POSTER PRESENTATION Open Access

Experimenting with database segmentation size vs time performance for mpiBLAST on an IBM HS21 blade cluster

Daniel Harris[1], Jerzy W Jaromczyk[1*], Christopher L Schardl[2]

From UT-ORNL-KBRIN Bioinformatics Summit 2010
Cadiz, KY, USA. 19-21 March 2010

Background

Large-scale genomic projects such as the Epichloë festucae Genome Project require regular use of bioinformatic tools. When using BLAST in conjunction with larger databases, processing complex sequences often uses substantial computation time. Parallelization is considered a standard method of curbing extensive computing requirements and parallel implementations of BLAST, such as mpiBLAST, are freely available.

Materials and methods

In this experiment, the implementation segments a database into smaller databases so that BLAST queries can be more effectively performed in parallel on smaller database segments. Since there are overhead costs from distributing tasks and merging the results from each parallel run, we investigate how the usefulness of database segmentation changes as the size and the number of the database segments change. When segmentation curbs time-performance, we ask the question: “How many segments will yield the best performance or will adding processors always help?” Specifically, we consider three different times: a one-time preprocessing (segmentation of database), queue wait-time, and CPU-time. We conducted experiments to monitor time-performance as the number of database segments vary on an IBM HS21 blade cluster running mpiBLAST against fungal protein sequences from the Epichloë festucae Genome Project. The cluster has 340 computer nodes (1,360 cores, 12.8 Teraflops) whose resources are shared with other researchers and are controlled through the SLURM

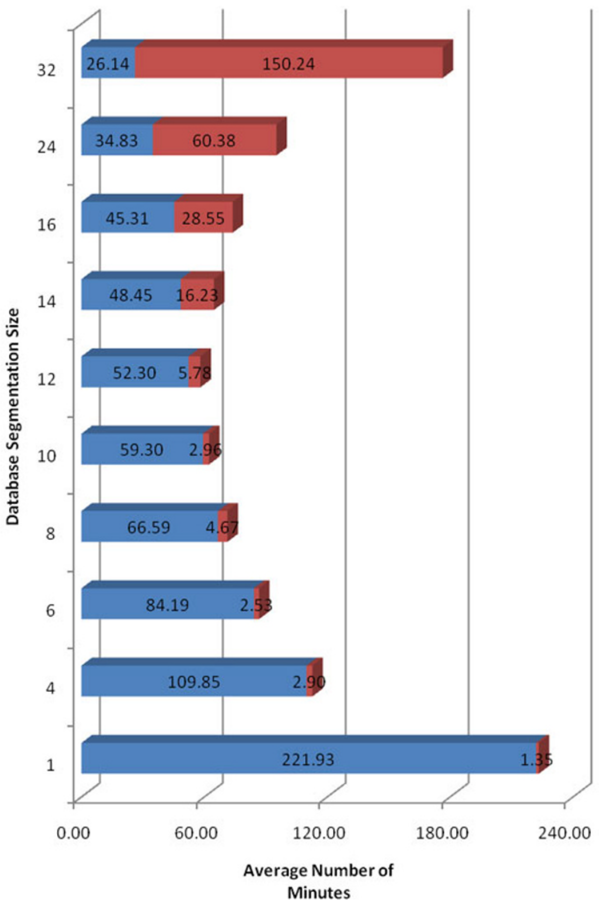

Figure 1 CPU-time and wait-time composite. Figure 1 shows the summation of CPU-time (blue) and queue wait-time (red) in minutes as the number of nodes and database segments increase.

* Correspondence: jurek@cs.uky.edu
[1]Department of Computer Science, University of Kentucky, Lexington, KY 40506, USA

batch-job resource-manager and scheduled through the Moab batch-job scheduler.

Results and conclusion

We observe that the shared nature of computing resources with multiple users has a direct consequence when determining what database segmentation configuration to use in practice. For example, in our experiment, the average CPU-time (in minutes) for one node is 221.93, for twelve nodes is 52.30, and for 32 nodes is 26.1; the average queue wait-time (in minutes) for one node is 1.35, for twelve nodes is 5.78, and for 32 nodes is 150.24 (Figure 1). Therefore, the composite time (in minutes) for one node is 223.28, for twelve nodes is 58.08, and for 32 nodes is 176.38 (Figure 1). Thus, the composite time for twelve nodes is the shortest for our experiment. Additionally, the preprocessing (segmenting database) required a fixed one-time cost of approximately three days. The collected data allows us to execute efficient planning and scheduling of our mpiBLAST experiments in an environment with uncontrollable variables such as queue wait-time. This work is based upon research supported by the NSF under Grant No. 0814194 and NIH Research Project Grant Program (R01) from the Joint DMS/BIO/NIGMS Math/Bio Program under Grant No. 1R01GM086888-01.

Author details
[1]Department of Computer Science, University of Kentucky, Lexington, KY 40506, USA. [2]Department of Plant Pathology, University of Kentucky, Lexington, KY 40506, USA.

Published: 23 July 2010

doi:10.1186/1471-2105-11-S4-P9
Cite this article as: Harris *et al.*: Experimenting with database segmentation size vs time performance for mpiBLAST on an IBM HS21 blade cluster. *BMC Bioinformatics* 2010 **11**(Suppl 4):P9.