whitepaper

Software-Defined GPU

Making GPUaaS work for service providers and the mainstream AI hosting market

hosted•ai

Introduction

Traditional GPU virtualization methods do not suit the service provider use case or the needs of end users, and yet they form the basis of most “GPUaaS” offerings today.

As AI becomes mainstream and on-demand GPU resources are required for many applications, a new approach to GPU virtualization is needed.

This short paper explains why, and how the hosted·ai platform transforms GPUaaS for service providers by introducing true software-defined GPU, with the same technical and commercial flexibility as IaaS cloud.

Contents

hosted·ai

Why GPU virtualization ≠ GPUaaS

Traditional GPU virtualization methods can work well in certain scenarios, but do not suit the multi-tenant service provider use case.

They can also be extremely inefficient, with suboptimal utilization, increased hardware, power and cooling costs, and high cost to the end user. These methods are used for most GPU virtualization today, either behind the enterprise firewall or through service providers using traditional IaaS platforms.

Traditional GPU virtualization options

GPU passthrough

An entire GPU is made available exclusively to one user. The user can use the full resources of the GPU.

GPU instancing (MIG)

A GPU is divided into isolated instances. A workload can securely access whatever fraction of GPU VRAM and cores is available in the instance (1/4 or 1/8, for example).

GPU time-slicing

A GPU is made available to multiple workloads by dividing its compute resource into time slices, but without secure memory isolation or contention management.

Example platforms: VMware, OpenStack, Virtuozzo, Proxmox, in-house projects

Example services: CoreWeave, Lambda, Nebius, AWS, GCP

Drawbacks

Performance	Passthrough and instancing guarantee performance for a single user up to the limits of the GPU or instance. Time-slicing runs the risk of contention issues impacting performance - without memory management a single task can consume all available RAM.
Security	Passthrough and instancing offer GPU compute and memory isolation for a single tenant only. Time-slicing has no memory or fault isolation and doesn't suit multi-tenant services.
Utilization	Passthrough and instancing restrict utilization to one GPU per user or one instance per user. Utilization is never 100% over time. For AI inference, average utilization can be as low as 15%; for tuning, 50%, and for training, 70%. Time-slicing can bring utilization to 100% but only by impacting performance for other users.
Scalability	Passthrough and instancing only allow scaling of available resources by purchasing new GPU hardware, creating additional partitions/shares or renting more GPU services from a provider. Time-slicing may afford temporary "scale" in trusted environments by enabling a team to share a GPU, but again, only up to the limit of one GPU.
ROI/commercials	Enterprise ROI may be easy to demonstrate for predictable single-tenant GPU requirements. For service provider GPUaaS based on passthrough/instancing, the high cost, limited scalability and under-utilization means high investment, low margins, and high pricing/long-term commits for end users.

hosted•ai

What does service provider GPUaaS need?

Traditional virtualization approaches can make sense when predictable GPU resource is needed for a predictable time – e.g. for research HPC, enterprise VDI, or long-term AI model training tasks.

They do not make sense for unpredictable as-a-service workloads, such as AI inferencing (hosted agents and bots), short-term provision of GPU for tuning or testing models, or on-demand services like AI site builders, image generation, audio and video generation, or gaming.

And yet, passthrough and instancing form the basis of current "GPUaaS" offerings: with low utilization and high hardware cost, current GPUaaS offerings are expensive to build, deliver low margins for service providers, and force users to pay high per-hour prices or sign long-term commitments to achieve some measure of affordability.

Introducing the Software-Defined GPU

These are the problems that the hosted·ai platform was built to overcome, by introducing the fully software-defined GPU – so that GPU virtualization is as flexible and affordable as CPU virtualization, and GPUaaS has the same characteristics as IaaS from a service provider and end customer viewpoint:

- Secure multi-tenancy
- On-demand GPU resources, not just cards or instances
- Resource sharing across GPUs
- Overcommit/overprovisioning without resource contention
- 100% hardware utilization
- Lower hardware CAPEX
- Lower power/cooling costs
- Lower end user pricing
- Higher service provider margins

Software-Defined GPU requirements

To enable software-defined GPU, the hosted·ai platform introduces multi-tenant vGPU core execution, adaptive scheduling and Intelligent Memory Management for user tasks.

