



NVIDIA®

NVIDIA Grace Hopper Superchip Architecture

Performance and Productivity for Strong-Scaling HPC and Giant AI Workloads

Table of Contents

Inside NVIDIA’s First GPU-CPU Superchip	5
NVIDIA Grace Hopper Overview	6
NVIDIA HGX Grace Hopper Superchip Platform	8
NVIDIA HGX Grace Hopper Superchip with InfiniBand	9
NVIDIA HGX Grace Hopper Superchip with NVLink	9
NVIDIA HGX Grace Hopper Comparisons.....	10
Enhance HGX Grace Hopper with NVIDIA BlueField-3 DPUs.....	11
NVIDIA Grace Hopper Superchip Architecture.....	13
NVLink C2C.....	13
NVLink Switch System.....	13
Accelerating Applications with Extended GPU Memory	14
Flexible Architecture Built for Peak Performance	14
NVIDIA Grace Hopper Programming Model.....	16
Hardware Accelerated Memory Coherency.....	17
Memory Access in NVLink Switch System.....	19
NVIDIA CUDA Platform.....	20
NVIDIA Grace Hopper Accelerated Applications	22
Natural Language Processing Networks	23
Recommender Systems	24
Graph Neural Networks	25
Databases	27
Partially Accelerated Applications.....	28
Molecular Dynamics: GROMACS.....	31
Multi-Grid Linear Solvers.....	33
Appendix A: NVIDIA CUDA Platform	35
CUDA System Software	35
CUDA Profilers and Debuggers	36
CUDA Documentation and Training	37
CUDA Language and Compilers	37

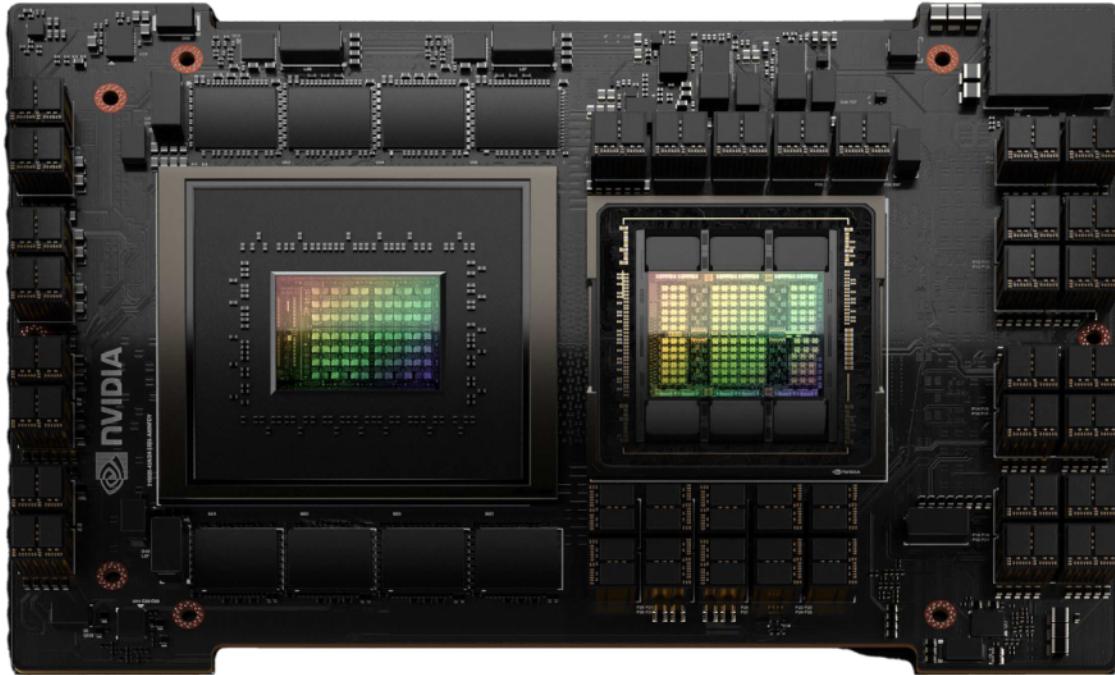
List of Figures

Figure 1. NVIDIA Grace Hopper Superchip Logical Overview	7
Figure 2. NVIDIA HGX Grace Hopper Superchip Platform.....	8
Figure 3. NVIDIA HGX Grace Hopper Superchip system with InfiniBand networking for scale-out ML and HPC workloads.....	9
Figure 4. NVIDIA HGX Grace Hopper Superchip System with NVLink Switch System for strong-scaling giant ML and HPC workloads.....	10
Figure 5. Memory Accesses across NVLink-connected Grace Hopper Superchips.....	14
Figure 6. NVIDIA Grace Hopper Superchip Programming Model.....	16
Figure 7. NVIDIA Hopper System with Disjoint Page Tables.....	17
Figure 8. ATS in an NVIDIA Grace Hopper Superchip System.....	18
Figure 9. Access-Frequency-Based Automatic Memory Migration	19
Figure 10. GPU threads address peer memory from other superchips in the NVLink Switch network	20
Figure 11. NVIDIA CUDA Platform and its ecosystem.....	21
Figure 12. Performance Simulations for Grace Hopper vs x86+Hopper for end-user applications. Datasets and details described in the individual application section below.....	22
Figure 13. A performance simulation for P-tune GPT3 175B NLP Model implemented with offloading	24
Figure 14. A performance simulation for large DLRM network model with 4.5 TB and 27 TB model size, 128 GPUs, and embeddings offloaded to CPU memory.....	25
Figure 15. A performance simulation shows normalized runtime per batch of a GraphSAGE model for an augmented ogbn-products dataset of 626 M nodes and 31 B edges.....	26
Figure 16. Performance simulations for Hash Join with input tables in CPU Memory (left) and host-to-device transfer of pageable host-resident memory (right)	27
Figure 17. A performance simulation for ABINIT with NVBLAS featuring Titanium 255 Atoms using the LOBPCG algorithm.....	29
Figure 18. A performance simulation for OpenFOAM HPC Motorbike L (34 M cells)	30
Figure 19. Performance simulations for GROMACS stmv Benchmark on Grace Hopper Superchip with NVLink Switch System	32
Figure 20. Speedup of NVLink Switch System on Grace Hopper versus InfiniBand NDR400 on NVIDIA Grace Hopper for 3D FFTs of different sizes.....	33
Figure 21. Grace Hopper Simplifies Multigrid Linear Solvers.....	34
Figure 22. Compiling high-level Languages to PTX with libNVVM	38

List of Tables

Table 1. NVIDIA Grace Hopper Superchip Key Features.....	7
Table 2. NVIDIA HGX Grace Hopper Superchip vs. NVIDIA x86+Hopper.....	11

Inside NVIDIA's First GPU-CPU Superchip



The NVIDIA® Grace Hopper architecture brings together the groundbreaking performance of the NVIDIA Hopper GPU with the versatility of the NVIDIA Grace™ CPU, connected with a high bandwidth and memory coherent NVIDIA NVLink Chip-2-Chip (C2C)® interconnect in a single Superchip, and support for the new NVIDIA NVLink Switch System.

NVIDIA NVLink C2C is NVIDIA's memory coherent, high-bandwidth, and low-latency interconnect for superchips. It is the heart of the Grace Hopper Superchip and delivers up to 900 GB/s total bandwidth. This is 7x higher bandwidth than x16 PCIe Gen5 lanes commonly used in accelerated systems.

NVLink C2C memory coherency increases developer productivity, performance, and the amount of GPU-accessible memory. CPU and GPU threads can now concurrently and transparently access both CPU and GPU resident memory, allowing developers to focus on algorithms instead of explicit memory management. Memory coherency allows developers to only transfer the data they need, and not migrate entire pages to and from the GPU. It also enables lightweight synchronization primitives across GPU and CPU threads by enabling native atomics from both the CPU and GPU. NVLink C2C with Address Translation Services (ATS) leverages NVIDIA Hopper DMA engines for accelerating bulk transfers of pageable memory across host and device.

NVLink C2C enables applications to oversubscribe the GPU's memory and directly utilize NVIDIA Grace CPU's memory at high bandwidth. With up to 512 GB of LPDDR5X CPU memory per Grace Hopper Superchip, the GPU has direct high-bandwidth access to 4x

more memory than available with PCIe. Combined with NVIDIA NVLink Switch System, all GPU threads running on up to 256 NVLink-connected GPUs can now access up to 150 TB of memory at high bandwidth. Fourth generation NVLink allows accessing peer memory using direct loads, stores, and atomic operations, enabling accelerated applications to solve larger problems more easily than ever.

The Grace Hopper Superchip is the first true heterogeneous accelerated platform for high-performance computing (HPC) and AI workloads. It accelerates applications with the strengths of both GPUs and CPUs while providing the simplest and most productive heterogeneous programming model to date, enabling scientists and engineers to focus on solving the world's most important problems. Together with NVIDIA networking technologies, NVIDIA Grace Hopper Superchips provide the recipe for the next generation of HPC supercomputers and AI factories, enabling customers to take on larger datasets, more complex models, and new workloads, solving them more quickly than before.

This whitepaper highlights NVIDIA Grace Hopper's key features, its programming model, and the performance improvements they deliver to the most demanding HPC and AI applications.

NVIDIA Grace Hopper Overview

NVIDIA Grace CPU is the first NVIDIA data center CPU, and it is built from the ground up to create HPC and AI superchips. The NVIDIA Grace CPU uses 72 Arm Neoverse V2 CPU cores to deliver leading per-thread performance, while providing higher energy efficiency than traditional CPUs. Each NVIDIA Grace CPU delivers up to a 370 (estimated) score on SPECrate®2017_int_base, ensuring high-performance for both HPC and AI workloads. Up to 512 GB of LPDDR5X memory provides the optimal balance between memory capacity, energy efficiency, and performance with up to 546 GB/s of memory bandwidth per CPU. Its Scalable Coherency Fabric provides up to 3.2 TB/s of total bisection bandwidth to realize the full performance of CPU cores, memory, system IOs, and NVLink C2C.

NVIDIA Hopper is the ninth-generation NVIDIA data center GPU and is designed to deliver order-of-magnitude improvements for large-scale AI and HPC applications compared to previous NVIDIA Ampere GPU generations. Thread Block Clusters and Thread Block Reconfiguration improve spatial and temporal data locality, and together with new Asynchronous Execution engines, enable applications to always keep all units busy.

NVIDIA Grace Hopper fuses an NVIDIA Grace CPU and an NVIDIA Hopper GPU into a single superchip via NVIDIA NVLink C2C, a 900 GB/s total bandwidth chip-to-chip interconnect. NVLink C2C memory coherency enables programming of both the Grace CPU Superchip and the Grace Hopper Superchip with a unified programming model.

Figure 1 shows the logical overview of the NVIDIA Grace Hopper Superchip and Table 1 lists its key features.

