





Balancing Performance and Power Efficiency in Modern GPU Architecture

Shivakumar Udkar¹, Muthukumaran Vaithianathan², Manjunath Reddy³ and Vikas Gupta⁴

¹*Design Engineering, AMD Inc., Colorado, U.S.A.*

²*Samsung Semiconductor Inc., San Diego, U.S.A.*

³*Qualcomm Inc., San Diego, U.S.A.*

⁴*System Design, AMD Inc., Texas, U.S.A.*

Keywords: GPU Architecture, Performance Optimization, Real-Time Workload Analysis, Dynamic Resource Allocation, Thermal Management.

Abstract: This study introduces a novel adaptive performance-power management system that has the potential to improve the efficiency and performance of current GPU systems. Conventional methods of managing these factors frequently fail due to their inability to adjust to changing demands. By utilizing operational characteristics and GPU resources, the proposed solution overcomes this constraint by analysing duties in real-time. The framework has the potential to enhance performance in high-demand situations and decrease power consumption in less demanding duties because of its real-time adaptability. The experimental evaluations indicate that the framework outperforms conventional methods by up to 15% while consuming 20% less power. The framework's ability to manage GPU architectures is illustrated by the results, which contribute to improved power efficiency without compromising performance.

1 INTRODUCTION

GPUs have reached previously imagined performance, meeting computers' rising needs. The importance of graphics processing units (GPUs) in data analytics, AI, VR, and gaming makes balancing performance and power economy more important than ever. Modern GPUs efficiently do complicate and simultaneous calculations, but they may also use a lot of power. Research shows that HPC systems with new deep learning applications need unique architectural alterations to maintain equilibrium (Ibrahim, Nguyen, et al. , 2021). Controlling power usage while doing intensive tasks is difficult. Optimization strategies for traditional GPU power-performance control are frequently too coarse-grained or static to meet current workloads' dynamic demands. These solutions are dominated by fixed operational factors clock rates and allocations making it difficult to dynamically and programmatically handle GPU-intensive applications' complicated computing needs. This stiffness reduces performance and battery efficiency, particularly for dynamic workloads. GPU performance and power consumption optimization often uses static or coarse-

grained techniques. According to studies, GPU performance and power efficiency depend on interconnects like PCIe and NVLink (Li, Song, et al. , 2020). Static approaches that rely on GPU factors like clock rates and core allocations struggle to meet different processing needs. Furthermore, efficient connection networks regulate power and performance, especially in deep neural network accelerators (Nabavinejad, Baharloo, et al. , 2020). When coarse-grained solutions use general power management tactics that do not account for duty-specific factors, they may perform poorly and waste power. Performance and power metrics have been improved to efficiently handle huge datasets using quick GPU interconnects (Lutz, Breß, et al. , 2020) Modern innovations like DVFS and power gating are more flexible. They adjust operational settings for the task. In streaming multiprocessor allocation, power-aware approaches have improved GPU performance and reduced power usage (Tasoulas, Anagnostopoulos, et al. , 2019). These strategies fail to balance power efficiency and performance because they use set criteria or infrequent tweaks. Python can improve GPU compute speed and energy efficiency, but researchers have found it difficult to make the

system user-friendly (Holm, Brodtkorb, et al. , 2020). Complex and diverse computer tasks need sophisticated and adaptable management systems. Research comparing AI accelerators emphasizes the need for enhanced management systems to balance performance and processing capacity (Wang, et al. , 2020). Real-time load analysis to create an adaptive performance-power management framework is a revolutionary solution. GPU design and programming, especially in distributed systems, are difficult, and the performance-power trade-off is complicated (Cheramangalath, Nasre, et al. , 2020). This framework may dynamically adjust GPU resources and operating factors to meet current needs by evaluating GPU responsibilities. According to GPU processing capability and performance models, dynamic management may boost efficiency (Payvar, Pelcat, et al. , 2021). Real-time adaptation allows accurate power economy and performance optimization, compensating for static and coarse-grained approaches. Parallelism-aware microbenchmarks may separate GPU architecture components to better align adaptive approaches with hardware (Stigt, Swatman, et al. , 2022). Workload analysis is crucial to the framework. It tracks computational intensity, memory access, and parallelism. An adaptive resource management component uses this data to dynamically adjust the GPU's CPU cores, memory bandwidth, and clock rates based on job attributes. A power efficiency optimization module optimizes operating settings to decrease power usage without affecting performance. Early studies show that this technique outperforms state-of-the-art technologies while using less power. Dynamically aligning GPU resources with task needs may enhance computational performance and minimize energy consumption, meeting the increasing need for effective GPU management in modern computing environments.

2 LITERATURE REVIEW

Wang et al (Wang, Karimi, et al. , 2021) This study introduces sBEET, a scheduling paradigm for real-time GPUs that employs spatial multiplexing to improve efficiency without sacrificing performance. It utilizes GPU benchmarks and actual hardware to demonstrate that it is more efficient and schedulable than existing techniques, and it reduces energy consumption and deadline violations while making scheduling decisions in runtime. Busato et al (Busato, and, Bombieri, 2017) The proposed research examines a variety of GPU workload division

