

**IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
AUSTIN DIVISION**

ONESTA IP, LLC,
Plaintiff,

V.

NVIDIA CORPORATION,
Defendant.

C.A. No. 1:25-cv-586

JURY TRIAL DEMANDED

ONESTA IP, LLC COMPLAINT FOR PATENT INFRINGEMENT

Pursuant to Federal Rule of Civil Procedure 15(a)(1)(B), Plaintiff Onesta IP, LLC (“Plaintiff” or “Onesta”) files this Complaint for patent infringement (“Complaint”) and for a jury trial against Defendant NVIDIA Corporation (“Defendant” or “NVIDIA”). Onesta alleges as follows:

THE PARTIES

1. Onesta is a Delaware Corporation with a principal place of business located at 230 Sugartown Road, Suite 100, Wayne, PA 19087.
2. Onesta is the owner by way of assignment of U.S. Patent No. 8,854,381 (“the ’381 Patent”) attached as Exhibit A, U.S. Patent No. 9,519,943 (“the ’943 Patent”) attached as Exhibit B, U.S. Patent No. 11,741,019 (“the ’019 Patent”) attached as Exhibit C, U.S. Patent No. 11,841,803 (“the ’803 Patent”) attached as Exhibit D, and U.S. Patent No. 9,116,809 (“the ’809 Patent”) attached as Exhibit E. All of the aforementioned patents are collectively referred to herein as the “Asserted Patents.”
3. Defendant NVIDIA is a corporation organized under the laws of the State of

Delaware and has a regular and established place of business within this judicial District, including, at least, at 11001 Lakeline Boulevard, Building 2, Suite 100, Austin, TX 78717.

4. Upon information and belief, NVIDIA designs and makes graphics processing units (“GPUs”) and related technologies, and “develops a platform for scientific computing, AI, data science, autonomous vehicles, robotics, metaverse, and 3D internet applications, as well as focuses on PC graphics.”¹ According to NVIDIA’s 2024 Annual Report, NVIDIA “specialize[s] in markets where our computing platforms can provide tremendous acceleration for applications[,]” including “Data Center, Gaming, Professional Visualization, and Automotive” markets.²

THE ASSERTED PATENTS

5. The ’381 Patent is entitled “Processing Unit That Enables Asynchronous Task Dispatch,” and issued on October 7, 2014, to inventors Michael Mantor and Rex McCrary. The ’381 Patent issued from United States Patent Application No. 12/874,134, which was filed on September 1, 2010.

6. The ’943 Patent is entitled “Priority-Based Command Execution,” and issued on December 13, 2016, to inventors Philip J. Rogers, David Gotwalt, Tom Frisinger, and Rex McCrary. The ’943 Patent issued from United States Patent Application No. 13/171,979, which was filed on June 29, 2011.

7. The ’019 Patent is entitled “Memory Pools in a Memory Model for a Unified Computing System,” and issued on August 29, 2023, to inventors Anthony Asaro, Kevin Normoyle, and Mark Hummel. The ’019 Patent issued from United States Patent Application No. 17/471,552,

¹ See, e.g., <https://www.bloomberg.com/profile/company/NVDA:US> (accessed 04/14/2025); NVIDIA 2024 Annual Report at 104-05, available at: https://s201.q4cdn.com/141608511/files/doc_financials/2024/ar/NVIDIA-2024-Annual-Report.pdf (accessed 04/14/2025).

² https://s201.q4cdn.com/141608511/files/doc_financials/2024/ar/NVIDIA-2024-Annual-Report.pdf at 104-105 (accessed 04/14/2025).

which was filed on September 10, 2021.

8. The '803 Patent is entitled "GPU Chiplets Using High Bandwidth Crosslinks," and issued on December 12, 2023, to inventors Skyler J. Saleh, Samuel Naffziger, Milind S. Bhagavat, and Rahul Agarwal. The '803 Patent issued from United States Application No. 16/456,287, which was filed on June 28, 2019.

9. The '809 Patent is entitled "Memory Heaps in a Memory Model for a Unified Computing System," and issued on August 25, 2015, to inventors Anthony Asaro, Kevin Normoyle, and Mark Hummel. The '809 Patent issued from United States Patent Application No. 13/724,879, which was filed on December 21, 2012.

10. The Asserted Patents are each valid and enforceable.

11. Onesta owns all right, title, and interest in the Asserted Patents.

JURISDICTION AND VENUE

12. This is an action for patent infringement under the patent laws of the United States, Title 35 of the United States Code. Accordingly, this Court has exclusive subject matter jurisdiction over this action under 28 U.S.C. §§ 1331 and 1338(a).

13. This Court has personal jurisdiction over NVIDIA. NVIDIA is subject to this Court's specific personal jurisdiction pursuant to due process and/or the Texas Long Arm Statute, at least in part, because NVIDIA conducts business in this judicial District. *See* Tex. Civ. Prac. & Rem. Code §§ 17.041-.045. Moreover, Onesta's causes of action arise, at least in part, from NVIDIA's contacts with and activities in the State of Texas and this judicial District. *Id.*

14. NVIDIA has done business in and has committed acts of infringement in this District by offering (directly and through intermediaries), inducing third parties to offer, and contributing to the offering of its products, including those accused of infringement here, to customers and potential customers located in this District.

15. NVIDIA has committed acts within this judicial District giving rise to this action and has established sufficient minimum contacts with the State of Texas such that the exercise of jurisdiction would not offend traditional notions of fair play and substantial justice. Furthermore, personal jurisdiction over NVIDIA in this action comports with due process. For example, in addition to NVIDIA regularly conducting business within this District, it has purposefully availed itself to the privileges of conducting business in this District and has sought protection and benefit from the laws of the State of Texas. Having purposefully availed itself of the privilege of conducting business within this District, NVIDIA should reasonably and fairly anticipate being brought into court here.

16. NVIDIA has also repeatedly acknowledged this Court has personal jurisdiction over it. *See, e.g., Vantage Micro LLC v. NVIDIA Corp.*, Case No. 6:19-cv-00582-RP, Dkt. No. 22 (W.D. Tex. Jan. 4, 2020) (admitting to personal jurisdiction in this District); *Ocean Semiconductor LLC v. NVIDIA Corp.*, Case No. 6:20-cv-01211-ADA, Dkt. No. 14 (W.D. Tex. Mar. 12, 2021) (same); *Xockets, Inc. v. NVIDIA Corp.*, Case No. 6:24-cv-00453, Dkt. No. 172 at 7 (W.D. Tex. Jan. 21, 2025) (admitting “NVIDIA maintains an office in Austin, Texas”). Further, NVIDIA has admitted “it is subject to this Court’s general personal jurisdiction.” *Ocean Semiconductor*, Case No. 6:20-cv-01211-ADA, Dkt. No. 14.

17. Venue in this judicial District is proper under 28 U.S.C §§ 1391(b), (c), and 1400 because NVIDIA has regular and established places of business within this District and has committed acts of infringement in this District.

18. NVIDIA is a registered business in Texas and has regular and established places of business in this District, including at least such offices and/or facilities at 11001 Lakeline Boulevard, Building 2, Suite 100, Austin, TX 78717.

19. Upon information and belief, NVIDIA has hundreds of employees based in the Western District of Texas and does business in this judicial District and across the State of Texas.³ NVIDIA's employees in this judicial District include, for example, software engineers, support engineers, design verification engineers, physical design engineers, SoC architects, power management firmware engineers, and CPU verification engineers.⁴ Upon information and belief, these personnel have expertise in, among other aspects of the infringing NVIDIA products described *infra*, including testing, validation, and qualification activities relating to such products, including to satisfy customer specific requirements. Accordingly, upon information and belief, witnesses and documents relevant to this action are located in this judicial District.

20. Upon information and belief, NVIDIA sells infringing NVIDIA products including, but not limited to, the accused NVIDIA GeForce RTX 4060 Ti graphics card, within this judicial District including, at least, at the following Best Buy locations: 4970 US-290, Austin, TX 78735, 9607 Research Boulevard Suite 500, Austin, TX 78759, and 1201 Barbara Jordan Boulevard Suite 100, Austin, TX 78723.

21. Similarly, upon information and belief, NVIDIA offers for sale the infringing NVIDIA Blackwell GB200 Superchip circuit board assembly that is available for purchase throughout the United States and within this judicial District.⁵

³ See, e.g., <https://www.linkedin.com/company/nvidia/people/?facetGeoRegion=104472865%2C90000064> (accessed 04/14/2025).

⁴ See, e.g., <https://nvidia.wd5.myworkdayjobs.com/NVIDIAExternalCareerSite?locationHierarchy1=2fcb99c455831013ea52fb338f2932d8&locations=91336993fab910af6d702b631b94c2de> (accessed 04/14/2025); <https://www.linkedin.com/company/nvidia/people/?facetGeoRegion=104472865%2C90000064> (accessed 04/04/2025).

⁵ Upon information and belief, the infringing NVIDIA Blackwell GB200 Superchip is available by direct bilateral contract with NVIDIA and/or its distributors. See, e.g., <https://nvidianews.nvidia.com/news/nvidia-blackwell-platform-arrives-to-power-a-new-era-of-computing> (accessed 04/14/2025); <https://www.coreweave.com/products/nvidia-blackwell> (accessed 04/14/2025); <https://lambda.ai/nvidia-blackwell-gpus> (accessed 04/14/2025).

ALLEGATIONS OF PATENT INFRINGEMENT

22. Onesta incorporates the allegations of all the foregoing paragraphs as if fully restated herein.

23. As set forth below, the infringing products consist of and/or incorporate, without any license from Onesta, products protected by patents owned by Onesta, including, without limitation, integrated circuits, circuit board assemblies, graphics cards, and computers. Onesta respectfully seeks relief from this Court for NVIDIA's infringement.

24. Onesta reserves the right to accuse any forthcoming NVIDIA technology or products not yet commercially available, and any products about which it learns additional relevant information.

25. Onesta is the sole and exclusive owner of all right, title, and interest in and to the Asserted Patents, and hold the exclusive right to take all actions necessary to enforce their rights to the Asserted Patents, including the filing of this action. Onesta also has the right to recover all damages for past, present, and future infringement of the Asserted Patents.

26. NVIDIA has directly infringed, and continues to directly infringe, the Asserted Patents under 35 U.S.C. §§ 271 (a) and (g) by making, having made, using, selling, offering for sale, importing, and/or having imported, the products that infringe in this District and elsewhere in the United States, as described in further described in Counts I-V *infra*.

27. NVIDIA has been placed on actual notice of the Asserted Patents at least as early as the filing of this Complaint, which constitutes notice of the Asserted Patents in accordance with 35 U.S.C. § 287.

28. NVIDIA has also indirectly infringed and continues to indirectly infringe the Asserted Patents under 35 U.S.C. § 271(b) by actively inducing its subsidiaries, affiliates, retail partners, customers, and/or other third parties to infringe the Asserted Patents by selling the

products that infringe to such third parties for use in end-user products in a manner that infringes the Asserted Patents, as described in detail in Counts I-V *infra*.

29. Additionally, NVIDIA has indirectly infringed and continues to indirectly infringe the Asserted Patents under 35 U.S.C. § 271(c) by materially contributing to infringement of the Asserted Patents by its subsidiaries, affiliates, retail partners, customers, and/or other third parties by selling the products that infringe to such third parties for use in end-user products in a manner that infringes the Asserted Patents, as described in detail in Counts I-V *infra*.

30. NVIDIA's acts of infringement have caused damage to Onesta. Onesta is entitled to recover from NVIDIA the damages incurred by Onesta as a result of NVIDIA's wrongful acts.

31. NVIDIA's acts of direct and indirect infringement are willful and have also caused, and will continue to cause, substantial damage and irreparable harm to Onesta, and Onesta has no adequate remedy at law. NVIDIA performed and continues to perform the acts that constitute direct and/or indirect infringement, with knowledge or willful blindness that the acts would constitute direct and/or indirect infringement of the Asserted Patents. Notwithstanding NVIDIA's knowledge of the Asserted Patents since at least as early as the filing of the present Complaint, NVIDIA has and continues to willfully infringe the Asserted Patents.

COUNT I

Infringement of the '381 Patent

32. Onesta incorporates the allegations of all of the foregoing paragraphs as if fully restated herein.

33. NVIDIA has directly and indirectly infringed, and continues to directly and indirectly infringe, the '381 Patent by making, having made, using, selling, offering for sale, and/or importing into the United States products that infringe the '381 Patent. The products that infringe one or more claims of the '381 Patent include, but are not limited to, at least the products identified

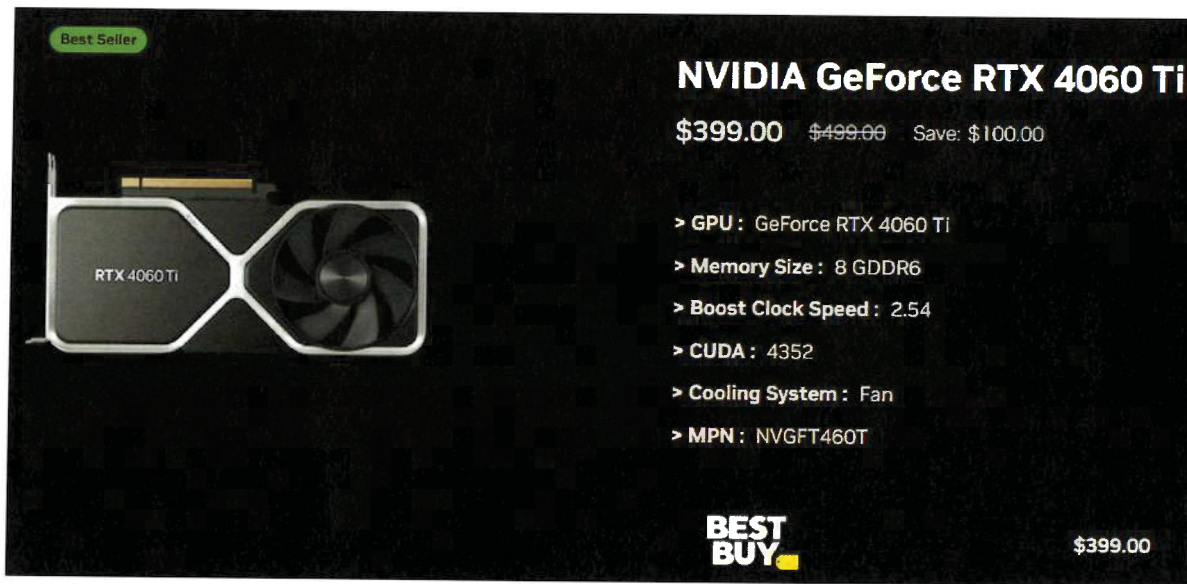
herein. Further discovery may reveal additional infringing products and/or models.

