Seeking Super-Linearity Speedup in birgHPC Environment

Pujianto Yugopuspito[1], Henry Sumargo[2]
Research and Development Computer Laboratory[1], Informatics Department[2]
Universitas Pelita Harapan
pujianto.yugopuspito@uph.edu[1], henrysumargo90@gmail.com[2]
Tel : +62-8174969846 Fax : +62-21-5460910

Abstrak

Tujuan dari penelitian ini adalah menunjukkan apakah super-linear speedup dapat atau tidak dapat dicapai pada komputer biasa dalam suatu laboratorium komputer di sebuah universitas dengan menggunakan cluster yang tidak menggunakan banyak konfigurasi berdasarkan pada system operasi pre-boot execution environment (PXE). Super-linear speedup adalah sebuah kasus speedup dimana waktu eksekusi dari sekuensial dibagi dengan eksekusi parallel yang nilainya lebih dari jumlah compute node. Yaitu dengan membuat lingkungan cluster High Performance Computing berdasar pada Bioinformatics Research Group High Performance Computing (birgHPC) sebuah perubahan dari PelicanHPC. Lingkungan ini terbatas pada 32 buah node Pentium4 3.2 GHz CPU, 2.5 GB RAM, 80 GB hard-disk. Kasus Basic Local Alignment Search Tool (BLAST) dan Message Passing Interface BLAST (mpiBLAST) dikerjakan, yaitu mencari kesamaan urutan basis data 8.9 GB yang tidak terformat dibandingkan dengan beberapa query genome dengan beberapa konfigurasi lingkungan. Tantangannya adalah untuk menanggulangi 7 jam lebih waktu komputasi dari mesin tunggal dari BLAST. Dari pengujian dapat disimpulkan bahwa super-linearity adalah sebuah kasus yang jarang terjadi pada sebuah lingkungan yang terbatas dan sederhana. Walaupun dengan mengubah lingkungan pengujian dengan menurunkan ukuran dari RAM compute node dan menaikkan ukuran urutan basis data yang dicari, sehingga efek cache dapat dirasakan secara lebih signifikan.

Kata kunci: *Super-Linearity, High Performance Computing, Bioinformatics, mpiBLAST*

Abstract

The aim of this research is to show whether a super-linearity speedup can be or cannot be achieved within normal personal computer in university laboratory with an effort-less cluster, based on a pre-boot execution environment (PXE) operating system. Super-linearity is a case of speedup where the time execution ratio of sequential versus parallel is more than the number of computing node. We create the High Performance Computing cluster environment based on Bioinformatics Research Group High Performance Computing (birgHPC), a remake of PelicanHPC. The environment is limited to 32 regular personal computers as PXE computing nodes, Pentium4 3.2GHz CPU. The head-node is equipped in Pentium4 3.2 GHz CPU, 2.5 GB RAM, 80 GB hard-disk. Basic Local Search Tool (BLAST) and Message Passing Interface BLAST (mpiBLAST) problems are conducted, searching sequences in 8.9GB unformatted database, within several environment settings. The challenge is to beat 7+ hours of computing time in a single machine of BLAST problem. It is concluded that super-linearity is a rare case that can be shown in simple limited environment. Even the experiment environment is tweaked: downsize the computing node RAM and upsize the searching sequence such that the cache effect can be significant.

Keywords*: Super-Linearity, High Performance Computing, Bioinformatics, mpiBLAST*

1. Introduction

High performance computing (HPC)solved many massive calculation problems,and needed for solving one of bioinformatics problems i.e. Deoxyribonucleic Acid (DNA) local alignment sequence matching. A new sequence genome can be identified within aregistered DNA sequence.Lately, the size of this registered sequence database increase dramatically. One important problem is DNA alignment using Basic Local Search Tool (BLAST), [1]. BLAST is an embarrassingly parallel problem. This research implements a parallel version of BLAST, mpiBLAST [2] whicha magical combination of BLAST in high performance computing of message passing interface (MPI).

This research uses an instant HPC, birgHPC [3]. It is a remix of PelicanHPC Linux Live CD. The idea is to easily create an instant high performance computing system on-the-fly on top of a regular university laboratory. This technique employs networked booting, pre-boot execution environment (PXE). The created HPC cluster is limited to a single user with network file system enable. The node installation is a kind of booting operating system via a computer network, without changing the real installed operating system on that personal computer. So it will be a perfect utilization of idle computing power for university laboratory personal computers.

