RETROSPECTIVE: Aladdin: a Pre-RTL, Power-Performance Accelerator Simulator Enabling Large Design Space Exploration of Customized Architectures

Yakun Sophia Shao
University of California, Berkeley

Brandon Reagen
New York University

Gu-Yeon Wei
Harvard University

David Brooks
Harvard University

I. What is Aladdin?

Aladdin [1] is a pre-RTL, power/performance simulator designed to enable rapid design space search of accelerator-centric systems. This framework takes high-level language descriptions of algorithms as inputs (C or C++), and uses dynamic data dependence graphs (DDDG) as a representation of an accelerator without having to generate RTL. Starting with an unconstrained program DDDG, which corresponds to an initial representation of accelerator hardware, Aladdin applies optimizations as well as constraints to the graph to create a realistic model of accelerator activity. We validated Aladdin against RTL implementations of accelerators from both handwritten Verilog and a commercial high-level synthesis (HLS) tool for a range of applications. Our results showed Aladdin can model power, performance, and area with high accuracy, well within 10% when compared to accelerator designs generated by traditional RTL flows, while providing these estimates with much less design effort and time.

Aladdin captures accelerator design trade-offs, enabling new architectural research directions in heterogeneous systems comprising accelerators, general-purpose cores, and the shared memory hierarchy seen, for example, in mobile SoCs. In particular, Aladdin allows users to explore customized and shared memory hierarchies for accelerators in a heterogeneous environment. As an example, in a case study with the GEMM benchmark, Aladdin uncovered significant, high-level, design trade-offs by evaluating a broader design space of the entire system. We envisioned that Aladdin could be used both as an accelerator simulator and a design space exploration tool for future many-accelerator systems.

II. Our motivation for Aladdin

Looking back, motivations that led to the Aladdin work are three-fold. It was well known by the early twenty teens that Dennard scaling had ostensibly run out and process technology scaling was slowing down. In order to continue to scale performance, the computing industry and academics had begun to explore specialization in the form of hardware accelerators to maximize computing efficiency. In fact, ITRS 2007 had predicted that systems would comprise hundreds to thousands of accelerators by 2022. The authors often use the growing number of perceived blocks observed across multiple generations of Apple's SoC die photos, plotted in Figure 1, to illustrate that trend. However, design of many-accelerator systems presented several challenges in terms of identifying the best/right mixture of accelerators within a system and understanding their vast and complex interactions. In other words, we needed a design space exploration framework and relying on RTL-level modeling of different accelerator designs was overly cumbersome to implement and prohibitively slow to simulate.

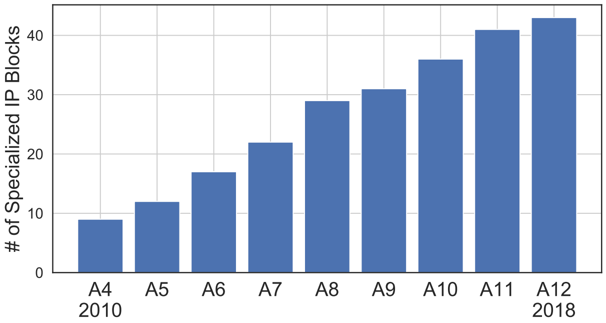

Fig. 1. Number of specialized IP blocks across generations of Apple's SoCs.

Within our research group at Harvard, there were a series of projects and publications that laid the steps towards Aladdin. Co-authors Brooks and Wei began their nearly 20-year research collaboration with their 2005 ISCA paper [2], "An ultra low power system architecture for sensor network applications." At the time, wireless sensor network (WSNs), now commonly known as "Internet of Things" or "IoT" was a hot emerging field. Designing computing systems for WSNs focused on applications where energy consumption was the most important target and characterized by ultra low duty cycle operation. We turned to hardware accelerators as the most energy-efficient solution that also minimized area and, consequently, leakage. Our follow-up workshop paper in 2009 [3], "*Navigo*: An early-stage model to study power-constrained architectures and specialization," postulated that "specialization is the answer to circumvent the power density limit that curbs performance gains and resumes traditional 1.58× performance growth trends." This early accelerator-centric research then led us to develop "Accelerator Store: A shared memory framework for accelerator-based systems" [4]. As accelerator-based systems gained momentum broadly, we realized that it was becoming more and more difficult to perform architecture research that explored realistic systems comprising many accelerators beyond a handful of toy examples in narrow domains. The mainstay of architecture research was to rely on high-level CPU and GPU models at the time, and therefore there was also a need for the ability to develop and simulate a broad swath of accelerator design quickly and accurately. The confluence of these prior works motivated the Aladdin work.

III. What We Accomplished, and What We Didn't

Aladdin enabled accelerator design and benchmarking. Aladdin broke the development cost barrier in accelerator research, enabling researchers to focus on higher-level questions and deeper optimizations rather than writing RTL. An example of this was Minerva [5], one of the initial AI accelerators for ubiquitous inference. Minerva used Aladdin to rapidly characterize the accelerator design space, evaluating hundreds of design alternatives in a matter of hours. This freed us to explore more complex tradeoffs, e.g., fault-tolerance techniques to lower operating voltage and rapidly quantify their benefits with simple Aladdin extensions (e.g., bitwidths and low-voltage). Minerva, thanks to Aladdin, demonstrated how accelerators are not a one-time win and that with algorithm-hardware co-design, performance, and efficiency can be improved over multiple chip generations. Aladdin also exposed the research community's unpreparedness for the era of accelerator-centric architectures. In early 2014, no accelerator benchmarks existed. Recognizing this issue, we then developed MachSuite [6], a collection of accelerator workloads and implementations to standardize research in the community, stress-test infrastructure (e.g., HLS and Aladdin), and usher in the era of accelerator-centric computing. Since its release, MachSuite has become the standard for accelerator-centric research and used in hundreds of papers.