34. For example, and without limitation, the infringing products infringe one or more claims of the '381 Patent, including but not limited to claim 5. The '381 Accused Products fall within the scope of and include, either literally or under the doctrine of equivalents, all of the elements of at least claim 5 of the '381 Patent.

35. The products that infringe include all the limitations of at least claim 5 of the '381 Patent. Specifically, the '381 Patent claims, *e.g.*, an apparatus comprising: a plurality of engines associated with a first processing unit and configured to receive, from a scheduling module associated with a second processing unit, a plurality of tasks and to load a state data associated with each of the plurality of tasks; a shader core associated with the first processing unit and configured to receive the plurality of tasks from at least one of the plurality of engines and to execute a first task from the plurality while executing a second task from the plurality of tasks based on respective state data associated with each of the first and second tasks; and, wherein the first task comprises a graphics-processing task and the second task comprises a general-computing task.

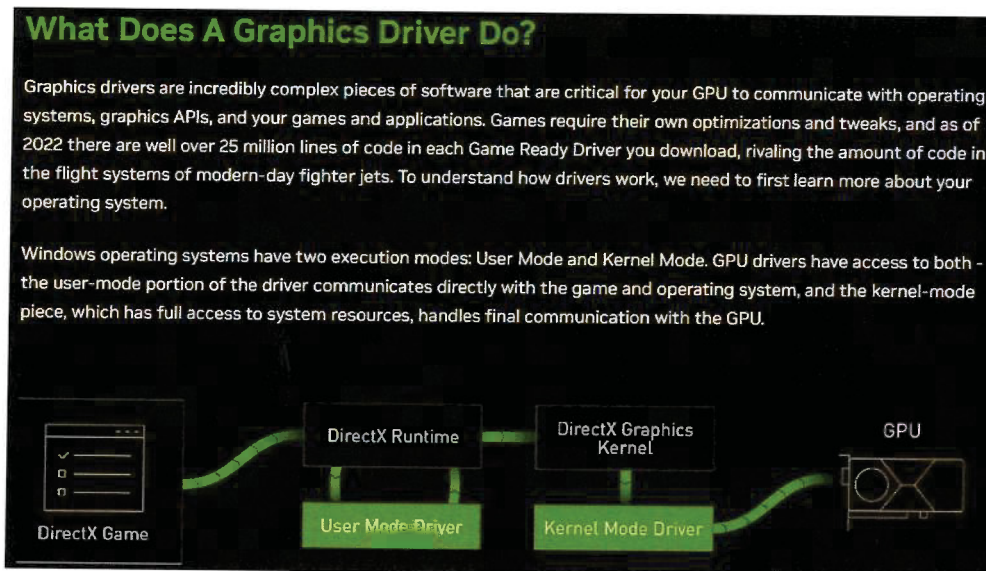
36. With respect to exemplary devices, the NVIDIA GeForce RTX 4060 Ti graphics card, made or sold by NVIDIA, incorporating the NVIDIA GeForce RTX 4060 Ti graphics processing unit (GPU) and associated software, made or sold by NVIDIA, directly infringes at least claim 5 of the '381 Patent.

37. For example, the NVIDIA GeForce RTX 4060 Ti graphics card includes the GeForce RTX 4060 Ti GPU and associated software, such as graphics drivers.



<https://marketplace.nvidia.com/en-us/consumer/graphics-cards/nvidia-geforce-rtx-4060-ti/>

(accessed 04/14/2025).



<https://www.nvidia.com/en-us/geforce/news/game-ready-driver-program-ultimate-gaming-platform/> (accessed 04/14/2025).

38. Upon information and belief, the GeForce RTX 4060 Ti GPU includes a plurality of engines associated with a first processing unit and configured to receive, from a scheduling

module associated with a second processing unit, a plurality of tasks and to load state data associated with each of the plurality of tasks. For example, the GeForce RTX 4060 Ti GPU includes the “GigaThread Engine,” “GPU front end (command processor),” graphics engine, compute engine, 3D engine, and/or “asynchronous engines,” which, on information and belief, are configured to receive a plurality of tasks from NVIDIA’s graphics driver run on a host CPU and to load state data associated with each of a plurality of tasks.



<https://images.nvidia.com/aem-dam/Solutions/Data-Center/l4/nvidia-ada-gpu-architecture-whitepaper-v2.1.pdf> (accessed 04/14/2025), at 7.

SYS Clock Frequency - `sys__cycles_elapsed.avg.per_second`

The average SYS clock frequency in hertz. The GPU front end (command processor), copy engines, and the performance monitor run at the SYS clock. On Turing and NVIDIA GA100 GPUs, the sampling frequency is based upon a period of SYS clocks (not time) so samples per second will vary with SYS clock. On NVIDIA GA10x GPUs, the sampling frequency is based upon a fixed frequency clock. The maximum frequency scales linearly with the SYS clock.

GR Active - `gr__cycles_active.sum.pct_of_peak_sustained_elapsed`

The percentage of cycles the graphics/compute engine is active. The graphics/compute engine is active if there is any work in the graphics pipe or if the compute pipe is processing work.

GA100 MIG - MIG is not yet supported. This counter will report the activity of the primary GR engine.

<https://docs.nvidia.com/nsight-systems/UserGuide/index.html#gms-available-metrics> (accessed 04/14/2025).

asyncEngines

INTEGER,

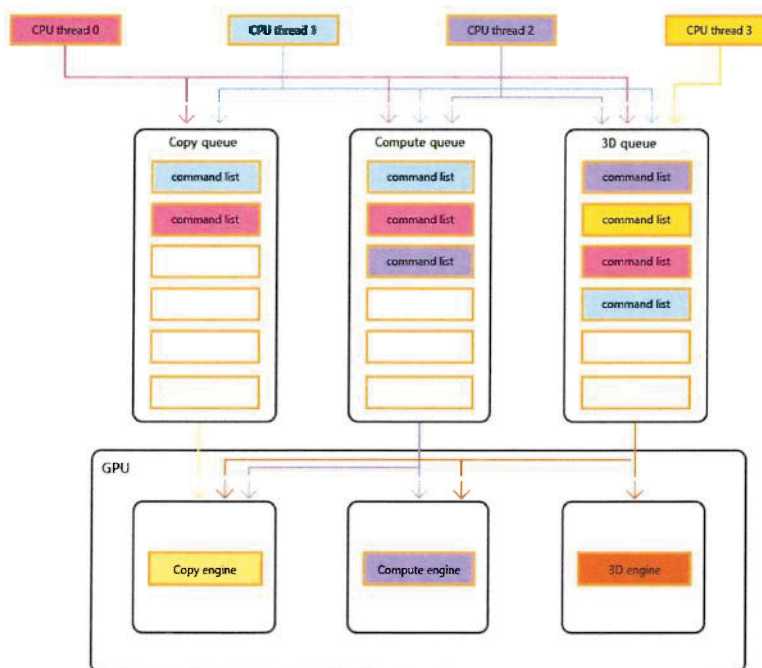
-- Number of asynchronous engines.

Id.

GPU engines

The following diagram shows a title's CPU threads, each populating one or more of the copy, compute, and 3D queues. The 3D queue can drive all three GPU engines; the compute queue can drive the compute and copy engines; and the copy queue simply the copy engine.

As the different threads populate the queues, there can be no simple guarantee of the order of execution, hence the need for synchronization mechanisms—when the title requires them.



<https://learn.microsoft.com/en-us/windows/win32/direct3d12/user-mode-heap-synchronization>

(accessed 04/14/2025); see also [https://docs.nvidia.com/nsight-](https://docs.nvidia.com/nsight-graphics/AdvancedLearning/index.html)

[graphics/AdvancedLearning/index.html](https://docs.nvidia.com/nsight-graphics/AdvancedLearning/index.html) (accessed 04/14/2025);

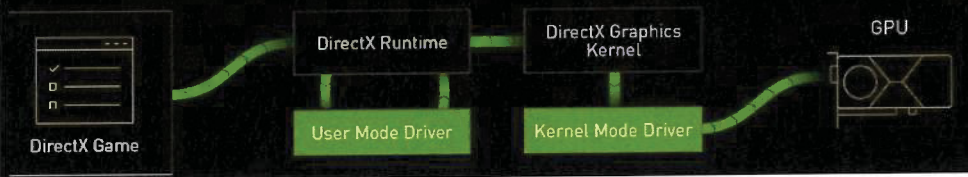
<https://vulkan.gpuinfo.org/displayreport.php?id=34756#queuefamilies> (accessed 04/14/2025);

https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf (accessed 04/14/2025).

What Does A Graphics Driver Do?

Graphics drivers are incredibly complex pieces of software that are critical for your GPU to communicate with operating systems, graphics APIs, and your games and applications. Games require their own optimizations and tweaks, and as of 2022 there are well over 25 million lines of code in each Game Ready Driver you download, rivaling the amount of code in the flight systems of modern-day fighter jets. To understand how drivers work, we need to first learn more about your operating system.

Windows operating systems have two execution modes: User Mode and Kernel Mode. GPU drivers have access to both - the user-mode portion of the driver communicates directly with the game and operating system, and the kernel-mode piece, which has full access to system resources, handles final communication with the GPU.



<https://www.nvidia.com/en-us/geforce/news/game-ready-driver-program-ultimate-gaming-platform/> (accessed 04/14/2025).

The CUDA Execution Model

Work Submission in CUDA

In the CUDA model, all work (i.e. CUDA tasks) are submitted to the GPU through CUDA streams, which execute their respective tasks in first-in, first-out (FIFO) order. In order to enable concurrency, applications may create multiple CUDA streams and enqueue work on separate streams to indicate that tasks can be processed in parallel. CUDA events can be used to identify dependencies across streams by recording an event on one stream and waiting on the event in another stream.

NVSHMEM stream-based operations (e.g. `nvshmemx_putmem_on_stream`) enqueue the corresponding operation on a CUDA stream and return immediately. The NVSHMEM operation is not performed until it reaches the head of the stream and is executed by the CUDA runtime. All NVSHMEM on-stream operations follow the NVSHMEM memory model. For example, `nvshmemx_quiet_on_stream` operations can be used to order or complete operations, respectively.

<https://docs.nvidia.com/nvshmem/api/cuda-interactions.html> (accessed 04/14/2025); *see also*

<https://par.nsf.gov/servlets/purl/10064498> (accessed 04/14/2025).

39. Further, upon information and belief, the GeForce RTX 4060 Ti GPU includes a shader core associated with the first processing unit and is configured to receive the plurality of tasks from at least one of the plurality of engines and to execute a first task from the plurality of

tasks while executing a second task from the plurality of tasks based on respective state data associated with each of the first and second tasks. For example, upon information and belief, the GeForce RTX 4060 Ti GPU includes “Graphics Processing Clusters (GPCs),” “Texture Processing Clusters (TPCs),” and “Streaming Multiprocessors (SMs)” and can “concurrently run compute and 3D” tasks.

The full AD102 GPU includes 12 Graphics Processing Clusters (GPCs), 72 Texture Processing Clusters (TPCs), 144 Streaming Multiprocessors (SMs), and a 384-bit memory interface with 12 32-bit memory controllers.



Note: The AD102 GPU also includes 288 FP64 Cores (2 per SM) which are not depicted in the above diagram. The FP64 TFLOP rate is 1/64th the TFLOP rate of FP32 operations. The small number of FP64 Cores are included to ensure any programs with FP64 code operate correctly, including FP64 Tensor Core code.

<https://images.nvidia.com/aem-dam/Solutions/Data-Center/l4/nvidia-ada-gpu-architecture-whitepaper-v2.1.pdf> (accessed 04/14/2025), at 7.

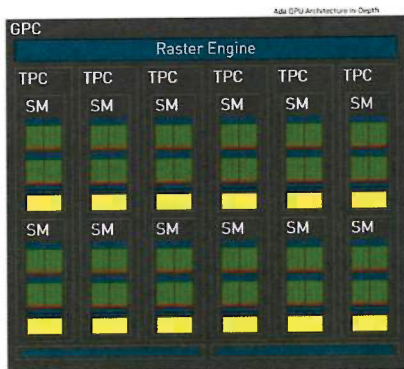


Figure 2. Ada GPC Block Diagram

The GPC is the dominant high-level hardware block within all AD10x Ada family GPUs, with all of the key graphics processing units residing within a GPC. Each GPC includes a dedicated Raster Engine, two Raster Operations (ROPs) partitions, with each partition containing eight individual ROP units, and six TPCs. Each TPC includes one PolyMorph Engine and two SMs.

Each SM in AD10x GPUs contain 128 CUDA Cores, one Ada Third-Generation RT Core, four Ada Fourth-Generation Tensor Cores, four Texture Units, a 256 KB Register File, and 128 KB of L1/Shared Memory, which can be configured for different memory sizes depending on the needs of the graphics or compute workload.

Id. at 8.



Figure 5 Ada Streaming Multiprocessor (SM)

Id. at 11; *see also* <https://developer.nvidia.com/blog/gpu-pro-tip-cuda-7-streams-simplify-concurrency/> (accessed 04/14/2025); https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf (accessed 04/14/2025); <https://par.nsf.gov/servlets/purl/10064498> (accessed 04/14/2025); <https://docs.nvidia.com/nsight-graphics/AdvancedLearning/index.html> (accessed 04/14/2025).

40. Upon information and belief, the GPCs and/or SMs are configured to receive the plurality of tasks from at least one of the plurality of engines and to execute a first task from the plurality of tasks while executing a second task from the plurality of tasks based on respective state data associated with each of the first and second tasks.

Heterogeneous computing is about efficiently using all processors in the system, including CPUs and GPUs. To do this, applications must execute functions concurrently on multiple processors. CUDA Applications manage concurrency by executing asynchronous commands in *streams*, sequences of commands that execute in order. Different streams may execute their commands concurrently or out of order with respect to each other. [See the post [How to Overlap Data Transfers in CUDA C/C++ for an example](https://developer.nvidia.com/blog/gpu-pro-tip-cuda-7-streams-simplify-concurrency/)]

<https://developer.nvidia.com/blog/gpu-pro-tip-cuda-7-streams-simplify-concurrency/> (accessed 04/14/2025).

6.2.8. Asynchronous Concurrent Execution

CUDA exposes the following operations as independent tasks that can operate concurrently with one another:

- ▶ Computation on the host;
- ▶ Computation on the device;
- ▶ Memory transfers from the host to the device;
- ▶ Memory transfers from the device to the host;
- ▶ Memory transfers within the memory of a given device;
- ▶ Memory transfers among devices.

https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf (accessed 04/14/2025), at 46.

