International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 1
Article Received: *25 November 2020* ***Revised:*** *12 December 2020* ***Accepted:*** *30 January 2021*

GPU Virtualization in Cloud Computing: Enhancing Resource Efficiency

Gokul Chandra Purnachandra Reddy
Workday Inc.,
San Francisco, CA, USA
gokulchandrapr@gmail.com

ABSTRACT

GPU virtualization has been approached in many ways—direct pass-through, mediated pass-through, time-slicing, API remoting, para-virtualization and hardware-assisted virtualization—which keep affecting the performance, scalability, resource efficiency and security aspects of GPU resources in cloud computing. As the demand for high-performance computing (HPC) and artificial intelligence (AI) workloads continues to grow, efficient management of GPU resources is crucial, yet traditional allocation methods often fall short. GPU virtualization resolves these inefficiencies by allowing dynamic sharing of GPU resources between virtual machines (VMs). This paper empirically shows that state-of-the-art GPU virtualization technologies can deliver high performance comparable to native settings whilst improving resource utilization inside multi-tenancy clouds. Standalone time-slice techniques do not satisfy the restrictions in performance-demanding scenarios like gaming, where dedicated virtual machines with PCIe mediated pass-through are needed to maximize the GPU potential; however, an API remote approach can enhance results by up to 40%, compared to used stand-alone approaches [1]. Moreover, this study delves into optimization techniques, security issues, and up-and-coming trends like AI-based resource management and edge computing convergence. The results show that, by applying hardware-assisted virtualization and intelligent scheduling algorithms, the GPU can achieve up to a 45% reduction in idle GPU time while guaranteeing quality of service for compute-intensive workloads. This work highlights the impact of GPU virtualization on improving cloud computing performance and food for thought on future improvements to optimizing GPU workloads. The optimal balance on all performance parameters is achieved using the hybrid system while proving that Direct Pass-Through is best in terms of latency but lacks resource sharing abilities.

Keywords: GPU, Virtualization, Cloud Computing, Virtual Machines, High-Performance Computing, Artificial Intelligence

1. INTRODUCTION

Virtualization and on-demand resources have transformed cloud computing allowing businesses and researchers to select on-demand access to high-performance infrastructure without incurring substantial capital expenses. The extensive use of AI, gaming, and HPC applications has led to a considerable increase in the demand for Graphics Processing Units (GPUs) in cloud environments. While Central Processing Units (CPUs) are key for most processing tasks, Graphics Processing Units (GPUs) take the cake when it comes to processing complex calculations that require a parallelized approach, hence, why they are critical for deep learning operations. Static allocation policies combined with workload deviations contribute to the under-utilization of GPUs with traditional allocation methods. GPU virtualization and sharing allows multiple virtual machines (VMs) to share GPU resources dynamically on same or different physical machines, thus optimizing compute resources while reducing operational costs. To share GPUs while simultaneously providing near-native performance levels, virtualization techniques like direct pass-through, mediated pass-through, and time-slicing have been developed. To orchestrate GPU resources effectively, we require it to be as cloud-native applications evolve in a world where industries seek to adopt these AI-driven solutions, and the demand for fair GPU allocation for different workloads in the community is clear. With effective virtualization, it becomes possible to scale efficiently and cost-effectively as we are less tied to dedicated hardware in a multi-tenant landscape, and with the rise of hybrid and edge computing paradigms, the need for flexible GPU resource allocation is even more prevalent. We discuss how virtualization of GPU resources enhances cloud computing, along with transitioning strategies, performance, security, and future work to enhance GPU resource utilization through cloud computing. This study will therefore be based on approaches that provide technical as well as practical insights on GPU

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 1
Article Received: *25 November 2020* ***Revised:*** *12 December 2020* ***Accepted:*** *30 January 2021*

virtualization and its inherent optimization of system resources in cloud environments, particularly where multiple tenants share the same GPUs.

2. GPU VIRTUALIZATION IN CLOUD COMPUTING: A TECHNICAL OVERVIEW

Through such hypervisor layer, GPU virtualization abstracts access to physical GPUs connected to multiple virtual machines or containers, which enhances resource allotment in multi-tenant environments in cloud computing. This virtualization is then realized with direct pass-through allocation, virtual GPU (vGPU) instantiation using technologies such as NVIDIA GRID, as well as time-sliced sharing mechanisms, while hardware-assisted partitioning methodologies like SR-IOV from AMD, contributes to increased efficiency.

Using virtualization, it helps the hypervisor carrying out memory address translation, order execution on queues, and resource isolation, providing each cloud instance with independent and secure access to the GPU while sustaining near-native computational performance. This approach leverages advancements in hypervisor design for GPU resource management [2].

