NVIDIA A100 Tensor Core GPU: Performance and Innovation

Jack Choquette, Wishwesh Gandhi, Olivier Giroux, Nick Stam, and Ronny Krashinsky, *NVIDIA, Singapore, 609927*

NVIDIA A100 Tensor Core GPU is NVIDIA's latest flagship GPU. It has been designed with many new innovative features to provide performance and capabilities for HPC, AI, and data analytics workloads. Feature enhancements include a Third-Generation Tensor Core, new asynchronous data movement and programming model, enhanced L2 cache, HBM2 DRAM, and third-generation NVIDIA NVLink I/O.

NVIDIA A100 TENSOR CORE GPU OVERVIEW

The diversity of compute-intensive applications in modern cloud data centers has driven the explosion of GPU-accelerated cloud computing. Such applications include AI deep learning (DL) training and inference, data analytics, scientific computing, genomics, edge video analytics and 5G services, graphics rendering, and cloud gaming. The NVIDIA Ampere architecture-based A100 GPU brings record-setting scale and novel capabilities to these workloads. The A100 GPU is 54 billion transistors built on TSMC's 7-nm process. It has 108 streaming microprocessors (SMs) with 6912 CUDA cores, 40 MB of L2 cache, 600 GB/s of NVIDIA NVLink interconnect bandwidth, and 1.6 TB/s of HBM2 memory bandwidth. It also has new elastic GPU capabilities including scale-out support with multi-instance GPU (MIG) virtualization, and scale-up support with a third-generation 50-Gb/s NVLink I/O interface connecting multiple A100s directly or through NVIDIA's NVSwitch. Inside the A100 SM are new third-generation tensor cores with support for fine-grain sparsity and new BFloat16 (BF16), TensorFloat-32 (TF32), and FP64 datatypes. The SM also adds new asynchronous data movement instructions and barriers which work together to efficiently stream data in a programmer-friendly way.

A comparison shows that A100 provides dramatically higher performance than currently available commercial designs and NVIDIA's previous generation V100 GPU.[1] A100 runs approximately 1.5× to over 2× faster than V100 on important HPC applications within molecular dynamics, physics, engineering ,and geo science areas (see Figure 1). For DL workloads, DGX super-pods with A100 have set records for the MLPerf benchmark,[2] handily surpassing all other commercially available systems, including Google's TPUv3 and Huawei's Ascend systems (see Figure 2). The benchmark also demonstrates A100's breadth of support for AI networks, the only system able to run all benchmarks, and run them with high performance.

Many new and innovative features in A100 contribute to its high performance and capabilities. This article will cover a few of those improvements: top to bottom features to support strong scaling, elastic GPU capabilities to support scale up and scale out, and asynchronous programming features to enable efficiency and programmer productivity.

STRONG SCALING: TOP TO BOTTOM

A typical deep neural network consists of long chains of interconnected layers (see Figure 3). While there is a massive amount of compute in a network, the parallelism is broken up into layers of smaller, sequentially dependent chunks of work. Each layer takes an input activation tensor and weight tensor, performs an operation similar to a matrix multiplication, and outputs an activation tensor. To leverage the large amount of compute available in a GPU, the output tensor is broken down into smaller tiles which are distributed across the different SMs.

A100's Tensor Core throughput is 2.5 times higher on dense FP16 data than the previous generation V100 GPU. In weak scaling, both the network size and parallelism must grow to match the increased throughput. In strong scaling, both the network size and available

0272-1732 © 2021 IEEE
Digital Object Identifier 10.1109/MM.2021.3061394
Date of publication 23 February 2021; date of current version 26 March 2021.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on September 28,2021 at 09:56:54 UTC from IEEE Xplore. Restrictions apply.