Asynchronous Compute

The only way to concurrently run compute and 3D is by simultaneously:

- › sending 3D work to the DIRECT queue
- › sending compute work to an ASYNC_COMPUTE queue

On Ampere, you can also dispatch concurrent compute workloads by dispatching it on both the DIRECT and ASYNC_COMPUTE queue.

You can detect whether a program is taking advantage of async compute in several ways:

The "Compute In Flight" row contains an "Async Compute In Flight" counter.

Observe when the compute warps executed on the SM Occupancy row, and determine if they were Sync or Async based on the color of the "Compute In Flight" row.



Look for multiple queue rows; the ASYNC_COMPUTE queue will appear as something other than Q0.

Compute will only run simultaneously with graphics if submitted on from an ASYNC_COMPUTE queue. This can disambiguate the SM Occupancy row.



<https://docs.nvidia.com/nsight-graphics/AdvancedLearning/index.html> (accessed 04/14/2025).

7.1. SIMT Architecture

The multiprocessor creates, manages, schedules, and executes threads in groups of 32 parallel threads called *warps*. Individual threads composing a warp start together at the same program address, but they have their own instruction address counter and register state and are therefore free to branch and execute independently. The term *warp* originates from weaving, the first parallel thread technology. A *half-warp* is either the first or second half of a warp. A *quarter-warp* is either the first, second, third, or fourth quarter of a warp.

When a multiprocessor is given one or more thread blocks to execute, it partitions them into warps and each warp gets scheduled by a *warp scheduler* for execution. The way a block is partitioned into warps is always the same; each warp contains threads of consecutive, increasing thread IDs with the first warp containing thread 0. *Thread Hierarchy* describes how thread IDs relate to thread indices in the block.

A warp executes one common instruction at a time, so full efficiency is realized when all 32 threads of a warp agree on their execution path. If threads of a warp diverge via a data-dependent conditional branch, the warp executes each branch path taken, disabling threads that are not on that path. Branch divergence occurs only within a warp; different warps execute independently regardless of whether they are executing common or disjoint code paths.

https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf (accessed 04/14/2025), at

135.

7.2. Hardware Multithreading

The execution context (program counters, registers, and so on) for each warp processed by a multiprocessor is maintained on-chip during the entire lifetime of the warp. Therefore, switching from one execution context to another has no cost, and at every instruction issue time, a warp scheduler selects a warp that has threads ready to execute its next instruction (the *active threads* of the warp) and issues the instruction to those threads.

In particular, each multiprocessor has a set of 32-bit registers that are partitioned among the warps, and a *parallel data cache* or *shared memory* that is partitioned among the thread blocks.

The number of blocks and warps that can reside and be processed together on the multiprocessor for a given kernel depends on the amount of registers and shared memory used by the kernel and the amount of registers and shared memory available on the multiprocessor. There are also a maximum number of resident blocks and a maximum number of resident warps per multiprocessor. These limits as well the amount of registers and shared memory available on the multiprocessor are a function of the compute capability of the device and are given in *Compute Capabilities*. If there are not enough registers or shared memory available per multiprocessor to process at least one block, the kernel will fail to launch.

Id. at 137.

41. Also, upon information and belief, the first task comprises a graphics-processing task and the second task comprises a general-compute task. <https://images.nvidia.com/aem->

[dam/Solutions/Data-Center/l4/nvidia-ada-gpu-architecture-whitepaper-v2.1.pdf](#) (accessed 04/14/2025); <https://developer.nvidia.com/blog/gpu-pro-tip-cuda-7-streams-simplify-concurrency/> (accessed 04/14/2025); https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf (accessed 04/14/2025); <https://par.nsf.gov/servlets/purl/10064498> (accessed 04/14/2025); <https://docs.nvidia.com/nsight-graphics/AdvancedLearning/index.html> (accessed 04/14/2025); <https://www.atlantic.net/gpu-server-hosting/gpu-parallel-computing-techniques-challenges-and-best-practices/> (accessed 04/14/2025).

42. NVIDIA has indirectly infringed, and continues to indirectly infringe, the '381 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the '381 Patent by others, such as its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties. These others directly infringe the '381 Patent by, for example, making, using, testing, selling, offering for sale throughout, or importing into the United States, including this judicial District, downstream products (for example, without limitation, personal computers) incorporating the products that infringe (for example, without limitation, the Exemplary NVIDIA GeForce RTX 4060 Ti graphics card). NVIDIA has induced and continues to induce such direct infringement by encouraging and facilitating others' infringing acts. For example, NVIDIA hires permanent sales and/or marketing personnel located throughout the United States, and in this judicial District. On information and belief, these sales and/or marketing personnel are engaged in activities, that are targeted to original equipment manufacturers, customers, resellers, retailers, and other parties, including original equipment manufacturers, customers, resellers, retailers, and other parties based on the United States, including instructions for making, using, testing, or selling the products that infringe.

43. NVIDIA specifically intended or intends these others, such as its subsidiaries and

affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties, to infringe the '381 Patent and knew or knows that the induced acts of these others constitute direct infringement. For example, although NVIDIA has notice of the '381 Patent, the scope of its claims, and the products covered thereby, NVIDIA nonetheless intentionally and knowingly encourages and facilitates these others to make, use, test, sell, offer for sale in, or import into, such covered products into the United States and thereby directly infringe the '381 Patent. Additionally, for example, NVIDIA designed the products that infringe (for example, without limitation, the Exemplary NVIDIA GeForce RTX 4060 Ti graphics card) such that they would each infringe the '381 Patent if made, used, tested, sold, offered for sale, or imported into the United States. NVIDIA provided, directly, or indirectly, the products that infringe to others, such as, but not limited to, its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties, knowing and intending that those others use, test, sell, offer for sale, and/or import in and into the United States downstream products (for example, without limitation, personal computers) that include the products that infringe (for example, without limitation the Exemplary NVIDIA GeForce RTX 4060 Ti graphics card) thereby directly infringing one or more claims of the '381 Patent.

44. NVIDIA has also indirectly infringed and continues to indirectly infringe the '381 Patent under 35 U.S.C. § 271(c) by materially contributing to infringement of the '381 Patent by its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties by selling the products that infringe to such third parties for use in downstream end-user products in a manner that infringes the '381 Patent. The products that infringe (for example, without limitation, the Exemplary NVIDIA GeForce RTX 4060 Ti graphics card) have no substantial non-infringing uses and are a material part of the invention. Moreover, the products

that infringe provide vital functionality to such downstream products (for example, without limitation, personal computers). Further, the products that infringe cannot be used without being incorporated into a downstream product. Thus, any manufacture, use, test, sale, offer for sale, or importation in or into the United States of a product that infringes or a downstream product incorporating a product that infringes, infringes the '381 Patent.

45. As a result of NVIDIA's infringement of the '381 Patent, Onesta is entitled to monetary damages in an amount adequate to compensate for NVIDIA's infringement, but in no event less than a reasonable royalty for the use made of the invention by NVIDIA, together with interest and costs as fixed by the Court.

46. Despite NVIDIA's actual notice of its infringement as of at least the filing of the Complaint, NVIDIA's acts of direct and indirect infringement of the '381 Patent are deliberate and willful, and have caused, and will continue to cause, substantial damage and irreparable harm to Onesta, and Onesta has no adequate remedy at law.

COUNT II

Infringement of the '943 Patent

47. Onesta incorporates the allegations of all of the foregoing paragraphs as if fully restated herein.

48. NVIDIA has directly and indirectly infringed, and continues to directly and indirectly infringe, the '943 Patent by making, having made, using, selling, offering for sale, and/or importing into the United States products that infringe the '943 Patent. The products that infringe one or more claims of the '943 Patent include, but are not limited to, at least the products identified herein. Further discovery may reveal additional infringing products and/or models.

49. For example, and without limitation, the infringing products infringe one or more claims of the '943 Patent, including but not limited to claim 11. The products that infringe fall

within the scope of and include, either literally or under the doctrine of equivalents, all of the elements of at least claim 11 of the '943 Patent.

50. The products that infringe include all of the limitations of at least claim 11 of the '943 Patent. Specifically, the '943 Patent claims, *e.g.*, a processing device, comprising: a set of queues, each queue of the set of queues being configured to hold commands received from a central processing unit (CPU), wherein each command is sent from a virtual device in the CPU; a command processor configured to retrieve the received commands from the set of queues, wherein the set of queues includes a high priority queue that holds high priority commands, wherein the command processor is configured to retrieve a high priority command held in the high priority queue before retrieving commands held in other queues of the set of queues, and wherein the processing device is configured to execute the received commands; and a processing core configured to execute the received command, wherein the command processor is configured to retrieve commands from the set of queues and send the retrieved commands to the processing core for execution.

51. With respect to exemplary devices, the NVIDIA GeForce RTX 4060 Ti graphics card, made or sold by NVIDIA, incorporating the NVIDIA GeForce RTX 4060 Ti graphics processing unit (GPU) and associated software, made or sold by NVIDIA, directly infringes at least claim 11 of the '943 Patent.

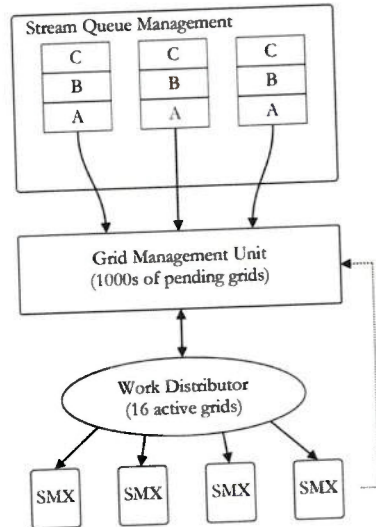
52. Upon information and belief, the GeForce RTX 4060 Ti GPU includes a set of queues, each queue of the set of queues being configured to hold commands received from a central processing unit (CPU), wherein each command is sent from a virtual device in the CPU. For example, the GeForce RTX 4060 Ti GPU includes the "Grid Management Unit," "Compute Front End," "Task Management Unit," and/or "Work Distribution Unit," which, on information and

belief, include a plurality of queues configured to hold commands sent from a virtual device in a CPU.

Grid Management Unit

Kepler GK110 introduces the Grid Management Unit, which creates multiple hardware work queues to reduce or eliminate false dependencies. With the GMU, streams can be kept as individual pipelines of work.

Also shown on the diagram is the feedback path from the SMXs to the Work Distributor, and the work creation path from the SMXs to the GMU. These components provide dynamic parallelism (see the CUDA Programming Guide for more information on dynamic parallelism).



https://developer.download.nvidia.com/compute/DevZone/C/html_x64/6_Advanced/simpleHyperQ/doc/HyperQ.pdf (accessed 04/14/2025), at 3.

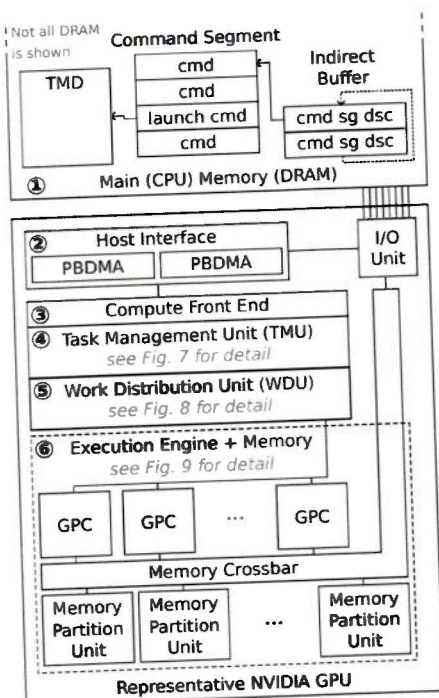


Fig. 6. High-level GPU hardware compute scheduling pipeline.

<https://www.cs.unc.edu/~jbakita/rtas23.pdf> (accessed 04/14/2025), at 6.

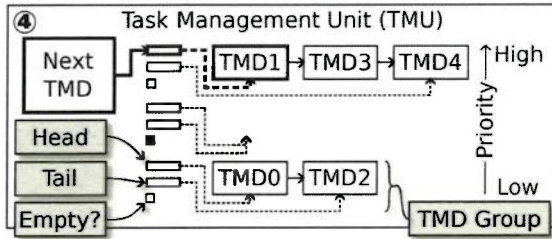


Fig. 7. A three-priority task management unit with five queued TMDs.

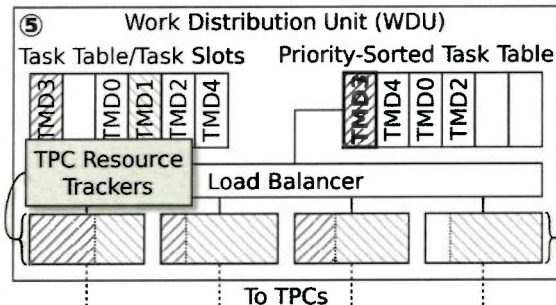


Fig. 8. A six-slot work distribution unit on a four-TPC GPU.

Id. at 7; see also *id.* at 7-8.

Hyper-Q enables multiple CPU threads or processes to launch work on a single GPU simultaneously, thereby dramatically increasing GPU utilization and significantly reducing CPU idle times. Applications that previously encountered false serialization across tasks, thereby limiting achieved GPU utilization, can see up to dramatic performance increase without changing any existing code.

https://developer.download.nvidia.com/compute/DevZone/C/html_x64/6_Advanced/simpleHyperQ/doc/HyperQ.pdf (accessed 04/14/2025), at 1.

53. Further, upon information and belief, the GeForce RTX 4060 Ti GPU includes a command processor configured to retrieve the received commands from the set of queues, wherein the set of queues includes a high priority queue that holds high priority commands, wherein the command processor is configured to retrieve a high priority command held in the high priority queue before retrieving commands held in other queues of the set of queues, and wherein the processing device is configured to execute the received commands. For example, upon information and belief, the GeForce RTX 4060 Ti GPU includes the “Grid Management Unit,” “Compute Front End,” “Task Management Unit,” and/or “Work Distribution Unit,” which, upon information and belief, configured to retrieve the received commands from the set of queues.

https://developer.download.nvidia.com/compute/DevZone/C/html_x64/6_Advanced/simpleHyperQ/doc/HyperQ.pdf (accessed 04/14/2025); <https://www.cs.unc.edu/~jbakita/rtas23.pdf> (accessed 04/14/2025). Upon information and belief, the set of queues includes a high priority queue that

holds high priority commands and the command processor is configured to retrieve a high priority command held in the high priority queue before retrieving commands held in other queues of the set of queues. *Id.*

③ **Compute Front End.** The Compute Front End relays TMD pointers from ②, the Host Interface to ④, the Task Management Unit. Further, we understand that this unit can orchestrate context switches in subsequent units.¹⁴ While this unit processes TMDs at a rate decoupled from the others, it may cause work to queue if the incorrect context is active. We focus on a single context in this work, and so do not further consider said issue. Further details on this unit are elusive.¹⁵

<https://www.cs.unc.edu/~jbakita/rtas23.pdf> (accessed 04/14/2025), at 7.