This research argues the existence of super-linearity speedup within birgHPC environment. Super-linearity is a case of speedup where the time execution ratio of sequential versus parallel is more than the number of computing node. Previous research [2] showed that super-linearity easily achieved in a big 240 nodes cluster, and start from 4 compute nodes, each node has a 667 MHz Transmeta TM5600, 640MB RAM and a 20GB hard disk. That research used 1.2GB binary database, about two times of the RAM. While Chew et al. [3] did not clearly state this phenomenon in instant deployment of birgHPC. This research is a further investigation for the existence of super-linearity speedup within birgHPC environment. The speedup is calculated by comparing the query time of single node BLASTversus two or more nodes mpiBLAST. Several setting of environments are implemented to seek the affected factors, i.e. database segmentation, number of nodes and RAM.

2. Research Method

This research is conducted as a laboratory experimental.The illustration of laboratory environment is shown in Figure 1. Some identical personal computers are used, with specifications: Intel Desktop Board D955XBK motherboard, Pentium4 CPU 3.2 GHz, 2.5 GB RAM and 80 GB HDD.All computers are connected to 10/100 Mbps switch. First experiment was a single node computation using sequential calculation based on original BLAST, and 8.9GB env_ntunformatted gnome database. This database to RAM ratio was more than three times.The query genomes are 78 KB and 816 KB of Saccaromyces_cerevisae_uid128_NC_001133, and_NC_001134 sequences. Then some parallel computations were conducted in mpiBLAST. Experiment variations were the number of computing nodes and database fragments: 2, 4, 8, 16 and 32, while the RAM variations were 2.5GB and 512MB.

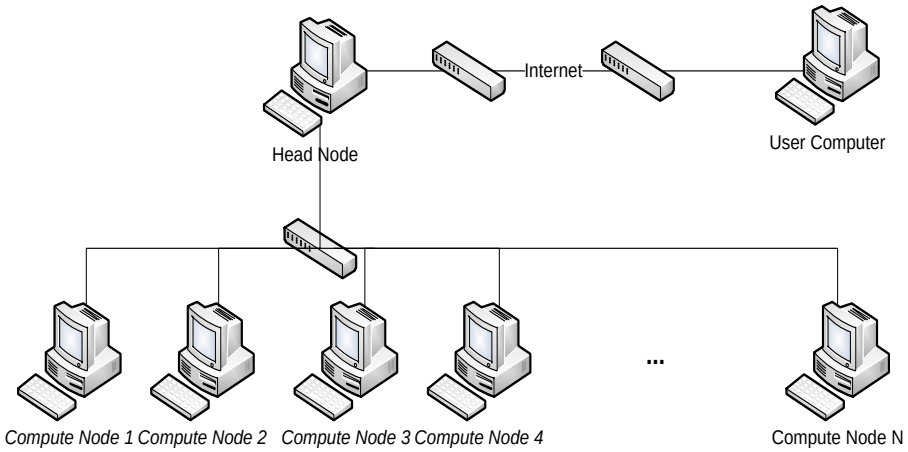

Figure 1. Laboratory environment setup.

mpiBLAST is first released in 2003 and is an open source implementation of BLAST using MPI as parallel interface. It works in a wide range of clusters and seems to beat its competitor in term of performance. It implements a fragmented approach, asdepicted in Figure 2. The fragment are pre-computed statically and then distributed to the slaves, compute nodes. There is one scheduler node that reads and broadcasts the

queries. mpiBLAST reduce disk access by making the fragment small enough that they fit in main memory of the workers nodes. The real data in birgHPC remains in the head node, the compute nodes have network file system (NFS) as their collaborative shared disk.