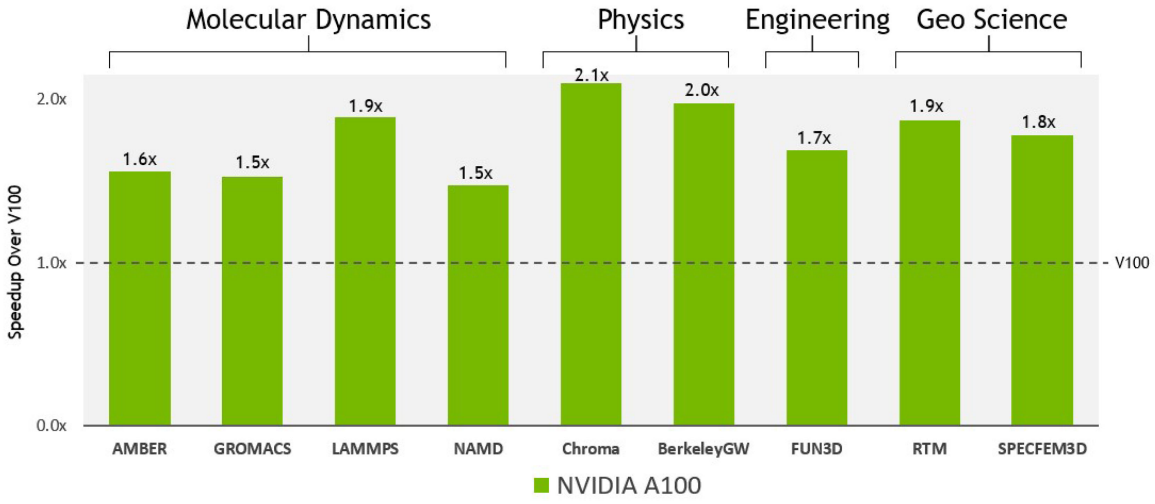

FIGURE 1. NVIDIA A100 HPC performance HPC relative to NVIDIA V100, normalized to per chip.

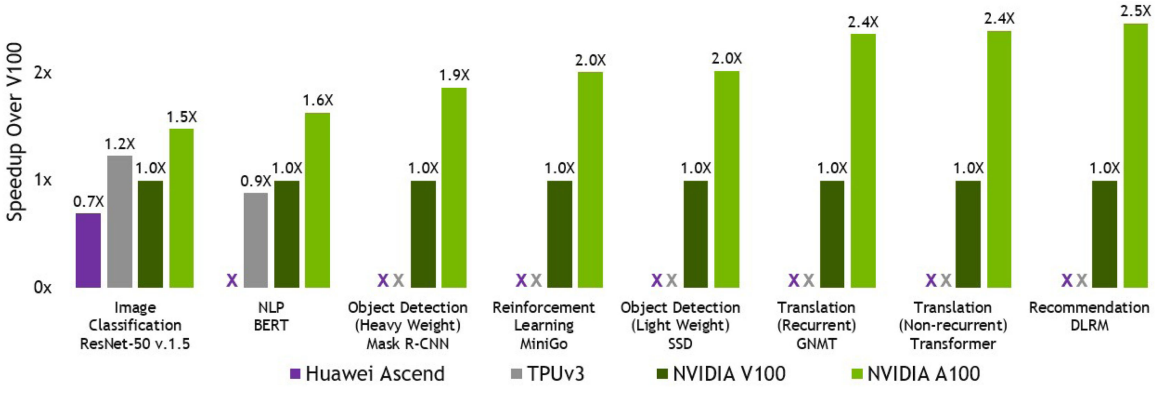

FIGURE 2. MLPerf v0.7 performance for NVIDIA A100 and other commercially available DL systems, relative to NVIDIA V100 & normalized to per chip.