④ **Task Management Unit (TMU).**¹⁶ Illustrated in Fig. 7, the Task Management Unit queues TMDs by priority and arrival order until ⑤, the Work Distribution Unit, is ready to receive them. Due to the explicit scheduling responsibilities of the TMU, we investigate it in depth.

The TMU is built around a series of priority-level, singly-linked lists, with one linked-list-head and -tail pointer tracked for each priority level. Each list is exclusively composed of TMDs. In Fig. 7, we show a unit supporting three priority levels, with TMDs illustratively numbered by arrival order. Three TMDs are in the highest-priority-level list, none are in the medium-priority-level list, and two TMDs are in the lowest-priority-level list. Each priority-level list is formally called a *TMD Group*.

These lists allow the TMU to reorder TMDs that it receives, such that higher-priority ones skip ahead of lower-priority ones. Upon TMD arrival from the Compute Front End, or elsewhere,¹⁷ the TMU reads the TMD's `GROUP_ID` field, and appends the TMD to the tail of the specified TMD Group.¹⁸ For example, in Fig. 7, TMD4 is the most-recently arrived high-priority TMD, and TMD2 is the most-recently arrived low-priority TMD.

As unit ⑤ in the scheduling pipeline signals readiness for another TMD, the TMU removes and passes the head of the highest-priority non-empty list. This is illustrated with the bold outlines and “Next TMD” box in Fig. 7. Given the TMU state of Fig. 7, TMDs would exit the TMU in the following order: TMD1, TMD3, TMD4, TMD0, and then TMD2.

The TMU is the final unit which can receive TMDs at a rate decoupled from the TMD completion rate. This causes TMDs to eventually accumulate in the TMU whenever kernels are dispatched by user space at a rate faster than they can complete.

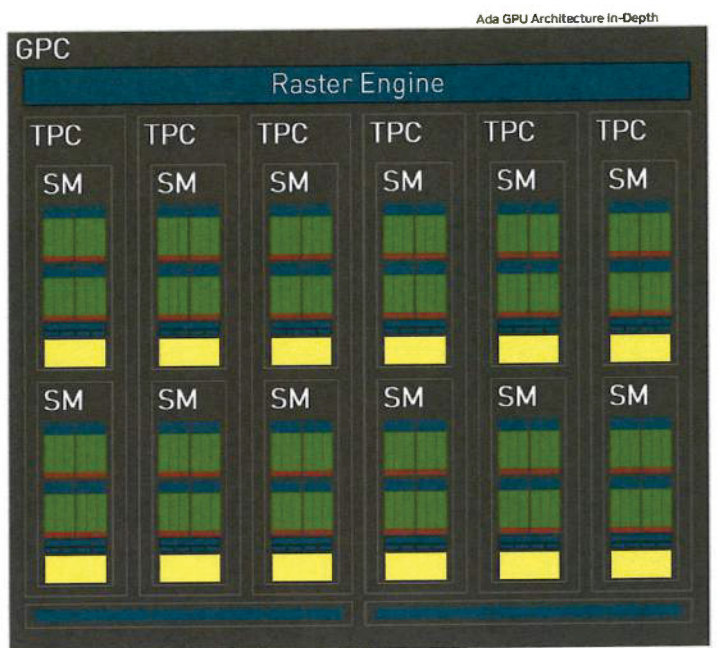
Id.

⑤ **Work Distribution Unit (WDU).**¹⁹ Illustrated in Fig. 8 the Work Distribution Unit dispatches TMDs from *task slots* to available TPCs. The number of task slots is hardware limited,²⁰ forcing the WDU to make scheduling decisions with incomplete information, eg. when the number of ready kernels exceeds the number of task slots.

The WDU signals unit ④, the TMU, for a new TMD whenever a task slot becomes available. It then inserts the TMD into a task slot in the Task Table, and inserts a reference to the TMD in the Priority-Sorted Task Table. The priority-sorted table is ordered first by priority, then by arrival time. The Load Balancer dispatches thread blocks from the TMD at the head of the priority-sorted table. Once all of a TMD’s blocks are launched, it is removed from the priority-sorted table, but left in the Task Table until all its blocks complete.

Id. at 7-8.

54. In addition, upon information and belief, the GeForce RTX 4060 Ti GPU includes a processing core configured to execute the received command, wherein the command processor is configured to retrieve commands from the set of queues and send the retrieved commands to the processing core for execution. For example, on information and belief, the GeForce RTX 4060 Ti GPU includes “Graphics Processing Clusters (GPCs),” “Texture Processing Clusters (TPCs),” and “Streaming Multiprocessors (SMs)” and can “concurrently run compute and 3D” tasks.



Ada GPC with Raster Engine, 6 TPCs, 12 SMs, and 16 ROPs (8 per ROP partition).

Figure 2. Ada GPC Block Diagram

The GPC is the dominant high-level hardware block within all AD10x Ada family GPUs, with all of the key graphics processing units residing within a GPC. Each GPC includes a dedicated Raster Engine, two Raster Operations (ROPs) partitions, with each partition containing eight individual ROP units, and six TPCs. Each TPC includes one PolyMorph Engine and two SMs.

Each SM in AD10x GPUs contain 128 CUDA Cores, one Ada Third-Generation RT Core, four Ada Fourth-Generation Tensor Cores, four Texture Units, a 256 KB Register File, and 128 KB of L1/Shared Memory, which can be configured for different memory sizes depending on the needs of the graphics or compute workload.

<https://images.nvidia.com/aem-dam/Solutions/Data-Center/l4/nvidia-ada-gpu-architecture-whitepaper-v2.1.pdf> (accessed 04/14/2025), at 8.

55. The GPCs and SMs are, upon information and belief, configured to execute the received command, wherein the command processor is configured to retrieve commands from the set of queues and send the retrieved commands to the processing core for execution. *Id.*; https://developer.download.nvidia.com/compute/DevZone/C/html_x64/6_Advanced/simpleHyperQ/doc/HyperQ.pdf (accessed 04/14/2025); <https://www.cs.unc.edu/~jbakita/rtas23.pdf> (accessed 04/14/2025).

56. NVIDIA has indirectly infringed, and continues to indirectly infringe, the '943 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the '943 Patent by others,

such as its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties. These others directly infringe the '943 Patent by, for example, making, using, testing, selling, offering for sale throughout, or importing into the United States, including this judicial District, downstream products (for example, without limitation, personal computers), incorporating the products that infringe (for example, without limitation, the Exemplary NVIDIA GeForce RTX 4060 Ti graphics card). NVIDIA has induced and continues to induce such direct infringement by encouraging and facilitating others' infringing acts. For example, NVIDIA hires permanent sales and/or marketing personnel located throughout the United States, and in this judicial District. On information and belief, these sales and/or marketing personnel are engaged in activities that are targeted to original equipment manufacturers, customers, resellers, retailers, and other parties, including original equipment manufacturers, customers, resellers, retailers, and other parties based on the United States, including instructions for making, using, testing, or selling the products that infringe.

57. NVIDIA specifically intended or intends these others, such as its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties, to infringe the '943 Patent and knew or knows that the induced acts of these others constitute direct infringement. For example, although NVIDIA has notice of the '943 Patent, the scope of the claims, and the products covered thereby, NVIDIA nonetheless intentionally and knowingly encourages and facilitates these others to make, use, test, sell, offer for sale in, or import into, such covered products into the United States and thereby directly infringe the '943 Patent. Additionally, for example, NVIDIA designed the products that infringe (for example, without limitation, the Exemplary NVIDIA GeForce RTX 4060 Ti graphics card) such that they would each infringe the '943 Patent if made, used, tested, sold, offered for sale, or imported into the United States.

NVIDIA provided, directly, or indirectly, the products that infringe to others, such as, but not limited to, its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties, knowing and intending that those others use, test, sell, offer for sale, and/or import in and into the United States downstream products (for example, without limitation, personal computers) that include the infringing products (for example, without limitation, the Exemplary NVIDIA GeForce RTX 4060 Ti graphics card), thereby directly infringing one or more claims of the '943 Patent.

58. NVIDIA has also indirectly infringed and continues to indirectly infringe the '943 Patent under 35 U.S.C. § 271(c) by materially contributing to infringement of the '943 Patent by its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties by selling the products that infringe to such third parties for use in downstream end-user products in a manner that infringes the '943 Patent. The products that infringe (for example, without limitation, the Exemplary NVIDIA GeForce RTX 4060 Ti graphics card) have no substantial non-infringing uses and are a material part of the invention. Moreover, the products that infringe provide vital functionality to such downstream products (for example, without limitation, personal computers). Further, the products that infringe cannot be used without being incorporated into a downstream product. Thus, any manufacture, use, test, sale, offer for sale, or importation in or into the United States of a product that infringes or a downstream product incorporating a product that infringes, infringes the '943 Patent.

59. As a result of NVIDIA's infringement of the '943 Patent, Onesta is entitled to monetary damages in an amount adequate to compensate for NVIDIA's infringement, but in no event less than a reasonable royalty for the use made of the invention by Defendant, together with interest and costs as fixed by the Court.

60. Despite NVIDIA's actual notice of its infringement as of at least the filing of the Complaint, NVIDIA's acts of direct and indirect infringement of the '943 Patent are deliberate and willful, and have caused, and will continue to cause, substantial damage and irreparable harm to Onesta, and Onesta has no adequate remedy at law.

COUNT III

Infringement of the '019 Patent

61. Onesta incorporates the allegations of all of the foregoing paragraphs as if fully restated herein.

62. NVIDIA has directly and indirectly infringed, and continues to directly and indirectly infringe, the '019 Patent by making, having made, using, selling, offering for sale, and/or importing into the United States products that infringe the '019 Patent. The products that infringe one or more claims of the '019 Patent include, but are not limited to, at least the products identified herein. Further discovery may reveal additional infringing products and/or models.

63. For example, and without limitation, the infringing products infringe one or more claims of the '019 Patent, including but not limited to claim 11. The products that infringe fall within the scope of and include, either literally or under the doctrine of equivalents, all of the elements of at least claim 11 of the '019 Patent.

64. The products that infringe include all of the limitations of at least claim 11 of the '019 Patent. Specifically, the '019 Patent claims, *e.g.*, a computer system comprising: a plurality of processors; a shared memory; and a mapper, wherein the mapper is configured to: receive a memory operation from a processor that references an address in the shared memory; map the received memory operation to at least one virtual memory pool to produce a mapping result, wherein the at least one virtual memory pool is associated with at least one memory resource; and provide the mapping result to the processor.

65. With respect to exemplary devices, the NVIDIA Blackwell GB200 Superchip, made or sold by NVIDIA, incorporating the NVIDIA Blackwell B200 graphics processing unit (GPU), NVIDIA Grace central processing unit (CPU), and associated software, made or sold by NVIDIA, directly infringes at least claim 11 of the '019 Patent.

66. The NVIDIA Blackwell GB200 Superchip implements the Blackwell architecture and “introduces groundbreaking advancements for generative AI and accelerated computing.”

Breaking Barriers in Accelerated Computing

The NVIDIA Blackwell architecture introduces groundbreaking advancements for generative AI and accelerated computing. The incorporation of the second-generation Transformer Engine, alongside the faster and wider NVIDIA NVLink™ interconnect, propels the data center into a new era, with orders of magnitude more performance compared to the previous architecture generation. Further advances in NVIDIA Confidential Computing technology raise the level of security for real-time LLM inference at scale without performance compromise. And Blackwell's new decompression engine combined with Spark RAPIDS™ libraries deliver unparalleled database performance to fuel data analytics applications. Blackwell's multiple advancements build upon generations of accelerated computing technologies to define the next chapter of generative AI with unparalleled performance, efficiency, and scale.

<https://resources.nvidia.com/en-us-blackwell-architecture/datasheet> (accessed 04/14/2025).

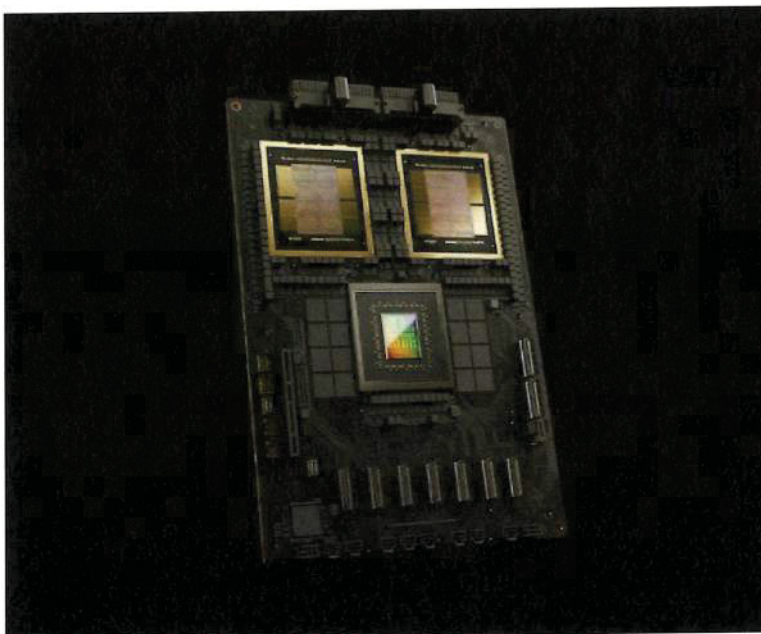


Figure 1. NVIDIA GB200 Superchip Incl. Two Blackwell GPUs and One Grace CPU

https://cdn.prod.website-files.com/61dda201f29b7efc52c5fbaf/6602ea9d0ce8cb73fb6de87f_nvidia-blackwell-architecture-technical-brief.pdf (accessed 04/14/2025), at 5.

