Workload Analysis Tool for DNA Sequence Benchmark

Kyeongyeol Lim, Geehan Park, Youjip Won
Division of Computer Science and Engineering, Hanyang University
{lkyeol, pghsky, yjwon}@hanyang.ac.kr

Abstract

The recent surge in the amount of genomic data created the need for high performance computing system and storage system to process the data. In this paper, we propose I/O workload analysis tool for bioinformatics application to find efficient configuration of storage system. In our experiment, we used I/O workload analysis tool to evaluate genome analysis pipeline, which analyzes sequence reads data, and analyzed I/O workload information from the experiment.

Keywords: Bioinformatics, Workload analysis, SSD.

1. Introduction

The development of biotechnology caused a huge increase in the speed of genomic data acquisition. Computing systems that manage, analyze, and process genomic data require petaflops level performance and storage systems that can manage data in petabytes.

Storage has been a bottleneck in computing system for a long time but recent development of flash-based SSDs (Solid State Drives) advanced the storage technology. SSDs have many advantages such as fast access speed, low power consumption, and robust durability. On the other hand, they also have disadvantages: higher cost per capacity than HDDs (Hard Disk Drives), impossible in-place-update, and limited number of erase times[1]. So both SSDs and HDDs are used in combination to take advantage of each device. The important matter is how to configure the two devices for better performance. Recently, there has been much progress in researches on this issue.

In this paper, we propose I/O workload analysis tool for genome analysis pipeline to configure storage system that is optimized for genome analysis pipeline.

2. Background: Genome Analysis Pipeline

Genome analysis pipeline[2], which is a method to extract disease information from genomic data, has three major steps: In sequence reads mapping step, aligning the human reference genome index is performed, which can be executed in parallel to reduce the execution time. In SNP (Single Nucleotide Polymorphisms) calling step, the result of sequence mapping is converted to SAM (Sequence Alignment/Map) format[4] and is aligned by genome location. In analysis of statistical information and extractiono of disease information step, the identified SNPs are analyzed for similarities, differences, insertions, deletions, and the possibility of frame-shift.

Genome analysis pipeline can consist of various applications. In this paper, we used BWA (Burrows-Wheeler Aligner)[5] and Samtools [6].

3. Workload Analysis Tool

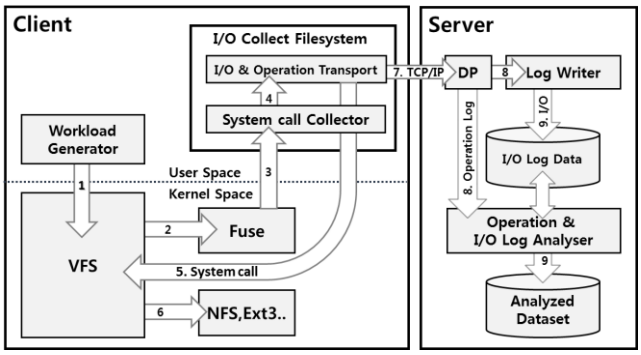

Figure 1: System Diagram

For genome analysis pipeline, we propose I/O workload analysis tool which can be operated in various environments including distributed computing. This tool consists of three components: First, a server which stores, manages, and analyzes captured workloads. Second, a client which captures I/O information in target applications. Third, WebGUI which manages and shows analyzed results.

Fig. 1 shows the system structure. To support distributed computing environments, a client component operates as an agency in each client and is implemented based on FUSE (Filesystem in Userspace)[7]. I/O information of each client is captured by the FUSE-based client component at the file system level, and then is transmitted to server through TCP/IP. The server, which receives I/O information, analyzes and stores the