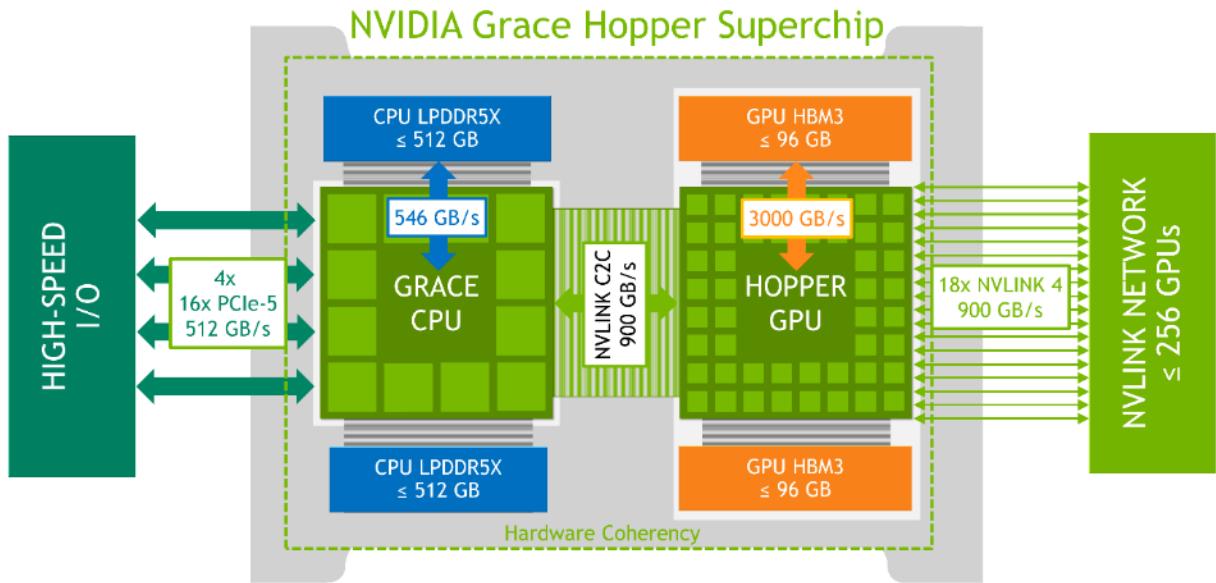


Figure 1. NVIDIA Grace Hopper Superchip Logical Overview

Table 1. NVIDIA Grace Hopper Superchip Key Features

Feature	Description
Grace CPU cores (number)	Up to 72 cores
CPU LPDDR5X bandwidth (GB/s)	Up to 546 GB/s
GPU HBM3 bandwidth (GB/s)	3 TB/s
NVLink C2C bandwidth (GB/s)	900 GB/s total, 450 GB/s per direction
CPU LPDDR5X capacity (GB)	Up to 512 GB
GPU HBM3 capacity (GB)	Up to 96 GB
PCIe Gen 5 Lanes	64x

NVIDIA HGX Grace Hopper Superchip Platform

Heterogeneous GPU-CPU Platform for Machine Learning and HPC

Grace CPU with up to:

- 72 Cores
- 512 GB LPDDR5X Memory @ 546 GB/s Bandwidth
- 64x PCIe Gen5 Lanes

Hopper GPU with up to:

- 96 GB HBM3 GPU Memory @ 3 TB/s Bandwidth

NVLink C2C with up to **900 GB/s Bandwidth**

1000W TDP

Air or Liquid cooled



Figure 2. NVIDIA HGX Grace Hopper Superchip Platform

The NVIDIA HGX Grace Hopper Superchip platform has a single Grace Hopper Superchip per node, with up to 512 GB LPDDR5X and up to 96 GB HBM3. It can be air or liquid cooled and has up to 1,000W TDP.

NVIDIA NVLink Switch System can connect 256 NVIDIA Grace Hopper Superchips at full 900 GB/s NVLink total bandwidth. NVLink-connected domains are networked with [NVIDIA InfiniBand](#) (IB) networking, e.g., NVIDIA ConnectX-7 NICs or NVIDIA BlueField-3 Data Processing Units (DPUs) paired with NVIDIA Quantum 2 NDR switches, and/or OEM-defined I/O solutions.

- **NVIDIA HGX Grace Hopper Superchip without the NVLink Switch System** is ideal for scale-out of traditional machine learning (ML) and HPC workloads. Combined with NVIDIA BlueField-3 DPUs, this platform is easy to manage and deploy and uses a traditional HPC/AI cluster networking architecture.
- **NVIDIA HGX Grace Hopper Superchip with NVLink Switch System** enables all GPU threads in the NVLink-connected domain to address up to 150 TB of memory at up to 900 GB/s total bandwidth per superchip, up to 450 GB/s all-reduce bandwidth, and up to 115.2 TB/s bisection bandwidth, in a 256 GPU NVLink connected system making this platform ideal for strong scaling the world's largest and most challenging AI training and HPC workloads.

NVIDIA HGX Grace Hopper Superchip with InfiniBand

Ideal for Scale-out Machine Learning and HPC Workloads

Figure 3 shows the HGX Grace Hopper Superchip without NVLink Switch System. NVLink C2C provides hardware coherency within a Grace Hopper Superchip. Each node contains one Grace Hopper Superchip and one or more PCIe devices like NVMe Solid-State Drives and BlueField-3 DPUs, NVIDIA ConnectX-7 NICs, or OEM-defined I/O. These nodes are designed for scale-out ML and HPC. With 16x PCIe Gen 5 lanes, an NDR400 InfiniBand NIC provides up to 100 GB/s of total bandwidth across the Superchips.

The configuration in Figure 3 simplifies cluster management. It is designed for workloads that can leverage the strong compute capabilities of NVIDIA Grace Hopper and are also not bottlenecked by the network communication overhead of InfiniBand, which is one of the fastest network interconnects available, but is still a traditional RDMA-accelerated network.

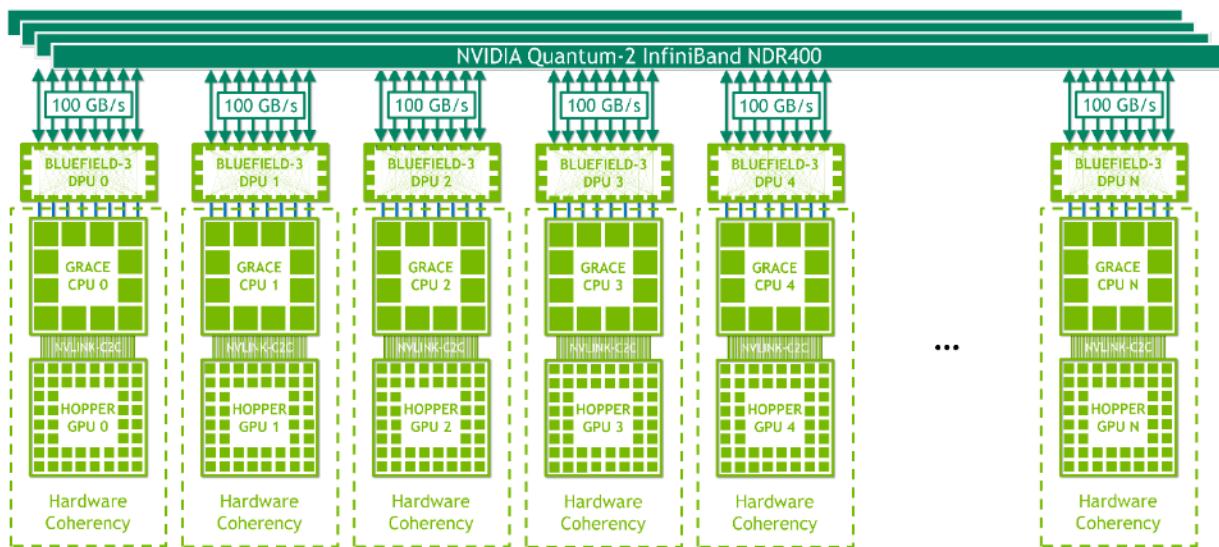


Figure 3. NVIDIA HGX Grace Hopper Superchip system with InfiniBand networking for scale-out ML and HPC workloads

NVIDIA HGX Grace Hopper Superchip with NVLink

Ideal for strong-scaling giant Machine Learning and HPC Workloads

The NVIDIA HGX Grace Hopper Superchip with NVLink Switch System shown in Figure 4 enables each Hopper GPU to communicate with any other GPU in the NVLink domain at 900 GB/s total bandwidth. NVLink TLBs enable a single GPU to address all the NVLink connected memory, that is, up to 150 TB of memory for a 256-node NVLink connected system. Up to 256 Superchips can be connected using NVLink in a pod, and InfiniBand NICs and switches connect multiple superchip pods together. NVLink C2C and the NVLink Switch System provide hardware consistency across all superchips within the NVLink-connected domain.

The high bandwidth and lower latencies of NVLink paired with memory consistency across all NVLink-connected GPUs and the ability to address up to 150 TB of memory using direct loads, stores, and atomic operations, makes this system configuration ideal for strong-scaling machine learning and HPC workloads, and training giant AI models.

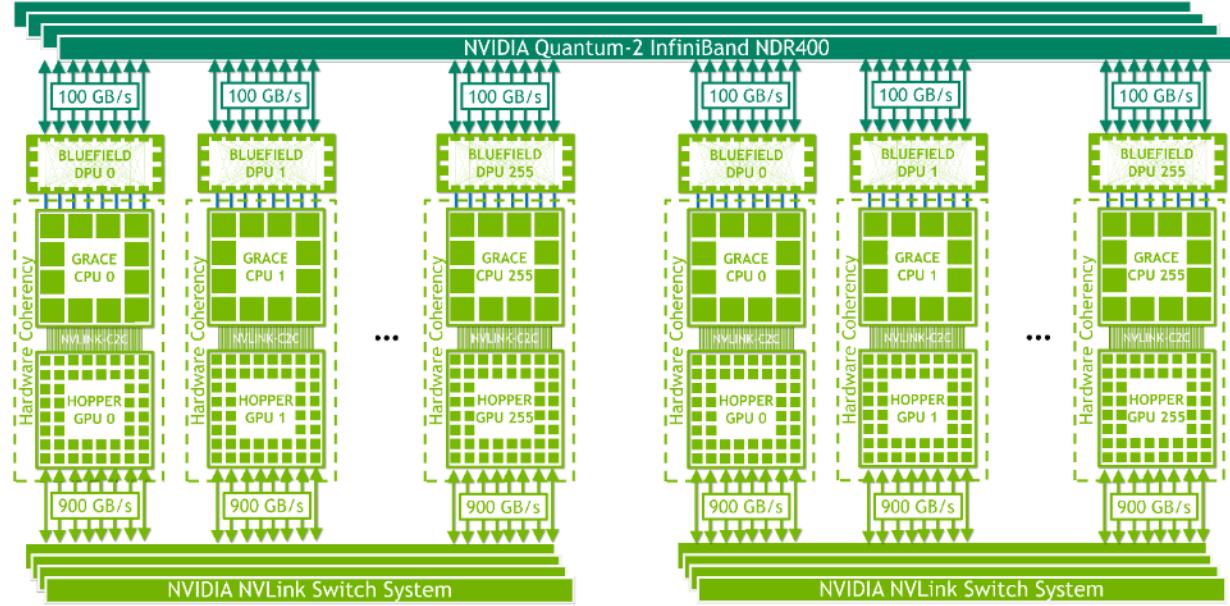


Figure 4. NVIDIA HGX Grace Hopper Superchip System with NVLink Switch System for strong-scaling giant ML and HPC workloads

NVIDIA HGX Grace Hopper Comparisons

Table 2 compares the speeds and feeds of x86+Hopper, NVIDIA HGX Grace Hopper Superchip, and NVIDIA HGX Grace Hopper Superchip with the NVLink Switch System, normalized by the number of Hopper GPUs in each system.

The NVIDIA HGX Grace Hopper Superchip delivers 3.5x higher CPU memory bandwidth per GPU than x86+Hopper due to the 1:1 GPU to CPU ratio and the high LPDDR5X bandwidth of NVIDIA Grace CPU. NVIDIA Grace Hopper's NVLink C2C provides 7x higher GPU-CPU link bandwidth per GPU over PCIe Gen 5. The NVLink Switch System achieves up to 9x higher GPU to GPU total bandwidth than InfiniBand NDR400 NICs connected via x16 PCIe Gen5 lanes. This significant reduction in the amount of compute required to hide memory transfers makes the system easier to program and improves the performance of GPU-CPU bandwidth bound applications.

Table 2. NVIDIA HGX Grace Hopper Superchip vs. NVIDIA x86+Hopper

Feature per GPU	HGX H100 4-GPU (x86)	NVIDIA HGX Grace Hopper	NVIDIA HGX Grace Hopper NVLink Switch
CPU Memory bandwidth (GB/s / GPU)	Up to 150	Up to 546	Up to 546
GPU Memory bandwidth (GB/s / GPU)	3000	3000	3000
CPU Memory bandwidth to GPU Memory bandwidth ratio	5%	18.2%	18.2%
GPU-CPU Link bi- directional bandwidth (GB/s / GPU)	128 (x16 PCIe Gen5)	900 (NVLink C2C)	900 (NVLink C2C)
GPU-GPU bi-directional bandwidth inter node (GB/s / GPU)	100 (InfiniBand NDR400)	100 (InfiniBand NDR400)	900 (NVLink 4)

These improvements in CPU ratio, and NVLink C2C and NVLink Switch System performance redefine how we achieve maximum performance from heterogeneous systems, enabling new applications, and efficient solutions to challenging problems.

