Comparative Analysis of GPU Cloud Providers for AI: Performance, Stability, and Cost-Efficiency

Jihoon Choi, Jehong Bae, Hyeokjin Doo, Young-Woo Kwon
Intelligent Software Systems Lab, Kyungpook National University

Introduction & Objectives

- The GPU Supply Crisis: The explosive growth of LLM training and serving has led to a global shortage of H100/A100 GPUs, creating a bottleneck for academic and cost-sensitive research.
- The DePIN Alternative: Decentralized Physical Infrastructure Networks (DePINs) utilize idle consumer-grade GPUs (e.g., RTX 4090), offering a potential cost reduction of up to 80% compared to AWS/GCP.
- The Challenge: Unlike centralized clouds, DePINs suffer from hardware heterogeneity, unverified stability, and "noisy neighbor" effects, necessitating a rigorous empirical evaluation.

- Vast.ai:
GPU Marketplace
(Direct Rental)

- Akash Network:
Decentralized Protocol
(On-chain)

- RunPod:
GPU Marketplace/
Dedicated Server
(Direct Rental)

- SaladCloud:
Distributed Cloud
(Containerized)

Methodology

To ensure fair comparison across heterogeneous decentralized networks, we developed a portable, automated Python benchmarking suite. As illustrated in Algorithm 1, the framework executes a sequential validation pipeline:

- System Integrity: Validates GPU availability (CUDA), VRAM capacity, and PCIe bandwidth (H2D/D2H) to detect hardware bottlenecks or PCIe x1 lane limitations.
- Storage & Network: Measures sequential/random Disk I/O latency and raw network throughput using standard HTTP libraries and HuggingFace Hub downloads.
- AI Inference (vLLM): Deploys the vLLM engine to measure Tokens Per Second (TPS) and Model Load Time under production-like serving workloads (Algorithm 2).
- Scoring Mechanism: Raw metrics are normalized against data-center baselines (e.g., 1000 MB/s Disk I/O, 80 TFLOPS) to calculate a weighted performance score (0-100).

Algorithm 1 Unified GPU Cloud Benchmarking Workflow

Require: Target Cloud Instance I
Ensure: Benchmark Results R, Score S
InstallDependencies(vLLM, Torch, HF_Hub)
$H_{info} \leftarrow$ MeasureSystemInfo(PCIe Bandwidth, VRAM)
$M_{disk} \leftarrow$ BenchmarkDisk(SeqWrite, RandRead, Concurrency)
$M_{net} \leftarrow$ MeasureNetwork(RawSpeed, ModelDownloadTime)
if GPU is Available **then**
  $M_{infer} \leftarrow$ RunInference(Model, vLLM) ▷ Measure TPS, Load Time
  $M_{train} \leftarrow$ SimulateTraining(GEMM Loop) ▷ Est. TFLOPS
end if
$S \leftarrow$ CalculateWeightedScore($H_{info}, M_{disk}, M_{infer}, M_{train}$)
return $R \leftarrow \{H_{info}, M_{disk}, M_{net}, M_{infer}, M_{train}\}, S$

Algorithm 2 Inference Measurement and Scoring Logic

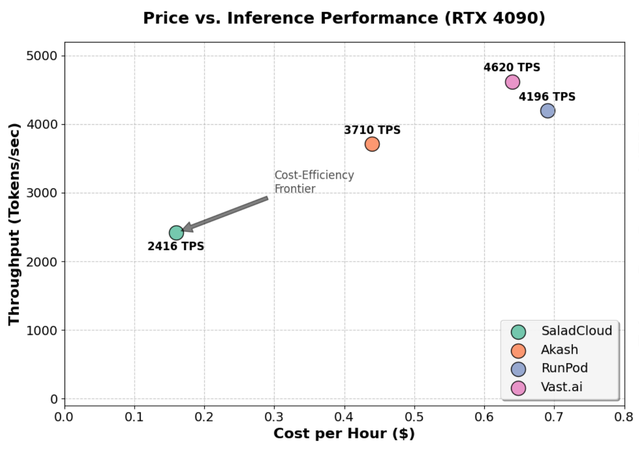
procedure RunInference($Model$)
  $t_{start} \leftarrow$ CurrentTime
  $E \leftarrow$ LoadEngine($Model$, vLLM)
  $T_{load} \leftarrow$ CurrentTime - t_{start}
  $O \leftarrow E$.Generate($Prompts, Params$)
  $N_{tokens} \leftarrow \sum$ CountTokens(O)
  $TPS \leftarrow N_{tokens}/Duration$
  return TPS, T_{load}
end procedure
▷ Score calculation with weighted normalization
procedure CalculateScore(M)
  $S_{train} \leftarrow \min(\frac{M_{flops}}{80.0}, 1.2) \times 50 + \ldots$
  $S_{infer} \leftarrow \min(\frac{M_{tps}}{100.0}, 1.2) \times 50 + \ldots$
  $S_{io} \leftarrow \min(\frac{M_{disk}}{1000.0}, 1.2) \times 40 + \ldots$
  return $\{S_{train}, S_{infer}, S_{io}\}$
end procedure