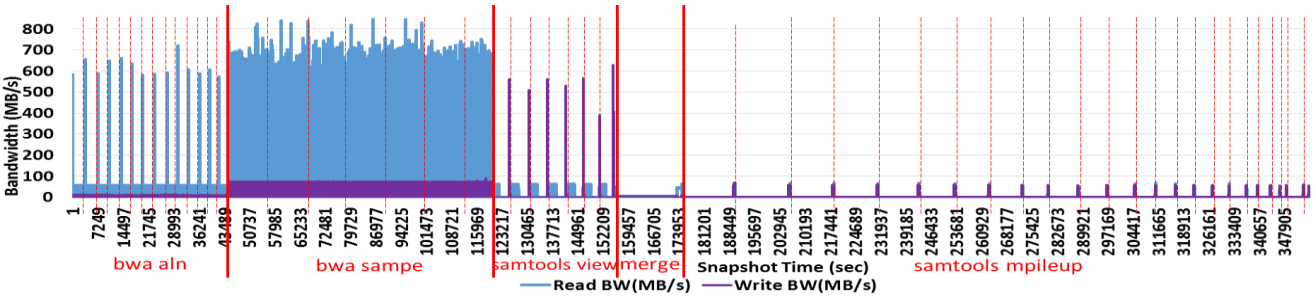

Figure 4: Bandwidth

information in DB. Users can monitor and manage the I/O information using WebGUI.

After completing the analysis of the overall workload, this system provides the following useful information: execution time, various access patterns, IOPS, bandwidth, access frequency, statistics of request size, and usage of CPU in each step of genome analysis pipeline.

4. Experiment

In this experiment, we used human reference genome and analyzed approximately five hundred million sequence reads data of human whole genome. Table 1 shows the size of the file set.

Table 1: File Set

	Count	Size
Human reference genome	10	11GB
Human whole genome sequencing data	14	218GB

The test bed consists of one client, one NFS (Network File System) server, and one analysis server; these are all connected by 10G network. Input data, temporal data, and result data are stored in NFS server.

We divided genome analysis pipeline into five steps according to the purpose of applications: bwa aln, bwa sampe, samtools view | sort & index & flagstat, samtools merge & index, and samtools mpileup.

Table 2 shows the number of create, delete, open operations and the size of write and delete files in each step

Table 2: File Operation Summery

	Create / Delete/ Open	Write(GB)	Delete(GB)
bwa aln	14 / 0 / 28	91.0	0
bwa sampe	25 / 14 / 4038	320.8	91.0
samtools sort	6 / 535 / 20	227.0	320.8
samtools merge	2 / 7 / 8	69.4	227.0
Samtools mpileup	155 / 7 / 133	323.4	0
Total	202 / 563 / 4227	1031	638.8

of pipelining. In each step, applications write temporal files to use in the next step. These temporal files are deleted when the next step completes.

Fig. 4 illustrates read/write bandwidth of genome analysis pipeline from start to completion. Bold solid lines divide the five steps. In each step, the target applications are executed repeatedly. The repetitions are separated by dashed lines. Each step displays unique pattern.

5. CONCLUSION

In this paper, we propose a tool that analyzes genome analysis pipeline I/O workload and operates independently of the file systems. We implemented prototype to experiment genome analysis pipeline.

In this 98-hour experiment, 1031.6 GByte file write and 638.8 GByte deletion took place. Also, unique I/O patterns were discovered in each step.

6. Acknowledgement

This work is sponsored by IT R&D program MKE/KEIT. [No.10035202, Large Scale hyper-MLC SSD Technology Development].

References

[1] J. Kang, H. Jo, J. Kim, and J. Lee, “A superblock-based flash translation layer for nand flash memory,” pp. 161–170, 2006.

[2] C. Bell, et al., “The medicago genome initiative: a model legume database,” Nucleic Acids Research, vol. 29, no. 1, pp. 114–117, 2001.

[3] E. Lander, et al., “Initial sequencing and analysis of the human genome,” Nature, vol. 409, no. 6822, pp. 860–921, 2001.

[4] A. McKenna, et al., “The genome analysis toolkit: a mapreduce framework for an-alyzing next-generation dna sequencing data,” Genome research, vol. 20, no. 9, pp. 1297–1303, 2010.

[5] H. Li and R. Durbin, “Fast and accu-rate short read alignment with burrows–wheeler transform,” Bioinformatics, vol. 25, no. 14, pp. 1754–1760, 2009.

[6] H. Li, et al., “The se-quence alignment/map format and samtools,” Bioinformatics, vol. 25, no. 16, pp. 2078–2079, 2009.

[7] FUSE, “Filesystem in userspace.” http://fuse.sourceforge.net/.