Enhance HGX Grace Hopper with NVIDIA BlueField-3 DPDUs

User-Defined Infrastructure for Intelligent Data Centers and HPC

Traditional supercomputing offers bare-metal performance without any isolation or secure infrastructure. The NVIDIA BlueField-3 Data Processing Unit (DPU) is a powerful “computer-in-front-of-a-computer” that is isolated and abstracted from the host system and therefore forms part of the trusted data-center infrastructure. Bluefield DPUs deliver high-speed networking connectivity (using EDR, HDR, NDR InfiniBand, or 100/200/400G Ethernet) to connect nodes, in-network computing offloads for HPC, as well as infrastructure offload and acceleration. Together with the NVIDIA DOCA (Data center-on-a-Chip) framework, BlueField DPUs enable software-defined, hardware accelerated infrastructure and data-center services for security (e.g., firewalls and data encryption), data protection, tenant isolation, and multi-tenancy on bare-metal environments. The DPUs also provide telemetry collection, storage virtualization, and system management functions that are fully isolated from the host, and therefore do not consume any CPU or GPU cycles.

Together, these technologies enable Cloud Native Supercomputing (CNSC), combining bare-metal performance and high data center efficiency with a modern zero-trust model for security isolation and multi-tenancy. CNSC includes storage virtualization for remote storage allocation as well as implementation of tenant Service Level Agreements (SLAs)

such as rate-limiting, bandwidth-guarantees, and reservation of network resources based on tenant isolation and other requirements.

Advanced visibility is critical when operating at large-scale. NVIDIA BlueField-3 extends traditional monitoring tools by providing deep and instantaneous real-time visibility into every node. The NVIDIA [DOCA Telemetry Service](#) (DTS) is a containerized telemetry agent that supports collecting and exporting data from a wide range of providers at the operating system and network levels, including core and non-core Performance Monitoring Units, and the Baseboard Management Controller (BMC). This enables the characterization of user workloads, and enables operators to detect and respond to system health and application performance issues as well as potential cyberattacks.

NVIDIA Grace Hopper Superchip Architecture

This section covers the main architectural features of the NVIDIA Grace Hopper Superchip including: NVLink C2C, NVLink Switch System, Extended GPU Memory (EGM), and the superchip flexible computing capabilities.

NVLink C2C

The NVLink Chip-2-Chip (C2C) interconnect provides a high-bandwidth direct connection between a Grace CPU and a Hopper GPU to create the Grace Hopper Superchip, which is designed for drop-in acceleration of AI and HPC applications. With 900 GB/s of bidirectional bandwidth, NVLink C2C provides 7x the bandwidth of x16 PCIe Gen links at lower latency. NVLink C2C also only uses 1.3 picojoules per bit transferred, which is greater than 5x more energy efficient than PCIe Gen 5.

Furthermore, NVLink C2C is a coherent memory interconnect with native hardware support for system-wide atomic operations. This improves the performance of memory accesses to non-local memory, such as CPU and GPU threads accessing memory resident in the other device. Hardware coherency also improves the performance of synchronization primitives, reducing the time the GPU or CPU wait on each other, increasing total system utilization. Finally, hardware coherency also simplifies the development of heterogeneous computing applications using popular programming languages and frameworks as elaborated in the NVIDIA Grace Hopper Programming Model section below.

NVLink Switch System

NVIDIA NVLink Switch System combines fourth-generation NVIDIA NVLink technology with the new third-generation NVIDIA NVSwitch. A single level of the NVSwitch connects up to eight Grace Hopper Superchips, and a second level in a fat-tree topology enables networking up to 256 Grace Hopper Superchips with NVLink.

Fourth generation NVIDIA NVLink enables GPU threads to address up to 150 TB of memory provided by all superchips in the NVLink network using normal memory operations, atomics, and bulk transfers. Communication libraries like MPI, NCCL, or NVSHMEM transparently leverage the NVLink Switch System when available.

NVIDIA NVLink Switch System connects each Grace Hopper Superchip to the network at 900 GB/s total bandwidth. That is, a Grace Hopper Superchip pair exchanges data at up to 900 GB/s. With up to 256 Grace Hopper Superchips, the network delivers up to 115.2 TB/s all-to-all bandwidth. This is 9x the all-to-all bandwidth of InfiniBand NDR400, and 4.5x the all-reduce throughput.

The incredible one FP16 exaflop delivered by 256x Grace Hopper Superchips combined with the 115.2 TB/s of all-to-all bandwidth provided by the NVLink Switch System and up to 150 TB of directly addressable memory enables training giant AI models and strong scaling HPC and AI workloads.

Accelerating Applications with Extended GPU Memory

The NVIDIA Grace Hopper Superchip is designed to accelerate applications with exceptionally large memory footprints, larger than the capacity of the HBM3 and LPDDR5X memory of a single superchip (see the NVIDIA Grace Hopper Accelerated Applications section below).

The Extended GPU Memory (EGM) feature over the high-bandwidth NVLink C2C enables GPUs to access all the system memory efficiently. EGM provides up to 150 TBs system memory in a multi-node NVSwitch-connected system. With EGM, physical memory in the system can be allocated to be accessible from any GPU thread. All GPUs can access EGM at the minimum of GPU-GPU NVLink or NVLink C2C speed.

Memory accesses within a Grace Hopper Superchip configuration go through the local high-bandwidth NVLink C2C at 900 GB/s total. Remote memory accesses are performed via GPU NVLink, and depending on the memory being accessed, also NVLink C2C as shown in Figure 5. With EGM, GPU threads can now access all memory resources available over the NVSwitch fabric, both LPDDR5X and HBM3, at 450 GB/s.

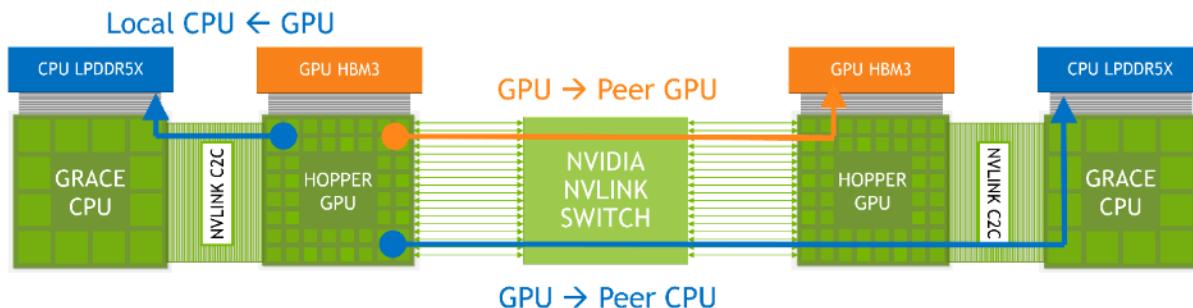


Figure 5. Memory Accesses across NVLink-connected Grace Hopper Superchips

Flexible Architecture Built for Peak Performance

The NVIDIA Grace Hopper Architecture is flexible and enables applications ranging from large scale-out deep learning training and HPC workloads, to small inference workloads requiring Quality of Service (QoS).

The NVIDIA Grace Hopper system can balance power between the NVIDIA Grace CPU and the Hopper GPU. The Grace Hopper Superchip enables building supercomputing systems with a 1:1 GPU-CPU ratio that are power efficient and excel at GPU-heavy, CPU-heavy, and truly heterogeneous workloads. NVIDIA Grace Hopper nodes enable uniform

systems to be built with higher peak performance, lower maintenance, and lower administration overheads.

The NVIDIA Grace CPU supports Memory Resource Partitioning and Monitoring (MPAM) features that provide performance isolation between jobs. MPAM enables users and administrators to partition the available LPDDR5X bandwidth and CPU cache usage. NVIDIA Multi-Instance GPU (MIG) allows partitioning of the Hopper GPU into smaller instances. Together MPAM and MIG can be used to partition system resources for improved QoS.

NVIDIA Grace Hopper Superchip Performance Monitoring Units (PMU) adhere to the ARM PMU architecture specification standards (Arm v8.5 PMUv3) for capturing performance metrics and are exposed via standard Linux performance tool interfaces like Linux perf. They provide a uniform and programmable approach for capturing performance metrics for the Grace CPU and Grace Hopper Superchip in a single pass. Single pass metric collection is performed at extremely low overhead with little CPU polling while supporting all features required for confidential computing. The metrics cover CPU core and caches, system caches, memory bandwidths, utilization, throughput, and latencies for GPU, CPU, NVLink C2C, PCIe, and DRAM.

NVIDIA Grace Hopper Programming Model

Traditional heterogeneous platforms with PCIe-connected accelerators require users to follow a complex programming model that involves manually managing device memory allocations and data transfer to and from the host.

The NVIDIA Grace Hopper Superchip is a heterogeneous platform that is easy to program, and NVIDIA is committed to making it accessible to all developers, independent of their programming language of choice. Both the Grace Hopper Superchip and the Platform are built to enable developers to pick the right language for the task at hand, and the NVIDIA libNVVM API enables developers to bring their preferred programming language to the CUDA platform with the same level of code-generation quality and optimizations as NVIDIA compilers and tools.

The languages provided by NVIDIA for the CUDA platform, as shown in Figure 6 include accelerated standard languages like:

- ISO C++
- ISO Fortran
- Python

And directive-based programming models like:

- OpenACC
- OpenMP
- CUDA C++
- CUDA Fortran

Accelerated Standard Languages	Incremental Portable Optimization	Platform Specialization
<pre>std::for_each(par, std::iota(n), [=](auto i) { y[i] += a*x; }); do concurrent (i = 1:n) y(i) = y(i) + a*x(i) enddo import cunumeric as np x = np.array(..) y = np.array(..) y += a*x</pre>	<pre>#pragma acc data copy(x,y) { std::for_each(par, std::iota(n), [=](auto i) { y[i] += a*x; }) } !acc data copy(x,y) do concurrent (i = 1:n) y(i) = y(i) + a*x(i) enddo !acc end data</pre>	<pre>__global__ void saxpy(int n, float a, float* x, float* y) { auto i = cg::grid_group::thread_rank(); if (i < n) y[i] += a*x[i]; } saxpy<<<(n+255)/256,256,s>>>(n, a, x, y); cudaStreamSynchronize(s);</pre>

Accelerated Libraries, Frameworks and SDKs

Figure 6. NVIDIA Grace Hopper Superchip Programming Model

NVIDIA is an important contributor to the ISO standardization processes of C++ and Fortran. We have worked together with these communities to enable ISO compliant C++ and Fortran applications to effectively program NVIDIA CPUs and NVIDIA GPUs without any language extensions.

This technology heavily relies on our hardware-accelerated memory coherency provided by NVIDIA NVLink C2C and NVIDIA Unified Virtual Memory to lift the restriction of traditional PCIe-connected accelerated platforms.

On Linux systems with the Heterogeneous Memory Management (HMM) extension, the NVIDIA CUDA platform provides the same unified programming model as NVIDIA Grace Hopper. When running on NVIDIA Grace Hopper, these applications transparently benefit from the higher-bandwidth, lower-latency, higher atomic throughput, and hardware acceleration for memory coherency provided by NVLink C2C.

Hardware Accelerated Memory Coherency

In PCIe-connected x86+Hopper systems, the CPU and the GPU have independent per-process page tables, and system-allocated memory is not directly accessible from the GPU (Figure 7). When a program allocates memory with the system allocator on the host, the page entry of the allocation is not available in the GPU's page table and accessing it from GPU threads fails¹.

