Enabling Large-Scale Human Genome Sequence Analysis on CloudLab

Praveen Rao
University of Missouri-Columbia
Columbia, MO, USA
praveen.rao@missouri.edu

Arun Zachariah
University of Missouri-Columbia
Columbia, MO, USA
azachariah@mail.missouri.edu

***Abstract*—Human genome sequences are very large in size and require significant compute and storage resources for large-scale analysis. Motivated by the need to speedup human genome processing, we developed software for large-scale human genome sequence analysis on CloudLab. We present the design of our software and the experimental setup on CloudLab. We also discuss the insights gained and lessons learned through our effort. Our software can be a valuable resource for researchers, educators, and students in computing and bioinformatics.**

I. Introduction

Genomics is regarded as a Big Data science [8]. It is projected that between 100 million-2 billion humans could be sequenced by 2025 producing up to 40 exabytes of genome data [8]. With the cost of human whole-genome sequencing (WGS) falling below $1,000, WGS has become economically feasible in large-scale studies and clinical practice. It has become a critical tool for accelerating scientific discovery in genomics and medicine. The COVID-19 pandemic has led to multiple genome sequencing initiatives worldwide,[1] providing further impetus to genomic medicine in a clinical setting.

In recent years, there has been a growing demand from hospitals and institutions to process large volumes of genomic data; getting back results in a few hours can potentially save a patient's life.[2] A single human genome sequence can consume 10's of GBs of storage [2]. Processing a large number of sequences poses technical challenges for efficient storage, processing, analysis, and data transfer. While access to a large number of computing and storage resources is possible today through cloud computing, reducing the cost of analyzing genomes continues to be a key challenge [4]–[6]. In recent years, open source projects have emerged (e.g., GATK4[3], ADAM-Cannoli [5], [6]) that employ cluster computing frameworks, Apache Spark [11] and Apache Hadoop [10], for analyzing human genomes. Companies such as Microsoft, Databricks, and NVIDIA are providing new tools and services to customers for accelerating analytics on genomics data. Thus, there is growing interest in advancing the state of the art in analyzing human genomes at scale.

[1]www.covidhge.com, www.cogconsortium.uk, www.genomecanada.ca

[2]https://blogs.microsoft.com/ai/microsoft-computing-method-makes-key-aspect-genomic-sequencing-seven-times-faster

[3]https://github.com/broadinstitute/gatk

Our effort is aimed at *democratizing human genome sequence analysis* using CloudLab [1]. Thus, any registered CloudLab user can process human genomes for research and educational purposes at no charge. We specifically focus on variant calling, which is a fundamental task that is performed to identify variants in an individual's genome compared to a reference human genome [9]. Identifying variants will enable better understanding of an individual's risk to diseases and lead to new advances in disease prevention and treatment.

The variant calling pipeline consists of several stages including reading the genome sequence, performing alignment of reads with a reference genome, additional pre-processing steps, and finally, invoking a variant caller to produce raw variants.[3] The raw variants are further processed by variant filtering and annotation steps.[3] The pipeline involves several computationally intensive tasks and requires significant computing and storage resources to analyze a large number of human genome sequences. Setting up the pipeline (based on best practices) and necessary datasets is tedious especially when executing in a commodity cluster. Therefore, we seek to lower the *barrier to entry* for human genome sequence analysis among computing/bioinformatics researchers, educators, and students. Thus, our effort can accelerate scientific discoveries in genomic medicine and enable new innovations in computer and network systems for large-scale genome analysis.

II. Our Software

Motivated by the growing demand for efficient genome analysis, we developed novel software called AVAH (**A**ccelerating **VA**riant Calling on **H**uman Genomes) to accelerate the variant calling pipeline on human genomes using a commodity cluster [7]. AVAH draws inspiration from asynchronous computations and the futures abstraction [3]. It distributes the task of executing the variant calling pipeline on the input sequences across the cluster nodes. It synergistically combines task parallelism and data parallelism for different stages in the pipeline by using futures and has minimal synchronization barriers among the stages. These asynchronous computations are executed in a sliding window manner on small groups of sequences to control the degree of parallelism and improve cluster utilization. AVAH is built atop Apache Spark and Apache Hadoop, and designed to leverage the APIs of existing Big Data Genomics software (e.g., Adam-Cannoli) to execute

the variant calling pipeline. Genome sequences, intermediate files produced by the pipeline, the reference genome, additional data needed during certain stages of the pipeline, and the raw variants can be stored in the Hadoop Distributed File System (HDFS). Thus, stages of the pipeline can be re-executed on different nodes if required. Failures can occur during the execution of the variant calling pipeline due to insufficient main memory for processes running outside of the Java Virtual Machine (JVM), Spark shuffle errors, and others. AVAH performs early re-execution of failed sequences when the cluster load falls below a user-specified threshold.

In terms of performance, AVAH was 3X-4.7X faster than ADAM-Cannoli in processing 98 genome sequences using a 16-node cluster on CloudLab. It yielded better cluster utilization than ADAM-Cannoli. Thus, AVAH can accelerate variant calling pipelines and lower the computational cost per genome. In the interest of space, the reader is referred to a previous publication [7] for complete details about AVAH. AVAH is available via `https://github.com/MU-Data-Science/EVA`.