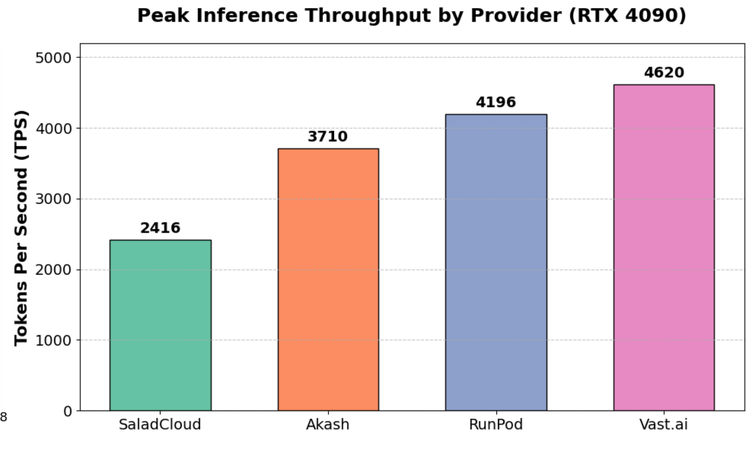
Quantitative Performance Summary

• Quantitative Performance Summary

Metric	Vast.ai	RunPod	Akash Network	SaladCloud
GPU Model	RTX 4090	RTX 4090	RTX 4090	RTX 4090 / 3090
Cost ($/hr)	$0.26 ~ $0.64	~$0.69	~$0.44	$0.16
Peak Inference (TPS)	4,620	4,196	3,710	2,416
Average PCIe Bandwidth	H2D : 18.99 GB/s D2H : 18.62 GB/s	H2D : 22.21 GB/s D2H : 19.14 GB/s	H2D : 16.30 GB/s D2H : 15.47 GB/s	H2D : 24.30 GB/s D2H : 24.42 GB/s
AVG Seq write (MB/s)	725.262	410.694	302.932	644.745
AVG Small write (MB/s)	163.543	29.917	22.524	83.215
AVG Random read (MB/s)	6021.712	72.303	5292.212	10421.02
AVG Small read (MB/s)	838.352	65.6	1060.499	3498.623

* H2D: Host to Device, D2H: Device to Host





• Detailed Provider Analysis

A. RunPod & Vast.ai: The Performance Leaders

- RunPod: Demonstrated exceptional stability and performance with the RTX 4090, achieving 4,196 TPS. Unlike previous trials with other models. However, since runpod utilizes external network drive for workspace, it shows significantly slower disk I/O speed.
- Vast.ai: Achieved the highest theoretical peak (4,620 TPS) on premium instances. However, lower-tier instances ($0.26/hr) frequently failed due to "Noisy Neighbor" effects (VRAM contention), requiring careful filtering.

B. Akash Network: The Decentralized Middle Ground

- Observation: Reliable performance (~3,710 TPS) sits comfortably between the high-end and budget options.
- Limitations: Suffers from significant Disk I/O bottlenecks (Seq Write as low as ~30 MB/s in some nodes) and a complex setup process (YAML), reducing rapid deployment capabilities.

C. SaladCloud: The Cost-Efficiency Champion

- Observation: Unbeatable price ($0.16/hr) with consistent, albeit lower, throughput (TPS ~2,200 - 2,800).
- Trade-off:
  - Provisioning: Long startup times ( )10 mins).
  - Access: No SSH access (Web terminal only), limiting advanced debugging.
  - Hardware: Limited to consumer-grade GPUs (No H100/A100).

Discussion

Operational Insights:

- 1. Stability vs. Performance: While Vast.aioffers the highest peak speeds, RunPod provided a more consistent "out-of-the-box" experience for the RTX 4090 workload without the severe noisy neighbor issues seen in Vast.ai's budget tier.
- 2. IO Variability: Both RunPod and Vast.aishowed variability in Disk I/O depending on the specific host machine, suggesting that applications requiring heavy data loading should verify storage specs (NVMe vs. Network/SATA) pre-deployment.

Conclusion & Recommendations

The choice of provider depends strictly on the deployment phase

1. For Production (Latency-Critical):
   - Recommendation: RunPod or Vast.ai(Premium Tier)
   - Reason: Delivers maximum throughput ( )4,000 TPS) and full SSH control. RunPod offers a slightly more polished UX, while Vast.aioffers potential cost savings if curated carefully.
2. For Development & Batch Processing:
   - Recommendation: SaladCloud
   - Reason: extremely low cost makes it perfect for long-running batch jobs where startup latency and SSH are not critical.
3. For Decentralization & Censorship Resistance:
   - Recommendation: Akash Network
   - Reason: Solid performance with true decentralization, suitable for users who prioritize infrastructure ownership over raw I/O speed.

Intelligent Software Systems Lab
KYUNGPOOK NATIONAL UNIVERSITY
KNU KYUNGPOOK NATIONAL UNIVERSITY

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2021-NR060080).

SCA 2026 Supercomputing Asia
Gathering the Best of HPC in Asia
HPC Asia 2026