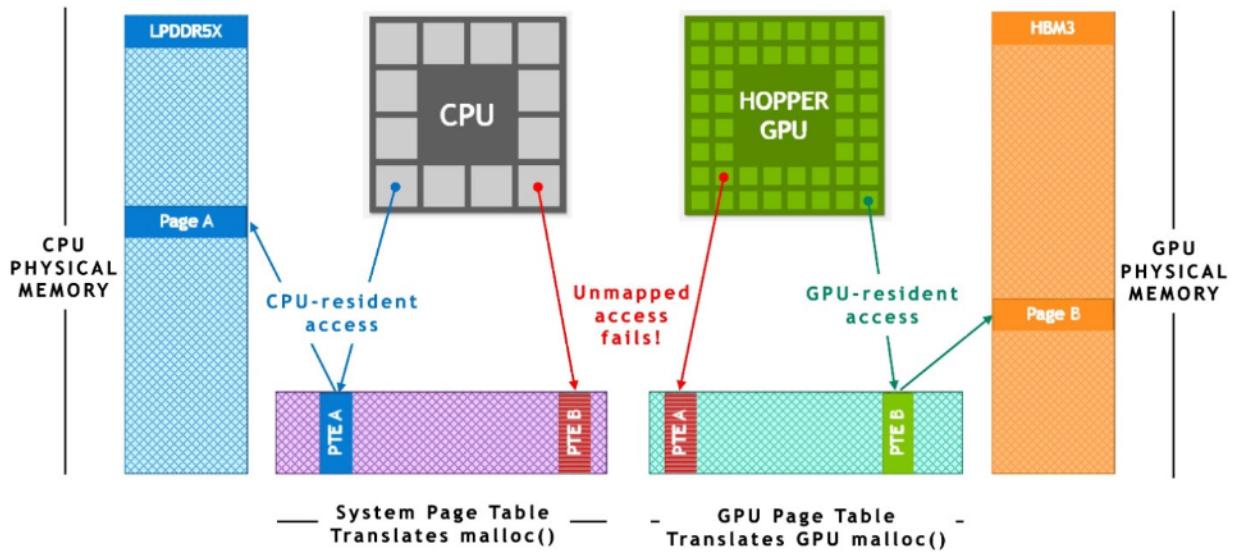


Figure 7. NVIDIA Hopper System with Disjoint Page Tables

In NVIDIA Grace Hopper Superchip-based systems, Address Translation Service (ATS) enables the CPU and GPU to share a single per-process page table, enabling all CPU and GPU threads to access all system-allocated memory (Figure 8), which can reside on

¹ Applications can lock the pages and register them with the CUDA driver using APIs like `cudaHostRegister`, but these API calls are expensive, they prevent memory migration, and the amount of pinned memory available is a scarce resource.

physical CPU or GPU memory. The CPU heap, CPU thread stack, global variables, memory-mapped files, and inter-process memory are accessible to all CPU and GPU threads.

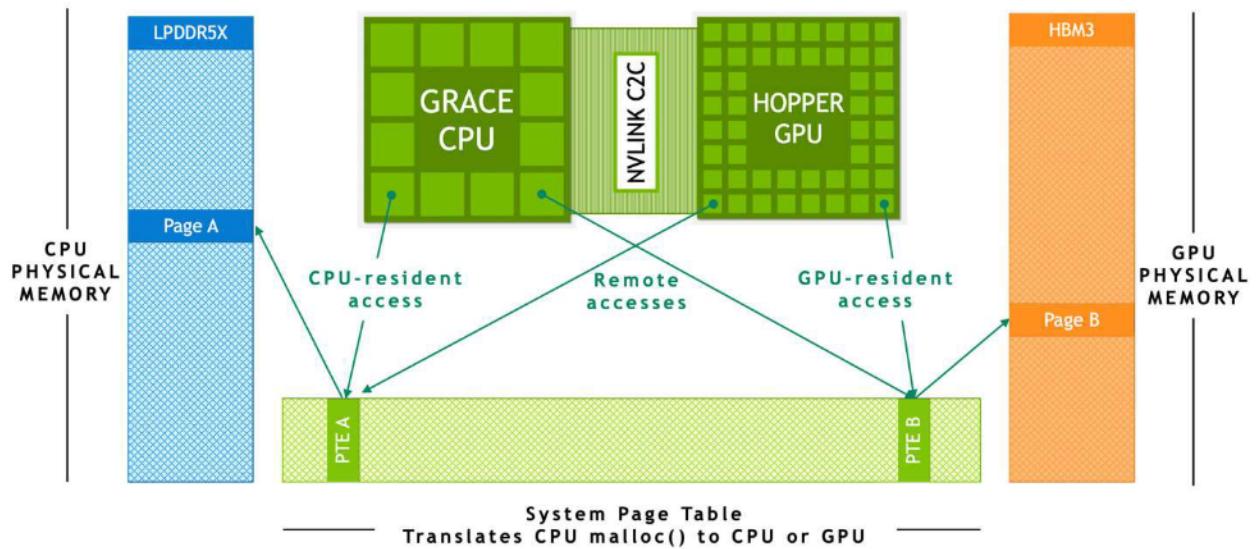


Figure 8. ATS in an NVIDIA Grace Hopper Superchip System

NVIDIA NVLink C2C hardware-coherency enables the Grace CPU to cache GPU memory at cache-line granularity and for the GPU and CPU to access each other's memory without page-migrations. NVLink C2C also accelerates all atomic operations supported by the CPU and GPU on system-allocated memory. Scoped atomic operations are fully supported and enable fine-grained and scalable synchronization across all threads in the system.

The runtime backs system-allocated memory with physical memory on first touch, either on LPDDR5X or HBM3, depending on whether a CPU or a GPU thread accesses it first. From an OS perspective, the Grace CPU and Hopper GPU are just two separate NUMA nodes.

System-allocated memory is migratable, i.e., the runtime can change its physical memory backing to improve application performance (Figure 9) or deal with memory pressure. Hardware access counters allow delayed migrations over a page-fault-based method so that only hot pages are migrated.

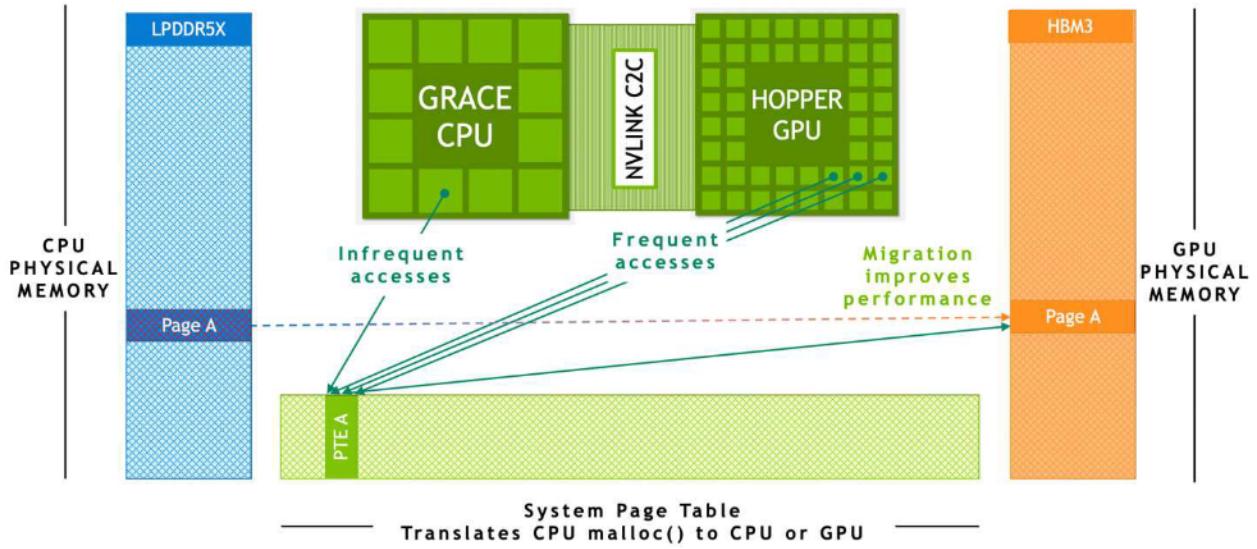


Figure 9. Access-Frequency-Based Automatic Memory Migration

Network and storage devices connected via non-coherent PCIe lanes have several methods for performing Direct Memory Access (DMA) and Remote DMA (RDMA) on system-allocated memory. On-Demand Paging (ODP) is an RDMA extension supported by NVIDIA InfiniBand Networking products like BlueField-3 and ConnectX-7 that allows devices to track pages being migrated. It enables communication and storage libraries such as MPI, [HPC-X](#), [NCCL](#), [NVSHMEM](#), [UCX](#), [MAGNUM IO](#), and [GPUDirect Storage](#) to perform efficient zero-copy I/O operations on system-allocated memory without having to stage transfers through separate buffers.

CUDA-specific memory APIs provide users with guarantees about where the memory resides, which threads can access it, whether it is migratable, and many other features that enable users to extract all the performance the hardware has to offer. Applications can hint the system about their memory access patterns, for example, using [CUDA](#) and/or [NUMA](#) APIs, to enable the users to perform application-specific optimizations. NUMA memory hints enable applications to inform the runtime about their memory access patterns.

Memory Access in NVLink Switch System

On Grace Hopper Superchips connected with NVLink Switch Systems, GPU threads can address peer HBM3 and LPDDR5X memory from other Grace Hopper Superchips in the NVLink network via an NVLink page table (Figure 10). CUDA APIs allow applications to map memory from remote nodes into the current process and then perform load, stores, atomics, and as well as bulk memory transfers to directly access the memory.

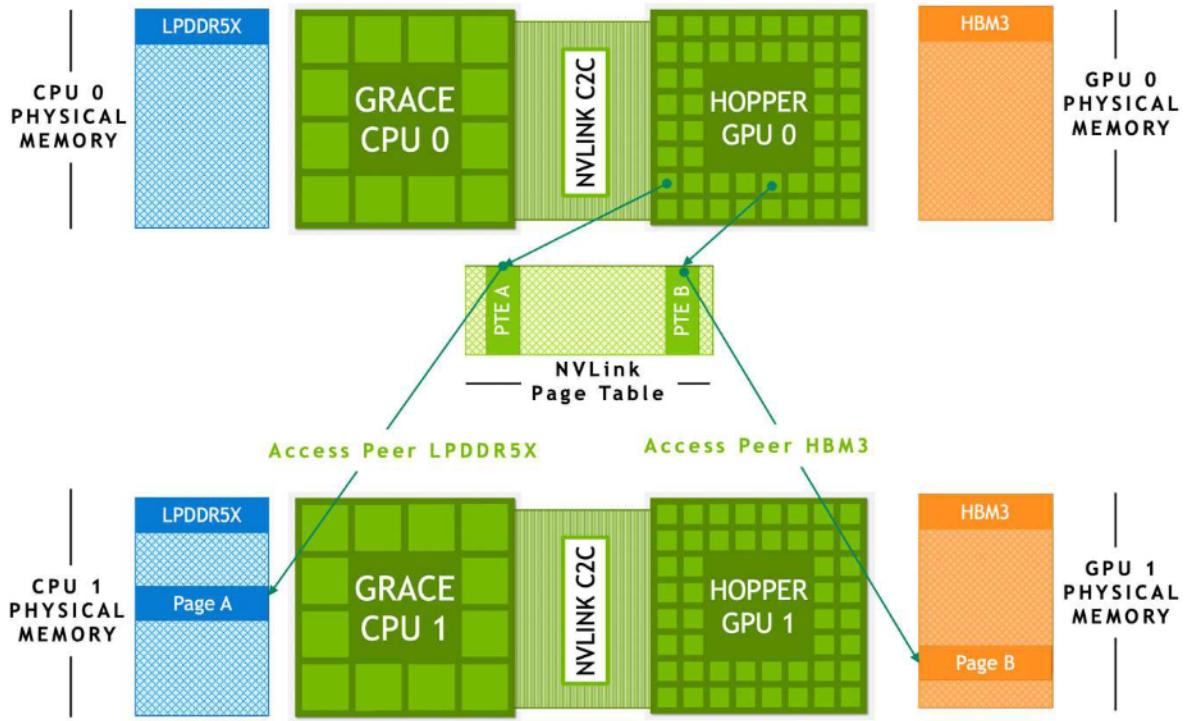


Figure 10. GPU threads address peer memory from other superchips in the NVLink Switch network

NVIDIA CUDA Platform

The NVIDIA CUDA platform (Figure 11) is optimized for Arm CPUs, the NVIDIA Grace CPU, and of course, the NVIDIA Grace Hopper Superchip and NVIDIA NVLink Switch System. NVIDIA CUDA is a comprehensive, productive, and high-performing platform for accelerated computing.

It accelerates end-user applications at all levels, from system software to application-specific libraries and frameworks, using all hardware available including GPUs, CPUs, DPUs, and in-network computing (Figure 11). The CUDA platform has mature and user-friendly toolchains, developer tools, and documentation. It provides the best developer experience for accelerating applications on heterogeneous platforms.