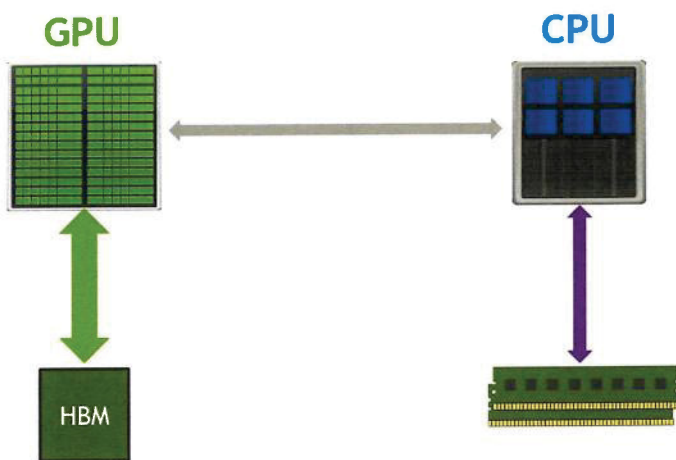
67. For example, upon information and belief, the NVIDIA Blackwell GB200 Superchip includes a plurality of processors. For example, the NVIDIA Blackwell GB200 Superchip “connects two NVIDIA B200 Tensor Core GPUs to the NVIDIA Grace CPU.”

The NVIDIA GB200 Grace Blackwell Superchip connects two NVIDIA B200 Tensor Core GPUs to the NVIDIA Grace CPU over a 900GB/s ultra-low-power NVLink chip-to-chip interconnect.

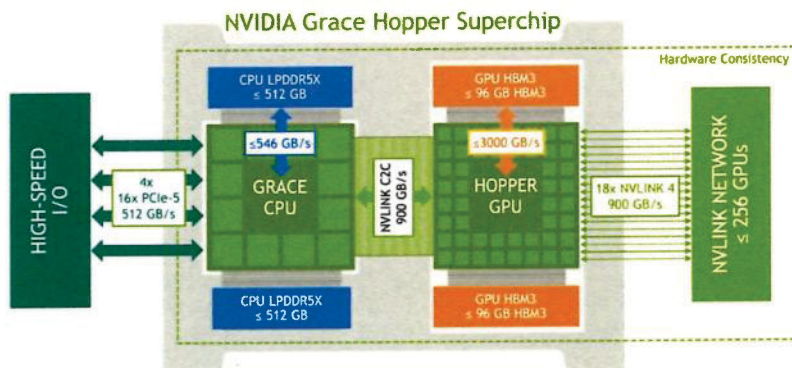
<https://nvidianews.nvidia.com/news/nvidia-blackwell-platform-arrives-to-power-a-new-era-of-computing> (accessed 04/14/2025); *see also* https://cdn.prod.website-files.com/61dda201f29b7efc52c5fbaf/6602ea9d0ce8cb73fb6de87f_nvidia-blackwell-architecture-technical-brief.pdf (accessed 04/14/2025); <https://developer.nvidia.com/blog/nvidia-gb200-nvl72-delivers-trillion-parameter-llm-training-and-real-time-inference/> (accessed 04/14/

2025); <https://developer.nvidia.com/blog/nvidia-grace-hopper-superchip-architecture-in-depth/> (accessed 04/14/2025).

68. Upon information and belief, the NVIDIA Blackwell GB200 Superchip also includes a shared memory. For example, the NVIDIA Blackwell GB200 Superchip includes a CPU LPDDR5X and a GPU HBM3 accessible by both the CPU and the GPU.



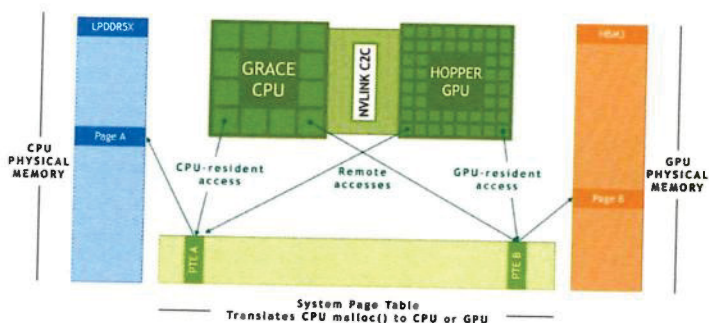
<https://developer.nvidia.com/blog/improving-gpu-memory-oversubscription-performance/> (accessed 04/14/2025).



<https://developer.nvidia.com/blog/nvidia-grace-hopper-superchip-architecture-in-depth/> (accessed 04/14/2025).

69. The NVIDIA Blackwell GB200 Superchip also includes a unified memory

accessible from any processor in the system.



Id.

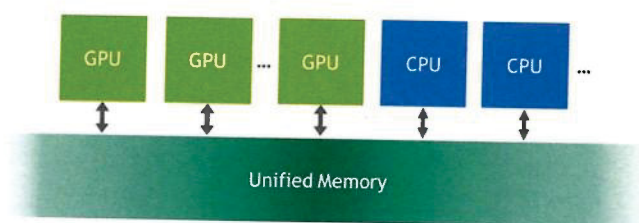
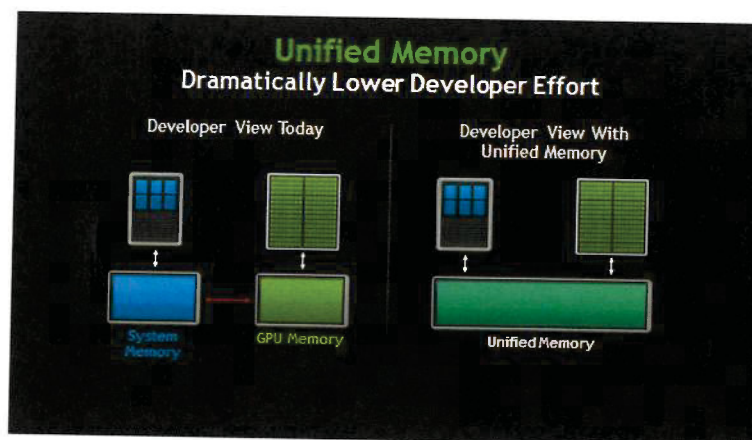


Figure 1. Unified Memory is a single memory address space accessible from any processor in a system

<https://developer.nvidia.com/blog/unified-memory-cuda-beginners/> (accessed 04/14/2025).



Unified Memory creates a pool of managed memory that is shared between the CPU and GPU, bridging the CPU-GPU divide. Managed memory is accessible to both the CPU and GPU using a single pointer. The key is that the system automatically *migrates* data allocated in Unified Memory between host and device so that it looks like CPU memory to code running on the CPU, and like GPU memory to code running on the GPU.

<https://developer.nvidia.com/blog/unified-memory-in-cuda-6/> (accessed 04/14/2025). See also

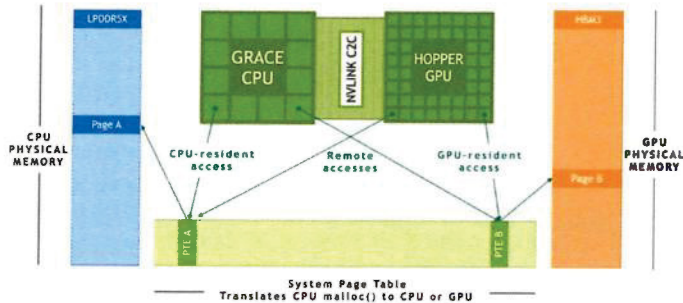
<https://developer.nvidia.com/blog/improving-gpu-memory-oversubscription-performance/>

(accessed 04/14/2025); <https://developer.nvidia.com/blog/nvidia-gb200-nvl72-delivers-trillion-parameter-llm-training-and-real-time-inference/> (accessed 04/14/2025); <https://developer.nvidia.com/blog/maximizing-unified-memory-performance-cuda/> (accessed 04/14/2025).

70. In addition, on information and belief, the NVIDIA Blackwell GB200 Superchip includes a mapper configured to receive a memory operation from a processor that references an address in the shared memory; map the received memory operation to at least one virtual memory pool to produce a mapping result, wherein the at least one virtual memory pool is associated with at least one memory resource; and provide the mapping result to the processor. For example, the NVIDIA Blackwell GB200 Superchip includes memory controller(s), memory management unit(s), and associated software that, upon information and belief, is configured to receive a memory operation from a processor that references an address in the shared memory; map the received memory operation to at least one virtual memory pool to produce a mapping result, wherein the at least one virtual memory pool is associated with at least one memory resource; and provide the mapping result to the processor.

Much like CPUs, GPUs have multiple levels of TLBs (Translation Lookaside Buffer: a memory cache that stores recent virtual to physical memory address translations) to perform address translations. When Pascal and Volta GPUs access a page that is not resident in the local GPU memory the translation for this page generates a fault message and locks the TLBs for the corresponding SM (on Tesla P100 it locks a pair of SMs that share a single TLB). This means any outstanding translations can proceed but any new translations will be stalled until all faults are resolved. This is necessary to make sure the SM's view of memory is consistent since during page fault processing the driver may modify the page table and add or revoke access to pages. The GPU can generate many faults concurrently and it's possible to get multiple fault messages for the same page. The Unified Memory driver processes these faults, remove duplicates, updates mappings and transfers the data. This fault handling adds significant overhead to streaming performance of Unified Memory on current generation GPU architectures.

<https://developer.nvidia.com/blog/maximizing-unified-memory-performance-cuda/> (accessed 04/14/2025).



<https://developer.nvidia.com/blog/nvidia-grace-hopper-superchip-architecture-in-depth/>

(accessed 04/14/2025).

__host__ cudaError_t cudaMallocManaged (void **devPtr, size_t size, unsigned int flags)

Allocates memory that will be automatically managed by the Unified Memory system.

Parameters

devPtr

- Pointer to allocated device memory

size

- Requested allocation size in bytes

flags

- Must be either `cudaMemAttachGlobal` or `cudaMemAttachHost` (defaults to `cudaMemAttachGlobal`)

Returns

`cudaSuccess`, `cudaErrorMemoryAllocation`, `cudaErrorNotSupported`, `cudaErrorInvalidValue`

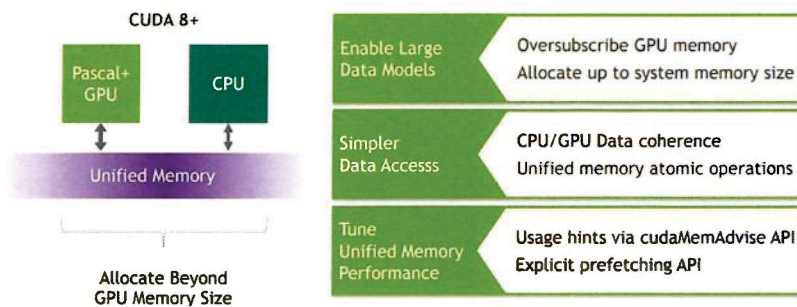
Description

Allocates `size` bytes of managed memory on the device and returns in `*devPtr` a pointer to the allocated memory. If the device doesn't support allocating managed memory, `cudaErrorNotSupported` is returned. Support for managed memory can be queried using the device attribute `cudaDevAttrManagedMemory`. The allocated memory is suitably aligned for any kind of variable. The memory is not cleared. If `size` is 0, `cudaMallocManaged` returns `cudaErrorInvalidValue`. The pointer is valid on the CPU and on all GPUs in the system that support managed memory. All accesses to this pointer must obey the Unified Memory programming model.

https://docs.nvidia.com/cuda/pdf/CUDA_Runtime_API.pdf (accessed 04/14/2025), at 446.

CUDA 8+: UNIFIED MEMORY

Demand Paging For Pascal and Beyond



https://www.olcf.ornl.gov/wp-content/uploads/2019/06/06_Managed_Memory.pdf (accessed

04/14/2025), at 6. See also

https://developer.ridgerun.com/wiki/index.php/NVIDIA_CUDA_Memory_Management

(accessed 04/14/2025); <https://developer.nvidia.com/blog/unified-memory-in-cuda-6/> (accessed

04/14/2025); <https://developer.nvidia.com/blog/unified-memory-cuda-beginners/> (accessed

04/14/2025); [https://developer.nvidia.com/blog/introducing-low-level-gpu-virtual-memory-](https://developer.nvidia.com/blog/introducing-low-level-gpu-virtual-memory-management/)

[management/](https://developer.nvidia.com/blog/introducing-low-level-gpu-virtual-memory-management/) (accessed 04/14/2025).

71. NVIDIA has indirectly infringed, and continues to indirectly infringe, the '019 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the '019 Patent by others, such as its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties. These others directly infringe the '019 Patent by, for example, making, using, selling, offering for sale throughout, or importing into, the United States, including this judicial District, downstream products (for example, without limitation, servers) incorporating the products that infringe (for example, without limitation, the Exemplary NVIDIA Blackwell GB200 Superchip). NVIDIA induced and induces such direct infringement by encouraging and facilitating others' infringing acts. For example, NVIDIA hires permanent sales and/or marketing personnel located throughout the United States, and in this judicial District. On information and belief, these

sales and/or marketing personnel are engaged activities that are targeted to original equipment manufacturers, customers, resellers, retailers, and other parties, including original equipment manufacturers, customers, resellers, retailers, and other parties based on the United States, including instructions for making, using, or selling the products that infringe.

72. NVIDIA specifically intended or intends these others, such as its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties, to infringe the '019 Patent and knew or knows that the induced acts of these others constitute direct infringement. For example, although NVIDIA has notice of the '019 Patent, the scope of its claims, and the products covered thereby, NVIDIA nonetheless intentionally and knowingly encourages and facilitates these others to make, use, test, sell, offer for sale in, or import into, such covered products into the United States and thereby directly infringe the '019 Patent. Additionally, for example, NVIDIA designed the products that infringe (for example, without limitation, the Exemplary NVIDIA Blackwell GB200 Superchip) such that they would each infringe the '019 Patent if made, used, tested, sold, offered for sale, or imported into the United States. NVIDIA provided, directly, or indirectly, the products that infringe to others, such as, but not limited to, customers, knowing and intending that those others use, test, sell, offer for sale, and/or import in and into the United States downstream products (for example, without limitation, servers) that include the products that infringe (for example, without limitation, the Exemplary NVIDIA Blackwell GB200 Superchip), thereby directly infringing one or more claims of the '019 Patent.

73. NVIDIA has also indirectly infringed and continues to indirectly infringe the '019 Patent under 35 U.S.C. § 271(c) by materially contributing to infringement of the '019 Patent by its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties by selling the products that infringe to such third parties for use in downstream end-

user products in a manner that infringes the '019 Patent. The products that infringe (for example, without limitation, the Exemplary NVIDIA Blackwell GB200 Superchip) have no substantial non-infringing uses and are a material part of the invention. Moreover, the products that infringe provide vital functionality to such downstream products (for example, without limitation, servers). Further, the products that infringe cannot be used without being incorporated into a downstream product. Thus, any manufacture, use, test, sale, offer for sale, or importation in or into the United States of a product that infringes or a downstream product incorporating a product that infringes, infringes the '019 Patent.