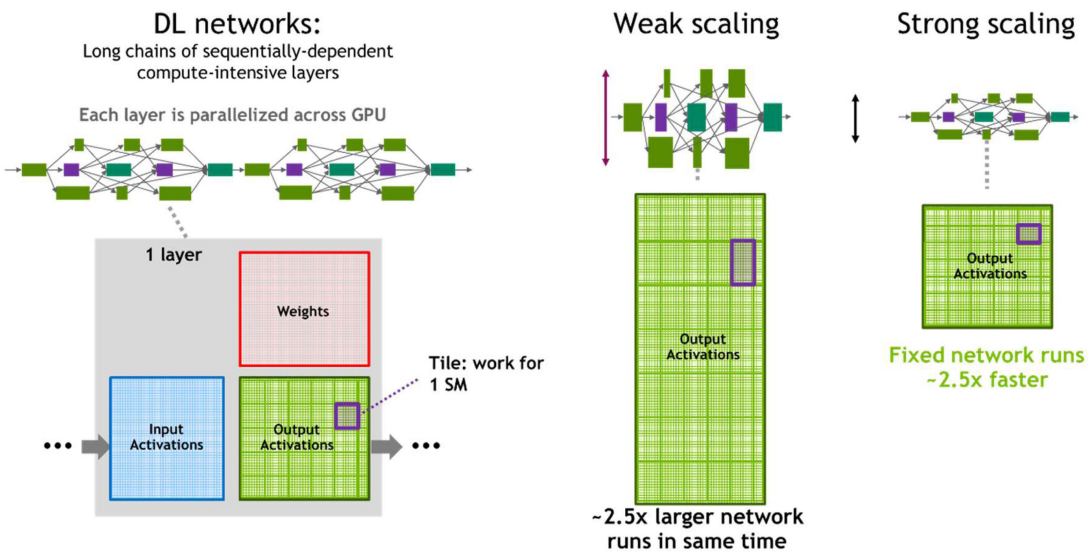

FIGURE 3. DL network mapping to GPU.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on September 28,2021 at 09:56:54 UTC from IEEE Xplore. Restrictions apply.

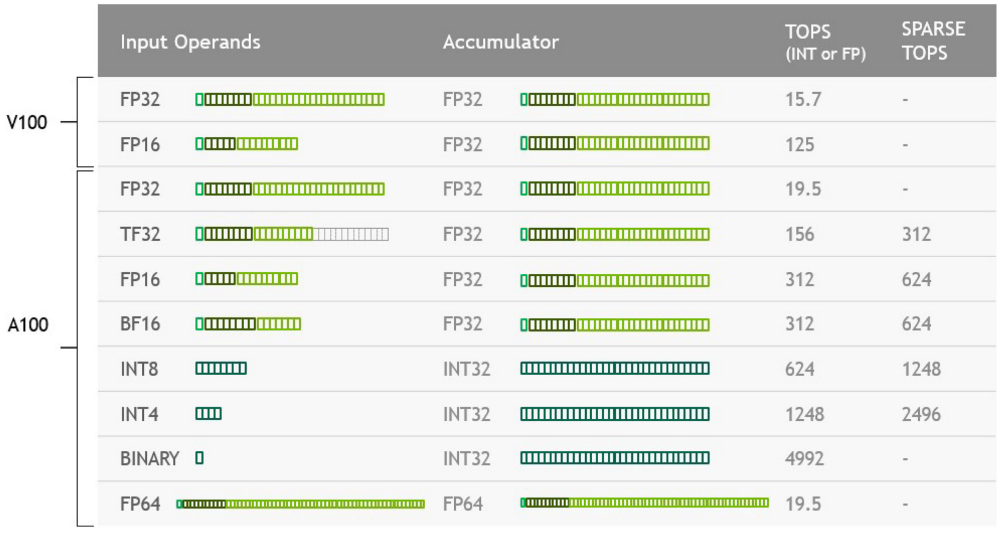

FIGURE 4. NVIDIA A100 & V100 tensor core formats and performance.

parallelism remain fixed, and A100 runs the network faster.

To achieve strong scaling, A100 needed to scale performance at all levels: from the tensor cores, through the cache hierarchy, through DRAM, and chip interconnect.

SM Core

A100's new tensor cores have increased the processing of dense FP16 data by 2x per SM and 2.5× per GPU over V100 (see Figure 4). The tensor cores added support for additional formats, including the TensorFloat-32 operation that improves the processing of FP32 data by 10× per GPU. A100's tensor cores also support fine-grain sparsity, which doubles the throughput when processing sparse data.

The 2× increase in FP16 math throughput per SM for dense data requires 2× more data bandwidth, and the effective 4× increase for sparse data requires 3× more data bandwidth.

In the SM core, multiple improvements in data delivery provide this increase in data bandwidth (see Figure 5). Inside the V100 and A100 SM, the network layer tile is further broken down into four smaller tiles which are each processed by a 32-thread warp. V100's tensor cores were designed to work at 8-thread granularity, and required the tiles to be further broken down to four smaller tiles per warp. Each of these tiles loads the tensor data from shared memory (SMEM), which in aggregate requires all data to be loaded four times. In A100, the tensor cores were reorganized and enhanced to work at 32-thread granularity. This reduces the number of times data needs to be loaded and reduces SMEM bandwidth by half.

Data transfer efficiency from the memory system was also improved. In V100, data must first be loaded into the register file and then stored into SMEM. In A100, a new asynchronous combined load-global-store-shared instruction was added which transfers data directly into SMEM, bypassing the register file and increasing efficiency.

