A clinical pathogen identification pipeline based on HPC platform

Haoran MA[1], Kenneth Hon Kim BAN[1,2] and Tin Wee TAN [1,2*]

[1]Dept of Biochemistry, Yong Loo Lin School of Medicine, National University of Singapore, [2]National Supercomputing Centre (NSCC), Singapore; Tin Wee TAN – Email: tinwee@bic.nus.edu.sg; * Corresponding author; Authors equally contributed

Abstract

The advent of next generation sequencing (NGS) has provided a new approach to identify, track or trace pathogens from an outbreak with great precision without laborious culture methods. However, the analysis of sequence for pathogens remains unnecessarily slow, given the large datasets requiring computationally intensive analysis using software tools which are not optimized for speed on high performance systems. Thus although some clinical pipelines such as SURPI can identify pathogens from NGS data, the slow speed makes it hard to be widely applied.

To address this, we show a computational pipeline on the petascale National Supercomputing Centre Singapore (NSCC). By taking advantage of the fast I/O, multicore, and large memory systems at NSCC, this pipeline is about 10 time faster than SURPI with the same input and output.

Availability

This pipeline has been written in BASH and is freely available for non-commercial users by request from the authors. Please contact the authors by E-mail: haoranma@u.nus.edu

Keywords

Pathogen Identification; HPC; Clinical pipeline

Introduction

Nowadays, next generation sequencing (NGS) technology based methods are widely used in clinical field. Compared with conventional means, NGS based technology is cheaper and faster with great precision without laborious culture methods, including monitoring the development of drug resistance. However, as the size of biomedical database such as NCBI increases in a booming speed, the challenge of applying NGS-based method into pathogen identification is the data analyzing step rather than the sequencing step. One recent cloud-based clinical pipeline SURPI tried to do pathogen identification by analyzing NGS data. **[1]** But the speed of analyzing data by SURPI is quite slow, which determines that this pipeline cannot be widely applied in clinic.

In this paper we introduce a HPC platform based clinical pathogen identification pipeline. By optimization and using the powerful HPC platform, this pipeline is about 10 times faster than SURPI with the same input file and database as well as same output file.

Methodologies

High Performance Computer (HPC) is a very powerful computing platform. Compared with cloud computing, HPC has more computing resources such as high memory, multiple nodes and faster disk. The HPC we choose to use is ASPIRE in NSCC Singapore. It provides 1200 servers with 128 Gigabyte memory per node and 10 Petabytes storage with very high I/O burst rate of up to 500 Gbps.

Similar with SURPI, our pipeline uses the same process thoughts, that is, firstly do file quality validation and remove the low quality sequences, then do nucleotide sequence alignment with human genome to remove this part of data, and finally do sequence alignment with NCBI NT and NR database. Our pipeline resolves the bottlenecks of SURPI in four aspects:

1. The sequence alignment step occupies most of the run time. As the size of database increases, the time of cost will also increase in a higher speed. To solve this problem, we adopted the scatter and gather approach. This approach fits HPC very well because it can make adequate use and show the advantages of HPC's multiple nodes. We separated the input file and database into 20 pieces to keep each piece in a proper size, built these databases on different machines of HPC, and ran these machines with one same input file and finally selected and gathered the results together.

2. When running the pipeline program, computer needs to repeat reading and writing data for many times. Thus besides the CPU number, the speed of reading and writing data is also very important. Normally we use disk to do this job. But even if HPC's speed of disk is much faster than normal server, it is still a limitation to get faster speed. To solve this problem as well as to make full use of HPC's resources, we put the data totally into memory to get the fastest speed we can reach as HPC's memory is very huge. Using shared memory, the speed of reading and writing with data are much faster than using normal disk, thus greatly increase the speed of the whole pipeline.

3. The pipeline contains many steps and are quite complex. Thus a simple way to run this pipeline is to do one job after the last job is done. However, many jobs are independent among each other in this pipeline. These jobs can run in the same time. So we split the whole pipeline into several parts and do parallelization of these tasks by using Bpipe (shown in figure 1).