It's a GPU vendor-agnostic implementation that is an integrated part of a hyperconverged GPU/CPU virtualization stack with extremely fast storage and network IO.

Multi-tenancy

Driving efficiency requires a single GPU to execute more than one task from different end users at the same time.

GPU core isolation

Control the amount of processing time or cores that any single task can consume on a given GPU.

GPU memory allocation isolation

Control the amount of GPU and system memory that any single task can consume.

User access isolation and security

Prevent malicious users from stealing data or resources from other users in the same pool.

Efficient task scheduling across GPU pools

Optimize the allocation of tasks to GPU cards to minimize wait time and maximize GPU utilization.

Fast Network IO

Remove network bottlenecks between client ML tools and the GPU task scheduler.

Fast Storage IO

Remove storage bottlenecks for reading data from persistent storage drives into memory .

Normalization of GPU cards

Provide a consistent semantic layer across different GPU cards, and cards from different vendors.

hosted·ai

hosted·ai GPUaaS architecture

The hosted·ai platform brings full virtualization to GPUs, so that service providers can deliver GPUaaS for variable workloads in a secure multi-tenant environment.

Multi-tenant GPUaaS

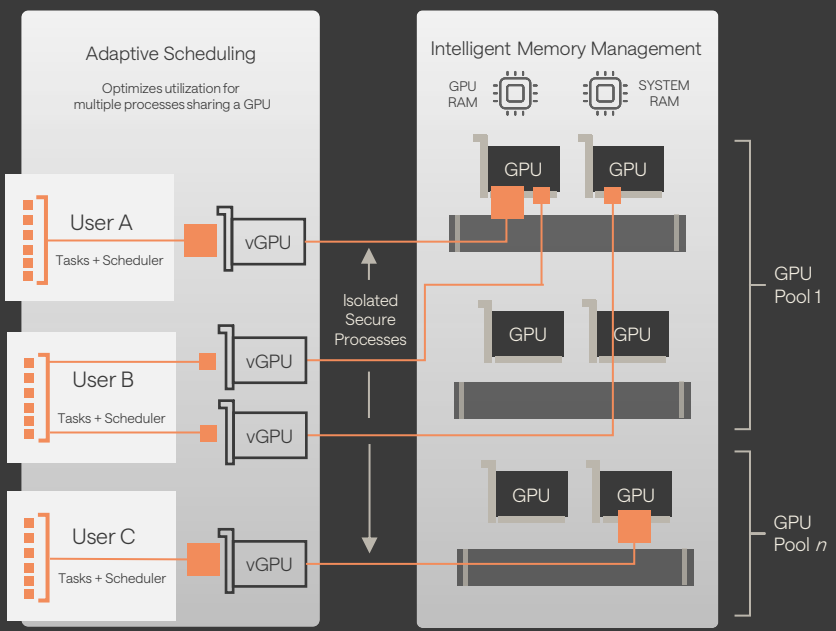
Many users run workloads across a pool of GPUs at the same time, with user tasks isolated and secured from other users.

Users see one or more virtual GPUs, even when their tasks execute on the same physical GPU as other user tasks (the same approach as CPU virtualization).

Instead of allocating entire GPUs or instances to users, GPU resources (TFLOPS/VRAM) are allocated to users from GPU pools.

Users pay for the GPU resources they consume, without requiring static reservations.

The platform uses adaptive scheduling to optimize utilization for multiple processes sharing a GPU.



Scaling

The GPU cloud administrator creates pools of GPUs of any size in the cluster:

- Public pools: any user can subscribe and submit tasks to the pool
- Private/reserved pools: access control is limited to specific users

With adaptive scheduling of tasks, the sum user workload can be scaled to 100% of GPU resource available in pools.

New GPUs can be added to increase the total physical scale available.

Overcommit/overprovisioning

Service providers can enable overcommit of GPUs and provision workloads with theoretical demands greater than the physical infrastructure available (just like vCPU provisioning).

Intelligent memory management and GPU core sharing is handled by the hosted.ai platform.

User tasks are scheduled and prioritized, and system RAM is used to supplement VRAM when required.