Combined with the Tensor Core organization changes, A100 reduces 6 L1+SMEM accesses (L1 read, SMEM write, 4 SMEM reads) down to only two SMEM reads. The asynchronous capability of the transfer of memory also helps the SM to continuously stream data, improving utilization throughout the memory system.

A100 L2 Cache

The A100 L2 cache is a shared resource for the SMs. Additional bandwidth from the L2 cache is necessary to achieve strong scaling, as the number of SMs is increased in A100 and each SM processes data at a faster rate. A100 delivers a 2.3x L2 bandwidth increase over V100, supported by a new structure to efficiently move this data.

The L2 cache is divided into two partitions to enable higher bandwidth and lower latency memory access (Figure 6). Each L2 partition localizes and caches data for memory accesses from SMs directly connected to the partition. Hardware cache-coherence maintains the CUDA programming model across the full GPU, and applications will automatically leverage the bandwidth and latency benefits of A100's new L2 cache.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on September 28,2021 at 09:56:54 UTC from IEEE Xplore. Restrictions apply.

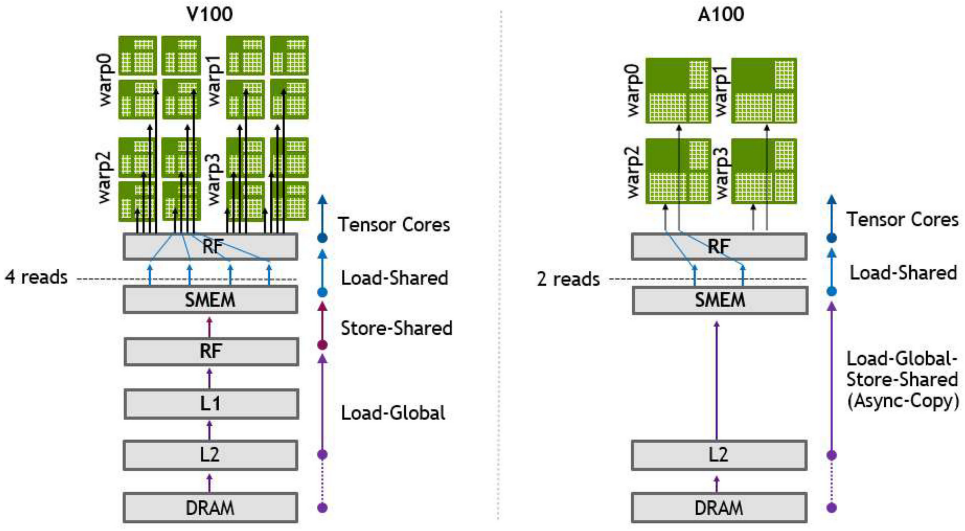

FIGURE 5. Data delivery for V100 and A100.

L2 Cache Residency Controls

A100 gives applications the capability to influence the persistence of data in the L2 cache, allowing higher bandwidth and lower latency accesses to the global memory. The so-called persistent accesses control the replacement policy to effectively set-aside a portion of L2 cache. Normal or streaming accesses to global memory can only utilize this portion of L2 when it is unused by persistent accesses.

There are two primary mechanisms that allow persistent data to be resident in the L2. In the first, an address-based window is specified where all the read/write accesses are persistently cached in the L2. Individual accesses are not tagged in this scheme. Alternatively, controls can be specified at a finer-grained, per-memory-operation basis.

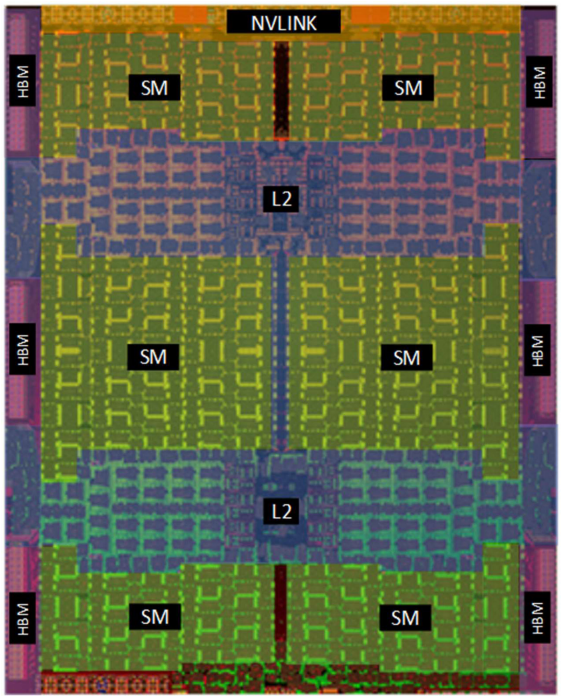

FIGURE 6. NVIDIA A100 die.