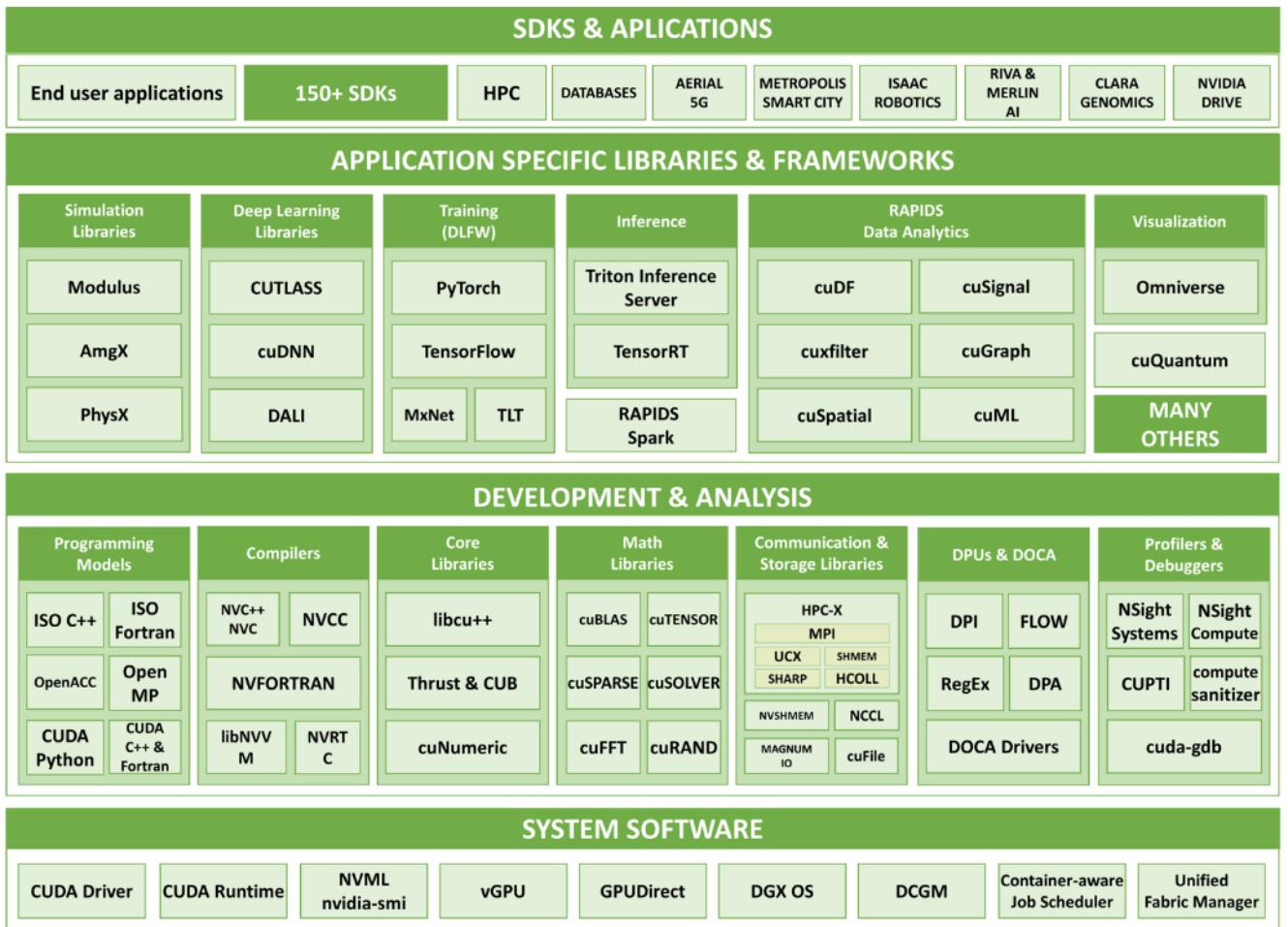


Figure 11. NVIDIA CUDA Platform and its ecosystem

The NVIDIA Grace Hopper system is designed from the ground up to accelerate applications on the CUDA platform. Applications that run correctly on the NVIDIA Arm HPC developer kit continue to do so in Grace Hopper-based systems and immediately benefit from its memory coherent high-performance NVLink C2C, NVLink Switch System, and high-bandwidth access to large amounts of memory.

NVIDIA is actively engaged with the broader developer community to ensure that the Arm ecosystem meets users' requirements. The whole NVIDIA software stack is available for Arm server CPUs, including NVIDIA Grace CPU. Every CUDA component available on x86 today has Arm-native installers and containers. The NVIDIA GPU Cloud™ (NGC) provides DL, ML, and HPC containers optimized for Arm platforms.

NVIDIA Grace Hopper Accelerated Applications



Figure 12. Performance Simulations for Grace Hopper vs x86+Hopper for end-user applications. Datasets and details described in the individual application section below.

NVIDIA Grace Hopper Superchip main architectural features are its NVLink C2C interconnect, which provides memory coherency within the superchip enabling Extended GPU Memory (EGM), and NVLink Switch System which extends EGM beyond a single superchip. NVLink Switch System and NVLink C2C are high-bandwidth and low-latency interconnects, providing up to 900 GB/s total bandwidth when accessing up to 150 TB of memory. NVIDIA Grace Hopper also provides a 1:1 GPU-to-CPU ratio, making the platform excel at GPU-heavy, CPU-heavy, and truly heterogeneous workloads that intensely engage both the GPU and the CPU.

This section highlights how these unique hardware features accelerate major applications shown in Figure 12, enable new applications, and improve developer productivity when writing new applications or incrementally accelerating already-existing ones on the GPU.

We highlight major algorithmic motifs like out-of-core processing, concurrent CPU and GPU processing, and applications with higher CPU performance requirements. We include examples within major fields like machine learning (natural language processing, recommender systems, Graph Neural Networks), databases, HPC (weather, climate, fluid dynamics, molecular dynamics, linear solvers) and the intersection of HPC with AI.

Natural Language Processing Networks

Higher CPU to GPU Bandwidth

State-of-the-art natural language processing (NLP) networks are very large, with the number of parameters reaching hundreds of billions. These models are usually pretrained for days on very large datasets using large GPU clusters.

To tune pretrained models for specific downstream tasks, prompt learning methods are used. During prompt learning - also referred to as p-tuning - the pre-trained language model is frozen. Then, a special prompt encoding layer is trained to generate virtual prompt tokens that are concatenated with the original input text before being fed to the pretrained language model. Prompt learning is typically characterized by small datasets and small batch sizes. Training is fast and there is no business need to scale an individual p-tuning training instance to a large GPU cluster. Instead, typical use cases involve continuously re-training multiple downstream NLP tasks with training data that evolves over time.

A simple and clean solution that lowers model maintenance and deployment cost is to p-tune a model on a single GPU. To fit a large NLP model on a single GPU, we can use the approach of offloading tensors to CPU memory during the training iteration.

With an x86 host CPU, accessing system memory for tensor offloading can become bottlenecked by PCIe. NVIDIA NVLink C2C provides the Hopper GPU with high-bandwidth access to LPDDR5X memory. This significantly reduces the exposed tensor offloading execution time in the critical path and enables fine-tuning a GPT-175B model on a single Grace Hopper Superchip up to 4x faster than on an x86+Hopper system (see Figure 13). With NVIDIA Grace Hopper, we can fine-tune a GPT-175B model on a single 80 GB HBM GPU with similar performance as if it had much larger 350GB of HBM capacity.

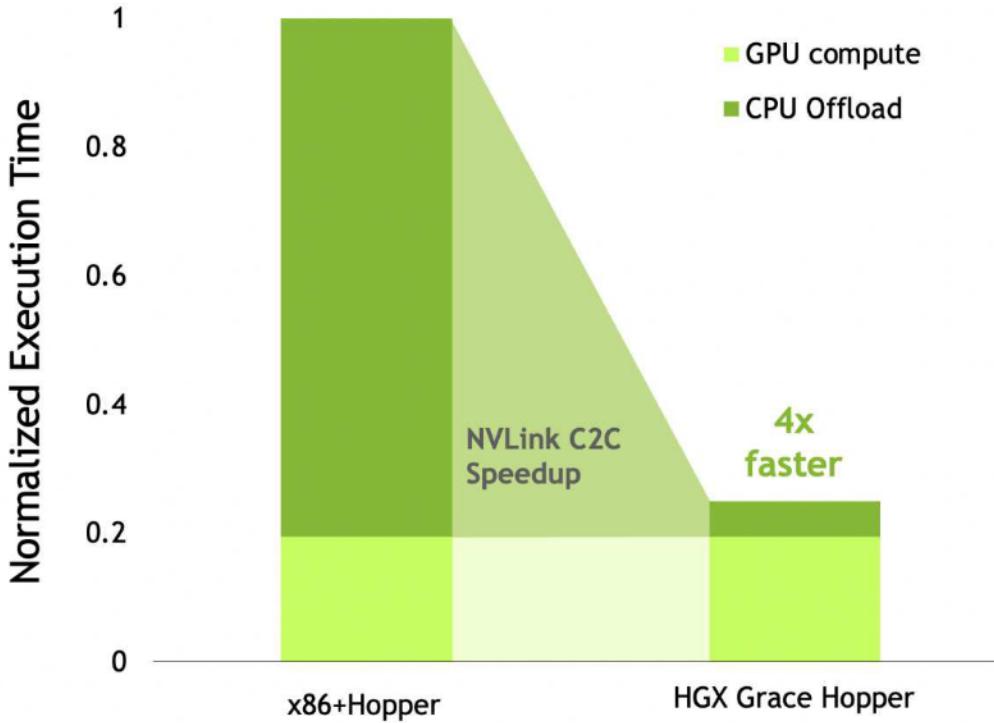


Figure 13. A performance simulation for P-tune GPT3 175B NLP Model implemented with offloading

Recommender Systems

Higher CPU to GPU Bandwidth using NVLink

Modern recommender system models require substantial amounts of memory for storing embedding tables. Embedding tables contain semantic representations for items and users' features, which help provide better recommendations to consumers.

Generally, these embeddings follow a power-law distribution for frequency of use since some embedding vectors are accessed more frequently than others.

NVIDIA Grace Hopper enables high-throughput recommender system pipelines that store the most frequently used embedding vectors in HBM3 memory, and the remaining embedding vectors in the higher-capacity LPDDR5X memory. The NVLink C2C interconnect provides Hopper GPUs with high-bandwidth access to their local LPDDR5X memory, while the NVLink Switch System extends this to provide Hopper GPUs with high-bandwidth access to all LPDDR5X memory of all Grace Hopper Superchips in the NVLink network.

Figure 14 shows the performance improvement delivered by the HGX Grace Hopper with NVLink Switch System over an equivalent x86+Hopper system for two Deep Learning Recommendation Model (DLRM) model sizes, 4.5 TB (1.5x speedup, left) and 27 TB (3.5x speedup, right).

For the 27 TB large model, in both the x86+Hopper and NVIDIA Grace Hopper cases, most network communication is hidden behind the computations accessing the embedding tables. However, when the embedding computation is accelerated with NVLink C2C, the communication must scale to avoid becoming the bottleneck. The NVLink Switch System accelerates all communication and achieves just that. For the smaller 4.5 TB model, a significant part of communication is exposed on the x86+Hopper system. This communication gets accelerated over the NVLink Switch system when strong scaling to 128 GPUs.