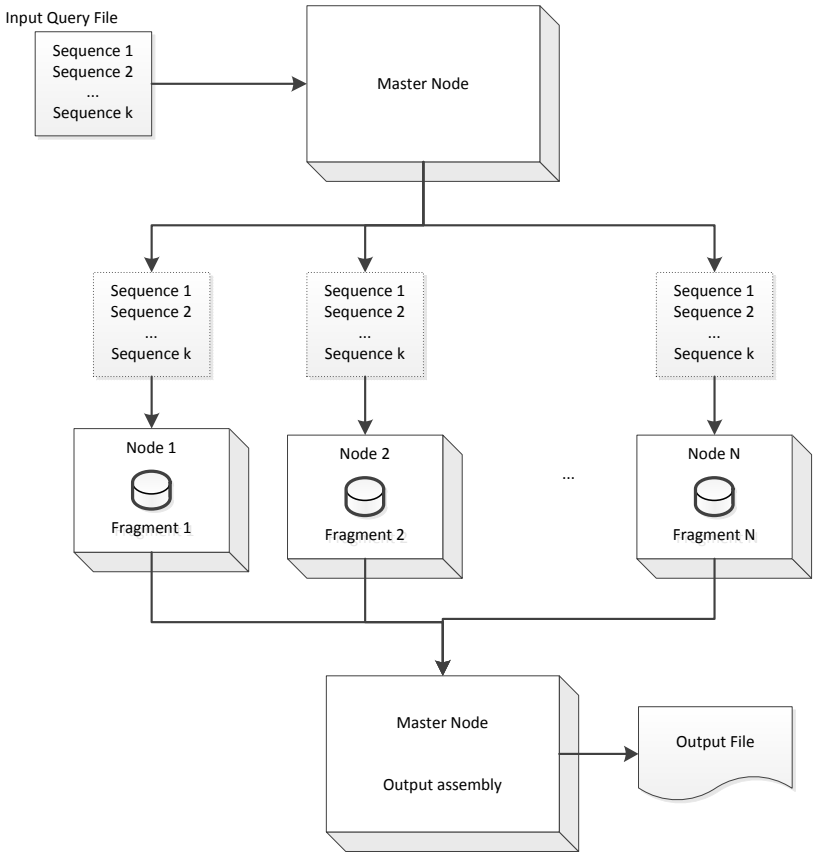

Figure 2. Fragmented approach.

3. Results and Analysis

The first experiment is to create the base line calculation, BLAST on a single processor and non-parallel. Start database preparation, the database should be in a binary file before conducting the query of DNA alignment.formatdb command is used to convert ASCII files from env_nt 8.9 GB unformatted sequence database FASTA format to binary files so that can be executed using BLAST or mpiBLAST. As a result the size of the database became 5.3 GB binary file. The query is conducted by executing, blastall –p blastn -d env_nt -1 NC_001133.fna –a 1. This means to execute a query in a nucleotide database for a nucleotide query type in a single threaded process. Table 1 shown the result of this experiment. The time formatting database using formatdb command for 2.5 GB RAM took 1537 seconds, average from ten executions. On the other settingof 512 MB RAM formatdb command took 1938 seconds, average from ten executions. With ten times doing the same job, average time execution is 2429 seconds with standard deviation 26. Smaller 2.5 GB RAM size compares to 5.3 GB sequence database caused long execution time using BLAST. After database preparation the querying of DNA alignment can be done. The query is a search of comparing env_nt FASTA format database against 78 KB of Saccaromyces_cerevisae_uid128_NC_001133 query genome.In 512 MB RAM, 2441 seconds spent. Using 2.5 GB RAM, 2429 seconds spent. The result shown that adding more RAM size did not give significant impact for the execution time result. It seems that within less RAM size, heavy disk I/O happened with a big portion of memory swapped.

Table 1. The first experiment of sequential BLAST at a single machine.

RAM size	formatdb time (s)	Query time (s)	Total (s)
2.49 GB RAM	1537	2429	3966
512 MB RAM	1938	2441	4379

The second experiment using mpiBLAST were conducted on 2, 4, 8, 16, 24, and 32 compute nodes. Sequence database used was the same as the BLAST experiment, env_nt 8.9 GB FASTA format uncompressed compared against 78 KB of Saccaromyces_cerevisae_uid128_NC_001133 query genome. As the BLAST experiment, mpiBLAST required a database preparation by using mpiformatdb command.It has the same function as formatdb, but mpiformatdbdivided the database segmented into many fragments. After using mpiformatdb, the total size of the database fragments decreased to 5.3 GB binary file. Each fragment will be sent to a compute node. Ideally for every thread exist, one thread search one fragment.After the database ready, the parallel query executions were conducted. This experiment showed that the number of fragments did affect execution time from 2, 4, 8, 16, 32, 48 and 64 fragments.Increasingnumber of fragments until 32 fragments will increase the performance by mean of decreasing execution time, butin the setting of 48 and 64 fragments, it increased execution time. Figure 3 shows the **execution time** result. It seems that a saturated fragment was happened for a certain point.It causeda high cost of input/output transaction for segmenting database.

