RETROSPECTIVE: Genesis: A Hardware Acceleration Framework for Genomic Data Analysis

Lisa Wu Wills
Duke University
Durham, NC 27701
lisa@cs.duke.edu

Tae Jun Ham
Google
Mountain View, CA 94043
taejunham@google.com

Jae W. Lee
Seoul National University
Seoul, South Korea
jaewlee@snu.ac.kr

Krste Asanović
University of California Berkeley
Berkeley, CA 94720
krste@berkeley.edu

I. Introduction

We began the work on "Genesis: A Hardware Acceleration Framework for Genomic Data Analysis" [6] in early 2017 while Lisa Wu Wills was a postdoc advised by Krste Asanović at UC Berkeley. Most accelerator work related to genomic analytics centered around specialized hardware for specific algorithms, including our own work in accelerating INDEL realignment in the cloud [12]. The cost of accelerator design and deployment cannot be justified if the accelerator is a one-off design with limited applicability. As genomic algorithms were still changing frequently, we began thinking of ways to amortize the cost of designing and deploying accelerated systems.

II. The Process

A. An Accelerator Development Framework

Around the same time we started exploring the acceleration of genomic analytics, Amazon Elastic Compute Cloud (EC2) launched its F1 instance, which is a rentable FPGA-in-the-cloud. The introduction of FPGAs in cloud environments provides an unprecedented opportunity to achieve hardware acceleration with little up-front cost. However, developing and evaluating hardware accelerators is a challenging task when compared to developing software. High-level synthesis promises one avenue for achieving highly productive hardware accelerator development. However, high-level tools often leave much to be desired in performance and energy efficiency. To simplify the workflow for hardware developers, we started exploring and implementing a reusable hardware development framework using Chisel [3].

With this framework, we tackled the most critical and time-consuming parts of accelerator development: the communications between the host and accelerators, the custom programming interface, and communications between the accelerator and its memory. The framework helps produce hardware that is automatically augmented with parallelism, efficient memory accesses, and easy-to-use high-level user interfaces in C/C++ between the host and the accelerator.

B. The Concept of a Hardware Library

While the framework simplifies development of an accelerated system, we also take from software the idea of libraries that are modular, reusable across applications, and easily updated and modified as needed, and developed a hardware library as part of Genesis. The vision is that a careful factoring of hardware libraries across multiple domains can allow the library components to be reused, composed, modified, and updated with much less effort within the framework.

Further, we believe that if a software algorithm can be mapped onto a hardware library to utilize the underlying accelerator(s) using an already-standardized software interface, while allowing the efficient mapping of such interface to primitive hardware modules (as we have demonstrated with Genesis), it will expedite the acceleration of domain-specific algorithms and allow easy adaptation to algorithm changes.

C. Data Manipulation Tasks Are Similar Across Analytics Domains

Investigating the genomic analytics algorithms allowed us to decompose them into primitive operators. We observed that there are ample similarities between the operations needed to perform genomic analytics and traditional big-data analytics. Specifically, analytics across domains share similar, if not identical, data manipulation operations. We proposed treating genomic data as traditional data tables and using extended SQL as a domain-specific language to perform genomic analysis. For example, performing *joins* between two tables using related information allows us to answer queries that are relevant to the related information. In genomic analytics, a patient's genomic reads table is joined with the known base-pair mismatch variants table to filter out non-alarming genomic variants. This step is essential when figuring out if there is a possibility that the patient has cancer (i.e., somatic variants). Other examples of data manipulation operations that are shared across domains are filtering, aggregation, and counting.

Conceptualizing the genomic data as a very large relational database allows us to reason about the algorithms and transform genomic data processing stages into simple extended SQL-style queries. Once the queries are constructed, Genesis facilitates the translation of the queries into hardware accelerator pipelines using the Genesis hardware library that accelerates primitive operations in database and genomic data processing. The same library can then be reused for accelerating database analytics tasks as well.

III. Subsequent Work

It has only been three years since the work was published but there have been many advancements in the relevant area.

First, genomic data analysis and the efficient use of computational hardware for it remains an important topic in both academia and industry. Specifically, as our work highlighted, recent advancements focus more on high-coverage acceleration to prevent unaccelerated operations from incurring an Amdahl's law bottleneck. For example, Illumina's FPGA-based genomic analysis accelerator DRAGEN [7] as well as NVIDIA's GPU-based genomic analysis solution Clara [10] supports a much wider range of operations to enable end-to-end acceleration for various use cases. In addition, academic works further explore the potential of leveraging in-memory or in-storage computing [4], [8] to overcome communication bottlenecks between processing units and memory or storage devices. Second, the idea of analyzing genomic data with database query languages has become even more popular. Cloud services like Amazon Omics [2] or Google Cloud Life Science [5] allow users to use query languages to analyze genomic data. Finally, using specialized hardware to accelerate database/data analytics operations is becoming commonplace. Amazon AQUA [1] utilizes FPGAs to accelerate their analytics workloads and startups like Neuroblade [9] and Speedata [11] explore the use of specialized application-specific integrated circuits for data analytics.