A100 DRAM Bandwidth

The A100 GPU includes 40 GB of fast HBM2 DRAM memory on its SXM4-style circuit board. The memory is organized as five active HBM2 stacks with eight memory dies per stack. With a 1215 MHz (DDR) data rate the A100 HBM2 delivers 1555-GB/s memory bandwidth, which is more than 1.7x higher than Tesla V100 memory bandwidth.

A100 Compute Data Compression

To boost DRAM bandwidth and L2 capacity, A100 implements a data-compression feature for sparsity and other compressible data patterns. Compression in L2 provides up to 4× improvement to DRAM read/write bandwidth, up to 4× improvement in L2 read bandwidth, and up to 2× improvement in L2 capacity.

Compression is initiated by marking buffers available for compressibility using new APIs in CUDA 11. Data written to these buffers is examined inside the L2 by the hardware. For specific data patterns that match the hardware algorithms, the data are compressed and written back to the L2. Any subsequent

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on September 28,2021 at 09:56:54 UTC from IEEE Xplore. Restrictions apply.

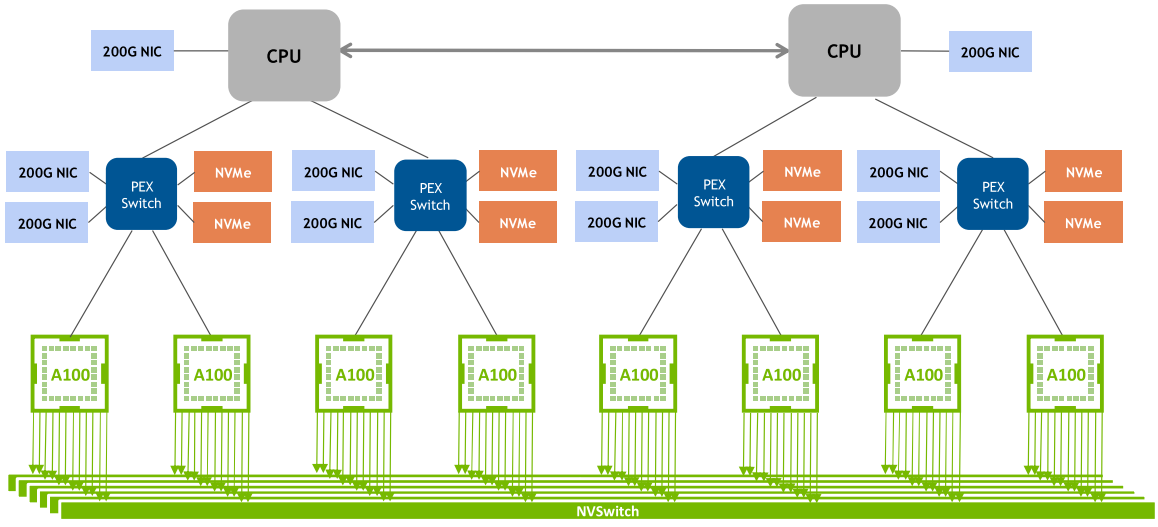

FIGURE 7. NVIDIA DGX-A100 system block diagram.

access will take advantage of the compressed data bandwidth. Compression helps both the write and read accesses to DRAM and increases the effective DRAM bandwidth available.

A100 Gen 3 NVLink

The third generation of NVIDIA's high-speed NVLink interconnect is implemented in A100 and the new NVSwitch. NVLink is a reliable, high bandwidth, low-latency memory interconnect, and includes resiliency features such as link-level error detection and packet replay mechanisms to guarantee successful transmission of data. The new NVLink has a data rate of 50 Gb/s per signal pair, nearly doubling the 25.78-Gb/s rate in Tesla V100.[3] Each link uses four differential signal pairs (four lanes) in each direction compared to eight signal pairs (eight lanes) in V100. A single link provides 25-GB/s bandwidth in each direction similar to V100 GPUs, but uses only half the signals compared to V100. The total number of NVLink links is increased to 12 in A100, versus six in V100, yielding 600-GB/s total bandwidth for an entire A100 versus 300 GB/s for Tesla V100. The twelve NVLink links in each A100 allow a variety of configurations with high-speed connections to other GPUs and switches.