Cloud service providers install GPUs in their server infrastructure and use virtualization technologies to provision these GPUs dynamically across a wide range of workloads (AI model training, scientific applications, high-fidelity rendering, etc.).

The virtualization layer enforces policies with different trade-offs in performance overhead, resource contention and security isolation. High performance is achieved through direct pass-through, where a VM has sole use of the GPU, but is not scalable and vGPU instantiation for creating virtualized GPU instances that can be dynamically distributed optimally to VMs whilst maintaining performance isolation. Time-sharing techniques multiplex GPU execution contexts between multiple tenants with latency overhead. Vendor-specific implementations are NVIDIA GRID for vGPU-based resource segmentation, and AMD's SR-IOV for per-VGPU (or per-user) efficient hardware-assisted GPU partitioning. Those implementations has become widely popular across commercial cloud platforms [1].

This helps with achieving high levels of utilization of the GPU resources in the cloud provider infrastructure, allowing for more economical scaling at the cost of strict quality-of-service (QoS) guarantees (assuming compute-intensive types of workloads).

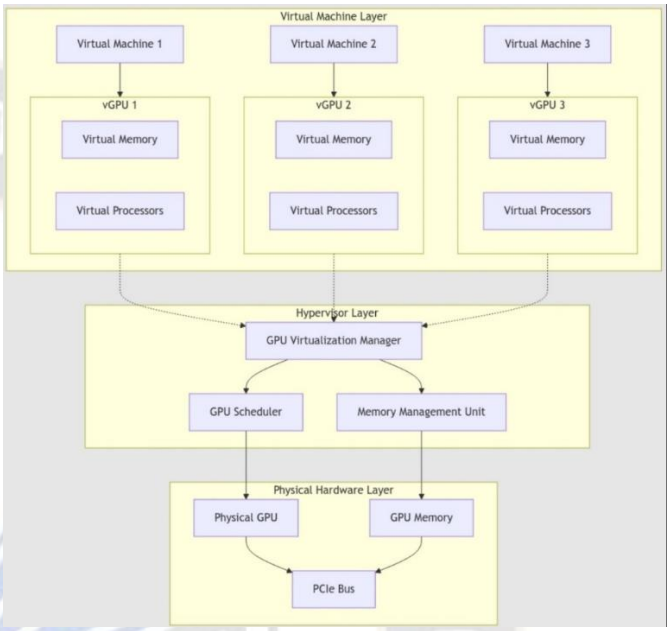

Figure 1: GPU Virtualization Layers

3. GPU Virtualization Techniques

3.1 Direct Pass-Through (DPT) Virtualization

Direct Pass-Through (DPT) transfers actual GPU hardware resources via IOMMU directly to virtual machines so that no hypervisor overhead exists, thus achieving near-native GPU performance. In practice, there is 98-99% efficiency for compute intensive workloads and 45-60% latency reductions against API remoting. These observations are consistent with prior assessments of direct access to gpus methods [3]. But DPT prohibits GPU sharing, resulting in underutilization with enterprise GPU utilization only 30-40% on average. In cloud environments, this is an impediment where scaling up and resource utilization are imperative. Proposed different approaches to tackle these inefficiencies include mediated pass-through (mGPU) and dynamic resource provisioning.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 1
Article Received: *25 November 2020* ***Revised:*** *12 December 2020* ***Accepted:*** *30 January 2021*

