Choosing the Right NVIDIA Cloud GPU: A Technical Guide

Contents

Chapter 01

What is Cloud GPU Compute?

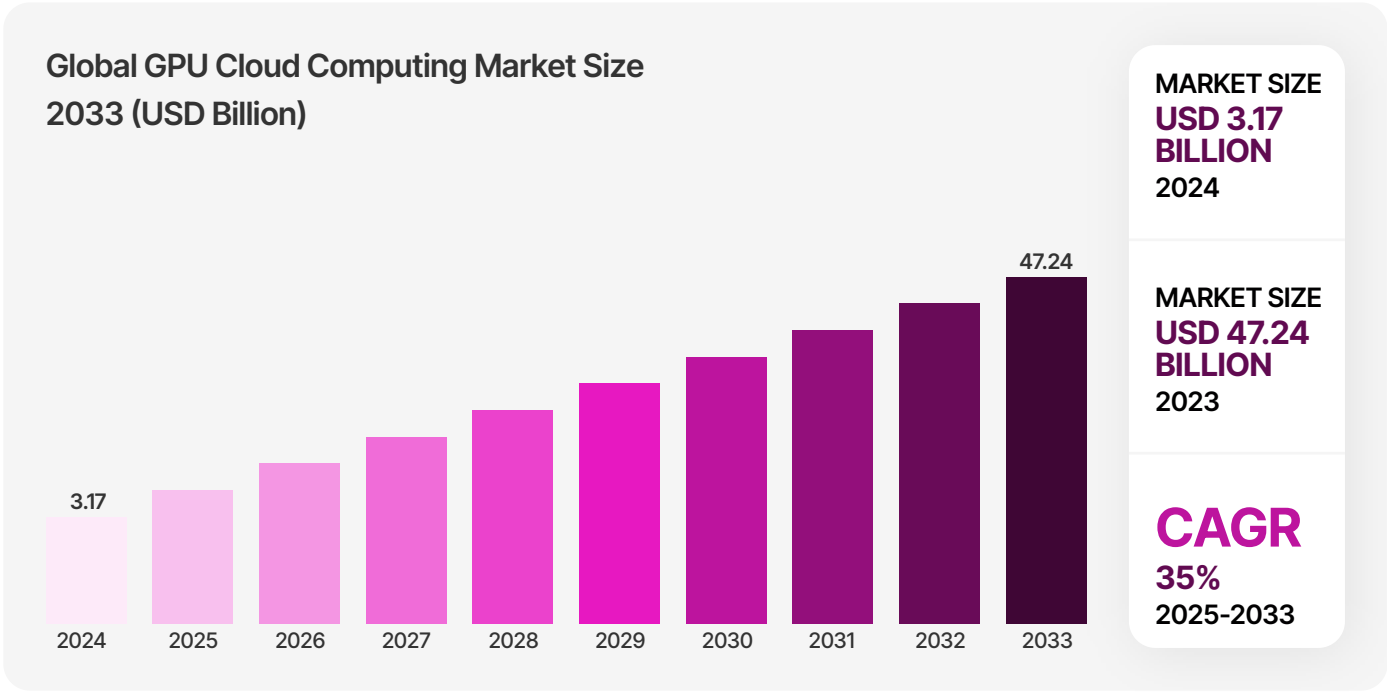
Cloud GPU compute refers to using Graphics Processing Units (GPUs), hosted in remote cloud data centers, to accelerate computations. Instead of setting up on-premise GPU hardware, you rent or access a virtual GPU over the internet from a cloud provider. These cloud GPUs are virtualized and delivered as a service, allowing users to perform graphics-intensive or parallel computations without owning the hardware. Cloud GPU compute leverages virtualization and high-speed networks, so developers can run GPU-accelerated workloads like AI model training, 3D rendering, and others on powerful remote servers, and then receive the results over the network.

Overview of Cloud GPU Technology

Public cloud providers (AceCloud, AWS, GCP, Azure) have GPU resources where users can access them. This represents a simple process of access to cloud GPU resources:

- In a typical setup, users will request access to GPU resources through a cloud platform.
- The cloud provider allocates the virtualized GPU based on the request. For instance, dedicated GPU resources for a single user or shared GPU resources.
- A virtual machine (EC2 instance) is setup and configured to access the GPU resources.
- The user then uploads workloads using cloud storage services such as S3 bucket or data ingestion tools. The GPU then processes these workloads.
- The GPU then performs parallel processing. The GPUs process complex calculations simultaneously. In the case of AI workloads, the tasks are divided into smaller operations and executed simultaneously. This allows for faster data processing.
- Once the processing is done, the user accesses the output through the cloud interface or a cloud storage service.
- After users achieve their objectives, they can shut down the instances to save costs and free up resources to be used by other users.

Chapter 02



Why Choose Cloud GPU Compute?

It is critical to put it into context. The significance of GPUs cannot be overstated, and this has been demonstrated by Business Research Insights. Based on these trends, it is evident that enterprises are massively adopting cloud GPUs to get the required compute power for AI development, enhanced gaming, and more.

Cloud GPUs offers several compelling benefits over On-Premise GPUs:

- **Scalability & Flexibility**

  Cloud platforms allow dynamic GPU resource adjustments to match workload demands. You can spin multiple GPU instances for intensive jobs and shut them down when done. For example, you might deploy a cluster of GPUs for a deep learning training run, then scale back to a single GPU for inference. It is ideal for different business users. For instance, those with significant workloads can increase the number of GPU resources without being concerned about additional hardware. This achieves flexible resource allocation. For AI firms, this avoids the costs of purchasing new hardware when demand arises. Moreover, when there is a reduced workload, enterprises avoid the cost of paying for GPU hardware. Cloud GPUs, thus, enable real-time resource-need matching, avoiding idle hardware while ensuring you can handle peak loads.

Tip: Use auto-scaling to add/remove GPU instances as your workload changes. This helps avoid idle costs.

- **Cost Efficiency (Pay-as-you-Go)**

  With cloud GPUs, you pay only for what you use. There is no large upfront capital expense to buy expensive GPU cards. Instead, you rent by-the-hour or second. This model is cost-effective for sporadic or project-based needs. For example, a short-term AI experiment, since you can rent a high-end GPU for hours or days instead of purchasing one. It also shifts expenses to Operational Expenditure (OpEx).

  Let's consider another example. When the workload is at its peak, such as deep learning tasks or machine learning work like inference, it is possible to increase the GPU instances. Moreover, the company can then scale down after completing AI workloads. This is significantly cost-efficient, as the company only pays when it needs the resources. It promotes efficient use of resources.

Note: Pay-as-you-go GPUs are ideal for startups or research teams that need burst compute without upfront hardware cost.

- **Access to Latest Hardware**

  As a user or enterprise, keeping up with the latest GPU updates will be costly. GPU hardware is updated every year, and new versions are released. The new versions are often better as they massively reduce the defective areas in the chips. It can be very expensive and even impractical for companies to keep up with the new technologies, especially when running on a tight budget.

  Cloud GPU providers frequently update their offerings to include the newest GPU models and architectures. As a user, you can access top-of-the-line NVIDIA GPUs like H100 or B100 immediately in the cloud without purchasing or upgrading hardware yourself.

- **Global Availability & Collaboration**

  Major cloud providers have data centers around the world. You can deploy GPU workloads in different regions to be closer to end-users or data sources, reducing latency. A major benefit of cloud GPUs is that it is possible to collaborate even for geographically dispersed teams.

