Retrospective on “In-Datacenter Performance Analysis of a Tensor Processing Unit”

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, **Boone Severson**, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon
Google, Mountain View, CA

Everyone knows the story of AlexNet blowing open the 2012 ImageNet Large Scale Visual Recognition Challenge [5]. After AlexNet, Google hired its authors and added them to the Google Brain team, which was already working in the areas of speech and image recognition. A key signal soon afterward was that matrix multiplication exceeded 1% of CPU fleet cycles in Google Wide Profiling. Another signal was the analysis by Jeff Dean (a Google Fellow, now the Chief Scientist) that processing a few minutes of speech or video by 100M users would require doubling or tripling the size of the CPU fleet. Other options were clearly required.

Dean’s observation made its way to Andy Phelps, who developed the initial TPU design, including the key decision to be systolic array-based. To lead the project, Dean recruited Norm Jouppi—who was leery about AI given past hyperbole—by noting “each time we try deep learning on something new, it works!”

Our vice president of engineering, Urs Hölzle, said something along these lines at a technical review of machine learning (ML):

> *I’ve seen this kind of thing happen before. I don’t know whether it’ll be in two days or two years, but someone from the Google Brain team is going to show up and tell me that if we had 10x the compute, it will be worth [a lot].*

Before Christmas of 2013, the TPU design ran as a two-track program, with one architecture targeting two platforms, FPGA and ASIC. The side length of the systolic array was parameterized in both our Verilog and our C++ codes. The FPGA design aimed for a side length of 64, as 4096 (64^2) multiply-accumulate cells slightly exceeding the number of DSP multiplier blocks in our target device, while the ASIC design aimed for a side length of 256, and became the TPU (later and fortunately, rebranded TPU v1).

In January of 2014, one colleague on the Speech team ported their MLP-based acoustic model to a GPU and used a high fraction of that GPU’s ~2000 ALUs. When combined with the GPU’s higher clock rate, it meant that the peak of the FPGA design had already been eclipsed by the GPU. We thus pivoted to an ASIC-first program. The FPGA still landed in the datacenter, six months before the ASIC was available; it served as a “pipe cleaner” for all of the Google deployment processes to support a new accelerator in production.

The TPU paper showed that it actually exceeded the original goals of an order of magnitude improvement, with 80x the performance/Watt of contemporary CPUs and 30x of contemporary GPUs [1]. We reflect now, ten years after the TPU project started:

- Early characterizations of TPUs incorrectly interpreted the term “ASIC” to mean a fixed-function device. There is a vast design space that includes general-purpose computers and highly optimized special functional units. The tricky part is to build a chip with enough flexibility—one that is programmable and flexible enough to serve the future market, but with enough specialization to confer an advantage. We considered but rejected building an AlexNet-specific machine, which would not have had the same impact.
- Because our forecasts for deep learning were imperfect, we put a number of “insurance policy” features into the TPU design—things that we were not sure we would use, but which might save us later, after the chip had taped out[1]. Sometimes we were just lucky—the element-wise multiplication and addition features were a late feature request by the Speech team, to support LSTMs. Because the element-wise operations let us mask and combine activations, we could support kernel striding for Inception and software transposes for AlphaGo—two critical features which our early customers had sworn in the definition phase that they would never need.
- The 8-bit quantized arithmetic in TPU v1 was both a basis for the 92 T operations/second peak and a huge programmer headache. In retrospect, we wish we had included numerical mathematicians in the project from the beginning, and that we had written and published a clean numerical basis for the calculations we performed. These issues persist today—loss scaling is a relative of the scale-factor management required by quantized arithmetic—around the standardization efforts for 8-bit floating-point formats and the quest for even smaller representations, whether fixed or floating point.
- We sometimes tout the TPU’s 15-month time from project kickoff to datacenter deployment, much shorter than is standard for production chips. Indeed, subsequent TPUs, which Google relies upon for production, have multiyear design cycles. The fast time-to-market was enabled by a singular schedule focus, not just in architecture—where the 700 MHz clock rate enabled easy timing closure and the aging 28nm process was fully debugged—but also in heroic work by our datacenter deployment teams.
- Programmability and flexibility remain vexing for accelerators today. TPU v1 was a mixture of hard- and easy-to-program facets. On the easy side, each chip had a single thread of control, where the programmer had complete

[1] Not using an insurance policy is not a failure—rather, a lack of unused insurance policies means that you were underinsured.

control over all resources and could reason analytically about how much time an operation would take. On the hard side, the quantized arithmetic and the schedule-rushed jumble of ISA features (at least three different integer representations) made seemingly simple tasks much harder ("why is it hard to do long multiplication?"). The tiny, CISC instruction set was a double-edged sword, preventing us from reusing general-purpose compilers but also entirely removing the need for many compiler optimizations.

- We identified memory bandwidth as a fundamental limitation in TPU v1, which often operated under the memory-bound part of the roofline [1]. Bandwidth remains a critical issue today, even with HBM3 and superpod-scale interconnects.
- We had low expectations for the impact of ML; maybe it could identify a few objects in images and help translation? Today's expectations for ML span a range from apocalypse to utopia; we hope very much to be closer to the latter.

Google's announcement in May 2016—that TPU v1 had 10x the performance/Watt of any FPGA and GPU for ML inference and was used by AlphaGo to beat Lee Sedol—was like an earthquake that shook Silicon Valley.