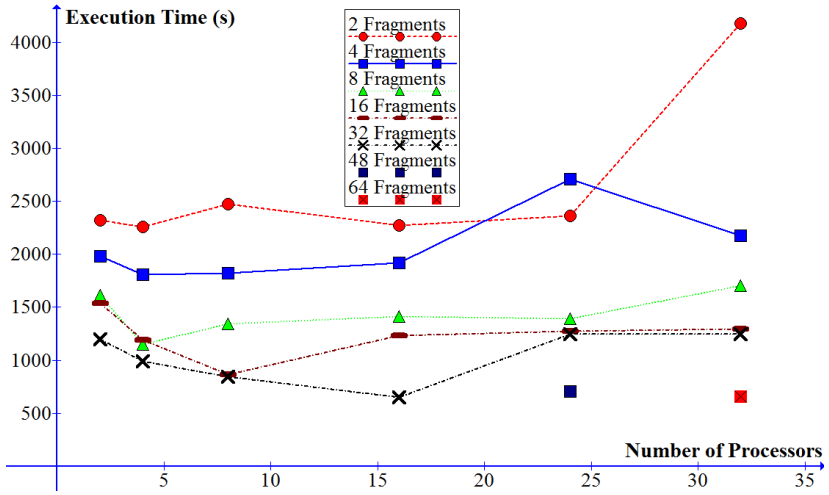

Figure 3. Results of compute nodes and database fragments variation.

It is noticed that an error occurred at 32 compute nodes. The NFS is seldom request- time-out. It could be happened because of the inadequate network interface card (NIC). It is suspected that headnode's NIC have reached its maximum utilization. As a result, NFS server could not serve request from all compute nodes. Since it is a TCP connection, the compute nodes transmitted will request to the headnode until it served. This will clog the network.

The **speedup** is calculated by comparing the time execution between sequential single machine andparallel machine:

$$S_{\mathrm{p}} = \frac{T_1}{T_{\mathrm{p}}}$$

where:

- P is the number of processor.
- T_1 is the execution time for sequential algorithm
- T_p is the execution time for parallel algorithm at p number of processors.

The result is depicted in Figure 4. It can be seen that super-linearity speedup cannot be obtained. Those speedup resultare based on search time for 78 KB query genome against 5.3 GB env_nt database. It stated that super-linearity speedup can be obtained if size of sequence database is much larger than RAM memory size, [2]. In this research, although size of sequence database is much larger than RAM memory size in this environment, super-linearity speedup did not happen. Furthermore decreasing the size of RAM from 2.5 GB to 512 MB still did not boost the speedup value.

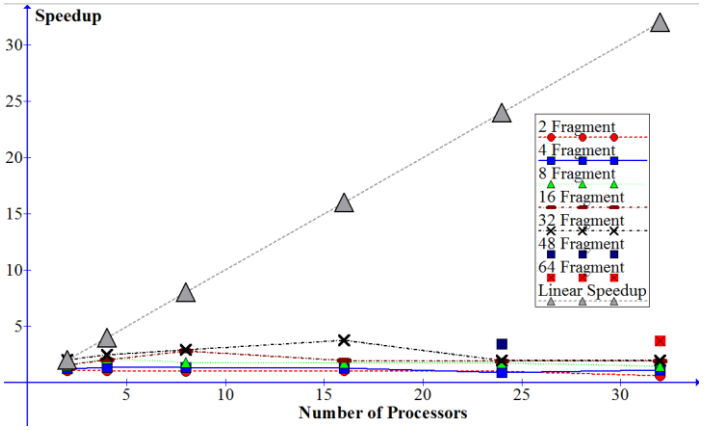

Figure 4. Result of speedup calculation for mpiBLAST in birgHPC environment

The gain is calculated by comparing the different execution time to the baseline execution time, as formulated:

$$Gain = \frac{New - Old}{Old}$$

where:

- Newis the new execution time.
- Oldis the baseline execution time.