  GPU resources are hosted on the cloud, and team members can participate and collaborate on GPU-accelerated projects as they can access the GPU instances. They only need a stable internet connection, and their PCs can contribute to development projects. Moreover, the cloud GPUs, especially used for development, are integrated with collaboration tools such as version control systems. With this, team members can track the progress, roll back the development from where they left off, and understand the changes made. This enables effective collaboration, and team members can iterate on models easily. This democratizes access to compute resources and fosters innovation among AI enthusiasts, researchers, and data scientists around the globe.

  This is critical for collaborating teams as they can access the same instances where workloads are deployed remotely. This allows them to collaborate on required tasks, thus improving performance.

- **Reduced Operational Overhead**

  Cloud GPU saves the company capital-intensive venture of setting up an on-premise GPU. Often, the cost of an on-premise GPU cluster is higher as there is a need for servers, networking tools, GPU hardware, and the resources to manage the data centers. The two major costs that a company struggles with include:

  - **Datacenter Costs:** To get the GPU cluster up and running, one needs a secure location with reliable power. The high cost of electricity, space, and cooling systems will ultimately drive-up operational costs.

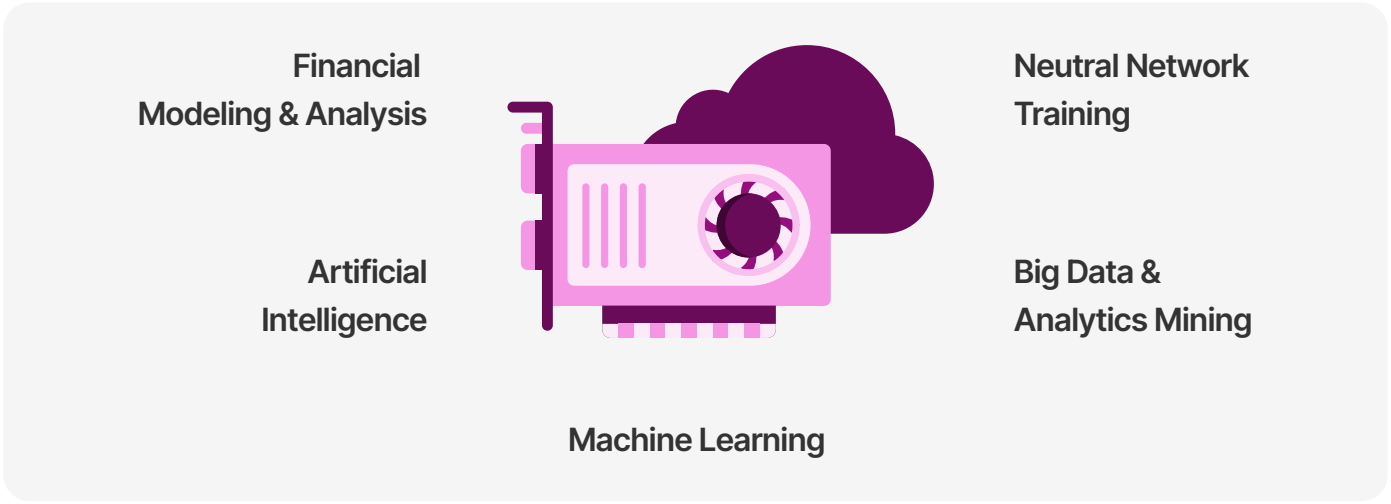
  - **Hardware Costs:** GPUs are costly, and keeping up to date with the latest GPU is even more costly and can become impractical in the long term. Apart from the GPU hardware, there is the cost of other hardware, such as server racks and storage, which will also be costly in the long run.

With cloud GPUs, the provider manages physical hardware, cooling, updates, and repairs. You don't need to worry about installing GPUs, driver updates, or machine downtime due to hardware failures, which offloads a lot of IT maintenance. Your team can focus on developing applications or running experiments while the cloud provider ensures the GPU servers are running optimally.

- **Flexible Configuration**

  Cloud GPU instances often come with configurable CPU, memory, and storage. Most public cloud providers have a user-friendly interface, and users can create virtual machines, deploy workloads, and connect to GPU resources. Cloud GPUs usually come pre-configured, making it easy to deploy workloads. With GPU instances, users can configure resources depending on their needs. For example, they can select the desired number of GPUs, memory size, and CPU cores based on their computational demands. You can choose an instance type that balances GPU power with the right amount of CPU/RAM for your task.

Chapter 03



Business Use Cases for GPU Compute

GPUs excel at parallel processing and number-crunching. So, any computationally intensive workload that involves processing large matrices, images, or simulations can leverage this feature. Here are some common business use cases that require GPUs:

- **AI/ML and Deep Learning**

  Training complex machine learning models (especially deep neural networks) is one of the primary use cases for GPU compute. GPUs dramatically accelerate the training of models in computer vision, natural language processing, recommendation systems, and more, by performing matrix math in parallel. This speed-up shortens development cycles for AI products. GPUs are also used for inference in production. Inference represents a moment of truth for a model, where the model is presented with new data in a real-world scenario.

  In this case, the model is expected to make quick decisions based on the learnings. This is especially visible in autonomous vehicles, where data is processed in real-time, and decisions are made to ensure safety. Thus, the GPU resources accelerate the inference process as multiple calculations are performed quickly, reducing latency and serving models in real-time.

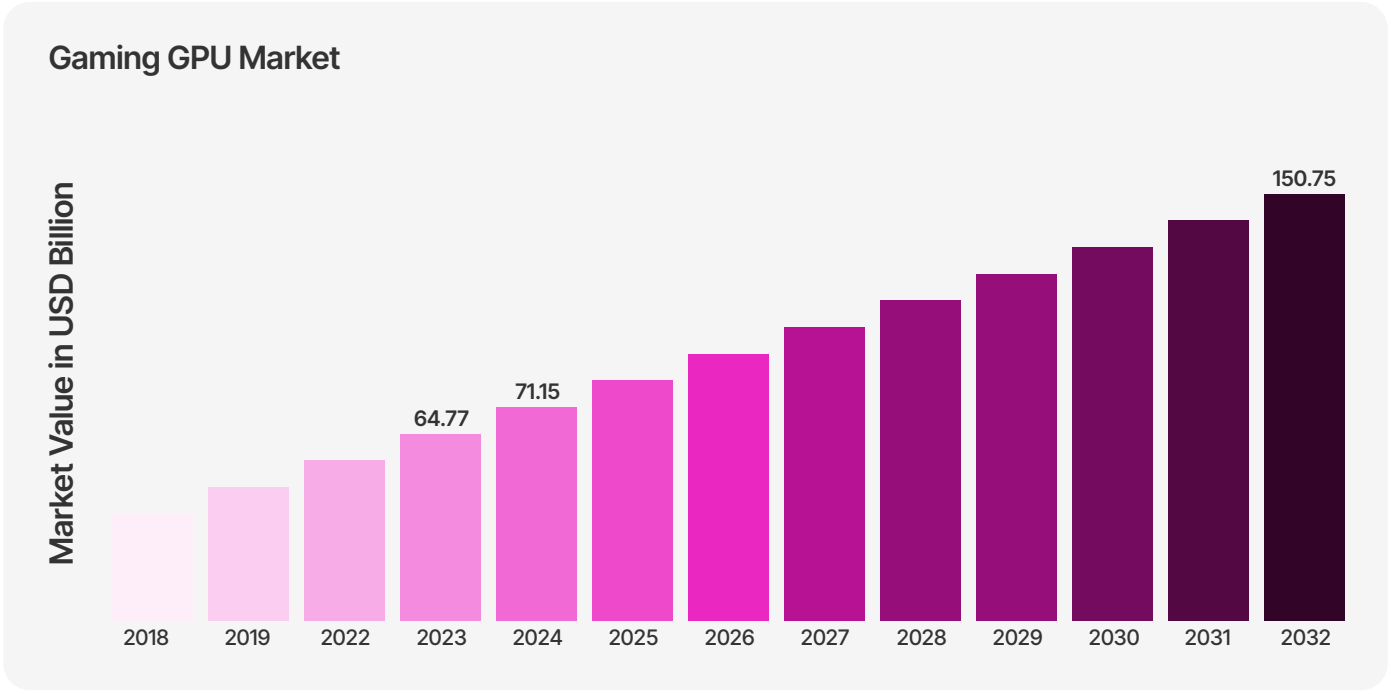
Tip: Use gradient checkpointing and mixed precision to reduce GPU memory usage during deep learning model training.

