Accelerating AI Performance in an Open Software Ecosystem

By Linley Gwennap
Principal Analyst

September 2019

The Linley Group

www.linleygroup.com

Accelerating AI Performance in an Open Software Ecosystem

By Linley Gwennap, Principal Analyst, The Linley Group

Although many new vendors are challenging Nvidia in AI accelerators, Nvidia's software stack, based on its proprietary CUDA, creates a high barrier to entry. Duplicating CUDA is a difficult task, but customers need the flexibility of its programming environment. SYCL is the only open alternative to CUDA that has multivendor support, enabling customers to move to higher-performance hardware while retaining software flexibility. Codeplay sponsored this white paper, but the opinions and analysis are those of the author. Trademark names are used in an editorial fashion and are the property of their respective owners.

Introduction

After years of relying on Nvidia chips for running most applications, artificial-intelligence (AI) researchers have a growing number of hardware options. Later this year, Intel plans to release its first accelerators based on its Nervana NNP technology. AMD has improved the AI performance of its GPU chips. Xilinx released FPGA-based cards that come with preprogrammed acceleration logic. Startups such as Graphcore, Habana, and Wave are shipping accelerators today, and others will soon join them.

Many of these emerging vendors have made extravagant performance claims, often demonstrating their accelerators running ResNet-50 or similar neural networks several times faster than Nvidia's best accelerators. For example, Xilinx's Alveo U280 achieved 4,127 inferences per second (IPS) on GoogleNet at a batch size of 1, more than 2.7x the speed of an Nvidia T4 card. Habana measured its Goya card at 15,393 IPS on ResNet-50, four times faster than the T4's top speed.

These performance gains have caught the attention of many AI developers, but so far we have seen little deployment of alternative solutions. Software is a critical problem. New hardware vendors often demonstrate their accelerators using a proprietary graph compiler with limited integration to frameworks such as TensorFlow and PyTorch. Some of these vendors support one or two frameworks, but only with a limited set of operations that is insufficient to run many modern neural networks. As developers evaluate these new products, they often find their production networks, which are typically far more complex than GoogleNet or ResNet, don't achieve the expected performance or fail to compile at all.

As AI-accelerator vendors attempt to port open frameworks to their new architectures, one challenge is that these frameworks are written atop Nvidia's CUDA software stack, which runs only on Nvidia GPUs. Thus, accelerator vendors must duplicate the extensive CUDA APIs, a time-consuming task. To help solve this problem, the Khronos Group created SYCL (pronounced "sickle"), a royalty-free cross-platform software stack that provides a close analog to the CUDA APIs, simplifying the porting of frameworks and other AI applications. Khronos Group members, including Nvidia, allow designers to use their IP to implement Khronos standards, providing SYCL users with greater legal protection than users of proprietary CUDA-like software.

Codeplay now offers a commercial version of SYCL, branded ComputeCpp, and provides support for vendors wishing to develop SYCL-based software stacks on new architectures. This approach, along with other SYCL implementations under development at major companies such as Intel and Xilinx, enables accelerator vendors to more quickly support a broad range of AI applications on their platforms.

Current Software Environment

The current wave of AI research, based on deep neural networks (DNNs), is still in its early stages. DNNs first came to note when AlexNet smashed the ImageNet accuracy record in 2012. Within a few years, newer DNNs surpassed human accuracy on the ImageNet challenge. To tackle more complex tasks, researchers have devised many different types of DNNs, such as convolutional networks, recurrent networks, long/short-term memory, multilayer perceptrons, sparse networks, and adversarial networks. Some are better suited to images, others to voice, and so on. To improve performance, developers continue to modify and evolve these models and even create new ones. The biggest data-center operators, such as Google and Baidu, develop their own AI software, deploying a variety of networks that are each optimized for a specific service (such as image search or language translation).

In this constantly changing environment, developers rely on a standard and flexible software platform. To test new algorithms, they may develop code using CUDA, which offers a broad set of C++ libraries that implement computation and AI functions, simplifying application development. Over time, open frameworks have become the most popular DNN development path, as they are much faster and easier than coding by hand.

These frameworks are generally developed on Nvidia GPUs using CUDA; this approach greatly reduces development time compared to creating all of the code from scratch. As a result, Nvidia easily supports all popular frameworks on its accelerators. Furthermore, the accelerator vendor has a large software team that has spent years expanding and optimizing CUDA to run efficiently on its GPU chips. Thus, these frameworks are easy to compile for Nvidia cards, and Nvidia's software team constantly optimizes leading frameworks to perform well on its hardware. In fact, DNN performance on the same hardware can improve dramatically when Nvidia releases new software. For example, a new software release last year boosted the company's DGX-1 system from 4,200 IPS to 7,850 IPS on ResNet-50.

