic application fits into the memory of each client, a small tool called WRAPID (Wrapper for RApid Parallelized Instruction Dispatching) can be used. It speeds up large numbers of small jobs, such as comparing all protein sequences of
an organism against one another or aligning a batch of sequences, through parallelization on a cluster. The installation effort was kept at a minimum and the application range as broad as possible. The script was written in Perl and only needs to be run on the initiating node—no installation is necessary on the other computers. The only client requirements comprise remote login, a shared directory holding clients’ input files and results, and the availability of Perl. Once this is given, many common bioinformatics programs can be easily parallelized through WRAPID. Its usage involves a simple prepaning to a valid execution statement, e.g.

```
wrapid.pl ssearch -Q -b 500 -d 0 -H -m 9 -p -S -E 1 /fullpath/query /fullpath/database
```

```
wrapid.pl clustalw /fullpath/file_of_filenames
```

Additional command line options or configuration files can be used to adjust the process to different set ups, changing work loads and varying requirements of different applications. So far the wrapper has been successfully tested with BLAST and FASTA, as well as with MPSRCH and ClustalW. WRAPID also provides some extra features, e.g. comprehensive checks for prerequisites are carried out on each node, dynamic assignment of nodes, a load balancing mechanism, and job completion is checked.

**CONCLUSION**

Parallelization of bioinformatic jobs offers improved performance and scalability. Although communication overheads are likely to affect performance as the cluster becomes larger, we observe up to 19-fold improvement in search times on a 20 node cluster (see documentation). When the database is too big for a single node to search, BLAST should be invoked through BLAST.pm. In cases where a single user has many queries and the entire database can be searched efficiently by a single node, WRAPID should be used. WRAPID can also be viewed as a batch queueing system for many applications, but differs from BEOBLAST (Grant et al., 2002) in that it is best suited for the single user with multiple queries. WRAPID differs from other queueing systems such as LSF (Zhou, 1992) or PBS (Henderson, 1995) in having an installation process that is simple enough to be carried out by a normal user. However, it will not have the advanced job management functions of LSF and PBS.

**ACKNOWLEDGEMENTS**

This work was aided by HRB Ireland, HEA Ireland, a Trinity College Dublin High Performance Computing studentship, and Science Foundation Ireland. Thanks to ISREC Bioinformatics for helpful discussions.

**REFERENCES**