- **High-Performance Computing (HPC) & Scientific Simulations**

  Research labs and industries run simulations that model physical phenomena like climate modeling, computational fluid dynamics, molecular dynamics (drug discovery protein folding), astrophysics simulations, and more. These HPC tasks involve large mathematical computations, often using linear algebra, which GPUs can accelerate significantly. When high accuracy is needed, scientific computing often benefits from GPU support for double-precision arithmetic (FP64). Businesses in finance (risk simulations), engineering (FEA, CFD), and science rely on GPU-accelerated HPC to get results in reasonable timeframes.

- **Gaming and Real-Time Graphics (Cloud Gaming)**

  The growth of immersive and core games with surreal graphics has driven the demand for GPUs. This trend has been on an upward trajectory, a testament to GPUs' compute capabilities and improved graphic rendering.



Source: Market Research Future

GPUs excel in video games, performing intense calculations to render graphics. To really put GPUs' power into context, let's look at the number of calculations performed per second to render realistic graphics for a video game.

A video game like Mario from 1995 required 100 million calculations per second. What about modern video games like Minecraft? Well, that one required 100 billion calculations per second. But that was more than a decade ago. With modern games, which render superb and real-life graphics, you will need a GPU that can perform trillions of calculations per second.

The gaming industry uses powerful GPUs to render complex 3D graphics. Cloud gaming services run video games on GPU servers in the cloud and stream the video output to users, allowing high-end games to be played on low-end devices. These remote game servers need robust GPUs to render high frame rates and resolutions. GPUs also enable virtual reality (VR) and augmented reality (AR) streaming experiences from the cloud.

Did You Know? Cloud gaming platforms can dynamically assign GPU resources based on game load, resolution and player concurrency.

- **3D Rendering and Visualization**

  Industries like media & entertainment, architecture, engineering, and construction (AEC) utilize GPUs for rendering images and animations. 3D rendering (using engines like Blender, Maya, 3ds Max, or renderers like Pixar Renderman) can be sped up by cloud GPU farms. To build a 3D world, there are multiple objects, each with a model space. GPUs perform millions of calculations, making it possible to simulate lighting, shading, and texture. Instead of waiting hours for CPU-based rendering, artists can offload jobs to dozens of GPUs in the cloud and get results in minutes.

  GPUs have significantly changed rendering. GPU technologies, including Unreal Engine and Octane Render, leverage parallel processing capabilities to speed up rendering. Consequently, architects can get faster results, increasing output and giving more time for additional testing during the design process. Multiple testing architectures guarantee quality results.

Note: Batch rendering with cloud GPUs can be automated using render queue schedulers, improving throughput and reducing idle time.

- **Video Transcoding and Processing**

  Another GPU use case is handling video at scale. Video transcoding (converting videos between formats or resolutions) can leverage specialized GPU encoding/decoding hardware to process streams much faster than a CPU. For instance, cloud streaming platforms use GPU-accelerated encoders to compress live video into various formats (H.264, HEVC, AV1) in real-time for viewers. One GPU can transcode multiple streams concurrently by utilizing its parallel hardware encoders.

Chapter 04

NVIDIA GPU Categorization for Different Use Cases

NVIDIA provides a wide range of GPU models, especially in their data center lineup. Each GPU model is generally optimized for certain workloads. When choosing a cloud GPU, it is important to understand the differences so you pick a GPU instance that matches your use case (whether it is training an AI model, rendering graphics, or running a simulation). Below is a comparison of some popular NVIDIA GPUs (L4, L40s, A100, H100, RTX A6000, A2, A30, A10, A40) and how they fit different needs:

GPU Model (Architecture)	Memory	Key Features	Ideal Use Cases
NVIDIA L4 (Ada Lovelace)	24 GB GDDR6	Low-power (≈72 W) small form factor; 4th-gen Tensor Cores optimized for FP16/INT8; AV1 encode/decode support	AI inference at scale, e.g., serving models, recommendation systems, video streaming/transcoding (efficient video encoding), and edge deployments where power is limited.
NVIDIA L40s (Ada Lovelace)	48 GB GDDR6	High-performance Ada GPU with Tensor Cores and RT Cores; "Universal" data center GPU for both graphics and compute (Limited FP64 throughput compared to HPC GPUs)	AI training and large-scale deep learning (supports big models/LLMs); High-end graphics/rendering (real-time ray tracing via RT Cores); multi-workload environments requiring versatility.
NVIDIA A100 (Ampere)	40 GB or 80 GB HBM2e	Flagship Ampere compute GPU; very high memory bandwidth (~1.5–2 TB/s); strong FP64 performance (for scientific computing); 3rd-gen Tensor Cores for FP16/FP32; supports MIG (Multi-Instance GPU) partitioning; NVLink for multi-GPU	AI/ML training of large models (e.g., GPT-3, BERT). High-Performance Computing (scientific simulations requiring double precision) heavy data analytics. MIG allows splitting into up to 7 smaller instances for parallel inferencing tasks.
NVIDIA H100 (Hopper)	80 GB HBM3	Latest-gen (Hopper) GPU with ultra-high performance; extremely high memory bandwidth (>2 TB/s) and more Tensor Cores; introduces FP8 precision (Transformer Engine) for AI; supports MIG and NVLink (up to 256 GPUs in NVLink cluster)	Exascale AI training is designed for the largest neural networks and fastest training times. Large language models (LLMs) and advanced deep learning require maximum throughput and top-tier HPC workloads.

GPU Model (Architecture)	Memory	Key Features	Ideal Use Cases
NVIDIA A2 (Ampere)	16 GB GDDR6	Entry-level datacenter GPU; low-profile, low-power (40–60 W) PCIe card; fewer CUDA cores; 3rd-gen Tensor Cores (Ampere) for basic AI acceleration; no NVLink/ MIG	Basic AI inference and edge computing – good for lightweight models or small batch inferencing on the edge scenarios where power/ cooling is limited testing and development of AI models on a budget
NVIDIA RTX A6000 (Ampere)	48 GB GDDR6	Professional workstation GPU (the successor to Quadro series); high FP32 and ray-tracing performance (84 RT Cores); 336 Tensor Cores (for AI tasks) but no MIG; actively cooled in workstations (passive for servers); supports NVLink (pairing 2 GPUs)	Visual computing and content creation – 3D rendering, CAD, VR, visualization, running professional graphics software. Virtual GPU Workstations (VDI) for designers and engineers can also provide AI/ML training or inference when data center GPUs like A40/A100 are unavailable. The RTX A6000 offers the same 48GB memory and similar compute to the A40, making it suitable for both graphics and moderate compute tasks. However, by default, it doesn't have data-center features like MIG or ECC memory that A100/A40 have.
NVIDIA A30 (Ampere)	24 GB GDDR6	Mid-range Ampere tensor core GPU; high memory bandwidth HBM2; supports MIG; lower power (~165 W) than A100; strong FP64 relative to smaller cards (but less than A100)	Versatile enterprise AI – suitable for both AI inference at scale and moderate training tasks. HPC applications that need good double-precision but don't require the full A100 ideal where a mix of training and inference or data analytics will run on the same GPU. NVIDIA calls A30 the "most versatile mainstream compute GPU" because it can handle a variety of workloads cost-effectively.