Intel went on a buying spree of companies with a domain specific architecture (DSA) for ML: Nervana, Movidius, MobilEye, and Habana. Hyperscalers like Alibaba and Amazon and end users like Tesla started their own inference DSA chips. The venture community also reacted, investing $3B per year from 2016 to 2020 in more than 100 ML DSA startups. Some startups bet on novel ideas that didn't succeed for general purpose computing such as analog computation, asynchronous logic, and wafer scale hardware. (It's unclear today if the startups made the same investment in their software stacks for ML as they did in their hardware.) Inside Google, we've announced four successors to TPU v1 so far. Drawing an historical analogy to Helen of Troy—"the face that launched a thousand ships"—we say tongue-in-cheek that TPU v1 "launched a thousand chips."

The TPU v1 paper [1] itself has had research impact as well. At the time of this writing, it has 4365 citations, which despite being published recently is the second most cited paper over the 50 years of ISCA and the most cited over the last 5 years [10]. One unusual feature is its 76 authors. We tried to include everyone who participated in any phase of the hardware or software design or deployment. One regret, corrected in this retrospective, is that we still missed some of our collaborators, including Boone Severson. Later TPU papers followed ACM policy that authors must participate in the writing of the paper, moving those who didn't to the acknowledgements.

One consequence of publishing the TPU v1 paper is that it made us realize that there were no well-established benchmarks for ML. Thus, some of us helped start an effort to create an ML benchmark, now known as MLPerf [8]. MLPerf is now the ML equivalent of the SPEC benchmark for CPUs [6], and we quote MLPerf results in all the subsequent TPU papers [2,3,4,7].

While this paper on TPU v1 got rave reviews at ISCA 2017, ASPLOS rejected a subsequent submission on TPU v2 that focused on training, and ISCA rejected a revision that added TPU v3. One opined there were no new ideas and another that it couldn't be research since it was already built! We published papers on TPU v2/v3 elsewhere [2,7], so you can decide if you agree or not with the program committees of 2019. (We don't!)

After commenting at the next ISCA business meeting that current program committees might not appreciate papers on real architectures from industry as they did in the past, Dave Patterson was charged with developing an ISCA industry track [9]. It started in 2021 and has become a popular feature at ISCA, with the industry session typically opening or closing the conference. Given program committees that valued innovative hardware built by industry, later TPU papers were well received [3,4].

We're delighted that Google management lets us write retrospective papers about novel hardware after deployment, and we encourage other architects in industry to follow suit. The lure of prestigious publication helps get busy people to take the time to do a thorough postmortem of design decisions and comparison to external alternatives, which can be rare in industry.

A reason such a post-deployment analysis is rare is that the architects of the last computer are already working hard on the next one. The subsequent publication of such an analysis is even rarer because unlike academia, no one gets promoted for publishing; authors have to be self-motivated. Many architects at Google have PhDs, so they've been indoctrinated early to value papers even if their managers don't reward publication.

Fred Brooks' advice—"Plan to throw one away. You will anyway"—was a guiding principle during the design of TPU v1. It allowed us to choose good-enough solutions while keeping to our schedule. We are proud that today, tens of thousands of these fully amortized, low-power inference DSAs are still running ML jobs in Google data centers eight years after their initial deployment.

REFERENCES

[1] Jouppi, N.P., et al., 2017. In-datacenter performance analysis of a tensor processing unit. In 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA), pp. 1-12.

[2] Jouppi, N.P., Yoon, D.H., Kurian, G., Li, S., Patil, N., Laudon, J., Young, C. and Patterson, D., 2020. A domain-specific supercomputer for training deep neural networks. *Communications of the ACM*, 63(7), pp. 67-78.

[3] Jouppi, N.P., Yoon, D.H., Ashcraft, M., Gottscho, M., Jablin, T.B., Kurian, G., Laudon, J., Li, S., Ma, P., Ma, X., Norrie, T., Patil, N., Prasad, S., Young, C., Zhou, Z., and Patterson, D. 2021. Ten lessons from three generations shaped Google's TPUv4i. In 2021 ACM/IEEE 48th Annual ISCA, pp. 1-14.

[4] Jouppi, N.P., Kurian, G., Li, S., Ma, P., Nagarajan, R., Nai, L., Patil, N., Subramanian, S., Swing, A., Towles, B., Young, C., Zhou, X., Zhou, Z., and Patterson, D., 2023. TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. in 2023 ACM/IEEE 50th Annual ISCA.

[5] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), pp. 84-90.

[6] Mattson, P., Cheng, C., Diamos, G., Coleman, C., Micikevicius, P., Patterson, D., Tang, H., Wei, G.Y., Bailis, P., Bittorf, V. and Brooks, D., 2020. MLPerf training benchmark. Proceedings of Machine Learning and Systems, 2, pp. 336-349.

[7] Norrie, T., Patil, N., Yoon, D.H., Kurian, G., Li, S., Laudon, J., Young, C., Jouppi, N. and Patterson, D., 2021. The design process for Google's training chips: TPUv2 and TPUv3. *IEEE Micro*, 41(2), pp. 56-63.

[8] Patterson, D., Diamos, G., Young, C., Mattson, P., Bailis, P., and Wei, G.-Y., MLPerf: A benchmark suite for machine learning from an academic-industry cooperative, Artificial Intelligence Conference, May 2, 2018.

[9] Patterson, D., Genesis and Reflections on the Return of Industry Products to ISCA, Computer Architecture Today, July 15, 2020

[10] Patterson, D., What Are the Most Cited ISCA Papers?, Computer Architecture Today Blog, June 15, 2023.