All writes in the third-generation NVLink now require an acknowledgement from the destination. This allows synchronization to be performed at the requester, and error attribution to be returned to a specific execution context. Writes are allowed to be pipeline to the destination while the requestor is waiting on a response. New features to improve the efficiency of small payload writes and data-less responses were also added.

ELASTIC GPU: MULTI-GPU SCALE UP

The twelve NVLink links in each A100 allow a variety of configurations with high-speed connections to other GPUs and switches. To meet the growing computational demands of larger and more complex DNNs and HPC simulations, the new NVIDIA DGX A100 system (Figure 7) includes eight A100 GPUs connected by the new NVLink-enabled NVSwitch.

Multiple DGX A100 systems can be connected via a networking fabric like NVIDIA Mellanox InfiniBand and Mellanox Ethernet to scale out data centers, creating powerful supercomputer-class systems. More powerful NVIDIA DGX POD and NVIDIA DGX SuperPOD systems will include multiple DGX A100 systems to provide much greater compute power with strong scaling.

ELASTIC GPU: MULTI-INSTANCE GPU (MIG)

While many data center workloads continue to scale, both in size and complexity, some acceleration tasks are not as demanding, such as early-stage development or inference on simple models at low batch sizes. Data center managers aim to keep resource utilization high, so an ideal data center accelerator not only needs to efficiently handle a large workload—it also efficiently accelerates many smaller workloads.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on September 28,2021 at 09:56:54 UTC from IEEE Xplore. Restrictions apply.

```
#include <cassert>
__shared__ int buf[1024];
__device__ void example_1(int* src)
{
  buf[threadIdx.x] = src[threadIdx.x];
  // ^^^ the latency of this copy must resolve before the barrier
  __syncthreads();
  // ^^^ this barrier operation immediately blocks execution
  // after unblocking, we can use the data in some computation:
  auto idx = (threadIdx.x+1) % blockDim.x;
  assert(buf[idx] == src[idx]);
}
```

FIGURE 8. Traditional CUDA C++ program with exposed latency.

The new MIG feature can partition each A100 into as many as seven GPU Instances for optimal utilization, effectively expanding access to every user and application. The A100 GPU's new MIG capability can divide a single GPU into multiple GPU partitions called GPU Instances. Each instance's SMs have separate and isolated paths through the entire memory system—the on-chip crossbar ports, L2 cache banks, memory controllers, and DRAM address busses are all assigned uniquely to an individual instance. This ensures that an individual user's workload can run with predictable throughput and latency, with the same L2 cache allocation and DRAM bandwidth even if other tasks are thrashing their own caches or saturating their DRAM interface. Using this capability, MIG can partition available GPU compute resources to provide a defined quality of service (QoS) with fault isolation for different clients (such as VMs, containers, processes, and so on). It enables multiple GPU Instances to run in parallel on a single, physical A100 GPU. MIG also keeps the CUDA programming model unchanged to minimize programming effort.

MIG enables users to see and schedule jobs on virtual GPU Instances as if they were physical GPUs. MIG works with Linux operating systems and their hypervisors.

PRODUCTIVITY: ASYNCHRONOUS PROGRAMMING

The goal of CUDA is to compile intuitive C++ sources into high-performance executable programs for the GPU. In this pursuit, a compiler's goals are in tension: to maximize operation-level parallelism, but never to alter the program's semantics. NVIDIA's work on joint hardware & programming system codesign directly addresses this tension.

Updates to CUDA for A100 expand the expressiveness of C++ for data and computation pipelining, which we identified as a growing source of difficulty for CUDA programmers. The goal of pipelining in software is the same as in hardware: to keep execution resources busy by overlapping the latency of different phases of computation. This is difficult to express in C++ due to its conservative requirements around memory consistency.

The example program in Figure 8 shows that operation-level parallelism is prevented by synchronization semantics. Software pipelining depends on having the opportunity to execute independent work while synchronization is resolving at the boundaries of stages.

The first programming model innovation borrows from asynchronous programming: separate the arrival and waiting steps, as in phasers.[4] Early in development, NVIDIA recognized this innovation would benefit programmers beyond CUDA, so NVIDIA offered both specifications and implementations to the community; it is now part of ISO C++ 20[5] and is available in LLVM (libcxx) today.