GPU Model (Architecture)	Memory	Key Features	Ideal Use Cases
NVIDIA A30 (Ampere)	24 GB GDDR6	Mid-range Ampere tensor core GPU; high memory bandwidth HBM2; supports MIG; lower power (~165 W) than A100; strong FP64 relative to smaller cards (but less than A100)	Versatile enterprise AI – suitable for both AI inference at scale and moderate training tasks. HPC applications that need good double-precision but don't require the full A100 ideal where a mix of training and inference or data analytics will run on the same GPU. NVIDIA calls A30 the "most versatile mainstream compute GPU" because it can handle a variety of workloads cost-effectively.
NVIDIA A10 (Ampere)	24 GB GDDR6	Single-slot, full-height GPU; 2nd-tier Ampere with 24GB memory; provides both significant CUDA/Tensor performance and has display outputs (for virtual workstation use); no MIG; ~150 W TDP.	Flexible GPU for mixed workloads – designed for both Virtual Desktop Infrastructure (VDI) and AI inference/training tasks (Which GPU is best for me?). It's often used in cloud instances that serve as virtual workstations (graphics for CAD, 3D apps) during the day and run AI/data jobs off-hours. A10 is a cost-effective alternative to A100 for many models, and it excels in situations needing GPU acceleration but not the absolute highest compute.

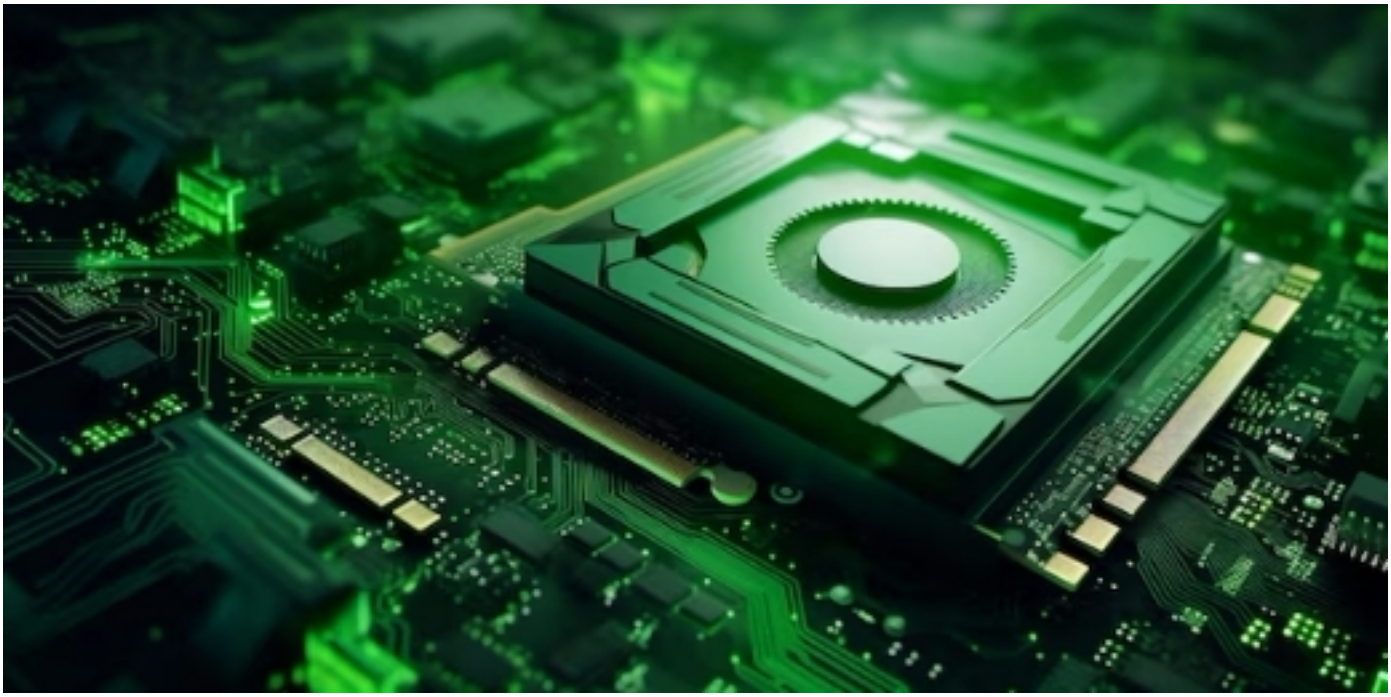
Did You Know? The Hopper architecture supports transformer engine optimizations that boost LLM performance up to 6×over A100.

GPU Model (Architecture)	Memory	Key Features	Ideal Use Cases
NVIDIA A40 (Ampere)	48 GB GDDR6	Datacenter version of RTX A6000 (48GB); passively cooled for servers; 10,752 CUDA cores, 336 Tensor Cores (same as A6000); supports NVLink (pair 2 for 96GB); no MIG (Ampere GA102 architecture); very high graphics and compute performance, but memory bandwidth (~696 GB/s) is about half of A100's (HBM2e) ([Right-Sizing Training Workloads with NVIDIA A100 and A40 GPUs	State-of-the-art GPU solution for workloads in rendering, AI accelerations, and real-time ray tracing. A40 fits into versatile medium-size workloads. With NvLink capabilities, A40 delivers high performance in multi-GPU systems.

Table 0.1: The table addresses some of the most popular NVIDIA GPUs and the segments in which these serve best.

Tip: The A100 supports NVLink and MIG simultaneously enabling both large-scale training and multi-tenant inference.