74. As a result of NVIDIA's infringement of the '019 Patent, Onesta is entitled to monetary damages in an amount adequate to compensate for NVIDIA's infringement, but in no event less than a reasonable royalty for the use made of the invention by NVIDIA, together with interest and costs as fixed by the Court.

75. Despite NVIDIA's actual notice of its infringement as of at least the filing of the Complaint, NVIDIA's acts of direct and indirect infringement of the '019 Patent are deliberate and willful, and have caused, and will continue to cause, substantial damage and irreparable harm to Onesta, and Onesta has no adequate remedy at law.

COUNT IV

Infringement of the '803 Patent

76. Onesta incorporates the allegations of all of the foregoing paragraphs as if fully restated herein.

77. NVIDIA has directly and indirectly infringed, and continues to directly and indirectly infringe, the '803 Patent by making, having made, using, selling, offering for sale, and/or importing into the United States products that infringe the '803 Patent. The products that infringe one or more claims of the '803 Patent include, but are not limited to, at least the products identified

herein. Further discovery may reveal additional infringing products and/or models.

78. For example, and without limitation, the infringing products infringe one or more claims of the '803 Patent, including but not limited to claim 1. The products that infringe fall within the scope of and include, either literally or under the doctrine of equivalents, all of the elements of at least claim 1 of the '803 Patent.

79. The products that infringe include all of the limitations of at least claim 1 of the '803 Patent. Specifically, the '803 Patent claims, *e.g.*, a computer system comprising: a plurality of processors; a shared memory; and a mapper, wherein the mapper is configured to: receive a memory operation from a processor that references an address in the shared memory; map the received memory operation to at least one virtual memory pool to produce a mapping result, wherein the at least one virtual memory pool is associated with at least one memory resource; and provide the mapping result to the processor.

80. With respect to exemplary devices, the NVIDIA Blackwell GB200 Superchip, made or sold by NVIDIA, incorporating the NVIDIA Blackwell B200 graphics processing unit (GPU) and NVIDIA Grace central processing unit (CPU), made or sold by NVIDIA, directly infringes at least claim 1 of the '803 Patent.

81. The NVIDIA Blackwell GB200 Superchip implements the Blackwell architecture and “introduces groundbreaking advancements for generative AI and accelerated computing.”

Breaking Barriers in Accelerated Computing

The **NVIDIA Blackwell architecture** introduces groundbreaking advancements for generative AI and accelerated computing. The incorporation of the second-generation Transformer Engine, alongside the faster and wider **NVIDIA NVLink™** interconnect, propels the data center into a new era, with orders of magnitude more performance compared to the previous architecture generation. Further advances in **NVIDIA Confidential Computing** technology raise the level of security for real-time LLM inference at scale without performance compromise. And Blackwell's new decompression engine combined with Spark RAPIDS™ libraries deliver unparalleled database performance to fuel data analytics applications. Blackwell's multiple advancements build upon generations of accelerated computing technologies to define the next chapter of generative AI with unparalleled performance, efficiency, and scale.

<https://resources.nvidia.com/en-us-blackwell-architecture/datasheet> (accessed 04/14/2025).

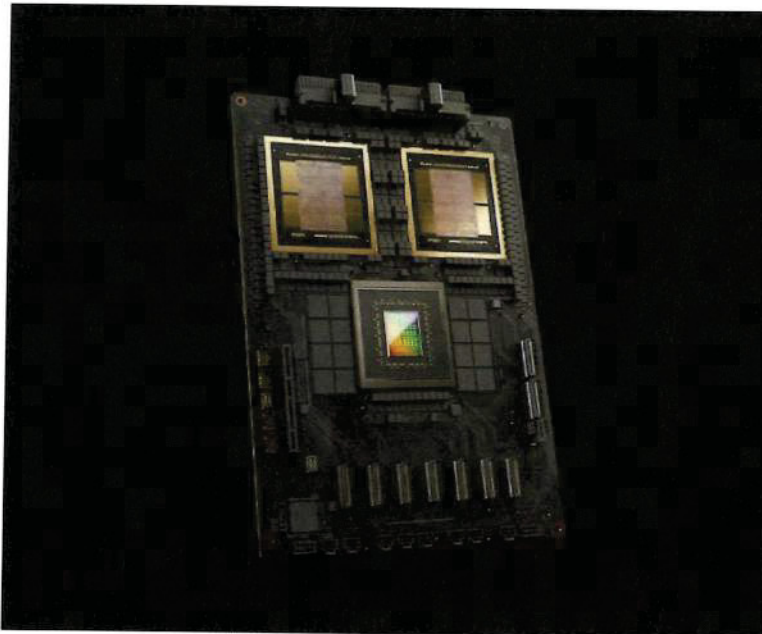


Figure 1. NVIDIA GB200 Superchip Incl. Two Blackwell GPUs and One Grace CPU

https://cdn.prod.website-files.com/61dda201f29b7efc52c5fbaf/6602ea9d0ce8cb73fb6de87f_nvidia-blackwell-architecture-technical-brief.pdf (accessed 04/14/2025), at 5.

82. For example, upon information and belief, the NVIDIA Blackwell GB200

Superchip includes a central processing unit (CPU) communicably coupled to a first graphics processing unit (GPU) chiplet of a GPU chiplet array, wherein the GPU chiplet array includes: the first GPU chiplet communicably coupled to the CPU via a bus. For example, the NVIDIA Blackwell GB200 Superchip includes a NVIDIA Grace CPU coupled to the NVIDIA B200 GPU via a bus.

The NVIDIA GB200 Grace Blackwell Superchip connects two NVIDIA B200 Tensor Core GPUs to the NVIDIA Grace CPU over a 900GB/s ultra-low-power NVLink chip-to-chip interconnect.

<https://nvidianews.nvidia.com/news/nvidia-blackwell-platform-arrives-to-power-a-new-era-of-computing> (accessed 04/14/2025).

NVIDIA® NVLink®-C2C extends the industry-leading NVLink technology to a chip-to-chip interconnect. This enables the creation of a new class of integrated products with NVIDIA partners, built via chiplets, allowing NVIDIA GPUs, DPUs, and CPUs to be coherently interconnected with custom silicon.

<https://www.nvidia.com/en-us/data-center/nvlink-c2c/> (accessed 04/14/2025).

	GB200
Type	Grace Blackwell Superchip
Interconnects	2x NVLink 5 (1800GB/sec) 2x PCIe 6.0 (256GB/sec)
GPU	2x "Blackwell GPU"
GPU Transistor Count	416B (2x2x104B)
TDP	2700W
Manufacturing Process	TSMC 4NP
Interface	Superchip
Architecture	Grace + Blackwell

<https://www.anandtech.com/show/21310/nvidia-blackwell-architecture-and-b200b100-accelerators-announced-going-bigger-with-smaller-data> (accessed 04/14/2025).

83. In addition, upon information and belief, the NVIDIA Blackwell GB200 Superchip includes a second GPU chiplet communicably coupled to the first GPU chiplet via a passive crosslink, wherein the passive crosslink is dedicated for inter-chiplet communications. For example, the NVIDIA B200 GPU includes two dies.

This architecture is able to incorporate a significant amount of computing power by merging two dies into a single, unified GPU. Each of the two dies are the largest die possible within the limits of reticle size, as big as can possibly be built today. The two dies are connected and unified with a single 10 terabyte-per-second (TB/s) chip-to-chip NVIDIA High-Bandwidth Interface (NV-HBI), providing one fully coherent, unified GPU.

<https://nvidianews.nvidia.com/news/nvidia-blackwell-platform-arrives-to-power-a-new-era-of-computing> (accessed 04/14/2025).

The B200 is composed of about 1600 square millimeters of processor on two silicon dies that are linked in the same package by a 10 terabyte per second connection, so they perform as if they were a single 208-billion-transistor chip.

<https://spectrum.ieee.org/nvidia-blackwell#:~:text=The%20B200%20is%20composed%20of,208%2Dbillion%2Dtransistor%20c>
[hip](https://resources.nvidia.com/en-us-blackwell-architecture) (accessed 04/14/2025); *see also* <https://resources.nvidia.com/en-us-blackwell-architecture>;
<https://spectrum.ieee.org/nvidia-blackwell> (accessed 04/14/2025).

84. Upon information and belief, the two dies are coupled via a passive crosslink, wherein the passive crosslink is dedicated for inter-chiplet communications. *Id.*

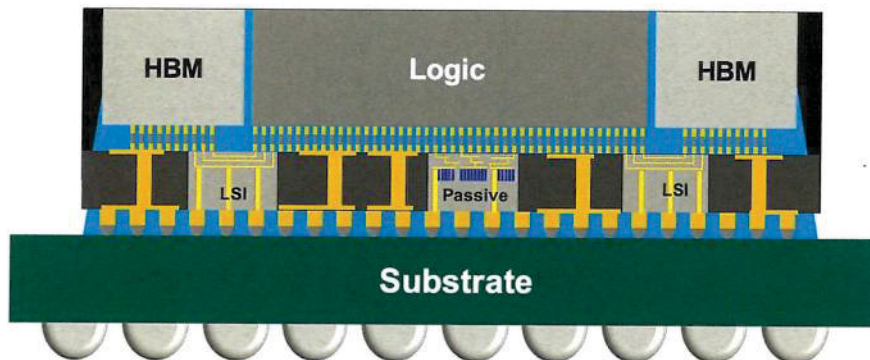
B200 GPUs are the first processors to use TSMC's CoWoS-L packaging, which connects chiplets using a redistribution layer (RDL) interposer with local silicon bridges.

<https://www.3dincites.com/2024/10/iftle-607-why-nvidias-blackwell-is-having-issues-with-tsmc-cowos-l-technology> (accessed 04/14/2025).

CoWoS[®]-L is one of the chip-last packages on the CoWoS[®] (Chip on Wafer on Substrate) platform. It combines the merits of CoWoS[®]-S and InFO (Integrated Fan-Out) technologies to provide the most flexible integration using an interposer with a Local Silicon Interconnect (LSI) chip for die-to-die interconnect and RDL layers for power and signal delivery.

Key CoWoS[®]-L features include:

1. LSI chips for a high routing density die-to-die interconnect through multiple layers of sub-micron copper lines. LSI chips can feature a variety of connection architectures, e.g., System on Chip (SoC)-to-SoC, SoC-to-chiplet, SoC-to-High Bandwidth Memory, within each product, and can be used repeatedly in multiple products. The corresponding metal types, layer counts, and pitches align with the offering from CoWoS[®]-S.



<https://3dfabric.tsmc.com/english/dedicatedFoundry/technology/cowos.htm>

(accessed

04/14/2025).

85. NVIDIA has indirectly infringed, and continues to indirectly infringe, the '803 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the '803 Patent by others, such as its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties. These others directly infringe the '803 Patent by, for example, making, using, testing, selling, offering for sale throughout, or importing into, the United States, including this judicial District, downstream products (for example, without limitation, personal computers) incorporating the products that infringe (for example, without limitation, the Exemplary NVIDIA

Blackwell GB200 Superchip). NVIDIA induced and induces such direct infringement by encouraging and facilitating others' infringing acts. For example, NVIDIA hires permanent sales and/or marketing personnel located throughout the United States, and in this judicial District. On information and belief, these sales and/or marketing personnel are engaged in activities that are targeted to original equipment manufacturers, customers, resellers, retailers, and other parties, including original equipment manufacturers, customers, resellers, retailers, and other parties based on the United States, including instructions for making, using, testing, or selling the products that infringe.

86. NVIDIA specifically intended or intends these others, such as its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties, to infringe the '803 Patent and knew or knows that the induced acts of these others constituted direct infringement. For example, although NVIDIA has notice of the '803 Patent, the scope of its claims, and the products covered thereby, NVIDIA nonetheless intentionally and knowingly encourages and facilitates these others to make, use, test, sell, offer for sale in, or import into, such covered products into the United States and thereby directly infringe the '803 Patent. Additionally, for example, NVIDIA designed the products that infringe (for example, without limitation, the Exemplary NVIDIA Blackwell GB200 Superchip) such that they would each infringe the '803 Patent if made, used, tested, sold, offered for sale, or imported into the United States. NVIDIA provided, directly, or indirectly, the products that infringe to others, such as, but not limited to, its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties, knowing and intending that those others use, sell, offer for sale, and/or import in and into the United States downstream products (for example, without limitation, personal computers) that include the products that infringe (for example, without limitation, the Exemplary NVIDIA

Blackwell GB200 Superchip), thereby directly infringing one or more claims of the '803 Patent.

87. NVIDIA has also indirectly infringed and continues to indirectly infringe the '803 Patent under 35 U.S.C. § 271(c) by materially contributing to infringement of the '803 Patent by its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties by selling the products that infringe to such third parties for use in downstream end-user products in a manner that infringes the '803 Patent. The products that infringe (for example, without limitation, the Exemplary NVIDIA Blackwell GB200 Superchip) have no substantial non-infringing uses and are a material part of the invention. Moreover, the products that infringe provide vital functionality to such downstream products (for example, without limitation, personal computers). Further, the products that infringe cannot be used without being incorporated into a downstream product. Thus, any manufacture, use, test, sale, offer for sale, or importation in or into the United States of a product that infringes or a downstream product incorporating a product that infringes, infringes the '803 Patent.

88. As a result of NVIDIA's infringement of the '803 Patent, Onesta is entitled to monetary damages in an amount adequate to compensate for NVIDIA's infringement, but in no event less than a reasonable royalty for the use made of the invention by NVIDIA, together with interest and costs as fixed by the Court.

89. Despite NVIDIA's actual notice of its infringement as of at least the filing of the Complaint, NVIDIA's acts of direct and indirect infringement of the '803 Patent are deliberate and willful, and have caused, and will continue to cause, substantial damage and irreparable harm to Onesta, and Onesta has no adequate remedy at law.

COUNT V

Infringement of the '809 Patent

90. Onesta incorporates the allegations of all of the foregoing paragraphs as if fully

restated herein.

91. NVIDIA has directly and indirectly infringed, and continues to directly and indirectly infringe, the '809 Patent by making, having made, using, selling, offering for sale, and/or importing into the United States products that infringe the '809 Patent. The products that infringe one or more claims of the '809 Patent include, but are not limited to, at least the products identified herein. Further discovery may reveal additional infringing products and/or models.

92. For example, and without limitation, the infringing products infringe one or more claims of the '809 Patent, including but not limited to claim 1. The products that infringe fall within the scope of and include, either literally or under the doctrine of equivalents, all of the elements of at least claim 1 of the '809 Patent.