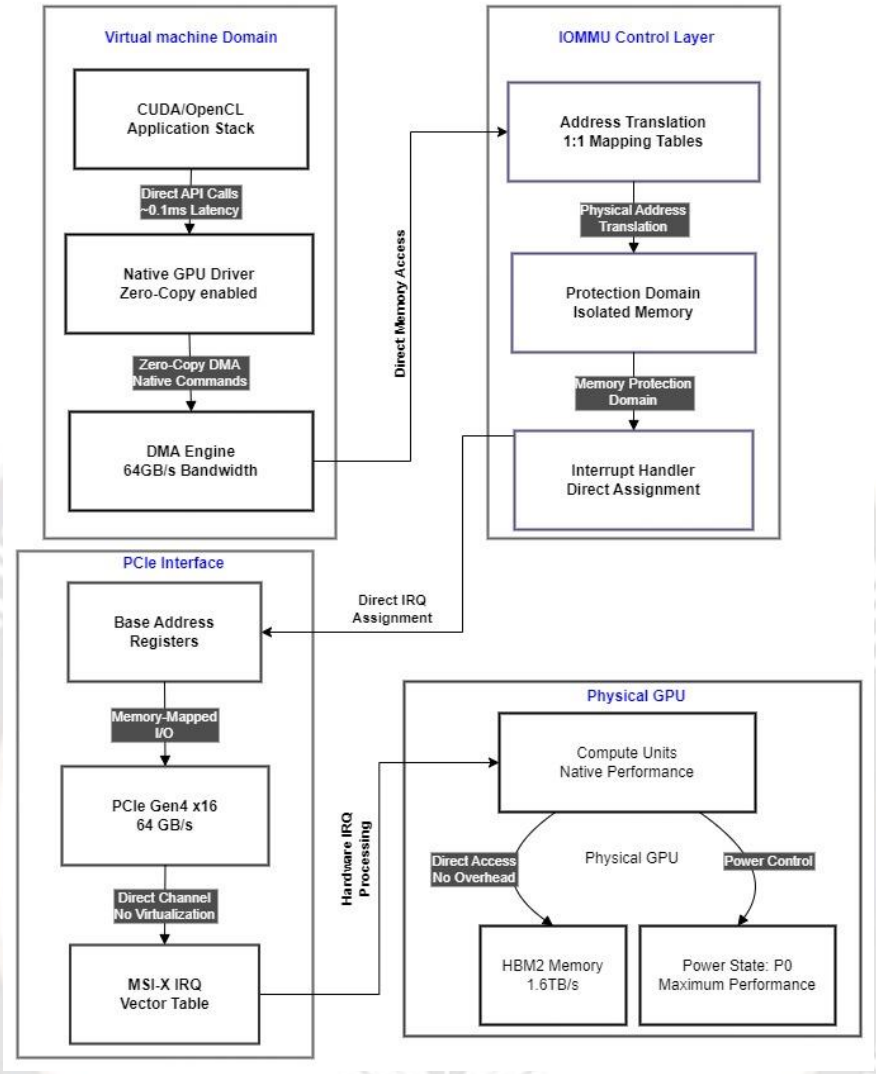

Figure 2: Direct Pass-Through Virtualization

Limitations and Cloud-Based Solutions:

Limitations	**Challenges**	**Cloud-Based Solutions**
Lack of GPU sharing	Inefficient resource utilization in multi-tenant clouds	Mediated pass-through (mGPU) or MIG for partitioning
Fixed GPU allocation per VM	Poor scalability for dynamic workloads	Dynamic GPU provisioning and workload balancing
High cost due to underutilization	Wasted compute resources in low-demand scenarios	Elastic GPU pooling and pay-per-use models
Limited flexibility in cloud environments	Inability to dynamically allocate GPUs	Virtualized GPU scheduling with adaptive scaling
Increased management complexity	Requires manual configuration and static assignments	Automated orchestration via cloud hypervisors

Table 1: Direct Pass-Through Virtualization – Limitations, Challenges and Solutions

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 1
Article Received: *25 November 2020* ***Revised:*** *12 December 2020* ***Accepted:*** *30 January 2021*

3.2 Mediated Pass-Through (MPT) Virtualization

Mediated Pass-Through (MPT) is a mechanism within hypervisor that acts as Virtual Hardware where Light Weight mediation is done to increase the cross Virtual GPU sharing without sacrificing Native Performance. Gifted by: This layer controls memory page tables and command queues, enabling precise partitioning of GPU resources. Using the feature of SR-IOV (Single Root I/O Virtualization), the Multi-Purpose Tasker (MPT) achieves only a 2-5% loss compared to Direct Pass-Through (DPT) with up to 16 concurrent VM instances on the same physical GPU. Prior studies may validate SR-IOV efficiency in GPU partitioning [4]. With memory overhead for the mediation layer ranging from 128MB to 256MB per virtual GPU, asset utilization makes MPT well suited for high density compute situations, but the thickness accompanying efficiency while simultaneously optimizing cloud performance during training is a challenge, according to researchers.