It's an exciting time where continuous developments in algorithms and applications drive the need for new customized hardware, and advancements in customized hardware enable further innovations in algorithms and applications – a virtuous circle. However, what remains a challenge is the transfer of prototype technology from academia to real-world products in this area. Specifically, for algorithms in healthcare and genomics domain, it is critical that specialized hardware be able to handle different configurations and variants of algorithms effectively. We look forward to seeing how future hardware-acceleration researchers in this domain handle this challenge, hopefully using a composable accelerator construction methodology as outlined in our original work to significantly reduce development and deployment effort.

IV. What Happened to the Authors

Lisa Wu Wills continued work on architecting accelerators, including finding ways to significantly simplify the process of developing and deploying accelerated systems. She joined Duke University as an assistant professor of Computer Science in 2019, where she has worked on accelerating Transformer models for Natural Language Processing (NLP) tasks and protein discovery as well as leveraging NLP models to predict synthesis results. She received a Clare Boothe Luce Professorship in 2019, a Facebook Research Award in 2020, a Google "Rising" Systems Faculty Award in 2020, and an NSF CAREER Award in 2021.

Tae Jun Ham was a postdoc researcher at Seoul National University (SNU) at the time paper was written. He later joined Google and currently working as a software engineer focusing on datacenter performance analysis and optimizations.

Jae W. Lee was an associate professor of Computer Science and Engineering at SNU back then. He has since received tenure, the SNU Educator Award, co-chaired the IEEE Micro Top Picks Selection Committee (2022), and spent some time at Google as a visiting researcher.

Krste Asanović continues to be a professor in the UC Berkeley EECS department, as well as chairman of the board of RISC-V International and co-founder and Chief Architect at SiFive.

David Bruns-Smith is an EECS PhD candidate at UC Berkeley. His research focuses on computer science and economics, specifically looking at the dynamics of wealth, income, and debt and their implications for public policy. Brendan Sweeney was an undergraduate researcher at UC Berkeley at the time this paper was published. Brendan is now an ECE PhD candidate at UT Austin. Yejin Lee and Young H. Oh from the SNU team have since received their PhDs. Yejin is working as a postdoctoral researcher at Meta, and Young is a staff researcher at Samsung. Seong Hoon Seo and U Gyeong Song are still with SNU, as a PhD candidate and undergraduate student, respectively.

References

[1] Amazon, "AQUA (Advanced Query Accelerator) – A Speed Boost for Your Amazon Redshift Queries," 2021. [Online]. Available: https://aws.amazon.com/blogs/aws/new-aqua-advanced-query-accelerator-for-amazon-redshift/

[2] Amazon, "Genomic Data Analysis - Amazon Omics," 2023. [Online]. Available: https://aws.amazon.com/omics/?nc1=h_ls

[3] "Chisel hardware construction language," https://chisel.eecs.berkeley.edu/.

[4] S. Diab, A. Nassereldine, M. Alser, J. Gómez Luna, O. Mutlu, and I. El Hajj, "A framework for high-throughput sequence alignment using real processing-in-memory systems," *Bioinformatics*, vol. 39, no. 5, 03 2023.

[5] Google, "Process genomic data by using Cloud Life Sciences," 2023. [Online]. Available: https://cloud.google.com/life-sciences/docs/process-genomic-data

[6] T. J. Ham, B. S. David Bruns-Smith, Y. Lee, S. H. Seo, U. G. Song, K. A. Young H. Oh, J. W. Lee, , and L. W. Wills, "Genesis: A hardware acceleration framework for genomic data analysis," in *Proceedings of the International Symposium on Computer Architecture (ISCA)*, June 2020.

[7] Illumina, "Dragen," 2023. [Online]. Available: https://developer.illumina.com/dragen

[8] N. Mansouri Ghiasi, J. Park, H. Mustafa, J. Kim, A. Olgun, A. Gollwitzer, D. Senol Cali, C. Firtina, H. Mao, N. Almadhoun Alserr, R. Ausavarungnirun, N. Vijaykumar, M. Alser, and O. Mutlu, "Genstore: A high-performance in-storage processing system for genome sequence analysis," in *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2022.

[9] Neuroblade, "Introducing the SQL Processing Unit (SPU)," 2023. [Online]. Available: https://www.neuroblade.com/product/#benefits

[10] NVIDIA, "Clara for Genomics," 2023. [Online]. Available: https://www.nvidia.com/en-us/clara/genomics/

[11] Speedata, "Speedata: Orders-of-magnitude hardware acceleration for Spark and Presto without code changes," 2023. [Online]. Available: https://www.speedata.io/

[12] L. Wu, D. Bruns-Smith, F. A. Nothaft, Q. Huang, S. Karandikar, J. Le, A. Lin, H. Mao, B. Sweeney, K. Asanović, D. A. Patterson, and A. D. Joseph, "FPGA accelerated INDEL realignment in the cloud," in *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, 2019.