Competitors with smaller software teams find it difficult to match this rapid pace of development. Although most frameworks are available as open source, the underlying CUDA code is restricted to Nvidia chips. Thus, implementing a framework on a new architecture, such as Alveo or Nervana, requires building a new set of routines to replace the CUDA code. TensorFlow and similar frameworks offer hundreds of different operations; to simplify their porting efforts, accelerator vendors often start with a limited number. Even large companies such as Intel and Qualcomm support about 10% of all the TensorFlow operations that Nvidia supports, as Figure 1 shows.

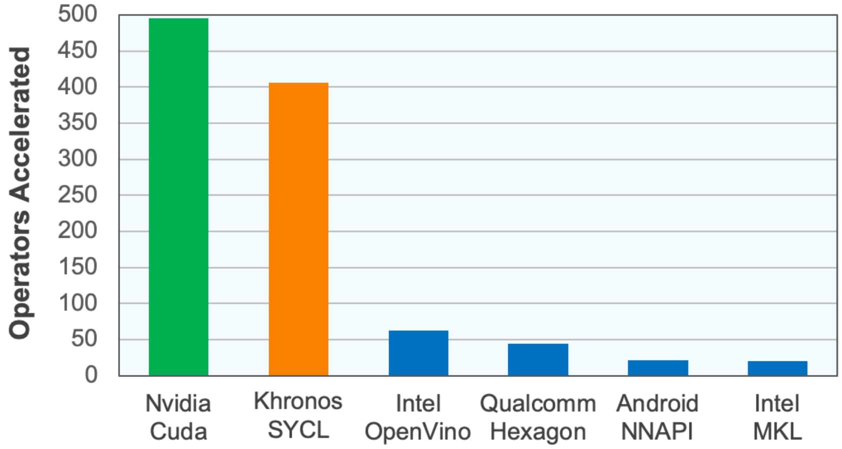

Figure 1. TensorFlow compatibility. Lacking CUDA, most competitors offer a small fraction of the TensorFlow operations that Nvidia supports. Only SYCL approaches Nvidia's compatibility. (Source: Codeplay)

These vendors focus on the most common operations, typically those required for a simple model such as ResNet. This approach allows the vendor to test and optimize performance on a few models, eventually leading to publishing an impressive score. Although ResNet was once state of the art, winning the ImageNet challenge in 2015, the fast-moving field has moved well beyond this level today, implementing many new models and new algorithms that require different calculations. If a modern model uses even a single function that an accelerator vendor hasn't implemented, it won't run on that platform.

Furthermore, CUDA is used for many tasks other than neural networks. A typical end application might combine a neural network with computer vision code (perhaps using OpenCV) and other computation that uses CUDA kernels. Even if an accelerator can run the neural network using a standard framework, the remaining code is harder to handle. This situation often keeps customers locked into CUDA—and Nvidia.

Moving to Open Standards

OpenCL (Open Computing Language) offers a C-language API for writing programs that can execute efficiently on different types of processors, including CPUs, GPUs, and DSPs. Originally developed by Apple, it is now an open standard administered by the Khronos Group. Most major CPU and GPU vendors, including Nvidia, now support OpenCL on their systems. (Ironically, Apple has recently deprecated OpenCL in favor of its proprietary Metal platform.) Although OpenCL is positioned as an open alternative to CUDA, it was never intended as a direct replacement. Its APIs offer similar functions but are not one-for-one compatible, and it is programmed in C rather than C++. Thus, code that is already written for CUDA must be rewritten for OpenCL.

AMD has attempted to break Nvidia's proprietary lock-in by developing HIP, an open-source copy of CUDA. It implements the same C++ API calls but (for legal reasons) renames them from, for example, CUDAMalloc to hipMalloc. Other vendors offer CUDA-like alternatives, including Intel OpenVINO and Xilinx Edge. Even if some are technically open source, they are essentially developed and maintained by a single vendor for a single platform, so customers that use these platforms instead of CUDA are simply trading one proprietary lock-in for another.

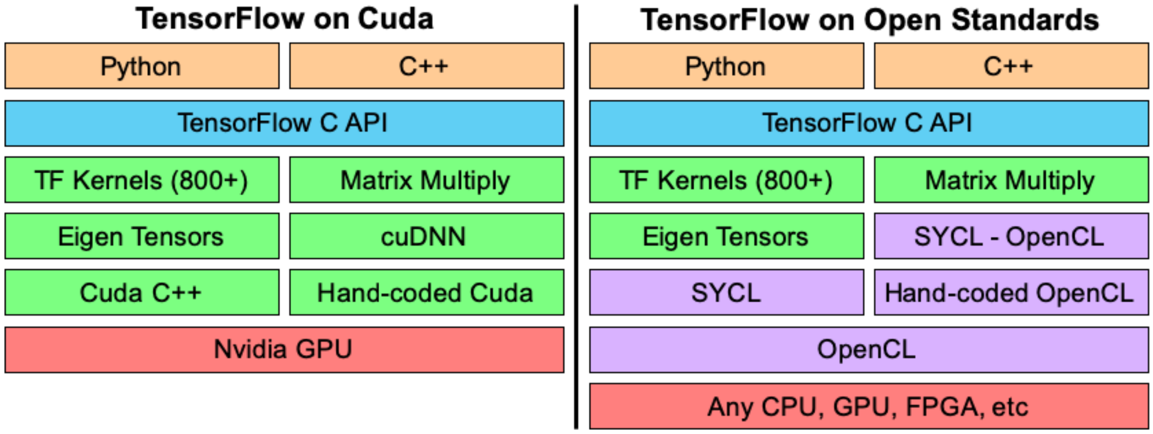

Figure 2. Porting TensorFlow to open standards. SYCL enables TensorFlow (TF) and other AI frameworks to run on any type of hardware using open software stacks. (Source: Codeplay)