   Bpipe is a platform for "running big bioinformatics jobs that consist of a series of processing stages - known as 'pipelines'". **[2]** Using Bpipe, we can change the structure of the whole pipeline and control the starting time to run specific jobs. After rewriting this pipeline by Bpipe, we save a lot of unnecessary cost time.

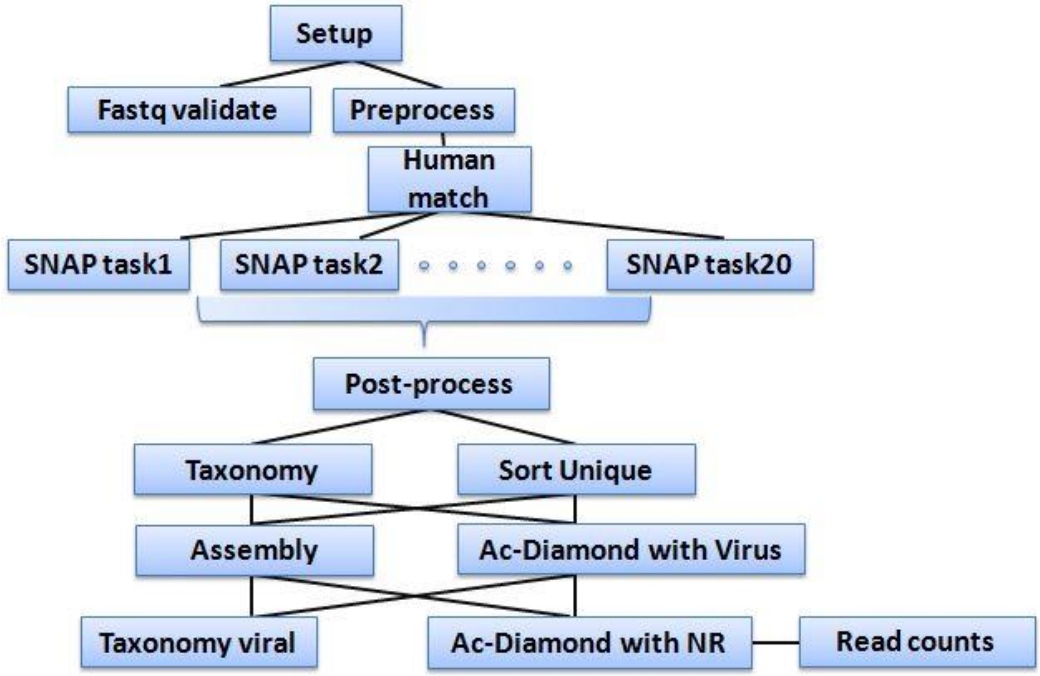

Figure 1: Parallelization of pipeline by using Bpipe

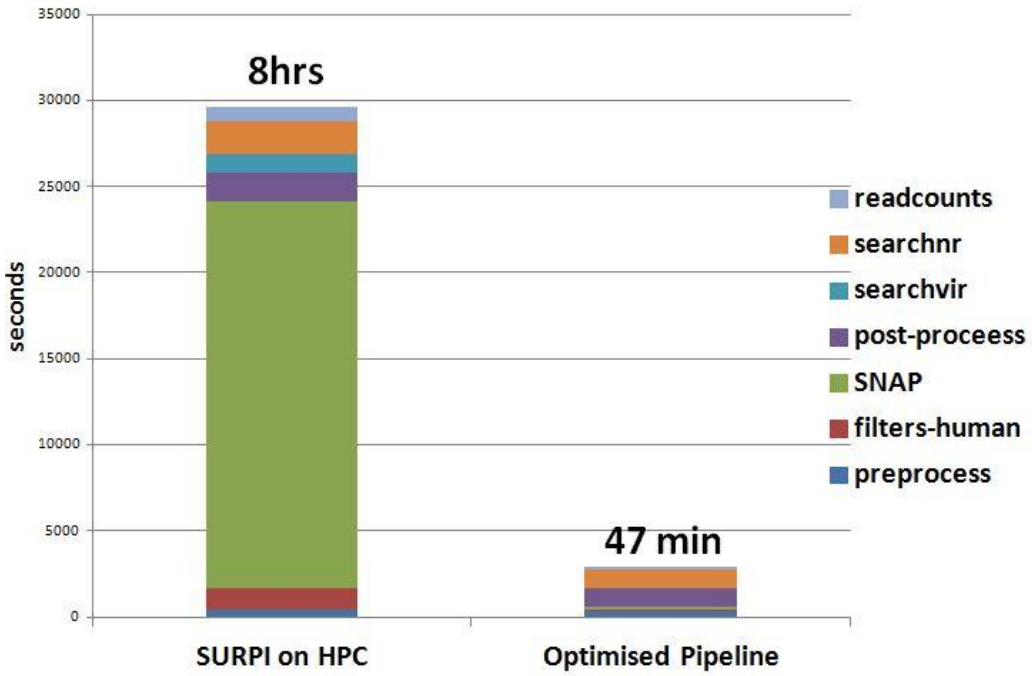

Figure 2: Time record for each step of SURPI and our pipeline

4. When developing the pipeline on HPC, it is necessary to use the proper software. As HPC has powerful computing ability, software that can use large memory and multi-cores are preferred. Thus, we replaced the SNAP with SNAP-aligner in this pipeline, and also replaced RAPSEARCH with AC-DIANMOND. **[3-5]**

Experiment Results

To test the performance of the pipeline and do a comparison with SURPI, we run both pipeline with the NCBI released dataset SRR1106123 and record the running time of each step.

As figure 2 shows, the total run time is 47 minutes, which is nearly 10 times faster than SURPI as it cost about 8 hours with the totally same output. Besides, a very impressive point is that SNAP step duration decreases from about 6 hours to about 8 minutes, which dues to the scatter and gather approach and using of shared memory. These results show that our strategies do work and this pipeline is a much faster pipeline to use.

Caveats and Future Development

Although the size of NT and NR are very large, the results that finally presented are only little part of them. One way to make pipeline more efficient is to do sequence clustering and train data with machine learning. Combined with these two methods, this pipeline could be smarter and does not have to do alignment with the whole database, which will greatly increase the speed.