The second innovation extends this foundation with asynchronous data movement capabilities. This example program in Figure 9 combines both asynchronous barrier and data movement operations, resulting in a remarkably precise expression of programming intent. By leveraging the relaxed semantics of asynchronous operations, there is a net reduction in the difficulty of compiling the program and performance is both higher and more predictable.

CONCLUSION

NVIDIA's A100 GPU is the largest and most advanced GPU developed by NVIDIA, and builds on the groundwork laid by previous generations of NVIDIA GPUs. It

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on September 28,2021 at 09:56:54 UTC from IEEE Xplore. Restrictions apply.

```
#include <cuda/barrier> //< CUDA's new asynchronous barrier
#include <cassert>
__shared__ int buf[1024];
using barrier = cuda::barrier<cuda::thread_scope_block>;
__device__ void example_2(barrier& b, int* src)
{
  memcpy_async(buf+threadIdx.x, src+threadIdx.x, sizeof(int), b);
  // ^^^ an asynchronous copy dislocates its latency from program order
  auto tok = b.arrive(); //< neither blocks, nor waits for the copy
  //< independent work placed here can hide barrier and copy latency
  b.wait(tok);
  // ^^^ blocks for thread arrivals on b, and copies associated with b
  auto idx = (threadIdx.x+1) % blockDim.x;
  assert(buf[idx] == src[idx]);
}
```

FIGURE 9. Asynchronous data movement and computation.

is the result of the work of thousands of engineers that worked together from transistors to standards and everything in between such as system integration and programming model design. A100 provides unprecedented acceleration at every scale, and adds powerful new features which deliver dramatically faster performance for HPC, AI, and data analytics workloads.

REFERENCES

1. J. Choquette, O. Giroux, and D. Foley, "Volta: Performance and programmability," *IEEE Micro*, vol. 38, no. 2, pp. 42–52, Mar./Apr. 2018.
2. P. Mattson *et al.*, "MLPerf training benchmark," 2019, *arXiv:1910.01500*.
3. A. Ishii *et al.*, "NVSwitch and DGX-2: NVLink-Switching chip and scale-up compute server," *Hot Chips*, 2018.
4. J. Shirako, D. M. Peixotto, V. Sarkar, and W. N. Scherer, "Phasers: A unified deadlock-free construct for collective and point-to-point synchronization," in *Proc. 22nd Annu. Int. Conf. Supercomput.*, 2008, pp. 277–288.
5. International Standard ISO/IEC 14882:2020 – Programming Language C++.

JACK CHOQUETTE is a Senior Distinguished Engineer with NVIDIA, where he has led the architecture development of NVIDIA's GPGPU Streaming Microprocessors for multiple generations. He has been leading CPU and system designs for over 25 years. Choquette received an M.S. in computer engineering from the University of Illinois Urbana-Champaign, Champaign, IL, USA. Contact him at jchoquette@nvidia.com.

WISHWESH GANDHI is a Senior Director of architecture at NVIDIA, Singapore. He has led the architecture development of the GPU memory system for multiple generations. He has been working with Integrated and Discrete GPU memory architecture for more than 20 years. Contact him at wgandhi@nvidia.com.

OLIVIER GIROUX is a Distinguished Architect at NVIDIA, and the ISO C++ Concurrency and Parallelism Chair. He has worked on ten GPU and six SM architecture generations, with a focus on clarifying the programming model of GPU threads. Giroux received an M.S. in computer science from McGill University, Montreal, QC, Canada. Contact him at ogiroux@nvidia.com.

NICK STAM is a Senior Technical Marketing Director with NVIDIA. His team provides tech support to press, and also produces our GPU white papers. Before NVIDIA, he worked at PC Magazine USA, and cofounded the ExtremeTech website. Stam received an M.S. in computer science from SUNY Binghamton, NY, USA. Contact him at nstam@nvidia.com.

RONNY KRASHINSKY is a Distinguished Engineer with NVIDIA where he has architected GPUs for 11 years. He began his NVIDIA career in Research, and later joined the Streaming Multiprocessor team. He now focuses on deep-learning compute architecture. Krashinsky received a Ph.D. in electrical engineering and computer science from Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. Contact him at rkrashinsky@nvidia.com.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on September 28,2021 at 09:56:54 UTC from IEEE Xplore. Restrictions apply.