93. The products that infringe include all of the limitations of at least claim 1 of the '809 Patent. Specifically, the '809 Patent claims, *e.g.*, a method of allocating memory to a memory operation executed by a processor in a computer arrangement having a first processor configured for unified operation with a second processor, comprising: receiving a memory operation from a processor; mapping the memory operation to one of a plurality of memory heaps, wherein the memory operation references a shared memory address (SMA) in a shared memory, and wherein the mapping includes mapping the memory operation to one of a plurality of memory heaps based on the SMA, wherein the mapping produces a mapping result; and providing the mapping result to the processor.

94. With respect to exemplary devices, the NVIDIA Blackwell GB200 Superchip, made or sold by NVIDIA, incorporating the NVIDIA Blackwell B200 graphics processing unit (GPU), NVIDIA Grace central processing unit (CPU), and associated software, made or sold by NVIDIA, directly infringes at least claim 1 of the '809 Patent.

95. The NVIDIA Blackwell GB200 Superchip implements the Blackwell architecture and “introduces groundbreaking advancements for generative AI and accelerated computing.”

Breaking Barriers in Accelerated Computing

The **NVIDIA Blackwell architecture** introduces groundbreaking advancements for generative AI and accelerated computing. The incorporation of the second-generation Transformer Engine, alongside the faster and wider **NVIDIA NVLink™** interconnect, propels the data center into a new era, with orders of magnitude more performance compared to the previous architecture generation. Further advances in **NVIDIA Confidential Computing** technology raise the level of security for real-time LLM inference at scale without performance compromise. And Blackwell's new decompression engine combined with Spark RAPIDS™ libraries deliver unparalleled database performance to fuel data analytics applications. Blackwell's multiple advancements build upon generations of accelerated computing technologies to define the next chapter of generative AI with unparalleled performance, efficiency, and scale.

<https://resources.nvidia.com/en-us-blackwell-architecture/datasheet> (accessed 04/14/2025).

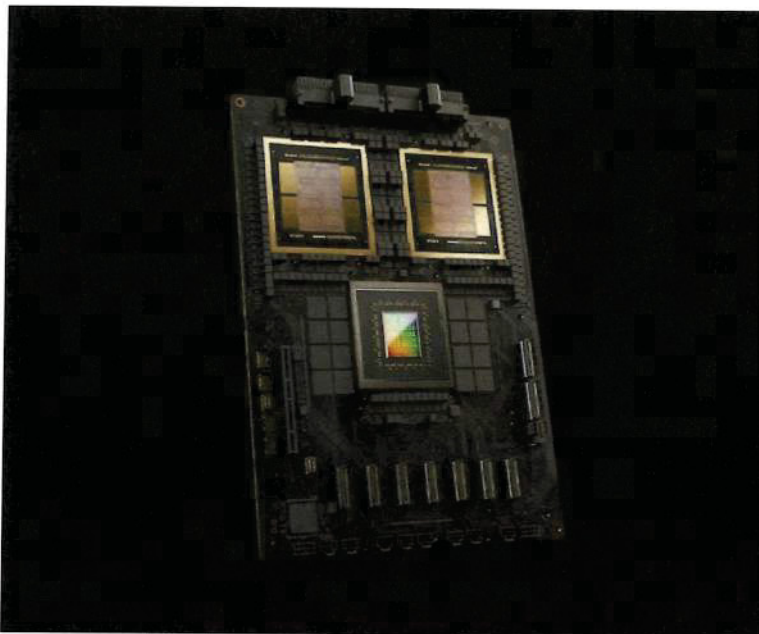


Figure 1. NVIDIA GB200 Superchip Incl. Two Blackwell GPUs and One Grace CPU

<https://cdn.prod.website->

[files.com/61dda201f29b7efc52c5fbaf/6602ea9d0ce8cb73fb6de87f_nvidia-blackwell-architecture-technical-brief.pdf](https://cdn.prod.website-files.com/61dda201f29b7efc52c5fbaf/6602ea9d0ce8cb73fb6de87f_nvidia-blackwell-architecture-technical-brief.pdf) (accessed 04/14/2025), at 5.

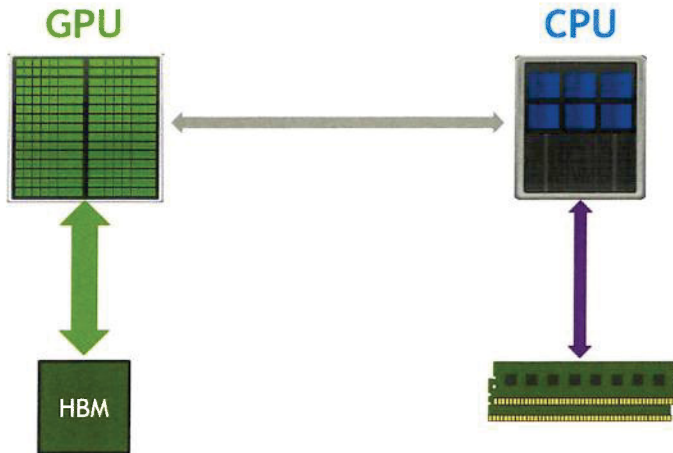
96. Upon information and belief, the NVIDIA Blackwell GB200 Superchip performs a method of allocating memory to a memory operation executed by a processor in a computer arrangement having a first processor configured for unified operation with a second processor. For example, the NVIDIA Blackwell GB200 Superchip “connects two NVIDIA B200 Tensor Core GPUs to the NVIDIA Grace CPU” and “power[s] a new era of computing.”

GTC—Powering a new era of computing, NVIDIA today announced that the NVIDIA Blackwell platform has arrived — enabling organizations everywhere to build and run real-time generative AI on trillion-parameter large language models at up to 25x less cost and energy consumption than its predecessor.

The NVIDIA GB200 Grace Blackwell Superchip connects two NVIDIA B200 Tensor Core GPUs to the NVIDIA Grace CPU over a 900GB/s ultra-low-power NVLink chip-to-chip interconnect.

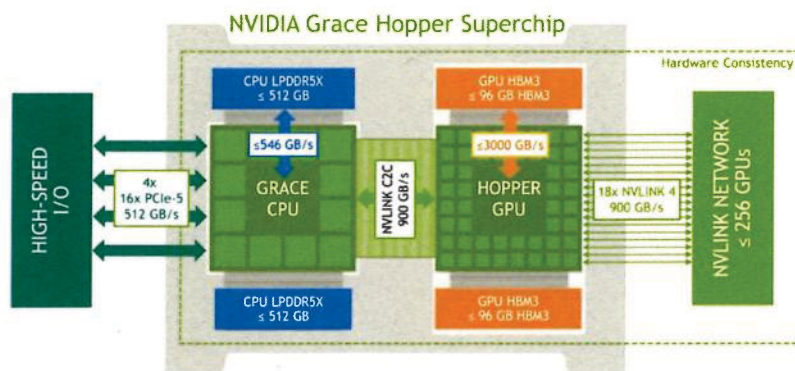
<https://nvidianews.nvidia.com/news/nvidia-blackwell-platform-arrives-to-power-a-new-era-of-computing> (accessed 04/14/2025); *see also* https://cdn.prod.website-files.com/61dda201f29b7efc52c5fbaf/6602ea9d0ce8cb73fb6de87f_nvidia-blackwell-architecture-technical-brief.pdf (accessed 04/14/2025); <https://developer.nvidia.com/blog/nvidia-gb200-nvl72-delivers-trillion-parameter-llm-training-and-real-time-inference/> (accessed 04/14/2025); <https://developer.nvidia.com/blog/nvidia-grace-hopper-superchip-architecture-in-depth/> (accessed 04/14/2025).

97. Upon information and belief, the NVIDIA Blackwell GB200 Superchip receives a memory operation from a processor, wherein the memory operation references a shared memory address (SMA) in a shared memory. For example, the NVIDIA Blackwell GB200 Superchip includes a CPU LPDDR5X and a GPU HBM3 accessible by both the CPU and the GPU.



<https://developer.nvidia.com/blog/improving-gpu-memory-oversubscription-performance/>

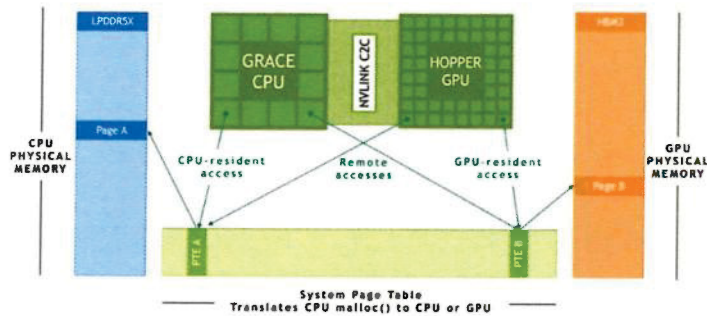
(accessed 04/14/2025).



<https://developer.nvidia.com/blog/nvidia-grace-hopper-superchip-architecture-in-depth/>

(accessed 04/14/2025).

98. The NVIDIA Blackwell GB200 Superchip also includes a unified memory “accessible to both the CPU and GPU using a single pointer.”



Id.

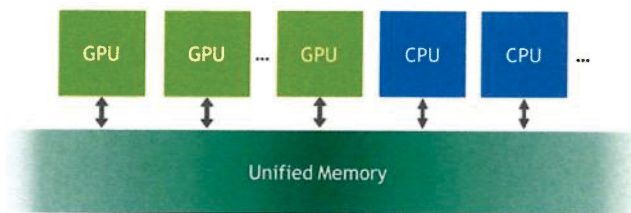
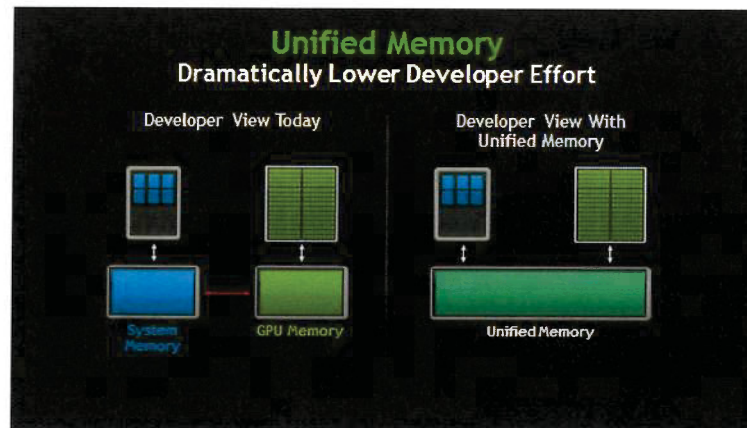


Figure 7. Unified Memory is a single memory address space accessible from any processor in a system.

<https://developer.nvidia.com/blog/unified-memory-cuda-beginners/> (accessed 04/14/2025).



Unified Memory creates a pool of managed memory that is shared between the CPU and GPU, bridging the CPU-GPU divide. Managed memory is accessible to both the CPU and GPU using a single pointer. The key is that the system automatically *migrates* data allocated in Unified Memory between host and device so that it looks like CPU memory to code running on the CPU, and like GPU memory to code running on the GPU.

<https://developer.nvidia.com/blog/unified-memory-in-cuda-6/> (accessed 04/14/2025). *See also*

<https://developer.nvidia.com/blog/improving-gpu-memory-oversubscription-performance/>

(accessed 04/14/2025); <https://developer.nvidia.com/blog/nvidia-gb200-nv172-delivers-trillion->

[parameter-llm-training-and-real-time-inference/](#) (accessed 04/14/2025);
<https://developer.nvidia.com/blog/maximizing-unified-memory-performance-cuda/> (accessed 04/14/2025).

99. In addition, on information and belief, the NVIDIA Blackwell GB200 Superchip maps the memory operation to one of a plurality of memory heaps, wherein the mapping includes mapping the memory operation to one of a plurality of memory heaps based on the SMA, wherein the mapping produces a mapping result, and provide the mapping result to the processor. For example, the NVIDIA Blackwell GB200 Superchip, upon information and belief, is configured to map the memory operation to one of a plurality of memory heaps, wherein the mapping is based on the SMA, and wherein the mapping produces a mapping result; and provide the mapping result to the Blackwell B200 GPU or the Grace CPU.

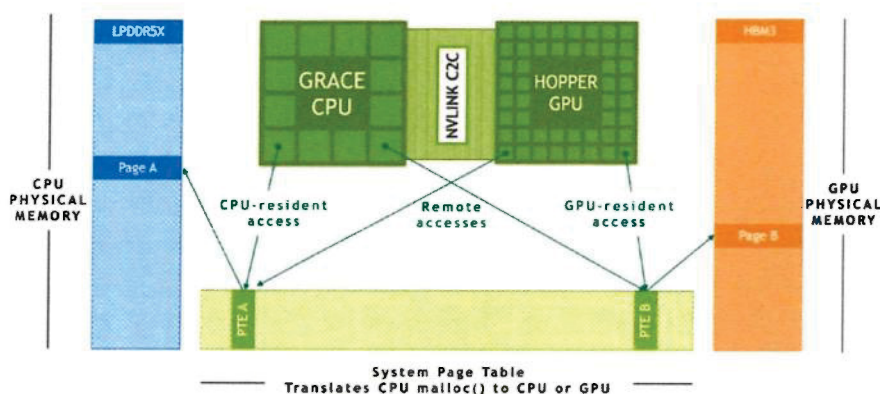
Page Migration Mechanism

Before diving into optimizations I want to explain what happens when a `cudaMallocManaged` allocation is accessed on the GPU. You can check out [my GTC 2017 talk](#) for more details. The sequence of operations (assuming no `cudaMemAdvise` hints are set and there is no thrashing) is:

1. Allocate new pages on the GPU;
2. Unmap old pages on the CPU;
3. Copy data from the CPU to the GPU;
4. Map new pages on the GPU;
5. Free old CPU pages.

Much like CPUs, GPUs have multiple levels of TLBs (Translation Lookaside Buffer: a memory cache that stores recent virtual to physical memory address translations) to perform address translations. When Pascal and Volta GPUs access a page that is not resident in the local GPU memory the translation for this page generates a fault message and locks the TLBs for the corresponding SM (on Tesla P100 it locks a pair of SMs that share a single TLB). This means any outstanding translations can proceed but any new translations will be stalled until all faults are resolved. This is necessary to make sure the SM's view of memory is consistent since during page fault processing the driver may modify the page table and add or revoke access to pages. The GPU can generate many faults concurrently and it's possible to get multiple fault messages for the same page. The Unified Memory driver processes these faults, remove duplicates, updates mappings and transfers the data. This fault handling adds significant overhead to streaming performance of Unified Memory on current generation GPU architectures.

<https://developer.nvidia.com/blog/maximizing-unified-memory-performance-cuda/> (accessed 04/14/2025).