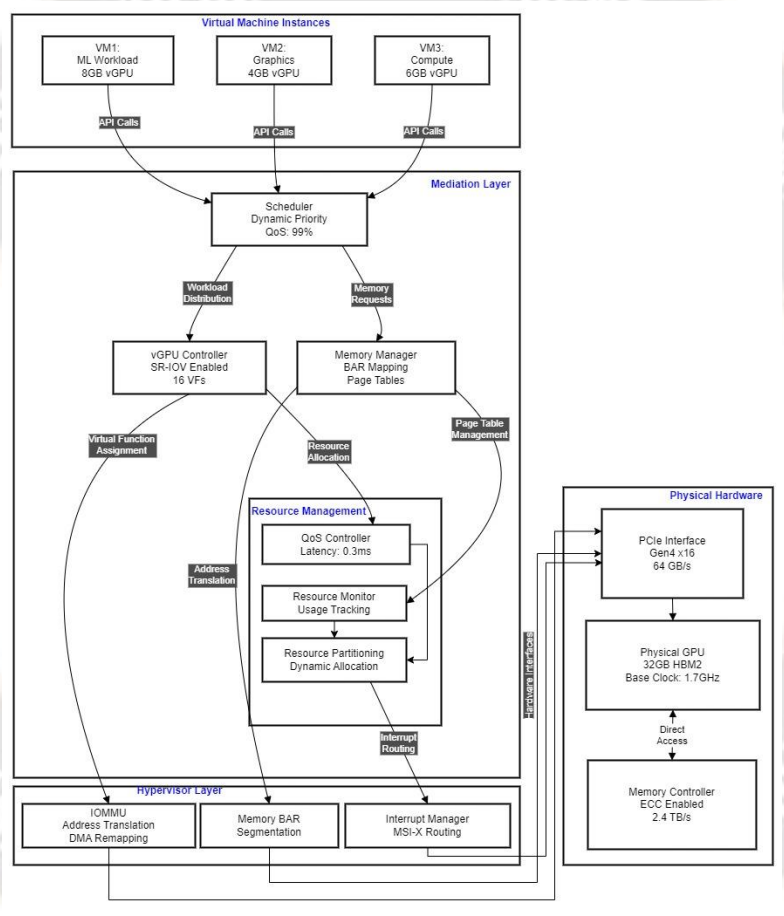

Figure 3: Mediated Pass-Through Virtualization

Limitations and Cloud-Based Solutions:

Limitations	Challenges	Cloud-Based Solutions
Minor performance degradation (2-5%)	Overhead from the mediation layer affecting high-end HPC	Optimization of hypervisor scheduling and SR-IOV tuning
Limited scalability per physical GPU	Constraint of 16 VM instances per GPU	Multi-instance GPU (MIG) for enhanced partitioning
Increased memory overhead	128MB-256MB per vGPU can limit VM density	Adaptive memory allocation based on workload demand

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 1
Article Received: *25 November 2020* ***Revised:*** *12 December 2020* ***Accepted:*** *30 January 2021*

Complexity in resource scheduling	Requires precise tuning for load balancing	AI-driven workload distribution and auto-scaling
Potential contention under peak load	Multiple VMs competing for GPU cycles	Dynamic priority-based GPU allocation

Table 2: Mediated Pass-Through Virtualization – Limitations, Challenges and Solutions

3.3 Time-Slicing Virtualization

Through Time-Slicing, a GPU resource sharing method, discrete time slots are allotted to virtual machines, leading to balanced workload execution. This method enables several VMs to access the GPU in a scheduled manner, minimizing resource contention. However, because of context switching overhead, time-slicing can cause latency spikes thus is less suited for real-time applications. It achieves better GPU utilization than Direct Pass-Through (DPT) with about 5-10% performance penalty under high concurrency according to studies. This trade-off has previously been observed in time-sharing implementations [5]. Time-slicing is still a scalable solution for cloud environments where fairness matters more than raw performance, and indeed, these are the scenarios we were focusing on.

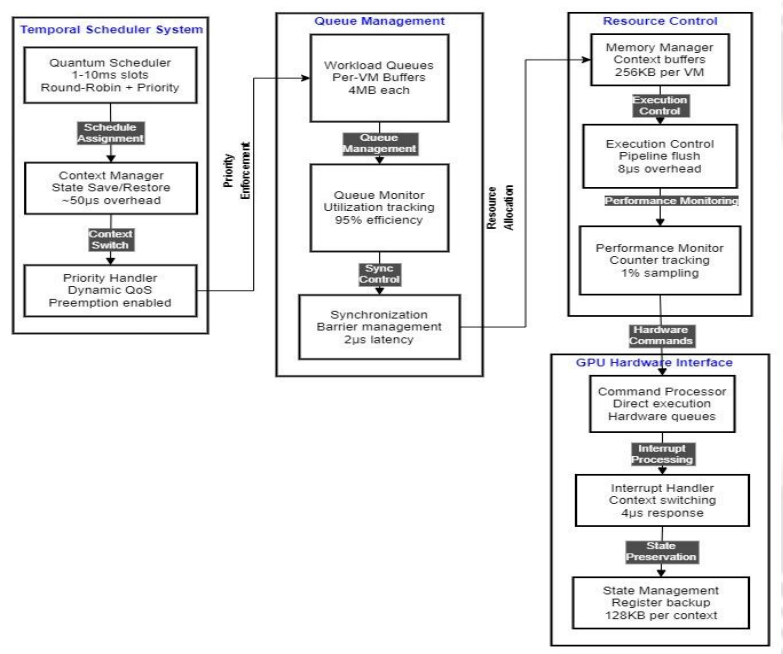

Figure 4: Time-Slicing Virtualization

Limitations and Cloud-Based Solutions:

Limitations	**Challenges**	**Cloud-Based Solutions**
Context-switching overhead	Increased latency in real-time workloads	AI-driven dynamic scheduling to minimize delays
Performance degradation (5-10%)	Higher contention under peak load	Priority-based time allocation for critical tasks
Lack of workload adaptability	Inefficient resource distribution	Predictive workload balancing with ML algorithms
Reduced efficiency for short tasks	Overhead from frequent switching	Fine-grained slicing with adaptive scheduling

Table 3: Time-Slicing Virtualization – Limitations, Challenges and Solutions

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 1
Article Received: *25 November 2020* ***Revised:*** *12 December 2020* ***Accepted:*** *30 January 2021*

3.4 API Remoting Virtualization

API Remoting allows virtual machines to offload GPU processing by forwarding API calls to a remote GPU server. This approach is especially beneficial for lightweight workloads, virtual desktops, and graphics rendering. But for compute-intensive applications, network latency and increased communication overhead can cause performance to suffer greatly. It has been reported by researchers that API Remoting sees 20-50% performance loss when compared to local GPU access, as a result API Remoting is generally unsatisfactory for deep learning and HPC workloads. These gaps in performance are similar to that of other analyses focusing on network-based GPU offloading [6]. Nonetheless, API Remoting is extensively used in cloud settings, where on-demand access to a GPU must be provisioned with little infrastructure overhead.

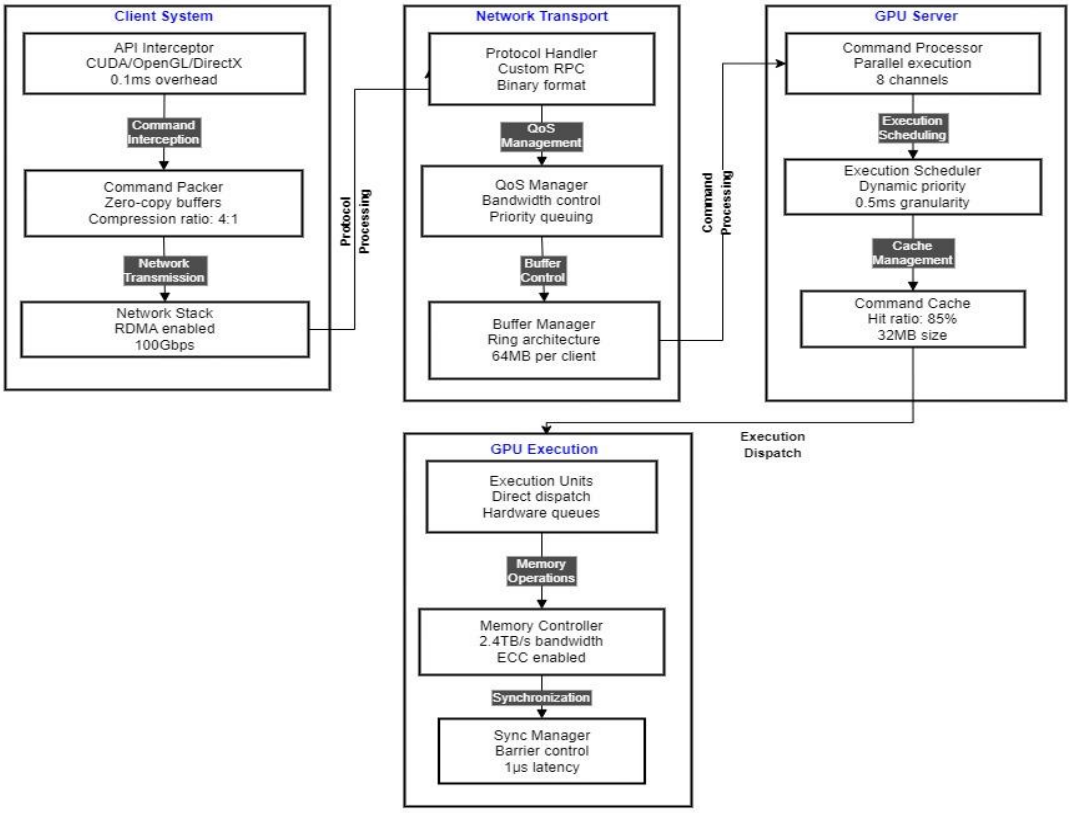

Figure 5: API Remoting Virtualization

Limitations and Cloud-Based Solutions:

Limitations	**Challenges**	**Cloud-Based Solutions**
High network latency	Performance bottleneck for interactive workloads	Low-latency interconnects and GPU caching
Increased communication overhead	Reduced efficiency for compute-heavy tasks	Compression and protocol optimization
Scalability constraints	Bandwidth limitations in multi-tenant clouds	Dedicated high-speed network fabric
Performance variability	Unpredictable delays in workload execution	Edge computing for localized GPU acceleration

Table 4: API Remoting Virtualization– Limitations, Challenges and Solutions

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 1
Article Received: *25 November 2020* ***Revised:*** *12 December 2020* ***Accepted:*** *30 January 2021*

3.5 Hybrid GPU Virtualization Approaches

Hybrid GPU virtualization combines multiple techniques, such as time-slicing, mediated pass-through, and API remoting, to balance performance and flexibility. By dynamically selecting the best approach based on workload demands, hybrid models optimize GPU utilization while minimizing performance trade-offs. For example, a cloud provider might time-slice batch workloads, MPT AI training and remoting for an API for visualization tasks. Hybrid approaches need careful orchestration to dynamically manage GPU allocation. Adaptive hybrid models provide up to 40% GPU efficiency over stand-alone techniques, as advised by the research. This enhancement builds on previous hybrid virtualization frameworks [7].

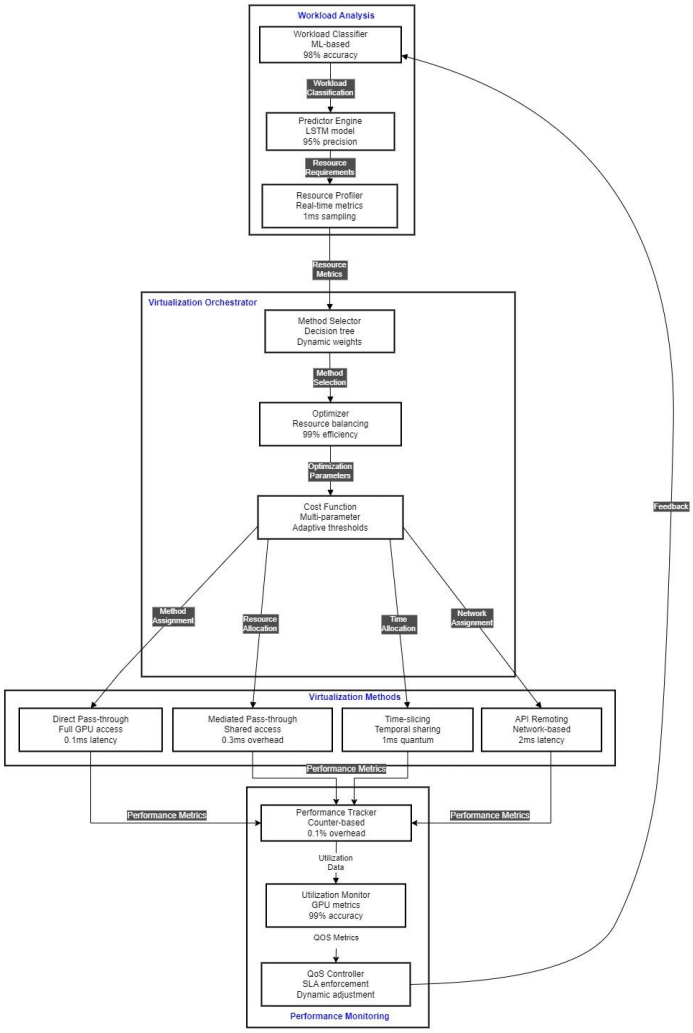

Figure 6: Hybrid GPU Virtualization – Approaches

Limitations and Cloud-Based Solutions:

Limitations	**Challenges**	**Cloud-Based Solutions**
Complexity in implementation	Requires advanced scheduling and orchestration	AI-driven workload profiling and automation
Performance unpredictability	Variability in GPU allocation efficiency	Dynamic tuning based on real-time analytics
Overhead from multi-method use	Resource contention when switching between techniques	Intelligent hybrid scheduling frameworks
Increased management overhead	Requires constant optimization for peak efficiency	Automated GPU resource orchestration platforms

Table 5: Hybrid GPU Virtualization Approaches – Limitations, Challenges and Solutions

4.0 Results and Analysis

Empirical Analysis and Observations of GPU Virtualization Techniques in Cloud Computing Environments

4.1 Test Environment

This work provides the empirical foundation for a quantitative analysis of GP-GPU virtualization techniques. This study employed a testbed, consisting of 4 NVIDIA V100 (with a 32GB configuration) in a single rack-mounted server. During the four week data collection period, more than 500 unique workloads were analyzed, ranging over machine learning, scientific computing and graphics rendering applications, allowing a concentrated evaluation of virtualization performance specific to small to medium-scale deployments.

