RETROSPECTIVE: A Configurable Cloud-Scale DNN Processor for Real-Time AI

Eric S. Chung  Doug Burger  Jeremy Fowers
Mahdi Ghandi  Gabriel Weisz  Sitaram Lanka  Steven K. Reinhardt

I. CONTEXT

As we reflect upon the work we started in 2015 and its influence over time, it’s illuminating to trace the trajectory of our thinking, the evolution of technology, and the impact it had on the field. When our work commenced, Tensor Processing Units (TPUs [8]) were just beginning to gain recognition, and Graphics Processing Units (GPUs), although used for deep learning, were often inadequate for some applications. The world was still exploring the potential of Convolutional Neural Networks (CNNs) for image processing and Recurrent Neural Networks (RNNs) for Natural Language Processing (NLP). We found ourselves on the verge where AI was increasingly significant, but not yet dominant.

In the original paper, we described the accelerator design of a deep learning serving stack called Project Brainwave [5], powered by our newly deployed Catapult fleet [4], our general infrastructure of direct-network-attached [3] Field-Programmable Gate Arrays (FPGAs). The genesis of our work here was driven by the escalating demands of Bing and its reliance on NLP, hence our emphasis on RNNs and optimization of the overall system, architecture and microarchitecture for these classes of models.

A critical challenge in deploying RNNs on conventional standalone accelerators and CPUs was balancing the simultaneous demand for low latency while maximizing utilization, particularly complicated given the dependence and parallelism structure of RNNs. Batching, a popular technique for driving up utilization, would not be able to accelerate the performance of a single query for better latency. Taking the predominant view of accelerators at the time as standalone co-processors to CPUs, it would have been impractical to inference long short-term memory (LSTM, a type of RNN) models within the expected tens of ms required of Bing’s real-time service level agreements (SLA). The issue was amplified as the models grew in size.

II. BRAINWAVE ARCHITECTURE

The Brainwave system architecture exploited model parallelism in a unique way where LSTM weights were carefully partitioned and pinned in on-chip SRAM across multiple Brainwave NPUs (each mapped to a single FPGA). The distribution across multiple FPGAs allowed scale up of LSTM models up to dozens of devices, while also enabling the local memory footprint of each device to fit within the on-chip SRAM. With this top-down strategy in mind, the Brainwave Neural Processing Unit (NPU) was then co-optimized for running on-chip SRAM-pinned GEMV kernels. It was uniquely optimized for low latency and featured chained instructions that fused multiple SIMD operations. These optimizations significantly improved area, allowing for increased resources for tensor and SIMD functional units. The Brainwave NPU also made efficient use of narrow-precision MSFP [6] datatypes, with as little as 4 bits of precision for weights and activations.

With these features, a single Brainwave NPU accelerator on a standard FPGA could deliver tens of effective TFLOPs, a stark contrast to the hundreds of GFLOPS achievable when running on a single device. This level of performance on memory-bound LSTMs was unprecedented at the time of the paper’s publication and could not be matched by contemporary CPUs and GPUs.

Since 2015, Brainwave has undergone significant advancements and evolutions. Over this period, we have developed and deployed at least five generations of soft NPUs, strategically integrating them across a fleet spanning three generations of FPGAs: Stratix V, Arria 10, and Stratix 10. These advancements have enabled us to scale production and effectively power over 100 models across Bing and Office, thereby supporting a wide range of services such as ranking, semantic vectorization, question and answer generation, image processing, and query intent understanding [5].

III. WHAT WE GOT RIGHT

Our ability to rapidly update and leverage the flexibility of FPGAs kept Project Brainwave relevant for AI in our cloud for over 7 years. It allowed our deployments to adapt in the rapidly evolving AI space, especially during the major transition points going from CNNs to LSTMs to Transformers. Utilizing flexible hardware also facilitated safe evolution, experimentation, and refinement of these IPs in large-scale production, within the span of weeks to months. This approach instilled boldness in our FPGA teams, fostering an environment where risk-taking and innovation in hardware IP were embraced.

One example of such IP that exists because of Project Brainwave, MSFP [6], [9], enabled us to fit LSTMs with millions of parameters into FPGAs with megabytes of SRAM using 4 bits per value. 4-bit MSFP was considered aggressive in 2015, since most other architectures didn’t have such stringent memory capacity constraints, and quantization had not yet to become mainstream. However, today’s large language models (LLMs) are facing similar pressures, which has resulted in a renaissance of new extreme narrow precision quantization...
techniques in the literature. MSFP still appears relevant for both training and serving state-of-the generative models [9].

Our strategy of insulating programmers from needing to program FPGAs directly in RTL also proved to be successful. We achieved this by defining a lightweight instruction set architecture on top of the ISA as an “overlay” that decoupled HW and SW development. Multiple versioned ISAs enabled software teams to preserve their agility, while rapid hardware evolution continued.

Lastly, the use of model parallelism for inferencing—not widely seen as obvious during 2015—has now become standard practice for scale up of large model inference and training. Distributed SRAM pinning has also become a widely known practice in many of today’s advanced AI accelerators optimized for models that fit within the envelope of a distributed collection of devices with high amounts of SRAM (e.g., [2], [10]).

IV. PITFALLS AND SURPRISES

Our project was not devoid of detours and surprises. The on-chip SRAM approach introduced by Project Brainwave gradually lost its relevance as the compute-bound Transformer-based models surpassed the capacity of on-chip memory hierarchies. These models no longer necessitated aggressive, SRAM-based model parallelism to achieve high utilization. Additionally, Nvidia GPUs aggressively scaled their High-Bandwidth Memory (HBM) and hardened tensor units to improve their AI performance, which eventually became predominant in most mainstream AI processors.

This limited the opportunity space for novel architectures to compete, and despite the development of the Stratix 10 NX FPGA [1], which introduced an HBM-enabled FPGA with hard tensor cores, the scope and scale at which Catapult FPGAs could competitively serve flagship AI models were reduced. Ultimately, the hardware lottery [7] effect made mainstream GPUs a much more attractive target for model evolution.

From an architecture perspective, the Brainwave NPU itself also had shortcomings. Our initial instruction set architecture (ISA) prioritized serial dependencies (critical for LSTMs, but almost irrelevant for Transformers) and lacked the generality of a full general-purpose ISA due to resource and time-to-market constraints. Moreover, flexibility over the RTL, while beneficial, also revealed itself to be a double-edged sword. Managing a diverse array of FPGA images at scale, along with maintaining bidirectional compatibility with software, presented significant deployment challenges.

V. FUTURE OUTLOOK

State-of-the-art AI algorithms are continuously evolving at a rapid pace, well surpassing the advancements made in the field back in 2015 when Project Brainwave was initiated. Recognizing the significance and value of favoring architectures that strike a balance between flexibility and performance during periods of algorithmic uncertainty, Project Brainwave serves as an excellent illustration. It demonstrates how a high-risk/high-reward approach (specifically, overfitting to a particular class of models) can momentarily secure an advantage in the hardware landscape. However, this advantage can just as easily be disrupted when the predominant model direction shifts.

As we relentlessly push the boundaries of large language models, which are trained and deployed on massive dense supercomputers, the future of hardware architectures for AI remains unclear. Will mainstream dense architectures like GPUs continue to scale and dominate, or will a new architectural paradigm eventually emerge to propel AI into its next stage? Moving forward, we continue to watch with great interest the introduction of new technologies, algorithms, and innovative computer architectures that will drive the next wave of breakthroughs in AI.

REFERENCES