Aladdin pioneered systematic integration of accelerators. The 2014 Aladdin paper laid the foundation for the "sea-of-accelerators" concept, foreseeing the future of heterogeneous SoCs. Despite drawing some skepticism at the time, its prediction has been validated almost a decade later. Today, companies, both large and small, are actively developing a diverse array of hardware accelerators spanning applications from edge to cloud. Motivated by this observation, the authors' immediate follow-up work on gem5-Aladdin [7] marked an important milestone in the study of accelerator memory systems and their interaction with the broader SoC memory hierarchy. Subsequently, recognizing the pervasiveness of machine-learning (ML) accelerators in modern SoCs, co-author Shao's group developed Gemmini [8], a full-system ML SoC generator, enabling seamless integration of ML accelerators into the entire SoC in a flexible and efficient manner. Another crucial aspect of accelerator simulation is how to capture the intricate inter-dependencies between hardware and software components in an end-to-end fashion. To address the knowledge gaps in this space, co-author Shao's group recently developed RoSÉ [9], a hardware-software co-simulation framework to enable end-to-end evaluation of domain-specific systems for closed-loop applications like robotics. These continued efforts in accelerator modeling and system integration have propelled the field forward, fostering new frontiers in hardware-software co-design and evaluation for accelerators.

The advances in domain-specific accelerator modeling and high-level synthesis (HLS) tools. What Aladdin did not anticipate is the rapid advancements in ML accelerator modeling and HLS tools. On the accelerator modeling side, instead of pursuing a general modeling approach that can be used for diverse applications like what Aladdin enabled, a notable portion of the field has focused on developing specialized modeling infrastructure for ML accelerators like Timeloop [10], which co-author Shao contributed to during her time at NVIDIA. Another important trend over the past decade is the significant improvement of HLS tools for accelerator development, particularly through the development of reusable libraries [11]. This progress has enabled the adoption of HLS-based methodologies in both academia and industry. Notably, all the Aladdin authors have leveraged HLS-based approaches in their subsequent tapeout activities, successfully building chip prototypes across different technology nodes and SoC complexities.

References

[1] Y. S. Shao, B. Reagen, G.-Y. Wei, and D. Brooks, "Aladdin: A Pre-RTL, Power-Performance Accelerator Simulator Enabling Large Design Space Exploration of Customized Architectures," in *ISCA*, 2014.

[2] M. Hempstead, N. Tripathi, P. Mauro, G.-Y. Wei, and D. Brooks, "An ultra low power system architecture for sensor network applications," in *ISCA*, pp. 208–219, 2005.

[3] M. Hempstead, G.-Y. Wei, and D. Brooks, "Navigo: An early-stage model to study power-constrained architectures and specialization," in *MoBS*, 2009.

[4] M. J. Lyons, M. Hempstead, G.-Y. Wei, and D. Brooks, "The accelerator store: A shared memory framework for accelerator-based systems," *ACM TACO*, vol. 8, jan 2012.

[5] B. Reagen, P. Whatmough, R. Adolf, S. Rama, H. Lee, S. K. Lee, J. M. Hernández-Lobato, G.-Y. Wei, and D. Brooks, "Minerva: Enabling Low-power, Highly-accurate Deep Neural Network Accelerators," in *ISCA*, 2016.

[6] B. Reagen, R. Adolf, Y. S. Shao, G.-Y. Wei, and D. Brooks, "MachSuite: Benchmarks for Accelerator Design and Customized Architectures," in *IISWC*, 2014.

[7] Y. S. Shao, S. Xi, V. Srinivasan, G.-Y. Wei, and D. Brooks, "Co-Designing Accelerators and SoC Interfaces using gem5-Aladdin," in *MICRO*, 2016.

[8] H. Genc, S. Kim, A. Amid, A. Haj-Ali, V. Iyer, P. Prakash, J. Zhao, D. Grubb, H. Liew, H. Mao, A. Ou, C. Schmidt, S. Steffl, J. Wright, I. Stoica, J. Ragan-Kelley, K. Asanovic, B. Nikolic, and Y. S. Shao, "Gemmini: Enabling Systematic Deep-Learning Architecture Evaluation via Full-Stack Integration," in *DAC*, 2021.

[9] D. Nikiforov, S. K. Dong, C. L. Zhang, S. Kim, B. Nikolic, and Y. S. Shao, "RoSÉ: A Hardware-Software Co-Simulation Infrastructure Enabling Pre-Silicon Full-Stack Robotics SoC Evaluation," in *ISCA*, 2023.

[10] A. Parashar, P. Raina, Y. S. Shao, Y.-H. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S. W. Keckler, and J. Emer, "Timeloop: A Systematic Approach to DNN Accelerator Evaluation," in *ISPASS*, 2019.

[11] R. Venkatesan, Y. S. Shao, M. Wang, J. Clemons, B. Keller, A. Klinefilter, A. Rekhi, Y. Zhang, B. Zimmer, W. J. Dally, J. S. Emer, S. W. Keckler, and B. Khailany, "MAGNet: A Modular Accelerator Generator for Neural Networks," in *ICCAD*, 2019.