Chapter 05



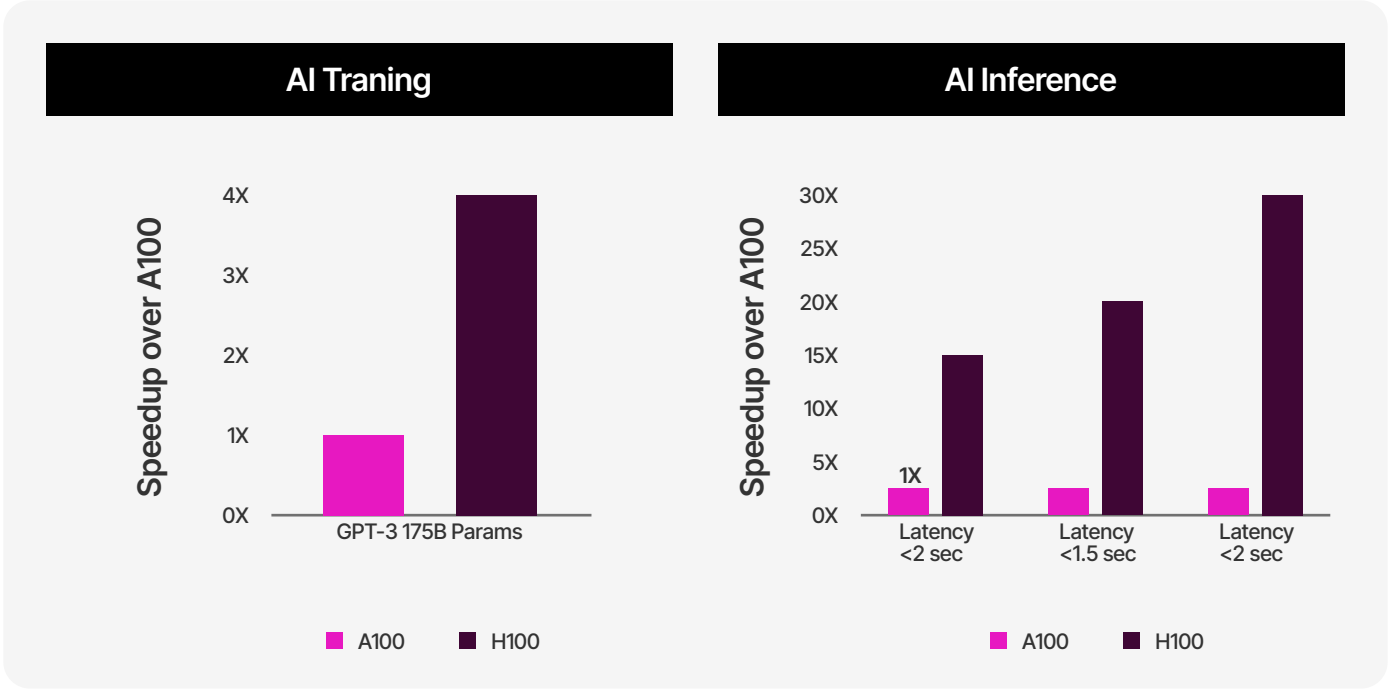
Key Points When Comparing NVIDIA GPUs

- **Double-Precision (FP64) Performance**

  For scientific computing or simulations that require high-precision math (FP64), A100 and H100 stand out. The A100 can deliver up to 19.5 teraflops of FP64 performance using Tensor Cores, and the H100 can do even more with its Hopper architecture. In contrast, GPUs like the L4 and L40S are not optimized for FP64 (L4/L40s have very limited FP64 capability) and thus are not ideal for HPC workloads that need double precision. For example, a CFD simulation for aerospace would favor A100/H100 for accuracy, whereas using an L40 might result in slower computation or lower precision. Always check if your workload needs FP64. If yes, lean towards the A100/H100 class (or older V100) rather than the L or RTX series.

- **Tensor Core and AI Throughput**

  Consider the Tensor Core generation and throughput if your main goal is AI model training or large-scale inference. Newer GPUs have more advanced Tensor Cores:



Source: Nvidia

- H100 (4th-Gen Tensor Cores) introduces FP8 and has significantly higher AI throughput (it is reportedly over 2X faster than A100 in many AI benchmarks).
- A100 (3rd-Ge Tensor Cores) is still an AI workhorse, and can be more cost-effective if H100 is scarce or expensive.
- L40S (4th-Gen Tensor, similar to H100 but with GDDR6 memory) provides excellent FP16/FP32 performance. NVIDIA touts it as a versatile training GPU for AI, but again, memory bandwidth and P64 are lower than the A100/H100.
- Smaller cards like the A10 or RTX A6000 have Tensor Cores too, but in fewer numbers. They are sufficient for inference or training medium models. But for cutting-edge model training (like billion-parameter models), the A100/H100 will train much faster. For example, the A100 has 432 Tensor Cores vs. the A40's 336, and much higher specialized throughput for matrix ops.

- **Memory Capacity and Bandwidth**

  GPU memory (VRAM) is a critical factor. Large models or high-resolution data require more VRAM:

	AI Training and Inference		AI Inference		
	H100 SXM	A100	H100 PCIe	L40	L4
RAM	80 GB	80 GB	80 GB	48 GB	24 GB
Bandwidth	3.35 TB/s	2 TB/s	2 TB/s	864 GB/s	300 GB/s
FP64, TFLOPS	67	19	51	-	-
TF32, TFLOPS	494	156	738	90	60
FB16, TFLOPS	989	312	757	181	121
BF16, TFLOPS	989	312	757	181	121
FP8, TFLOPS	1979	-	1513	362	242

- H100 and A100 80GB GPUs have enormous memory for the biggest models or very large batch sizes. They use HBM memory, which also offers huge bandwidth (over 1.5 TB/s). This high bandwidth quickly feeds the GPU cores with data, which is vital for massively parallel computations. If your model or dataset is huge, these GPUs might be the only option to avoid running out of memory or getting bottlenecked.

Tip: High Bandwidth Memory (HBM) is crucial for workloads like genomics, graph analytics and large batch inference.

- Mid-range 24–48GB GPUs (A10, A30, A40, RTX A6000, L40s) can handle most tasks. For example, a 48GB A40 can train many deep learning models that a 16GB GPU (like a T4 or A2) could not. These mid/high memory GPUs are often the choice for general workloads requiring serious memory, but not the extreme end. They also tend to be more available and cheaper per hour than A100s. For instance, one might choose a 48GB A40 to finetune a large model, whereas a smaller 16GB card would fail at a lower cost than an A100 40GB.

- Small 16GB GPUs (L4, A2, older T4) are fine for inference or smaller training jobs. They are very cost-efficient for running multiple parallel jobs on smaller models. If memory is a limiting factor, you would either pick a bigger GPU or use techniques like model parallelism. Cloud providers often offer choices like a single A100 40GB vs. multiple smaller GPUs. Multiple L4 GPUs might serve many concurrent inference requests depending on the scenario, whereas a single A100 could handle one big training job.

- **Graphics and Specialized Cores**

  Some GPUs include RT Cores (for ray tracing), and others are certified for certain graphics APIs, and more. For example, L40s and RTX A6000 have RT Cores that accelerate ray-traced rendering in real-time, which is beneficial for graphics-heavy applications. Those models make sense if you are running a GPU for a graphics or visualization workload. On the other hand, pure compute GPUs (A100/H100) do not have RT cores. They focus purely on compute and typically are used headless (no display). Also, professional GPUs (A40, A6000) support virtualization with multiple virtual displays for VDI use. So, if your cloud use case involves virtual workstations or 3D streaming, choosing a GPU like A40 or A6000 (which NVIDIA supports for virtualization drivers) would be appropriate, whereas an A100 might not even offer a video output.