<https://developer.nvidia.com/blog/nvidia-grace-hopper-superchip-architecture-in-depth/>

(accessed 04/14/2025).

**__host__ cudaError_t cudaMallocManaged (void **devPtr,
size_t size, unsigned int flags)**

Allocates memory that will be automatically managed by the Unified Memory system.

Parameters

devPtr

- Pointer to allocated device memory

size

- Requested allocation size in bytes

flags

- Must be either `cudaMemAttachGlobal` or `cudaMemAttachHost` (defaults to `cudaMemAttachGlobal`)

Returns

`cudaSuccess`, `cudaErrorMemoryAllocation`, `cudaErrorNotSupported`, `cudaErrorInvalidValue`

Description

Allocates `size` bytes of managed memory on the device and returns in `*devPtr` a pointer to the allocated memory. If the device doesn't support allocating managed memory, `cudaErrorNotSupported` is returned. Support for managed memory can be queried using the device attribute `cudaDevAttrManagedMemory`. The allocated memory is suitably aligned for any kind of variable. The memory is not cleared. If `size` is 0, `cudaMallocManaged` returns `cudaErrorInvalidValue`. The pointer is valid on the CPU and on all GPUs in the system that support managed memory. All accesses to this pointer must obey the Unified Memory programming model.

https://docs.nvidia.com/cuda/pdf/CUDA_Runtime_API.pdf (accessed 04/14/2025), at 446.

Starting with the Pascal GPU architecture, Unified Memory functionality is significantly improved with 49-bit virtual addressing and on-demand page migration. 49-bit virtual addresses are sufficient to enable GPUs to access the entire system memory plus the memory of all GPUs in the system. The Page Migration engine allows GPU threads to fault on non-resident memory accesses so the system can migrate pages on demand from anywhere in the system to the GPU's memory for efficient processing.

In other words, Unified Memory transparently enables oversubscribing GPU memory, enabling out-of-core computations for any code that is using Unified Memory for allocations (e.g. `cudaMallocManaged()`). It "just works" without any modifications to the application, whether running on one GPU or multiple GPUs.

Also, Pascal and Volta GPUs support system-wide atomic memory operations. That means you can atomically operate on values anywhere in the system from multiple GPUs. This is useful in writing efficient multi-GPU cooperative algorithms.

Demand paging can be particularly beneficial to applications that access data with a sparse pattern. In some applications, it's not known ahead of time which specific memory addresses a particular processor will access. Without hardware page faulting, applications can only pre-load whole arrays, or suffer the cost of high-latency off-device accesses (also known as "Zero Copy"). But page faulting means that only the pages the kernel accesses need to be migrated.

See NVIDIA, Unified Memory for CUDA Beginners, <https://developer.nvidia.com/blog/unified-memory-cuda-beginners/> (accessed 04/14/2025).

- ▶ `cudaMemAdviseSetAccessedBy`: This advice implies that the data will be accessed by device. Passing in `cudaCpuDeviceId` for device will set the advice for the CPU. If device is a GPU, then the device attribute `cudaDevAttrConcurrentManagedAccess` must be non-zero. This advice does not cause data migration and has no impact on the location of the data per se. Instead, it causes the data to always be mapped in the specified processor's page tables, as long as the location of the data permits a mapping to be established. If the data gets migrated for any reason, the mappings are updated accordingly. This advice is recommended in scenarios where data locality is not important, but avoiding faults is. Consider for example a system containing multiple GPUs with peer-to-peer access enabled, where the data located on one GPU is occasionally accessed by peer GPUs. In such scenarios, migrating data over to the other GPUs is not as important because the accesses are infrequent and the overhead of migration may be too high. But preventing faults can still help improve performance, and so having a mapping set up in advance is useful. Note that on CPU access of this data, the data may be migrated to host memory because the CPU typically cannot access device memory directly. Any GPU that had the `cudaMemAdviseSetAccessedBy` flag set for this data will now have its mapping updated to point to the page in host memory. If `cudaMemAdviseSetReadMostly` is also set on this memory region or any subset of it, then the policies associated with that advice will override the policies of this advice. Additionally, if the preferred location of this memory region or any subset of it is also device, then the policies associated with `cudaMemAdviseSetPreferredLocation` will override the policies of this advice. If the memory region refers to valid system-allocated pageable memory, then device must have a non-zero value for the device attribute `cudaDevAttrPageableMemoryAccess`. Additionally, if device has a non-zero value for the device attribute `cudaDevAttrPageableMemoryAccessUsesHostPageTables`, then this call has no effect.

See NVIDIA, CUDA Runtime API, https://docs.nvidia.com/cuda/pdf/CUDA_Runtime_API.pdf (accessed 04/14/2025), pp. 164-165.

```
__host__ cudaError_t cudaMemPrefetchAsync (const void  
*devPtr, size_t count, int dstDevice, cudaStream_t stream)
```

Prefetches memory to the specified destination device.

Parameters

devPtr

- Pointer to be prefetched

count

- Size in bytes

dstDevice

- Destination device to prefetch to

stream

- Stream to enqueue prefetch operation

Returns

[`cudaSuccess`](#), [`cudaErrorInvalidValue`](#), [`cudaErrorInvalidDevice`](#)

Description

Prefetches memory to the specified destination device. `devPtr` is the base device pointer of the memory to be prefetched and `dstDevice` is the destination device. `count` specifies the number of bytes to copy. `stream` is the stream in which the operation is enqueued. The memory range must refer to managed memory allocated via [`cudaMallocManaged`](#) or declared via `__managed__` variables, or it may also refer to system-allocated memory on systems with non-zero `cudaDevAttrPageableMemoryAccess`.

See NVIDIA, CUDA Runtime API, https://docs.nvidia.com/cuda/pdf/CUDA_Runtime_API.pdf

(accessed 04/14/2025), p. 198.

The need to support a variety of use cases makes low-level virtual memory allocation quite different from high-level functions like `cudaMalloc`. Instead of a single function, there are four primary functions that you'll work with, which we describe in more detail in later sections:

- `cuMemCreate`: Creates a physical memory handle.
- `cuMemAddressReserve`: Reserves a virtual address range.
- `cuMemMap`: Maps a physical memory handle to a virtual address range.
- `cuMemSetAccess`: Sets the memory access rights for each device to the allocation.

These functions can be used simultaneously with the runtime functions like `cudaMalloc` and `cudaMallocManaged`, but they require loading these entry points directly from the driver. For more information about how to interface with driver functions like these, see the [samples](#) included in this post or the various samples distributed with the CUDA Toolkit.

See NVIDIA, Introducing Low-Level GPU Virtual Memory Management,

<https://developer.nvidia.com/blog/introducing-low-level-gpu-virtual-memory-management/>

(accessed 04/14/2025).

Allocating physical memory

First, you need physical memory to operate on, for which you make use of the new function `cuMemCreate`. This function takes the handle `CUMemGenericAllocationHandle` that describes the properties of the memory to allocate, like *where is this memory physically located* or *what kind of shareable handles should be available*. For now, the only supported type of memory is pinned device memory on the current device but there are more properties to come in future CUDA releases.

Next, you need the size. Unlike `cuMemAlloc`, `cuMemCreate` only accepts sizes that are aligned to the granularity appropriate for the memory described by the handle. Use `cuMemGetAllocationGranularity` to get this granularity and use it to pad the requested size.

Now, you have all the information that you need to create a physical allocation, as the following code example shows:

```
size_t granularity = 0;
CUMemGenericAllocationHandle allocHandle;
CUMemAllocationProp prop = {};
prop.type = CU_MEM_ALLOCATION_TYPE_PINNED;
prop.location.type = CU_MEM_LOCATION_TYPE_DEVICE;
prop.location.id = currentDev;
cuMemGetAllocationGranularity(&granularity, &prop,
                             CU_MEM_ALLOC_GRANULARITY_MINIMUM);
padded_size = ROUND_UP(size, granularity);
cuMemCreate(&allocHandle, padded_size, &prop, 0);
```

You can use the allocation handle to map the allocated memory so that it's accessible to the rest of CUDA, as described in the next section. You can also export this allocation handle to something that can be used for inter-process communication, or even graphics interop. We come back to these use cases in later sections.

See NVIDIA, Introducing Low-Level GPU Virtual Memory Management,

<https://developer.nvidia.com/blog/introducing-low-level-gpu-virtual-memory-management/>

(accessed 04/14/2025); *see also*

https://developer.ridgerun.com/wiki/index.php/NVIDIA_CUDA_Memory_Management

(accessed 04/14/2025); <https://developer.nvidia.com/blog/unified-memory-in-cuda-6/> (accessed

04/14/2025); <https://developer.nvidia.com/blog/introducing-low-level-gpu-virtual-memory-management/> (accessed 04/14/2025).

100. NVIDIA has indirectly infringed, and continues to indirectly infringe, the '809 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the '809 Patent by others, such as its subsidiaries and affiliates, original equipment manufacturers, customers, resellers,

retailers, and other parties. These others directly infringe the '809 Patent by, for example, making, using, testing, selling, offering for sale throughout, or importing into the United States and this judicial District, downstream products (for example, without limitation, servers), incorporating the products that infringe (for example, without limitation, the Exemplary NVIDIA Blackwell GB200 Superchip). NVIDIA induced or induces such direct infringement by encouraging and facilitating others' infringing acts. For example, NVIDIA hires permanent sales and/or marketing personnel located throughout the United States, and in this judicial District. On information and belief, these sales and/or marketing personnel are engaged activities that are targeted to original equipment manufacturers, customers, resellers, retailers, and other parties, including original equipment manufacturers, customers, resellers, retailers, and other parties based on the United States, including instructions for making, using, testing, or selling the products that infringe.

101. NVIDIA specifically intended or intends these others, such as its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties, to infringe the '809 Patent and knew or knows that the induces acts of theses others constitute direct infringement. For example, although NVIDIA has notice of the '809 Patent, the scope of its claims, and the products covered thereby, NVIDIA nonetheless intentionally and knowingly encourages and facilitates these others to make, use, test, sell, offer for sale in, or import into, such covered products into the United States and thereby infringe the '809 Patent. Additionally, for example, NVIDIA designed the products that infringe (for example, without limitation, the Exemplary NVIDIA Blackwell GB200 Superchip) such that they would each infringe the '809 Patent if made, used, tested, sold, offered for sale, or imported into the United States. NVIDIA provided, directly, or indirectly, the products that infringe to others, such as, but not limited to, its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties,

knowing and intending that those others use, test, sell, offer for sale, and/or import in and into the United States downstream products (for example, without limitation, servers) that include the products that infringe (for example, without limitation, the Exemplary NVIDIA Blackwell GB200 Superchip), thereby directly infringing one or more claims of the '809 Patent.

102. NVIDIA has also indirectly infringed and continues to indirectly infringe the '809 Patent under 35 U.S.C. § 271(c) by materially contributing to infringement of the '809 Patent by its subsidiaries and affiliates, original equipment manufacturers, customers, resellers, retailers, and other parties by selling the products that infringe to such third parties for use in downstream end-user products in a manner that infringes the '809 Patent. The products that infringe (for example, without limitation, the Exemplary NVIDIA Blackwell GB200 Superchip) have no substantial non-infringing uses and are a material part of the invention. Moreover, the products that infringe provide vital functionality to such downstream products (for example, without limitation, servers). Further, the products that infringe cannot be used without being incorporated into a downstream product. Thus, any manufacture, use, test, sale, offer for sale, or importation in or into the United States of a product that infringes or a downstream product incorporating a product that infringes, infringes the '809 Patent.

103. As a result of NVIDIA's infringement of the '809 Patent, Onesta is entitled to monetary damages in an amount adequate to compensate for NVIDIA's infringement, but in no event less than a reasonable royalty for the use made of the invention by NVIDIA, together with interest and costs as fixed by the Court.

104. Despite NVIDIA's actual notice of its infringement as of at least the filing of the Complaint, NVIDIA's acts of direct and indirect infringement of the '809 Patent are deliberate and willful, and have caused, and will continue to cause, substantial damage and irreparable harm to

Onesta, and Onesta has no adequate remedy at law.

PRAYER FOR RELIEF

WHEREFORE, Onesta requests the Court grant the relief set forth below:

- a. Enter a judgment that NVIDIA has directly and/or indirectly infringed, and continues to directly and/or indirectly infringe, one or more claims of each of the Asserted Patents;
- b. Enter a judgment that NVIDIA's acts of patent infringement are willful;
- c. Order NVIDIA to account for and pay damages caused to Onesta by NVIDIA's unlawful acts of patent infringement;
- d. Award Onesta pre- and post-judgment interest on such damages and costs Onesta incurs in this action;
- e. Award Onesta increased damages and attorney fees pursuant to 35 U.S.C. §§ 284 and 285; and
- f. Grant Onesta other and further relief, including equitable relief, as the Court finds just and proper.

DEMAND FOR JURY TRIAL

Pursuant to Fed. R. Civ. P. 38, Onesta hereby demands a jury trial for all claims and issues deemed to be triable by a jury.

Dated: April 17, 2025

Respectfully submitted,

/s/ Michael T. Renaud

Michael T. Renaud (629783)

Adam Rizk (688305)

Samuel F. Davenport (636958)

William Meunier (*Pro hac vice* forthcoming)

Matthew A. Karambelas (691034)

Catherine Xu (*Pro hac vice* forthcoming)

Meena Seralathan (*Pro hac vice* forthcoming)

Courtney Herndon (693418)

Hannah M. Edge (*Pro hac vice* forthcoming)
Mintz Levin Cohn Ferris
Glovsky and Popeo PC
One Financial Center
Boston, MA 02111
Tel: 617-542-6000
Fax: 617-542-2241
mtrenaud@mintz.com
arizk@mintz.com
sfdavenport@mintz.com
wameunier@mintz.com
makarambelas@mintz.com
cxu@mintz.com
mserelathan@mintz.com
cherndon@mintz.com
hmedge@mintz.com

Yanyi Liu (*Pro hac vice* forthcoming)
Mintz Levin Cohn Ferris
Glovsky and Popeo PC
44 Montgomery Street, 36th Floor
San Francisco, CA 94104
Tel: (415) 432-6013
yliu@mintz.com

Of Counsel:

Andrea L. Fair
Texas State Bar No. 24078488
andrea@millerfairhenry.com
Garrett Parish
Texas State Bar No. 24125824
garrett@millerfairhenry.com
MILLER FAIR HENRY PLLC
1507 Bill Owens Parkway
Longview, TX 75604
(903) 757-6400 (telephone)
(903) 757-2323 (facsimile)

Counsel for Plaintiff
Onesta IP, LLC