SYCL is the only open alternative to CUDA that has multivendor support. Arm, Codeplay, IBM (Red Hat), Imagination, Intel, Renesas, and Xilinx support SYCL. A version called HipSYCL runs on AMD GPU. Like CUDA, SYCL allows developers to program in C++ while using API calls to access hardware-specific acceleration functions. SYCL converts these API calls and C++ code to an intermediate representation (called SPIR-V) that is compatible OpenCL, allowing it to run on the many hardware platforms that offer OpenCL support.

For example, Figure 2 illustrates how TensorFlow was ported from CUDA to SYCL. TensorFlow builds its compute kernels using Eigen, a high-level C++ library for mathematical operations that is typically implemented using CUDA. Codeplay created a compile-time tool that maps Eigen expressions to SYCL expressions whenever possible, focusing on the Eigen tensor modules. This approach handles about 500 of the 800+ TensorFlow kernels, providing much greater compatibility than any manual TensorFlow port. For better performance, Codeplay hand-optimized the frequently used matrix-multiply and convolution kernels, much as Nvidia has hand-coded these functions in CUDA.

Codeplay Enables Open AI

Codeplay has long experience in developing and optimizing software for hardware accelerators, dated back to its work on VectorC for the SonyPlaystation 2 in 2002. The company began working on vision-based software for Google's Project Tango in 2014 and has since ported SYCL to the Arm Mali GPUs, Renesas R-Car, Imagination PowerVR, and other hardware platforms. In addition to SYCL, it has driven several other open-source projects including SPIR-V tools for videogame acceleration, the Vulkan graphics API, and OpenCL for Android. Codeplay often works closely with hardware vendors to optimize open-source performance on their platforms.

The company offers a fully supported and optimized version of OpenCL as part of its ComputeAorta toolkit. This product actually supports both SPIR-V and Vulkan as well and runs on a variety of platforms including Android, Linux, and Windows on Arm, Mips, and x86 hardware. The toolkit is modular, so customers can build in only the standards that they need. It's built to use the LLVM compiler, so the code is easily ported to new architectures. Customers can license the source code to perform their own ports, or Codeplay can handling any porting and customization, simplifying the customer's development.

The company has developed an implementation of the SYCL standard called ComputeCpp. It recently announced plans to give software developers free access to this code, encouraging development of open-source software stacks. The current early-access version supports SYCL version 1.2 and runs Eigen and TensorFlow, as described above. ComputeCpp is built on OpenCL and supports a wide range of hardware including AMD and Intel CPUs and GPUs. Codeplay can assist customers in porting and tuning ComputeCpp for other hardware platforms.

Conclusion

New accelerator vendors, both large and small companies, are challenging Nvidia's lead in AI. Most of these vendors have focused on demonstrating high performance on a few simple neural networks. But customers need more than just fast hardware; they need to run many different types of networks. Furthermore, customers want to evolve their algorithms and even create new types of models in the future. To address these needs, Nvidia's software stack, based on its proprietary CUDA, provides a flexible programming environment.

Competing vendors have found it difficult to match Nvidia's software. Lacking CUDA, they have typically tried to hand-code a basic set of operations for TensorFlow and other popular AI frameworks, but this approach leaves 90% of the functions unsupported. Because of this limited support, customer applications written for Nvidia often fail to compile on other hardware, or the performance is far slower than anticipated.

An open software ecosystem addresses these shortcomings. Instead of each vendor having to start from scratch, they can adopt open standards that provide access to existing code. For example, SYCL provides an open alternative to CUDA that already has broad industry support. Whereas single-vendor efforts support only a few dozen TensorFlow operations, SYCL supports more than 400, greatly improving compatibility and enabling customers to innovate on non-Nvidia platforms. To further ease software development, Codeplay can assist accelerator vendors as they shift to these open standards, providing fully supported and optimized implementations of SYCL and OpenCL and helping to port them to new platforms.

Linley Gwennap is principal analyst at The Linley Group and editor-in-chief of Microprocessor Report. *The Linley Group offers the most-comprehensive analysis of microprocessors and SoC design. We analyze not only the business strategy but also the internal technology. Our in-depth articles cover topics including embedded processors, mobile processors, server processors, AI accelerators, IoT processors, processor-IP cores, and Ethernet chips. For more information, see our website at www.linleygroup.com.*