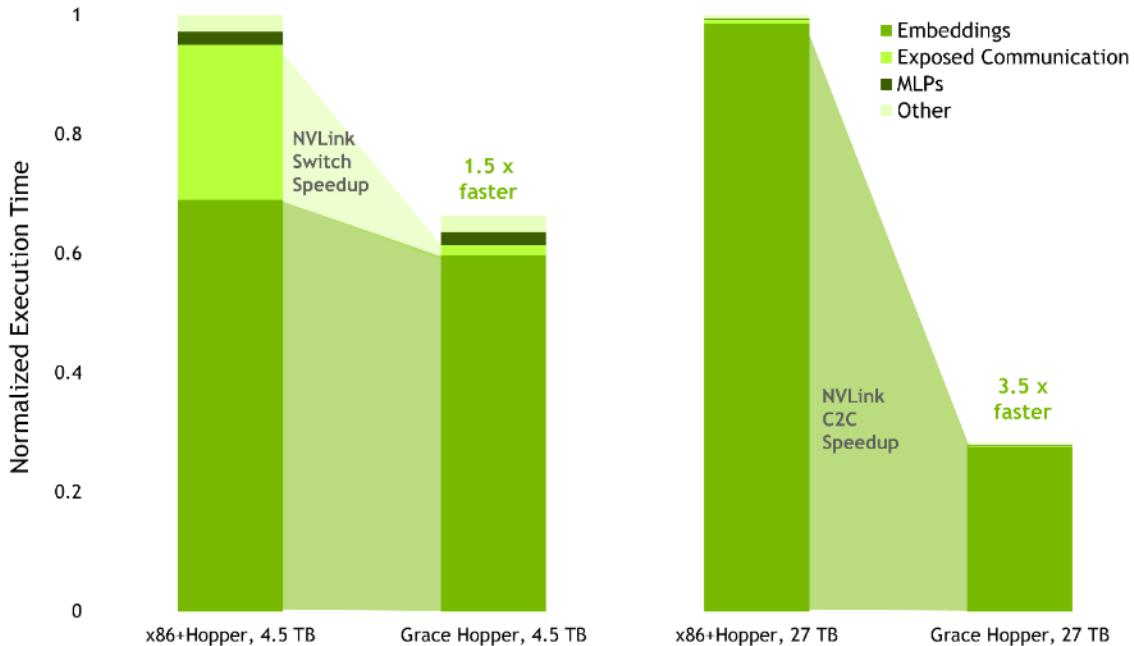


Figure 14. A performance simulation for large DLRM network model with 4.5 TB and 27 TB model size, 128 GPUs, and embeddings offloaded to CPU memory

Graph Neural Networks

Higher CPU to GPU Bandwidth

Graph Neural Networks (GNNs) leverage message passing to extract information from data that is formed from individual nodes and their relation to each other, by repeatedly gathering information from neighboring nodes and transforming the aggregated representations. While this scheme is somewhat related to convolutional neural networks (CNNs) applied to image data, GNNs do so for arbitrary neighborhoods representing nodes in large networks of interactions at a large scale. GNNs therefore must cope with large datasets consisting of hundreds of millions of nodes and billions of edges. Besides the sheer scale of graph data, the arbitrariness of graph neighborhoods leads to irregular memory accesses to contiguous embedding tables and makes

distributing the dataset across multiple GPUs challenging. Especially when data does not fit into the memory of a single GPU, or even of a single node.

We studied the performance of GraphSAGE which is a model for node property prediction, with potential applications in biomedical contexts (e.g., drug discovery) or the financial service industry (e.g., fraud detection). To demonstrate the benefits of the CPU-GPU NVLink of Grace Hopper, we show the performance on an augmented version of the ogbn-products dataset having 626M nodes and 31B edges, coming to about 500 GB in size.

NVIDIA Grace Hopper accelerates GNN training with its high-bandwidth access to LPDDR5X through its NVLink C2C interconnect. This allows large graph datasets to be stored in pinned memory and accessed efficiently. Our experiments show that Grace Hopper can provide 1.9x performance gains over x86+Hopper systems as shown in Figure 15.

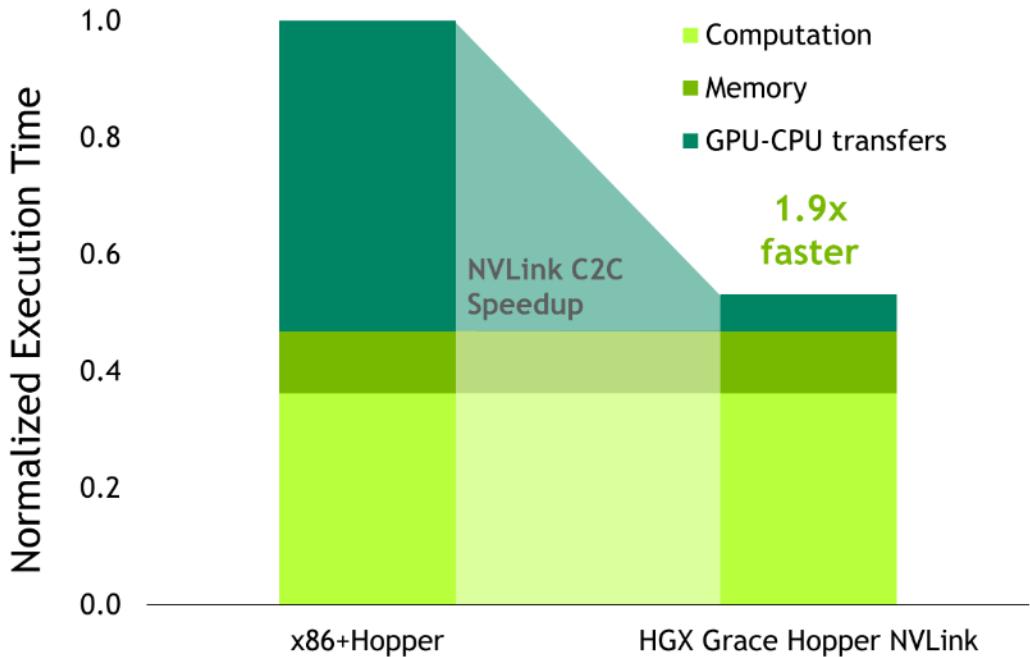


Figure 15. A performance simulation shows normalized runtime per batch of a GraphSAGE model for an augmented ogbn-products dataset of 626 M nodes and 31 B edges

Databases

Higher CPU to GPU Bandwidth with Address Translation Service (ATS)

Traditionally, GPU memory capacity limited the dataset sizes GPUs can operate on at high performance. Database workloads operate on exceptionally large input tables that cannot fit in GPU memory, so performance is often limited by the CPU-GPU data transfer due to low PCIe bandwidth.

The Grace Hopper NVLink C2C excels at databases, analytical workloads, and the Extract Transform Load (ETL) stages of ML applications. GPUs can now directly operate on datasets located in CPU memory at high speed, eliminating the PCIe bottleneck (Figure 16). Support for a rich set of atomic operations enables new coprocessing opportunities. For example, the CPU and the GPU can concurrently build a shared hash table for *join* and *group by* by accessing both HBM3 and LPDDR5X.

Because GPUs can access all system-allocated memory, integrating GPU applications with pre-existing databases running on the CPU is easier than ever and provides performance benefits.

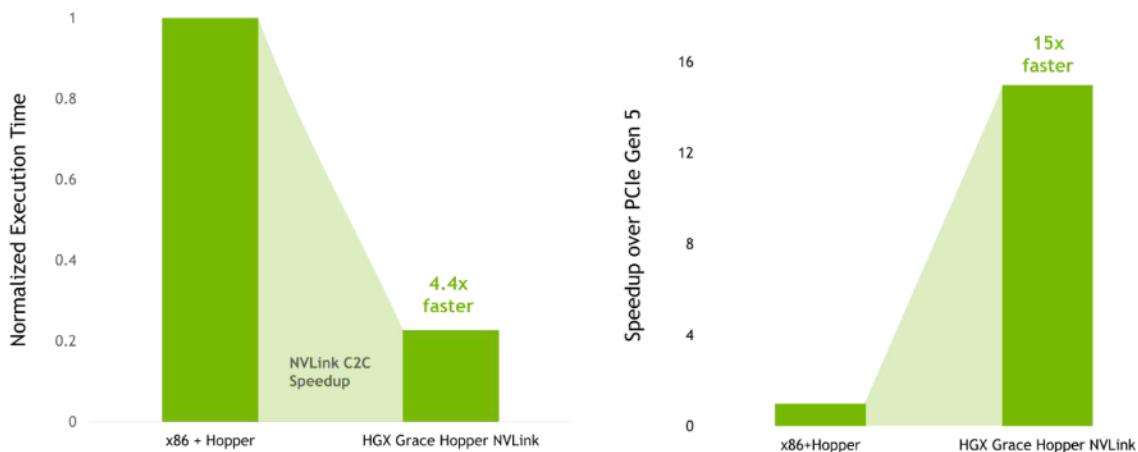


Figure 16. Performance simulations for Hash Join with input tables in CPU Memory (left) and host-to-device transfer of pageable host-resident memory (right)

Large GPU database applications often need to process databases larger than the system memory capacity and strongly prefer to map those as pageable memory. On platforms without Address Translation Services (ATS), DMA transfers between CPU resident pageable memory must be staged through a host-pinned buffer for correctness: the pageable memory could be paged to disk, and this would cause the DMA to access the wrong physical memory. That is, the performance of host-to-device memory transfers on non-coherent platforms is limited by: $\min(\text{single-threaded CPU memcpy Bandwidth}, \text{PCIe Bandwidth})$. The single-threaded CPU DRAM Bandwidth of modern CPUs is in the 20-40 GB/s range and barely suffices to saturate PCIe Gen4 BW.

PCIe Gen5 provides 64GB/s host-to-device bandwidth that these database applications cannot benefit from.

The ATS feature of the Grace Hopper Superchip enables the memory transfer accelerators to transfer pageable memory between the host and device without staging through a pinned memory buffer on the host using multiple CPU threads, or without pinning the memory buffer before the transfer using APIs like *cudaHostRegister*. With ATS, the performance of host-to-device memory transfers is limited only by the NVLink C2C bandwidth between Grace and Hopper. The performance impact is projected in Figure 16 and highlights how applications transparently benefit from Grace Hopper features like ATS without any application-side changes.

The performance simulation for the Grace Hopper Superchip system uses the NVLink C2C raw bandwidth. While the attainable bandwidth is often lower, it is not impacted by single-threaded CPU STREAM COPY Bandwidth.

Partially Accelerated Applications

Higher CPU to GPU Bandwidth with Easier Incremental Porting

Multiphysics, quantum chemistry, and climate applications are large and complex. For example, weather and climate applications contain physical models for different phenomena like wind, clouds, distinct types of precipitation, evaporation, groundwater, solar radiation, ice, and oceans. Accelerating these applications with GPUs is an incremental and time-consuming process. Moving the entire application to GPU is often complicated and the performance benefits often diminish once the most computationally expensive modules are ported to the GPU.

The new unified NVIDIA Grace Hopper programming model presents opportunities for incremental porting of computationally expensive parts to the GPU, while leaving computationally inexpensive parts for the CPU to process, such as rarely used codes, or code that do not generate enough GPU work.

The NVIDIA Grace Hopper programming model enables the whole application to work on the same memory, simplifying incremental acceleration. At the same time, NVLink C2C improves the performance of porting techniques, for example, redirecting BLAS library calls to NVBLAS.

ABINIT is a material science application that simulates Density Function Theory using standard BLAS calls. Figure 17 shows the performance impact of “drop-in” acceleration by executing standard BLAS calls on the GPU using NVBLAS. Because GPU threads access CPU memory at high bandwidth, porting techniques can avoid memory allocations and data transfer overheads. Grace CPU’s memory bandwidth and powerful compute performance deliver excellent performance to the parts of these applications that are better suited for CPUs, or not worth accelerating (see Table 1).

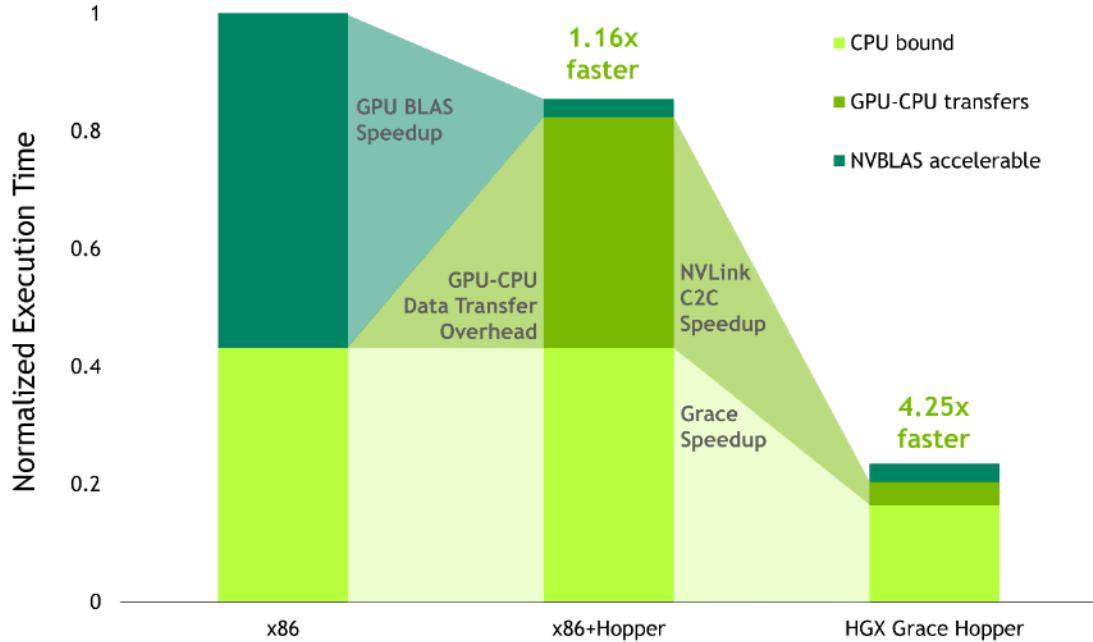


Figure 17. A performance simulation for ABINIT with NVBLAS featuring Titanium 255 Atoms using the LOBPCG algorithm.