III. Experimental Setup

AVAH can be executed in three different cluster settings on CloudLab: (a) single-site, homogeneous cluster (C_1); (b) single-site, heterogeneous cluster (C_2); and (c) multi-site cluster (C_3). In C_1, all the nodes are of the same hardware type. In C_2, two different hardware types can be chosen for the cluster nodes. In C_1 and C_2, the nodes are connected by a Gigabit Ethernet network (10 Gbps). In C_3, the nodes between the two sites (e.g., Clemson and Wisconsin) are connected by Internet2's Advanced Layer 2 Service (AL2S) using a VLAN (virtual LAN). CloudLab profiles are available for each cluster setting. The network bandwidth among cluster nodes can be configured to study how it impacts the processing time of AVAH. Typically, 16-24 nodes are sufficient to process 100's of human genome sequences. We recommend the Clemson or Wisconsin data centers as they have nodes with large amounts of RAM required for variant calling.

Due to limited root filesystem storage on the nodes, we mount additional local block storage (striped across multiple physical disks) on each node (about 1-2 TB per node). HDFS and the required software packages are set up on the local block storage of the nodes. The maximum transmission unit (MTU) value can be changed for the network interfaces (e.g., to use jumbo frames). During execution of AVAH, statistics such as CPU load average, network I/O, disk I/O, memory usage, etc., can be collected using `dstat`. Network traces can also be collected using `tcpdump` for further analysis.

Due to privacy laws, we allow only publicly available or de-identified genome sequences to be processed on CloudLab. A user can specify the URLs of the sequences to download via the Internet. AVAH downloads these sequences in parallel (using futures) and directly stores them in HDFS.

IV. Lessons Learned

Next, we report the lessons learned while enabling large-scale genome analysis on CloudLab. Security threats can arise when running experiments on CloudLab. Specifically, Apache Spark and Hadoop can be exploited by attackers (e.g., for cryptomining) through arbitrary code execution. To prevent such exploits, we disabled the Spark UI, enabled Spark's authentication using secret keys, and enabled access control in Hadoop's Yet Another Resource Negotiator (YARN).

Configuring the YARN settings was challenging. The maximum RAM used by YARN on each node had to be configured carefully. If too much memory was allocated to YARN, it caused failures when executing third-party genomic tools for alignment and variant calling that run outside of the JVM. Spark's executor memory had to be carefully configured so that genome sequences of varying sizes could be processed. A higher value of executor memory resulted in fewer YARN containers being launched, thereby leading to lower cluster utilization. Certain variant calling stages required more executor memory to execute successfully due to processing of additional genomic data files.

As HDFS is used to store genome sequences and intermediate files, network bandwidth can impact AVAH's performance. Higher bandwidth links (e.g., 1-5 Gbps) are required for fast network I/O during different stages of the variant calling pipeline. Otherwise, the cluster utilization deteriorates leading to slower execution of the pipeline by AVAH.

In conclusion, we hope our work will stir more innovation in human genome sequence analysis using cluster computing.

Acknowledgments*:* This work was supported by the National Science Foundation under Grant No. 2034247.

References

[1] D. Duplyakin, R. Ricci, A. Maricq, G. Wong, J. Duerig, E. Eide, L. Stoller, M. Hibler, D. Johnson, K. Webb, A. Akella, K. Wang, G. Ricart, L. Landweber, C. Elliott, M. Zink, E. Cecchet, S. Kar, and P. Mishra. The Design and Operation of CloudLab. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, pages 1–14, 2019.

[2] S. Goodwin, J. D. McPherson, and W. R. McCombie. Coming of Age: Ten Years of Next-Generation Sequencing Technologies. *Nature Reviews Genetics*, 17(6):333–351, 2016.

[3] R. H. Halstead. MULTILISP: A Language for Concurrent Symbolic Computation. *ACM TOPLAS*, 7(4):501–538, 1985.

[4] P. Muir, S. Li, S. Lou, D. Wang, D. J. Spakowicz, L. Salichos, J. Zhang, G. M. Weinstock, F. Isaacs, J. Rozowsky, and M. Gerstein. The Real Cost of Sequencing: Scaling Computation to Keep Pace with Data Generation. *Genome Biology*, 17(1):53, 2016.

[5] F. A. Nothaft. *Scalable Systems and Algorithms for Genomic Variant Analysis*. PhD thesis, UC Berkeley, ProQuest, 2017.

[6] F. A. Nothaft, M. Massie, T. Danford, Z. Zhang, U. Laserson, C. Yeksigian, J. Kottalam, A. Ahuja, J. Hammerbacher, M. D. Linderman, M. J. Franklin, A. D. Joseph, and D. A. Patterson. Rethinking Data-Intensive Science Using Scalable Analytics Systems. In *Proc. of the 2015 ACM SIGMOD Conference*, pages 631–646, 2015.

[7] P. Rao, A. Zachariah, D. Rao, P. Tonellato, W. Warren, and E. Simoes. Accelerating Variant Calling on Human Genomes Using a Commodity Cluster. In *Proc. of 30th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 3388–3392, 2021.

[8] Z. D. Stephens, S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai, M. J. Efron, R. Iyer, M. C. Schatz, S. Sinha, and G. E. Robinson. Big Data: Astronomical or Genomical? *PLOS Biology*, 13(7):1–11, 2015.

[9] A. Supernat, O. V. Vidarsson, V. M. Steen, and T. Stokowy. Comparison of Three Variant Callers for Human Whole Genome Sequencing. *Scientific Reports*, 8, 2018.

[10] T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 2009.

[11] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster Computing with Working Sets. In *Proc. of 2nd USENIX Conference on Hot Topics in Cloud Computing*, pages 1–7, 2010.