As for now, the pipeline is only available from the author. Thus a user-friendly web interface is needed to spread the use of this pipeline.

Acknowledgement

We thankfully acknowledge Mr Paul Hiew in NSCC Singapore for providing the computing resource and help, Dr Xie Chao in Human Longevity Inc for his invaluable help in this project and Department of Biochemistry, Yong Loo Lin School of Medicine, National University of Singapore.

Reference

[1] Naccache, S. N., Federman, S., Veeraraghavan, N., Zaharia, M., Lee, D., Samayoa, E., ... & Wadford, D. A. (2014). A cloud-compatible bioinformatics pipeline for ultrarapid pathogen identification from next-generation sequencing of clinical samples. Genome research, 24(7), 1180-1192.

[2] Sadedin, S. P., Pope, B., & Oshlack, A. (2012). Bpipe: a tool for running and managing bioinformatics pipelines. Bioinformatics, 28(11), 1525-1526.

[3] Zaharia, M., Bolosky, W. J., Curtis, K., Fox, A., Patterson, D., Shenker, S., ... & Sittler, T. (2011). Faster and more accurate sequence alignment with SNAP. arXiv preprint arXiv:1111.5572.

[4] Ye, Y., Choi, J. H., & Tang, H. (2011). RAPSearch: a fast protein similarity search tool for short reads. BMC bioinformatics, 12(1), 1.

[5] Mai, H., Li, D., Zhang, Y., Leung, H. C. M., Luo, R., Ting, H. F., & Lam, T. W. (2016, April). AC-DIAMOND: Accelerating Protein Alignment via Better SIMD Parallelization and Space-Efficient Indexing. In International Conference on Bioinformatics and Biomedical Engineering (pp. 426-433). Springer International Publishing.