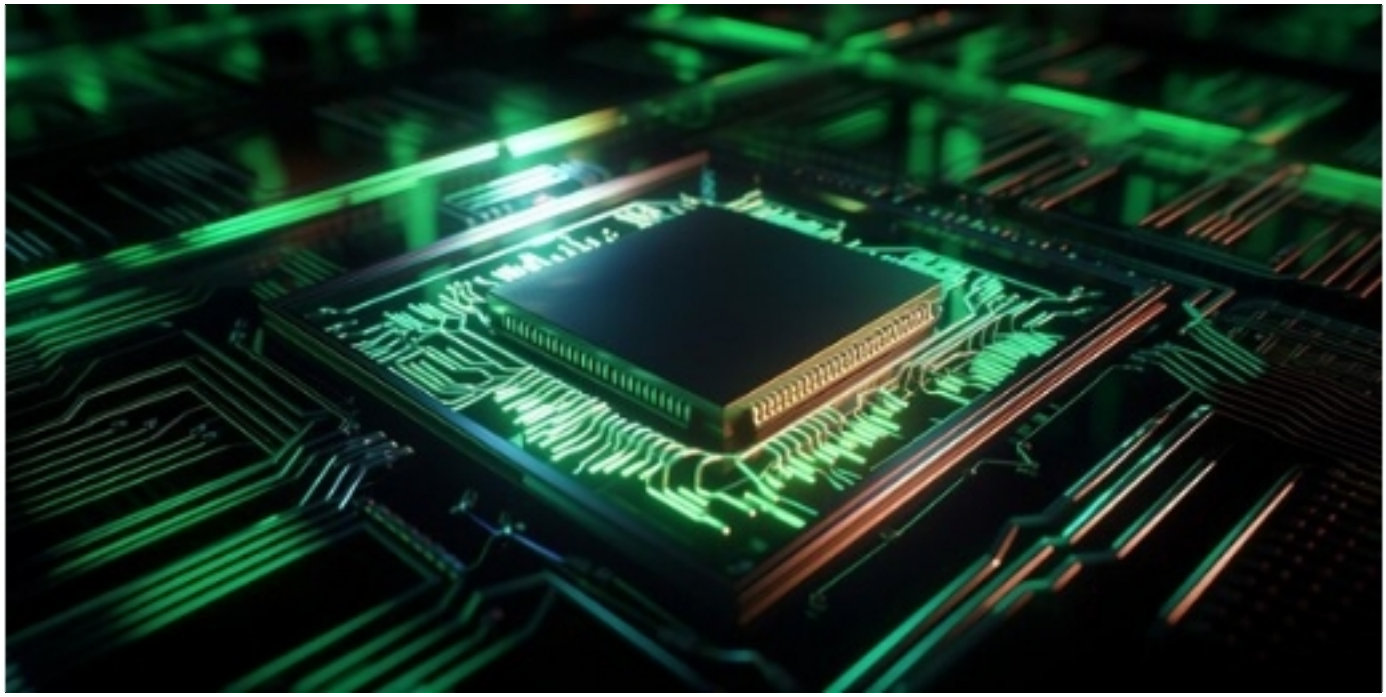
- **Multi-Instance and Multi-GPU**

  Look at MIG support if you need to split a GPU to run multiple jobs. A100 and H100 support MIG (allowing the GPU to be partitioned into up to 7 smaller instances). This is great for cloud environments to maximize utilization. For example, 7 users can each get a slice of an A100 with ~10GB and a portion of compute. L4, L40, A10, and others, do not support MIG (MIG is only on GPUs based on the GA100 chip – A100, A30 – and Hopper). Alternatively, if you need multiple GPUs working together on one job (multi-GPU training), consider NVLink availability. A100 (SXM form factor especially) can use NVLink/NVSwitch to join 4, 8, and 16 GPUs with fast interconnect for distributed training. Some cloud instances offer multi-GPU with NVLink (like 8x A100 servers). H100 pushes this further with support for giant clusters (up to 256 GPUs on NVLink with NVSwitch). In contrast, something like an RTX A6000 can only NVLink 2 cards, and L4 has no NVLink at all. So, A100/H100 is the go-to solution for large-scale parallel training across GPUs due to its interconnect capabilities.

In summary:

- Choose A100/H100 for cutting-edge AI training or HPC with large models and datasets, as they offer precision, memory, and cluster scalability.
- Choose L4 or A10 for cost-effective inference and moderate workloads. They excel at throughput per dollar for smaller models.
- Use L40s or A40/A6000 when you have a mix of graphics and compute needs or need lots of GPU memory for a bit lower cost.
- Use A30 if you want a balance for mainstream enterprise tasks or A2 if you just need a little GPU acceleration at minimal power. If doing scientific computing with heavy double precision, stick to the data center and compute GPUs like A100 or the older V100, which are designed for that.

Chapter 06



Calculating GPU Memory Needs

One of the most important considerations in choosing a GPU is ensuring it has enough VRAM (video memory) for your workload. Running out of GPU memory will cause your program to crash or force it to use slow disk swapping. Here's how to estimate GPU memory requirements for various workloads:

- **Deep Learning Model Training**

  The memory needed for training grows with model size, batch size, and training precision. A rough rule-of-thumb from experts is that training typically requires 14–18X the memory of just the model's parameters. This accounts for storing the model weights, gradients, optimizer state (like Adam moments), and intermediate activations during forward/backprop. For example, if your model has 500 million parameters (around 2GB if stored in 16-bit), you might need 30+ GB of GPU memory to train with a reasonable batch size.

	Bytes per parameter
Model Parameters (Weights)	4 bytes per parameter
Adam optimizer (2 states)	+8 bytes per parameter
Gradients	+4 bytes per parameter
Activations and temp memory (variable size)	+8 bytes per parameter (high-end estimate)
Total	**=4 bytes per parameter +20 extra bytes per parameter**

More formally:

- Model weights: each parameter is 4 bytes in FP32 (or 2 bytes in FP16/BF16).
- Gradients: another 4 bytes per param (accumulated in FP32).
- Optimizer state: e.g., Adam uses 8 bytes per param (two 4-byte moment vectors).
- Activations: depends on batch size and network architecture – larger batches or deeper networks produce more intermediate data to store for backprop. This can often equal a few times the model size.

As an example calculation, say a model has 100 million parameters (~0.4 GB in FP16). Training in mixed precision (FP16 weights, FP32 accumulate) might use ~6 bytes/param for weights+grad (per the rule above), plus 4–8 bytes/param for the optimizer, totaling ~10–14 bytes per param. That's ~1–1.4 GB for 100M params. Now add activation memory: if you train with a batch size that produces, say, another 1 GB of activations, the total might be ~2–2.5 GB. This fits in a 4 GB GPU. If the batch or model is bigger, requirements scale up. In practice, monitor GPU usage during training to refine this estimate or consult profiling tools. But the key is that bigger models and batches need exponentially more memory, So, choose a GPU with ample VRAM (or plan to use gradient checkpointing and smaller batches to compensate).

- **AI Inference and Deployment**

  Inference (using a trained model to make predictions) generally needs less memory than training the same model. You typically only need to load the model weights and some buffers for the input and output. A good estimate is to have GPU memory slightly larger than the model size. For example, a 2GB model will fit in a 4GB GPU for inference; however, if you plan to do batch inference (processing multiple inputs at once) or run multiple models concurrently, factor that in.

  Also, using lower precision for inference (such as INT8 or quantized models) can drastically cut memory needs. Many deployment scenarios use 8-bit or 16-bit weights, so a model that was 2GB in training (FP32) might be only 0.5GB in INT8 form. That said, also consider memory for pre-processing – if the GPU is also doing operations on input data (like image scaling or feature extraction), those will need memory. Rule: ensure the GPU memory is at least the model size + some headroom (say 25% extra) for safe measure. If running multiple models or very large batch queries, choose a GPU with more memory or use multiple GPUs (or MIG to split a big GPU for multiple models).

Tip: INT8 and sparsity-aware inference can increase throughput by 2-4X on NVIDIA Tensor Cores.