OpenFOAM is a popular C++ toolbox for developing CFD and multiphysics solvers. When running on x86 CPUs, the pressure solver of OpenFOAM takes roughly 35% of the execution time of the OpenFOAM HPC Motorbike. After accelerating the pressure solver using AmgX and Hopper GPU, only 15% of the total runtime uses the GPU. The remaining 85% of the execution time is CPU-only and involves a mix of preprocessing, matrix assembly, and small linear solves.

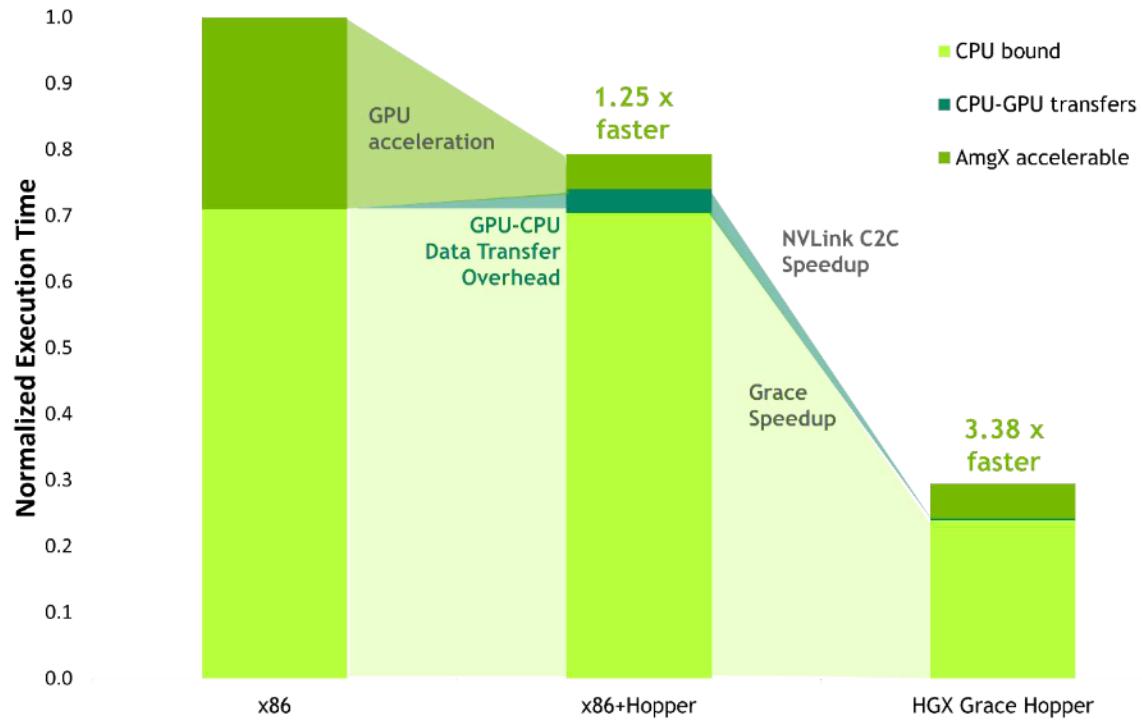


Figure 18. A performance simulation for OpenFOAM HPC Motorbike L (34 M cells)

The HPC Motorbike L case (a scaled-up version of the OpenFOAM Motorbike tutorial) is representative of typical engineering workloads, with a flatter profile and more time spent in turbulence calculations. The GPU portion of the code is still being optimized, but due to the substantial proportion of time spent in CPU-bound work, the Grace Hopper case provides significant speedup (Figure 18).

Molecular Dynamics: GROMACS

High-Bandwidth NVLink Switch System

NVIDIA is a strong GROMACS contributor with an excellent track record in accelerating molecular dynamics simulations. Together with our collaborators, we have recently accelerated the Particle Mesh-Ewald (PME) implementation with multiple GPUs. It is expected to be available in the GROMACS 2023 release.

The implementation assigns one GPU to PME for every three or four GPUs assigned to PP. PME rank uses the NVIDIA accelerated multi-node FFT library, cuFFTMp, which is part of the [NVIDIA HPC SDK](#), and is bottlenecked by inter-node bandwidth. Historically, InfiniBand has been the fastest and most efficient external networking method to connect nodes and servers, but significant performance gains are possible by connecting all the nodes using NVLink.

Figure 19 compares the projected performance improvements of Grace Hopper systems with NVLink Switch System for GROMACS stmv benchmark with 1,066,628 atoms, against the performance of an HGX-H100 platform with InfiniBand HDR-200 network connections across nodes. Figure 19 (left) shows the normalized speedups of the different components of a PME time-step using 32 GPUs. FFT and PME-PME communication are sensitive to inter-node communication bandwidth and benefit from a 1.3x NVLink Switch System speedup. Figure 19 (right) shows the total normalized speedup for a varying number of GPUs.

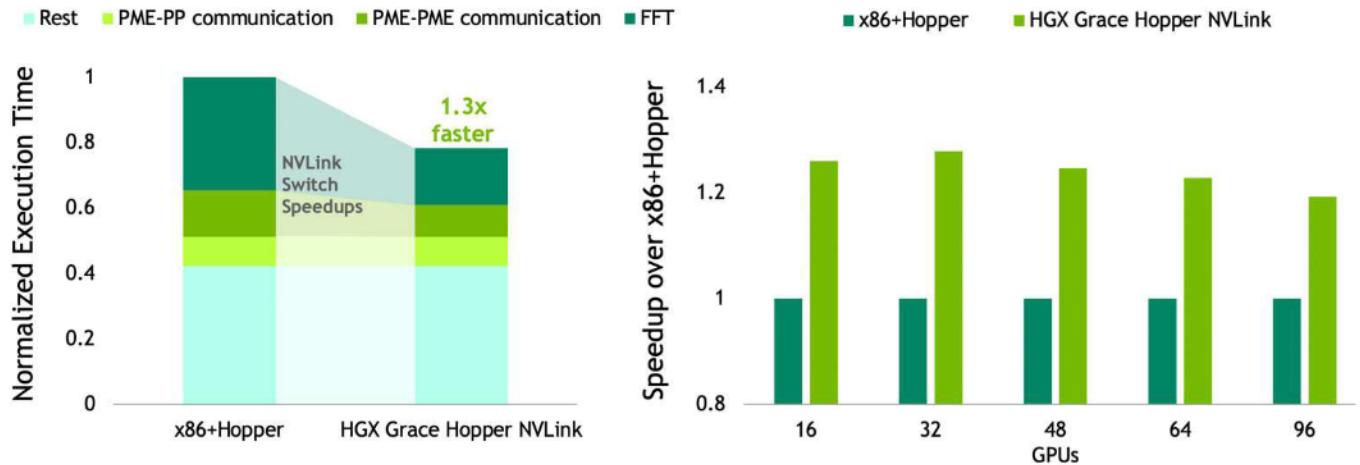


Figure 19. Performance simulations for GROMACS stmv Benchmark on Grace Hopper Superchip with NVLink Switch System

For more details about GROMACS GPU implementation, see [Pall et al. “Heterogeneous Parallelization and Acceleration of Molecular Dynamics Simulations in GROMACS”](#) and the NVIDIA Technical Developer Blog articles by Gray et al. [“Creating Faster Molecular Dynamics Simulations with GROMACS 2020”](#) and [“Maximizing GROMACS Throughput with Multiple Simulations per GPU Using MPS and MIG”](#).

Applications that perform FFTs are ideal candidates for being accelerated with Grace Hopper Superchip with NVLink Switch System. FFTs require high bandwidth all-to-all communications between GPU nodes. Figure 20 shows performance gains with Grace Hopper NVLink Switch System for FFTs of various sizes. It compares a Grace Hopper system with InfiniBand NDR400 versus a Grace Hopper NVLink Switch system.

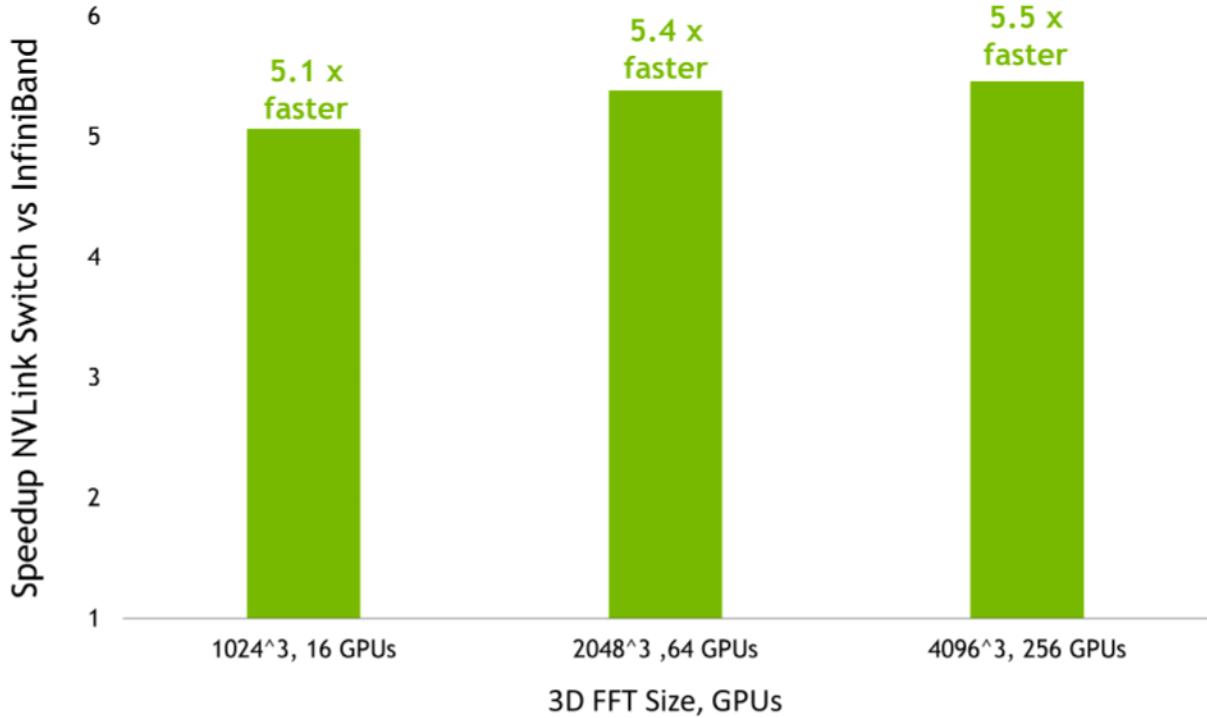


Figure 20. Speedup of NVLink Switch System on Grace Hopper versus InfiniBand NDR400 on NVIDIA Grace Hopper for 3D FFTs of different sizes

Multi-Grid Linear Solvers

Easier CPU and GPU Coordination

Linear solvers are probably the most common tool in scientific computing applications. Multigrid iterative methods deliver linear complexity by solving problems at different resolutions and smoothing low-frequency errors using coarser grids (Figure 21).

The V-cycle (left) iteratively smoothens the error, propagating the residual to the coarsest grid (left, bottom). A direct solver computes the error at the coarsest level, and the V-cycle iteratively interpolates and smoothens it up to the finer grids (left, top). The F-cycle (right) accelerates the convergence with successively finer V-cycles, resulting in coarser grid solves than in the classical V-cycle.