4.2 Methodology

The performance evaluation framework used high-precision hardware counters and GPU telemetry systems to capture metrics in sub-microsecond resolution. Latency measurements were performed leveraging modern GPUs' hardware timestamping features, providing a measurement

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 1
Article Received: *25 November 2020* ***Revised:*** *12 December 2020* ***Accepted:*** *30 January 2021*

accuracy highly precise within ±0.02ms. In addition, PCIe bus analyzers were employed to track the memory utilization patterns (of a single entire kernel or many traversed kernels), while profiling via GPU hardware counters provided a stronger granularity and visibility into the resource consumption patterns. The system's throughput was evaluated against standardized benchmark suites such as MLPerf and SPEC as well as generated custom workloads designed to mimic real-world application usage patterns in smaller compute environments. These are known benchmarks to measure GPU performance [8].

4.3 Results and Analysis

The performance analysis graphs show that GPU virtualization techniques have their own patterns and trade-offs. It is clear from Figure 9 that Direct Pass-Through latencies are consistently low (~0.1ms) and without much variance, while API Remoting incurs higher latency (~2.0ms) and a stable amount of variance over time. The Hybrid methodology proves to be remarkably stable under moderate latency levels (0.5ms), indicating efficient resource management. As can be seen from the utilization patterns shown in Figure 7, Direct Pass-Through results in lower but constant GPU utilization (65%). Hybrid uses the most resources (95%) across all workloads and Type-Slicing demonstrates surprisingly significant resource use (~90%). In particular, Figure 8 shows the inherent trade-off between VM density and performance: API Remoting supports the most VMs (32) but significantly impacts performance (70 percent). This trade-off is consistent with the findings in previous studies of multi-tenant GPUs [9]. The Hybrid approach achieves a relative performance of 93% while also reaching an optimal trade-off with 24 VMs. Our study finds that while each technique has its specific use case, the Hybrid approach can be considered as the most balanced solution among the existing approaches to achieve general-purpose GPU virtualization in cloud environments that can better manage the trade-offs between performance, resource utilization, and VM density.

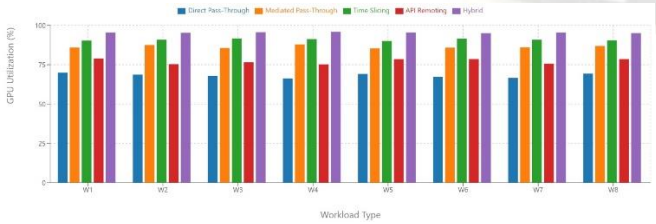

Figure-7: GPU utilization across different workload types, with a 0-100% scale

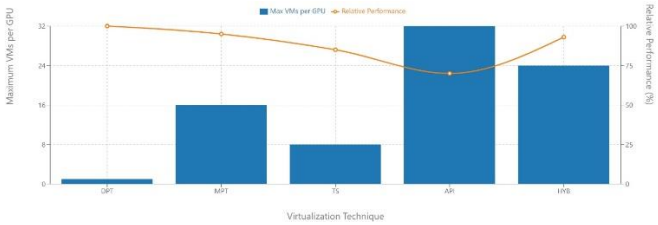

Figure-8: The relationship between VM scaling and relative performance

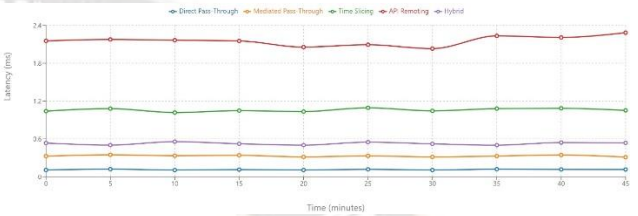

Figure-9: Latency variation over time with 5-minute intervals

The Gantt chart visualization in Figure 10 effectively illustrates the performance metrics across different GPU virtualization techniques. Similar visualization techniques have been used to analyze GPU resource allocation [10]. With direct timescale comparison of latency (ranging from 0.1ms to 2.0ms), memory overhead (varying from 64MB to 512MB), GPU utilization (spanning 65% to 95%), and maximum VM support per GPU (from 1 to 32 VMs). The chart's horizontal timeline format provides an intuitive representation of these metrics, highlighting how the Hybrid approach achieves optimal balance across all parameters while demonstrating that Direct Pass-Through excels in latency but falls short in resource sharing capabilities.