- **High-Performance Computing (Scientific) Tasks**

  HPC memory needs can vary widely based on the problem size (matrix sizes, grid resolution, and more). Many HPC applications allocate large arrays on the GPU. For instance, a weather simulation might allocate a 3D grid of atmospheric data – if the grid is 1000×1000×1000 and each point stores 8 bytes (double precision), that's 8 GB of data right there. So you would need a GPU with >8 GB just for that array. HPC users should calculate the memory footprint of their datasets or meshes. If an application demands more data than a single GPU's memory, you either need to use multiple GPUs (with domain decomposition) or use a GPU with more memory (like the 80GB A100). Memory bandwidth is also crucial (large data on GPU requires fast memory to be useful), so GPUs with HBM (A100/H100) might handle large scientific data more efficiently than ones with GDDR memory. As an example, a bioinformatics protein folding job like AlphaFold can require different GPU memory depending on input size – one study found that on a 12GB GPU, you could process protein sequences up to about 1000 amino acids, but longer sequences (~2000–3000) needed 32GB or 40GB GPUs. This illustrates that for HPC workloads, problem size (for example, sequence length, mesh detail) directly drives GPU memory requirements, and using a GPU with insufficient memory will simply not run the larger cases. Always check the memory per GPU and scale your problem or choose an instance accordingly.

- **Graphics & Rendering**

  In rendering and graphical workloads, GPU memory is used for things like textures, frame buffers, and geometry data. If you are doing high-resolution rendering or working with very detailed models, you will want a GPU that can fit all the textures and data in VRAM to avoid swapping. For example, a 3D scene with several high-res texture maps, 4096×4096 pixels each, could easily consume a few gigabytes of VRAM. Real-time engines also need memory for frame buffers, which grow with output resolution and anti-aliasing. If you plan on 4K rendering or VR, aim for GPUs with larger memory (16GB and up). Professional rendering tasks (using GPU renderers like Octane or Redshift) often scale well across multiple GPUs; each GPU still needs a copy of the scene data, so if the scene is 8GB in size, each GPU must have >8GB free. Cloud rendering farms typically use GPUs like RTX A6000 or A40 with 48GB to accommodate huge scenes.

Tip: If your scene doesn't fit in one GPU's memory, some rendering software can split across GPUs (or use NVLink to combine GPU memory), but performance might suffer. It's more straightforward to pick a GPU instance with enough VRAM to hold your entire project data.

- **Other Workloads (Video processing, etc.)**

  Video transcoding on GPUs uses some VRAM for buffering frames, but this is usually modest (a few hundred MB per 4K stream, for example). The main constraint is often not memory but encoder/decoder throughput. Still, if you plan to handle dozens of concurrent video streams on one GPU, ensure there is a few GB free for all the frame buffers. In data analytics (GPU-accelerated databases), memory is used to store chunks of the dataset on the GPU for processing – the more memory, the larger the chunk (which can reduce the need for data transfers). So, for analytics on big data, opt for GPUs with high memory so that more of your dataset can reside on the GPU at once.

  In summary, estimate your GPU memory needs by examining the size of your model or data and how it will be processed. Use known rules-of-thumb for your domain, for example, 14–18× model size for DL training or specific guidance from software documentation. Always allow some cushion; also, it is wise not to fill a GPU with 100% of its VRAM; keep maybe 10-20% headroom to avoid OOM errors. If unsure, test on the side of a GPU with more memory or a smaller sample to gauge usage. Cloud providers often allow easy switching to a larger GPU instance if you find memory is insufficient. And if your problem simply cannot fit in one GPU, consider techniques like model parallelism (for AI) or distributed computing (splitting data across GPUs), albeit with added complexity.

CPU Cores and System RAM Requirements for GPU Workloads

Selecting the right cloud GPU instance isn't only about the GPU – you must also ensure the CPU and system RAM paired with the GPU are adequate. In cloud offerings, a "GPU instance" typically includes a certain number of CPU cores and a certain amount of host memory. Here are guidelines for choosing cores and RAM based on workload:

- **AI/ML Workloads (Training and Inference)**

  Surprisingly, training neural networks is often not very CPU-intensive once data is prepared – the GPU does the heavy lifting. A common guideline is to have about 1 to 2 CPU cores per GPU for feeding the GPU and handling housekeeping tasks. This is usually enough to launch GPU kernels and perform any minor CPU-side processing. For example, if you have a server with 4 GPUs, a CPU with 8 cores (2 per GPU) is generally sufficient to keep them busy. Suppose your training involves a lot of data preprocessing, such as image augmentation and loading huge datasets from disks. In that case, you might want a few extra cores to handle those in parallel threads.

  However, more than 4 cores per GPU often yield diminishing returns for pure training. For inference, CPU needs can be even lighter per GPU – a single CPU core can sometimes handle dispatching for many inference calls on one GPU unless you are doing CPU-heavy pre/post-processing for each request. In summary, you don't need an extremely powerful multi-core CPU for GPU-centric ML tasks; prioritize clock speed for feeding data quickly and ensure at least one thread per GPU.

  For system RAM, consider the size of your dataset and any memory-mapped files. Ensure your CPU RAM is sufficient to hold the data being fed to the GPU. If you are training on a dataset that is, say, 100GB, you might not load it all in RAM at once, but you should have enough RAM for caching a portion and for any data structures, for example, a queue of batches. Typically, having host RAM equal to at least the GPU memory is recommended so you can stage data. For instance, if using a GPU with 16GB VRAM, 32GB system RAM can help buffer data and handle the OS and other processes. If multiple GPUs are running on one machine, scale the RAM accordingly. For example, a machine with 4×16GB GPUs might have 128GB RAM. This is not a hard rule, but memory is cheap compared to GPUs, and insufficient RAM can become a bottleneck or cause out-of-memory issues before the GPU even saturates. Also, note that some AI frameworks, such as large training checkpoint buffers or dataloader workers, consume a fair amount of CPU memory for bookkeeping.

- **High-Performance Computing (HPC) Simulations**

  HPC workloads often involve a tight coupling between CPU and GPU. Many HPC applications use a hybrid approach where CPUs handle certain parts of the computation and GPUs accelerate others. CPU core requirements for HPC can be significant: you may run an MPI rank on each CPU core and use GPUs as accelerators. If you have an HPC job that utilizes a GPU, having multiple CPU cores working in parallel can feed the GPU or perform non-accelerated parts of the code. For example, a molecular dynamics simulation might use CPU threads to calculate some interactions while the GPU does others. For HPC in the cloud, consider at least 4–8 CPU cores per GPU and sometimes one full CPU socket (8-16 cores) per GPU for balance – especially if the code isn't 100% offloaded. Many HPC cloud instances pair, say, a single A100 with 32 vCPUs, acknowledging that HPC workloads need strong CPU support.

  HPC system RAM also tends to be large because input datasets or intermediate results can be huge. It is common to see HPC nodes with 256 GB or more RAM when GPUs are involved. For instance, a GPU might hold a subset of data while the CPU holds a larger portion – if the CPU RAM is insufficient, you cannot feed the GPU effectively. If your simulation needs to read big files or store a large mesh, ensure your instance's host memory covers that. A good practice is to match what on-premise HPC clusters do: often, each GPU node has an order of 0.5 – 2TB of RAM shared among CPUs when there are multiple GPUs (this allows very large problems to reside in memory, and GPUs stream chunks as needed). In cloud terms, you might choose a "memory-optimized" GPU instance type if available for HPC workloads that need it.

- **Data Analytics and Big Data with GPUs**

  If you are using GPUs to accelerate analytics like GPU databases, Spark with GPU support, etc., don't skimp on the CPU either. The CPU will still handle parts of the query planning, aggregation, etc., and needs enough cores to keep up, especially if multiple users or threads are querying simultaneously. Aim for a balanced setup, such as a GPU node with 16–32 CPU cores for analytic workloads. Also, ensure the system RAM is much larger than the GPU memory in this scenario because not all data will typically fit on the GPU at once. The RAM can act as a cache or buffer. For example, if you have an 80GB GPU and 256GB of system RAM, a large dataset can be partitioned – the GPU processes one chunk at a time from RAM. If you only had 80GB RAM to match the GPU, you would not be able to stage the next chunk while the GPU works on the current one, leading to stalls.