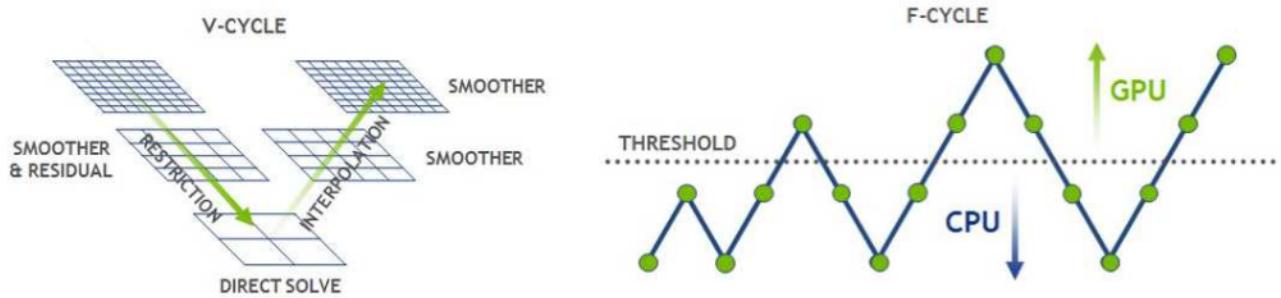


Figure 21. Grace Hopper Simplifies Multigrid Linear Solvers

Fine levels with many grid points run efficiently on throughput-oriented parallel architectures like GPUs. Coarse levels with few grid points are latency-limited on GPUs because they do not have enough work to utilize the available resources fully. A hybrid GPU-CPU scheme with a grid-size threshold ensures that coarse levels are executed on the latency-optimized Grace CPU (Figure 21 right).

When combined with adaptive mesh refinement and dynamic load balancing, data structures in multigrid solvers have deep hierarchies of pointers, and explicit data movement is challenging because of the indirect accesses. The NVIDIA Grace Hopper coherent programming model simplifies these applications and improves performance by allowing the GPU and CPU kernels to efficiently access all data and work on the same data structures without explicitly moving data around. This simplifies the implementation of switching between GPU and CPU kernels at a grid size threshold, while reducing the penalty for switching between processors, because both the GPU and CPU have fast access to system-allocated memory.

Appendix A: NVIDIA CUDA Platform

NVIDIA CUDA® is a comprehensive, productive, and high-performing platform for accelerated computing. It accelerates end-user applications at all levels, from system software to application-specific libraries and frameworks, using GPUs, CPUs, DPUs, and in-network computing. The CUDA platform has mature and user-friendly toolchains, developer tools, and documentation. It provides the best developer experience for accelerated heterogeneous applications.

CUDA System Software

The CUDA platform provides flexible system software components that help users deploy, manage, and optimize large heterogeneous systems productively and efficiently. The offering includes:

- Device drivers such as the CUDA driver
- Device management software such as NVML, NVIDIA System Management Interface, DCGM, and Unified Fabric Manager
- GPUDirect for heterogeneous network and file I/O
- Container-aware job-scheduling systems and operating systems such as DGX OS

High-Performance Libraries and Frameworks

NVIDIA Grace Hopper packs an immense amount of computing performance. A suite of heterogeneous libraries for compute-intensive applications complements the CUDA programming models to make this performance easily accessible. CUDA libraries maximize the performance of common math (CUDA Math Library), parallel algorithms (CUBLAS and Thrust), linear algebra (cuBLAS), dense and sparse linear solvers (cuSOLVER and cuSPARSE), FFTs (cuFFT), random number generation (cuRAND), tensor manipulation (cuTENSOR), image and signal processing (NPP), JPEG decoding (nvJPEG), and GPU management (NVML). cuNumeric transparently accelerates and distributes NumPy programs to machines of any scale through Legate and the Legion runtime without any code modifications. libcu++ provides heterogeneous synchronization and data-movement primitives to enable highly concurrent, heterogeneous, ISO-standard compliant C++ applications.

In addition, the CUDA platform communication libraries (Figure 11) enable standards-based scalable systems programming. HPC-X is a CUDA-aware MPI library with support for GPUDirect for sending and receiving GPU buffers directly using RDMA and GPU P2P. The NVIDIA Collective Communications Library (NCCL) implements highly optimized multi-node collective communication primitives. NVSHMEM is based on OpenSHMEM

and provides heterogeneous multi-node communication primitives for both host and device threads. cuFile and NVIDIA MAGNUM IO enable heterogeneous applications with high-performance file I/O through GPUDirect Storage.

An extensive suite of domain-specific libraries and frameworks further accelerates main algorithms in a wide range of application domains, for example:

- Deep neural networks ([cuDNN](#))
- Linear solvers for simulations and implicit unstructured methods ([AmgX](#))
- Quantum computing ([cuQuantum](#))
- Data science
- Machine learning ([RAPIDS](#))
- Data loading and preprocessing for machine learning ([DALI](#))
- Real-time 3D simulation and design collaboration ([Omniverse](#))

More than 150 Software Development Kits leverage these libraries to help developers become highly productive in a large set of application domains, including high-performance computing (NVIDIA HPC SDK), AI, Machine Learning, Deep Learning, and data science, genomics (NVIDIA CLARA), smart cities (NVIDIA Metropolis), autonomous driving (NVIDIA Drive SDKs), telecoms (NVIDIA Aerial SDK), robotics (NVIDIA Isaac SDK), Cybersecurity (NVIDIA Morpheus SDK), Computer Vision, and many more.

CUDA Profilers and Debuggers

The NVIDIA CUDA-GDB tool is an extension to GDB, the GNU Project debugger, and provides developers with a mechanism for debugging CUDA applications. The NVIDIA Compute Sanitizer is a functional correctness checking suite for highly concurrent CUDA kernels that precisely detects and attributes many common memory and thread safety errors like misaligned or out-of-bounds memory accesses, shared memory data races, uses of uninitialized memory, and invalid usages of synchronization primitives.

The NVIDIA NsightTM family of performance analysis tools help users identify coarse- and fine-grained optimization opportunities in their applications. NVIDIA Nsight Systems is a system-wide performance analysis tool designed to visualize an application's algorithms across many GPUs, CPUs, DPUs, Memory, Network I/O, and File I/O. NVIDIA Nsight Systems is fully integrated with the CUDA ecosystem; supporting tracing, sampling, and visualizing system, library, and framework API calls, e.g., CUDA, CUDA-X, RAPIDS, Magnum-IO, GPU Direct, MPI, UCX, OpenSHMEM, OpenMP, OpenACC, OS events, and even call-stack sampling. NVIDIA Nsight Compute is an interactive GPU kernel profiler that provides detailed performance and bottleneck analysis for optimizing single kernels towards peak GPU performance.

NVIDIA tools provide the same workflow experience on NVIDIA Grace CPU as on x86, and in addition support profiling and optimizing for the NVIDIA Grace Hopper architecture and its different single node and multi-node configurations.

NVIDIA profilers and debuggers are part of a larger tools and software ecosystem. Users should be able to use any tool they know and love on NVIDIA platforms. NVIDIA is a strong open-source contributor and Linux perf includes support for NVIDIA Grace Hopper Superchip Core and Uncore Performance Monitoring Units (PMUs), as well as ARM Statistical Profiling Extensions.

CUDA Documentation and Training

The large CUDA software ecosystem is complemented with excellent documentation for our programming models, for example, C++ parallel algorithms, libraries, [libcu++](#), frameworks, RAPIDS AI, and SDKs, (HPC SDK).

The NVIDIA Deep Learning Institute (DLI) offers self-paced and live trainings, for example, at conferences like Supercomputing and the International Supercomputing Conference, that enable individuals to advance their knowledge in AI, accelerated computing, accelerated data science, graphics, simulation, and more. DLI trains and certifies qualified educators as DLI Ambassadors, at research institutions and HPC centers, enabling them to teach and tailor the DLI content to their needs.

Beyond our official documentation, NVIDIA partners with different communities and HPC sites to provide the GPU Hackathon and Bootcamp program. It pairs teams of domain scientists and research software engineers (RSEs) with GPU mentors from NVIDIA and the HPC community to transfer the software development, parallel computing, and optimization skills required to effectively use modern heterogeneous computing systems. Every year, NVIDIA holds its [GPU Technology Conference \(GTC\)](#) with a focus on educating developers on the latest NVIDIA platform and technology. The talks cover NVIDIA programming models, hardware details, and the applications of accelerated computing to a wide range of domains. All these talks are recorded and available at NVIDIA On-Demand.

CUDA Language and Compilers

The CUDA platform exposes a unified and flexible compiler stack for generating highly optimized device binaries through the NVIDIA NVVM IR and the NVIDIA libNVVM library. NVVM IR is a compiler Intermediate Representation (IR), based on LLVM-IR, providing a front-end compiler target for generating GPU compute kernels. libNVVM is a library for compiling and optimizing NVVM IR too, the virtual ISA of NVIDIA GPUs. All NVIDIA Compute compilers use libNVVM to target NVIDIA GPUs (Figure 22) and it enables users and frameworks to bring their programming language of choice to the CUDA platform with the same code generation quality and optimization as CUDA C++ itself.

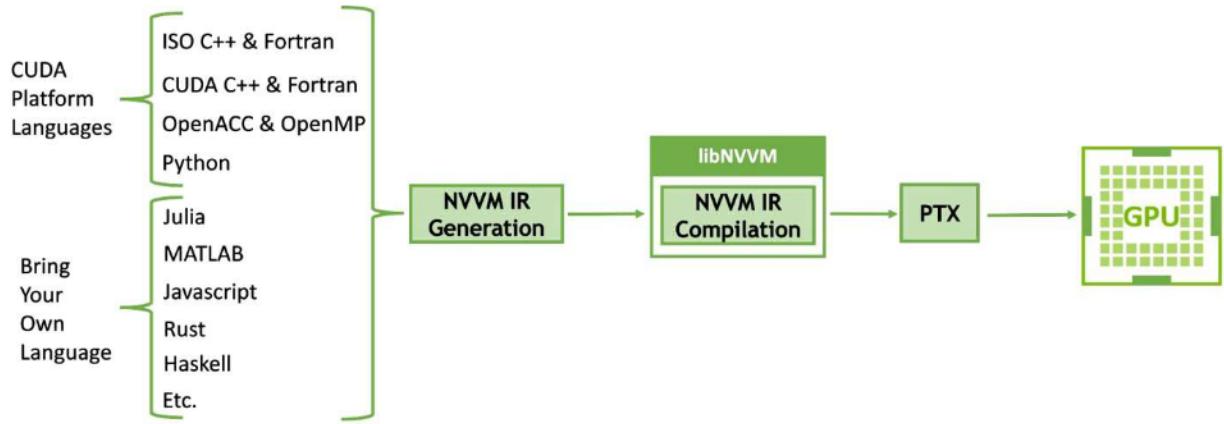


Figure 22. Compiling high-level Languages to PTX with libNVVM

PTX, the portable virtual ISA of NVIDIA GPUs, is a public ISA targeted by third-party producers to run efficiently on our target architectures. PTX also has the advantages of being forward compatible and can be assembled offline or at runtime.

In many applications, the GPU compute kernels to be generated depend on the program inputs. While these applications could generate NVVM IR, the NVIDIA Runtime Compiler significantly improves the productivity of these applications and their users by allowing them to generate familiar CUDA C++ instead. NVRTC compiles CUDA C++, at runtime to PTX using libNVVM or to native GPU binary code by using an embedded PTX assembler as well. This enables applications, for example, Python programs, to dynamically generate kernels for the program a user input and C++ programs, to specialize compute kernels at runtime depending on program inputs.

The NVIDIA HPC SDK is a set of toolchains for heterogeneous systems. NVCC is a CUDA C++ compiler and the NVIDIA HPC compilers: NVC, NVC++ and NVFortran. These toolchains enable users to pick the right compiler toolchain that is best suited for their application.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA, the NVIDIA logo, NVIDIA Grace GPU, CUDA, NVLink, NVIDIA GPU Cloud, and NSight are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

Arm

Arm, AMBA, and Arm Powered are registered trademarks of Arm Limited. Cortex, MPCore, and Mali are trademarks of Arm Limited. All other brands or product names are the property of their respective holders. "Arm" is used to represent Arm Holdings plc; its operating company Arm Limited; and the regional subsidiaries Arm Inc.; Arm KK; Arm Korea Limited.; Arm Taiwan Limited; Arm France SAS; Arm Consulting (Shanghai) Co. Ltd.; Arm Germany GmbH; Arm Embedded Technologies Pvt. Ltd.; Arm Norway, AS, and Arm Sweden AB.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Copyright

© 2022 NVIDIA Corporation. All rights reserved.