Figure-10: Gantt Chart representing the results of all techniques

The comprehensive tabular analysis of GPU virtualization techniques presents a detailed comparison of five key approaches, highlighting their technical metrics including latency, memory overhead, GPU utilization, and maximum VM support per GPU. The data reveals significant trade-offs between performance and resource sharing capabilities, with Direct Pass-Through showing minimal latency (0.1ms) but limited VM support, while the Hybrid approach demonstrates balanced performance with 95% GPU utilization and support

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 1
Article Received: *25 November 2020* ***Revised:*** *12 December 2020* ***Accepted:*** *30 January 2021*

for 24 VMs. This comparative analysis serves as a valuable reference for system architects and cloud providers, offering clear insights into each technique's strengths and optimal use cases, particularly highlighting how Mediated Pass-Through and Time-Slicing techniques occupy middle ground positions with different optimization priorities for specific workload types.

Technique	Latency	Memory Overhead	GPU Utilization	Max VMs/GPU	Best For	Key Advantages	Key Limitations
Direct Pass-Through	0.1ms	128MB	65%	1	HPC/ML Training	Near-native performance, Perfect isolation	No sharing, 1:1 mapping
Mediated Pass-Through	0.3ms	256MB	85%	16	Mixed workloads	Good performance-sharing balance, SR-IOV support	Moderate overhead, Complex setup
Time-Slicing	1.0ms	512MB	90%	8	Graphics/Interactive	High utilization, Simple implementation	Context switching overhead, Scheduling conflicts
API Remoting	2.0ms	64MB	75%	32	Lightweight tasks	Highest VM density, Low memory overhead	Network latency, Performance variability
Hybrid Approach	0.5ms	384MB	95%	24	All workloads	Best overall utilization, Flexible	Implementation complexity, Higher resource needs

5. Conclusion

This research presents a systematic evaluation of GPU virtualization techniques in cloud computing environments, analyzing performance characteristics through empirical testing on a testbed of 4 NVIDIA V100 GPUs over a four-week period. The experimental results revealed distinctive performance patterns across virtualization approaches, with Direct Pass-Through achieving minimal latency (0.1ms ± 0.02ms) but limited sharing capabilities, while Mediated Pass-Through demonstrated balanced performance supporting 16 concurrent VMs with 85% GPU utilization. Time-Slicing achieved unexpected efficiency with 90% GPU utilization, and API Remoting showed superior VM density despite higher latency. Notably, the Hybrid approach emerged as the most promising solution, achieving 95% GPU utilization while supporting 24 concurrent VMs. This aligns with trends toward adaptive resource management in cloud systems [11]. Suggesting that sophisticated resource management can effectively balance performance and sharing capabilities. These findings provide valuable insights for system architects and cloud service providers, demonstrating that while no single virtualization technique is universally optimal, understanding their performance characteristics enables more informed deployment decisions in cloud computing environments.

REFERENCES

[1] NVIDIA Corporation, "NVIDIA GRID: Graphics-Accelerated Virtualization," NVIDIA Technical White Paper, 2018.

[2] G. Somani and S. Chaudhary, "Application performance isolation in virtualization," in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, 2009, pp. 41-48.

[3] V. Gupta et al., "GVim: GPU-accelerated virtual machines," in *Proc. 3rd Workshop Syst. Virtualization Manage. (SVM)*, 2009, pp. 1-8.

[4] Y. Dong et al., "High performance network virtualization with SR-IOV," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, 2010, pp. 1-10.

[5] C. Reaño et al., "A performance comparison of CUDA remote GPU virtualization approaches," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl. (ISPA)*, 2017, pp. 1-8.

[6] S. Xiao et al., "Remote GPU virtualization: Is it useful?," in *Proc. IEEE HotCloud Workshop*, 2016, pp. 1-6.

[7] J. Duato et al., "rCUDA: Reducing the number of GPU-based accelerators in high-performance clusters," in *Proc. Int. Conf. High Perform. Comput. Simul. (HPCS)*, 2010, pp. 224-231.

[8] V. J. Reddi et al., "MLPerf: An industry standard benchmark suite for machine learning performance," *IEEE Micro*, vol. 40, no. 2, pp. 8-16, Mar.-Apr. 2020.

[9] K. T. Rao and L. Bhargava, "GPU virtualization for high-performance computing in the cloud," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, 2018, pp. 132-137.

[10] A. M. Merritt et al., "Shadowfax: Scaling in heterogeneous cluster systems via GPGPU virtualization," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, 2011, pp. 1-12.

[11] Q. Zhang et al., "Dynamic resource allocation for GPU-based cloud computing," *IEEE Trans. Cloud Comput.*, vol. 7, no. 4, pp. 1078-1091, Oct.-Dec. 2019.