Searching Bioinformatic Sequence Databases using UM-BLAST—A Wrapper for High-Performance BLASTs

Xue Wu, Chau-Wen Tseng [1]

Keywords: Bioinformatics, sequence comparison, parallel processing, cluster computing

1 Introduction

BLAST [1] is the most widely used search tool for screening large bioinformatic sequence databases, and accounts for a large portion of the computation performed in bioinformatics. As a result, researchers have developed many versions of BLAST to improve its performance. Among them, NCBI BLAST [2] and WU BLAST [3] support parallel processing on a variety of SMP computer architecture. mpiBLAST [4] is the most recent version of parallel BLAST for distributed memory architecture. While BLAST++ [5] tried to improve the throughput of BLAST by batching the query sequences and exploiting sharing of results on common subsequences of queries. In addition, several organizations (TurboWorx Inc., Paracel Inc. and SGI Inc.) and researchers [6, 7, 8] have also developed their own version of high performance BLASTs.

However, our experiments show that no single version of BLAST is able to achieve the best performance given variations in sequence database size, query batch size, and query sequence length. We find mpiBLAST is best at exploiting database partitioning over multiple nodes to keep large databases in memory. BLAST++ is best at amortizing search costs for multiple batched queries. Threaded BLAST is best at reducing search costs for very long single queries. Based on our evaluation, we design UM-BLAST, a wrapper capable of selecting the proper combination of threaded BLAST, BLAST++, and mpiBLAST to achieve good performance over a range of search parameters.

2 UM-BLAST Wrapper

UM-BLAST is designed to select and invoke the most efficient version of BLAST for the database size, batch size, and query length selected. The basic algorithm for UM-BLAST is as follows:

1. Pre-partition large sequence databases for mpiBLAST so that each partition fits in memory for a single node
2. For each batched sequence search query
   (a) If database is too large to fit in memory on a single node, use mpiBLAST
   (b) Else
      i. If (batch size $> B$) and (query length $< L$) use replicated BLAST++
      ii. Else use replicated threaded-BLAST
      iii. Combine outputs for batch from replicated BLASTs

[1] Computer Science Department, University of Maryland at College Park. E-mail: {wu,tseng}@cs.umd.edu

3 Performance Measurement and Results

We measured the performance of threaded BLAST, mpiBLAST, BLAST++ and UM-BLAST on a Linux PC cluster with 10 worker nodes (each with dual AMD Athlon 1.6 GHz processors and 1G bytes of memory). Detailed results are shown in Figure 1, 2 and Table 1.

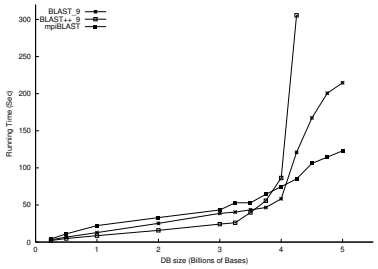

Figure 1: Impact of DB Size on BLASTs' Parallel Performance

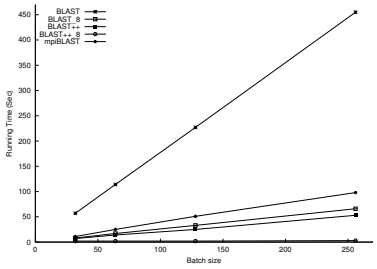

Figure 2: Impact of Batch Size on BLASTs' Performance

DB Type	Example DB	Batch Size	Query Length	UM-BLAST's Choice	Time(s)	Speedup
Large	4.5G	any	any	mpiBLAST	85.1	20 - 30
Medium	1.0G	256	3250	replicated threaded BLAST	438	1.4 - 100+
	1.0G	1	16384	mpiBLAST	6	5 - 6
	250M	256	64	replicated BLAST++	3	17.7 - 151.7
Small	13M	1024	64	BLAST++	2	46 - 123.5

Table 1: UM-BLAST's Performance

4 Conclusions

The results show that our UM-BLAST, a wrapper for high-performance BLASTs, can automatically select the appropriate version and configuration of BLASTs based on the target database size, query batch size and query sequence length. We feel UM-BLAST will be very useful to bioinformatics researchers seeting up their own BLAST servers.

References

[1] Altschul, SF, Gish, Miller, W, Myers, EW and DJ Lipman, A basic local alignment search tool, *Journal of Molecular Biology (1990)* 215:403-410

[2] NCBI BLAST, *http://www.ncbi.nih.gov/BLAST/*

[3] WU-BLAST, *http://blast.wustl.edu/blast/README.html*

[4] Aaron Darling, Lucas Carey, Wu-chun Feng, The Design, Implementation, and Evaluation of mpiBLAST, *ClusterWorld 2003*

[5] H. Wang, T.H. Ong, B.C. Ooi, K.L. Tan, BLAST++: A Tool for BLASTing Queries in Batches, *Proceedings of the 1st Asia-Pacific Bioinformatics Conference*, Feburary, 2003

[6] J. D. Grant, R. L. Dunbrack, F. J. Manion, and M. F. Ochs, BeoBLAST: distributed BLAST and PSI-BLAST on a Beowulf cluster, *Bioinformatics* Vol 18, pp. 765-766

[7] Michel Dumontier1,2 and Christopher W. V. Hogue, NBLAST: a cluster variant of BLAST for NxN comparisons, *BMC Bioinformatics*, 2002; 3(1):13

[8] Akira Naruse, Naoki Nishinomiya, Hi-per BLAST: High Performance BLAST on PC Cluster System, *Genome Informatics*, 2002; Vol 13, pp 254-255