Table 2 shows the gain of the experiment. The maximum gain can be obtain at 16 compute nodes with 32 fragments of database, i.e. -73%. This result is too big from the expected value, less than -94%. In the case of super-linearity speedup, the gain is more than -100%.

Table2.Gain from mpiBLAST experiment.

	2 Node	**4 Node**	**8 Node**	**16 Node**	**24 Node**	**32 Node**
2 fragment	-4%	-7%	2%	-6%	-3%	72%
4 fragment	-18%	-26%	-25%	-21%	12%	-10%
8 fragment	-33%	-53%	-45%	-42%	-43%	-30%
16 fragment	-37%	-51%	-64%	-49%	-47%	-47%
32 fragment	-51%	-59%	-65%	-73%	-49%	-49%
48 fragment	-	-	-	-	-71%	-
64 fragment	-	-	-	-	-	-73%

- not tested or failed

Based on the previous experiments, an addition experiment was set for the remedy. The aim is to see size of query genome factor to the speedup. It was limited to a BLAST, 2 and 16 compute nodes using mpiBLAST. The RAM size of headnode and compute node were 2.5 GB. The same database env_nt 5.3 GB binary file with 32 fragments was used. The query DNA alignment was 816 KB Saccaromyces_cerevisae_uid128_NC_001134 genome, about 10 times bigger.The result

is shown in Figure 5.Thespeedup can be improved but super-linearity speedup still can not be reached.

A further research is still required to investigate the super-linearity speedup phenomenon. Despite Darling et al. [2] showed a clear achievement of super-linearity speedup from 4 compute nodes, in a large cluster environment. This research did not find a super-linearity speedup up to 32 compute nodes. Even thoughthe binary database to RAM ratio was same, two times. This research used almost five times faster CPU speed.The NIC problem should be clarified by implementing GbE NIC, supported by GbE switch. Other possible model can be build within virtualization machine, i.e. in a virtual high performance environment.

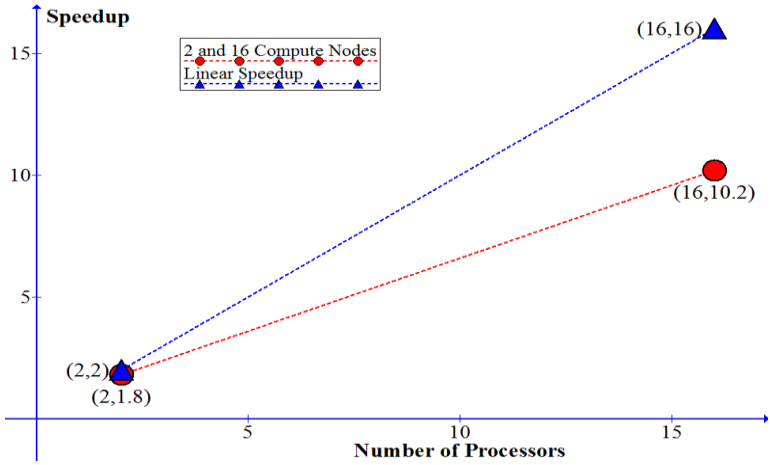

Figure 5. The Speedup of mpiBLAST on 2 and 16 compute nodes in birgHPC environment using 2.5 GB RAM on head node and each Compute Node.

4. Conclusion

After a successful implementation of instant HPC using birgHPC environment and conducted several experiments on a universitycomputer laboratory, it can be concluded that:

1. birgHPC is fairly stable to take advantage computer resource in normal laboratory university.
2. mpiBLAST improves performance from single machine BLAST.
3. For each compute node tested, increasing number of fragments generally improve performance. Number of MPI processes should not exceed number of fragments.
4. It takes longer time execution of mpiBLAST for a larger size of the database, as well as a larger size of query genome.
5. So far, the super-linearity speedup could not be existed within birgHPC enviroment.

References

[1] Altschul, S et al. Basic Local Alignment Search Tool, Journal of Molecular Biology. 1997; 215:403-410.

[2] Darling A.E, Carey L and Feng W.C, The Design, Implementation and Evaluation of mpiBLAST, 4th International Conference on Linux Cluster 2003.

[3] Chew, T.H. et al. birgHPC: Creating Instant Computing Cluster for Bioinformatics and Molecular Dynamics. Oxford Journals: Bioinformatics. 2011; 27(9):1320-1321