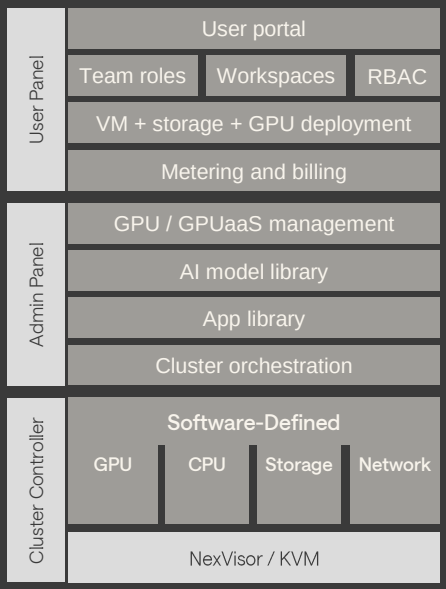
hosted·ai

The hosted·ai hyperconverged platform

These GPUaaS capabilities form an integrated part of a full hyperconverged software platform designed for service provider GPU cloud: a turnkey solution for modern AI workload hosting.

- Software-defined GPU, CPU, storage, networking
- Orchestration tools
- Metering and billing tools
- Application/model libraries
- Self-service user interfaces
- REST API
- Integration with billing engines (e.g. WHMCS)

Of particular importance to GPUaaS performance, hosted·ai has an extremely efficient type 1 hypervisor and a fast, highly resilient SDS and SDN stack, optimizing throughput for intensive GPU workloads.



Redundant and Scalable

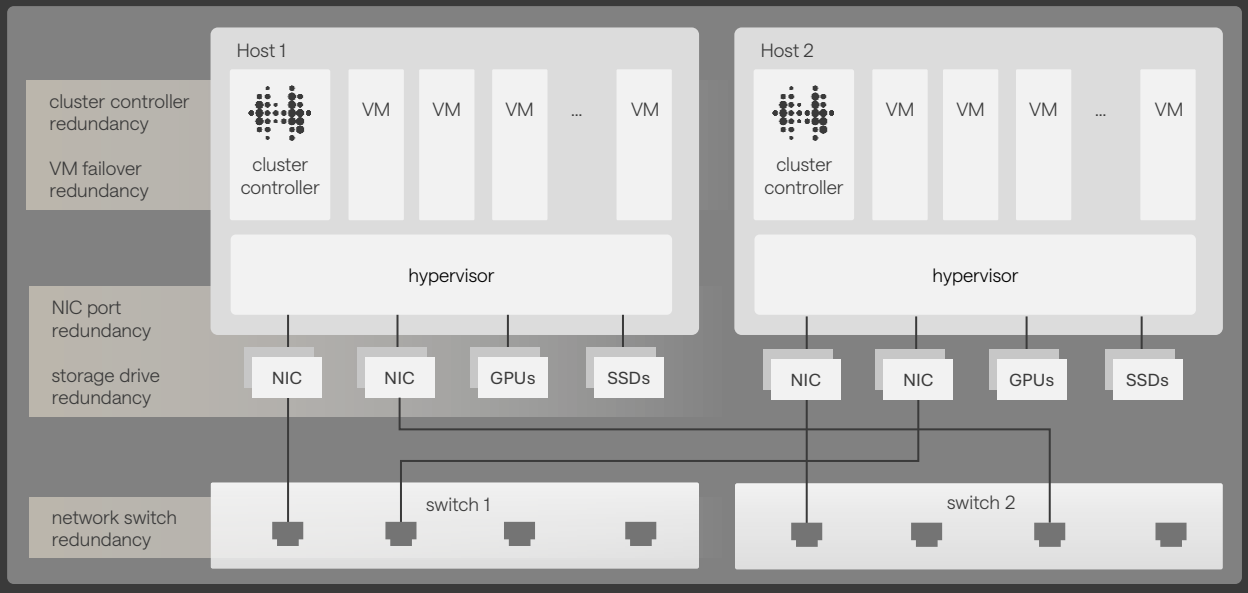
Redundancy at every level of the stack; failover algorithm built-in to the cluster, leverages network heartbeat via multi network paths.

Lightweight

Fully functional cluster deploys on any hardware from Edge to core DC with fully integrated Software Defined Storage, Network, Compute and GPU.

Simple Deployment

Installs on commodity hardware – 24 hours to GPU cloud. Wide range of storage, networking, server and GPU types supported.



hosted·ai

hosted•ai

For a demo, discussion, proof of concept:

https://hosted.ai
hello@hosted.ai