- **Graphics, Gaming, and VDI**

  These use cases require a balance of CPU and GPU. The GPU renders frames in a cloud gaming scenario, but the CPU is still responsible for running the game engine logic- AI, physics, and game world updates. Modern games can utilize multiple CPU cores – many titles are optimized for 4–8 cores. If you are setting up a cloud gaming server per user, you would allocate a share of CPU cores to that user as well. A reasonable guideline is 4 vCPU (cores) or more per gaming instance to meet the game's CPU needs. Some high-end games might even benefit from 6 or 8 cores if they scale well. The CPU should also be high clock speed, since game performance often depends on per-core performance. Cloud providers sometimes offer GPU instances with strong CPUs, for example, NVIDIA GeForce Now reportedly uses high-clock Intel CPUs with each GPU for gaming.

  For virtual desktop infrastructure (VDI), if a GPU is shared among multiple users, you also need enough CPU for each user's general OS and application needs. Typically, each VDI user might get 2–4 vCPUs. So, if 1 GPU is supporting 4 VDI users, the instance might have 16 vCPUs total.

  For system RAM in graphics/gaming, you need memory for the game or application that is similar to a normal PC. Many modern games recommend 16GB of RAM on a PC for smooth operation. So, a cloud gaming instance should have around 16GB RAM per game instance to be on a safer side in addition to what the OS needs. If multiple users share the machine, allocate accordingly. For example, 4 users might use a machine with 64GB RAM. In professional visualization, if you open very large models or datasets, you might need even more RAM to load the scene before parts are sent to the GPU. But for most AEC/design apps, the GPU's 48GB might be the cap, and having equal or double that in RAM, let's say 96GB ensures you can load everything.

  In summary, match a gaming/graphics cloud instance's CPU/RAM to a high-end workstation: multi-core CPU and lots of memory, so system limitations don't hold the GPU back.

- **Video Transcoding/Streaming**

  GPU video encoding (using NVENC on NVIDIA GPUs) offloads almost all the heavy work to the GPU's fixed-function hardware. The CPU role here is to handle the I/O: reading video files or streams, maybe some minor pre-processing, and mixing the output. You do not need many CPU cores for pure transcoding. Even 2-4 cores can handle multiple 1080p video streams when the encoding is done on GPU. However, if you are running many parallel transcodes, some additional cores help manage multiple threads, such as reading from disk for 10 videos at once and writing results. It is reasonable to have 1 CPU core per 5-10 video streams as a general guideline, depending on bitrate and complexity. So, if a single GPU can encode 20 streams, having at least 2-4 cores is recommended, so the CPU can feed those 20 parallel encoding sessions without lag.

System RAM is typically not a bottleneck for transcoding – each video stream might buffer a few frames in memory. Even for dozens of streams, a few gigabytes total is enough. So usually, if your instance has 16+ GB RAM, it is fine for video tasks; however, if you are also doing something like AI-based video enhancement, it then falls under AI guidelines.

Did You Know? The NVIDIA L4 GPU can handle over 1,000 concurrent AV1 video streams at 720p30 for mobile applications.

- **General Guidance**

Ensure the CPU does not become a bottleneck for your GPU. If you notice your GPU isn't fully utilized, but your CPU usage is 100%, that is a sign you need more CPU power for that task. For example, a complex data pipeline might require more CPU threads to keep the GPU fed. In cloud environments, you can often choose instance types like 1 GPU + 8 vCPU or 1 GPU + 32 vCPU based on your app's known CPU demands. Likewise, system RAM should be enough to comfortably handle the workload's data and overhead. Running out of system RAM can cause swapping to disk, which can cause issues. It is better to have too much RAM than too little.

Many recommend at least RAM equal to 2–4X GPU memory for balanced systems, especially for data-intensive tasks, to accommodate data structures and OS needs. So, if you have a GPU with 16GB, having 32–64GB RAM is healthy for heavy tasks. You can get away with less if the workload is minor or the dataset is small.

In cloud GPU instance specs, these ratios are often predefined, and are marketed as a GPU-optimized instance that might have a set CPU: GPU ratio. If possible, choose an instance flavor with more CPU/RAM than the bare minimum unless you are sure the CPU will not be used much. A well-balanced system will ensure the powerful GPU is fully utilized and not idle while waiting on CPUs or data.

Tip: Monitor GPU utilization metrics via NVIDIA DCGM or cloud-native tools to optimize instance selection.

Conclusion

New GPUs like NVIDIA's Ada and Hopper series bring new capabilities (for example, FP8 precision, AV1 encoding) that might benefit your use case. Regularly revisit the available instance types and consider benchmarking a sample of your workload on a couple of GPU types if possible. A few test runs can empirically reveal which GPU gives you the best performance per cost.

In summary, it is important to understand why and when you want to use a Cloud GPU. The next part is about how and which GPUs are used. With this comprehensive guide, you can better map out your workload and then size the GPU, CPU, and RAM requirements accordingly. Getting a cloud GPU will always be a tradeoff between delivery time and cost. When you choose the speed of delivery of your workload, the cost gets high, while when you choose the cost, you compromise on the speed of delivery of your workload. With the technical understanding from this guide, you will be well-equipped to make those decisions and choose the right cloud GPU for your needs, enabling your applications to run efficiently and ensure scalability in the cloud.

About AceCloud

AceCloud is a leading provider of end-to-end cloud computing solutions to global organizations across industries at scale. It offers a full spectrum of cloud services, including Public Cloud, Application Hosting, AWS Services, Managed Security Services, and Hosted Virtual Desktop Solutions.

AceCloud is a brand of **Real Time Data Services (RTDS)** group of companies - a leading provider of information technology capabilities specializing in Cloud Computing and Cloud Telephony. RTDS Group has an employee base of 600+ across India, the US, and the UK, supporting over 20,000+ customers and IT infrastructure in 10+ data centers spanning the globe. It offers industry-leading technological solutions that help customers streamline their operations and enhance efficiency.

Know more: acecloud.ai/

15+ Years of Experience | **20K+** Business Users | **100+** Awards

References

- https://www.coreweave.com/blog/right-sizing-training-workloads-with-nvidia-a100-and-a40-gpus
- https://acecloud.ai/resources/blog/cloud-gpus-vs-on-premises-gpus/
- https://www.hyperstack.cloud/blog/case-study/how-gpus-impact-cloud-computing
- https://www.nvidia.com/en-in/data-center/l4/
- https://www.nvidia.com/en-in/data-center/l40s/
- https://images.nvidia.com/content/Solutions/data-center/a40/nvidia-a40-datasheet.pdf
- https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf
- https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/quadro-product-literature/proviz-print-nvidia-rtx-a6000-datasheet-us-nvidia-1454980-r9-web%20(1).pdf
- https://www.nvidia.com/en-in/data-center/h100/
- https://www.horizoniq.com/blog/h100-vs-a100-vs-l40s/
- https://huggingface.co/docs/transformers/perf_train_gpu_one#anatomy-of-models-memory
- https://researchcomputing.princeton.edu/support/knowledge-base/memory

Get in Touch

Contact AceCloud Experts today to get a customized quote.

Visit Us

Our Location

809-A, Sector 19, Udyog Vihar, Phase 5, Gurugram - 122016, Haryana

Get Connect

publiccloud@acecloud.ai

sales@acecloud.ai