techniques, such as static, dynamic, and semi-dynamic methods, with a focus on energy efficiency, power consumption, and performance. It illustrates the influence of different strategies on overall efficiency in a variety of processing contexts by conducting testing on both regular and irregular datasets on desktop GPUs and low-power embedded devices. Shenoy et al (Shenoy, 2024) In this proposed research, this investigates the efficacy and power consumption of numerous GPU architectures, such as Fermi, Kepler, Pascal, Turing, and Volta. It emphasizes that while Volta provides the most optimal performance in most scenarios, Pascal is superior in certain applications due to its superior memory-level parallelism (MLP). The study indicates that the efficacy of graphics processing units (GPUs) from newer iterations is not always superior. This is attributable to the complexity of the factors that influence GPU efficacy. Foster et al (Foster, Taneja, et al. , 2023). By profiling ML benchmarks, the proposed research assesses the performance and power utilization of Nvidia's Volta and Ampere GPU architectures. The study examines the relationship between system performance and power efficiency and hyperparameters such as batch size and GPU count. The study illustrates that the PCIe communication overhead reduces the advantage of Ampere's 3.16x higher energy efficiency in comparison to Volta when scaled across multiple GPUs. Arafa et al (Arafa, Badawy, et al. , 2019) PPT-GPU, a simulation system that is both accurate and scalable, is introduced in the proposed work. It is designed to determine the performance of GPU applications across a variety of architectures. Performance Prediction Toolkit (PPT) has been enhanced by the inclusion of models for GPU memory hierarchies and instruction latencies. PPT-GPU demonstrates its utility to developers and architects by producing predictions within 10% accuracy, outperforming actual devices and GPGPU-Sim by a factor of up to 450.

3 PROPOSED WORK

3.1 System Architecture

The System Architecture Overview describes the design of the adaptive performance-power management system that was developed for the current GPU architectures as well as outlines an overview of its crucial components. Modular parts that make up this system work together to provide a happy middle ground between performance and

power consumption. Core modules of architecture include Adaptive Resource Allocation, Power Efficiency Optimization, and Real-Time Workload Monitoring. Fig 1 depicts the system architecture diagram. Fig. 1. System Architecture Diagram To ensure that the system is able to respond to shifting workload requirements, each module has a different yet dependent role. In an iterative process, the Real-Time Workload Monitoring Module captures and analyzes information about the computational load, memory access patterns, and parallelism needs of incoming tasks. The basis of the adaptive decision-making process in the system is the real-time and accurate insights of this module about the particular demands on the GPU. In response to this analysis of workload, the Adaptive Resource Allocation Module adjusts the resources of the GPU: core usage, memory bandwidth, and processing speed, keeping in mind the requirements of the jobs at hand. This is possible due to real-time allocation or throttling of GPU resources, which keeps up the power consumption without any losses in performance. Lastly, the Power Efficiency Optimization Module optimizes power consumption based on dynamic voltage and frequency adjustment. Alterations of these factors are made to deliver effective power reduction without any form of degradation in performance; this is through collaboration with the Adaptive Resource Allocation

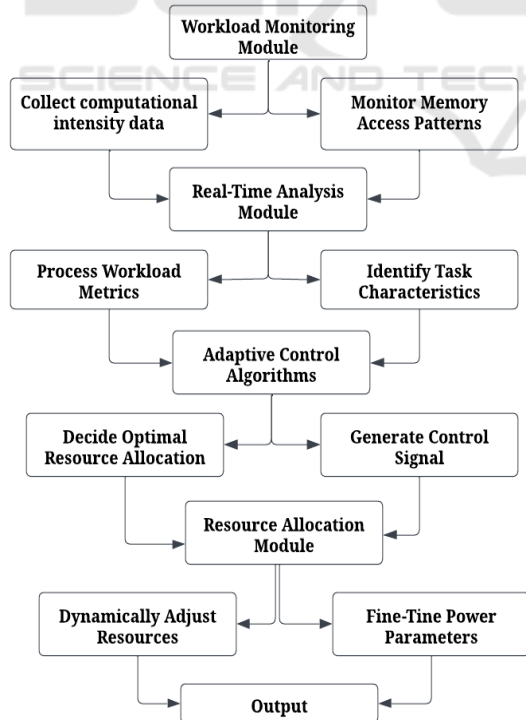


Figure 1: System Architecture

Module in using information from the workload monitoring system under different levels of load intensity for reduction without loss in performance.

3.2 Real-Time Workload Monitoring & Analysis

The key to an effective adaptive performance-power management system in GPU architecture is the real-time analysis and monitoring of workload. Dynamic adjustments in resource allocation and operational parameters are informed by the continuous collection and analysis of precise information on GPU workload characteristics. A state-of-the-art workload monitoring system is the foundation of this approach, as it captures a myriad of performance indicators in real-time, thereby providing a comprehensive understanding of how the GPU manages a variety of calculations. The monitoring system commences monitoring the main parameters of memory bandwidth usage, parallelism requirements, and intense computation. The computational intensity of a computer, or the quantity of computing capacity necessary to complete a task, can fluctuate significantly among different duties. The rate at which data is read or written to memory is measured as memory bandwidth utilization, which aids in comprehending the impact of memory access patterns on overall performance. It is imperative to ascertain the GPU's capacity to leverage its parallel processing capabilities by determining the efficiency with which the task can be distributed across multiple processor cores, a concept referred to as parallelism requirements. The system employs sensors and high-resolution performance monitors that are incorporated into the GPU's design to ensure precise and comprehensive monitoring. These components enable the precise examination of workload characteristics by collecting real-time data on core use, clock rates, and memory access. Complex algorithms may be employed to analyze this data to determine the GPU's performance under various circumstances. Patterns and trends are then employed to illustrate the results. A component of this monitoring procedure is the ability to promptly respond to fluctuating duties. By revising its analysis in real-time in response to changes in the GPU's state, the system adapts to variations in duty intensity and resource requirements. For instance, the system may inform you that an increase in computational intensity necessitates the addition of additional processor cores or higher frequency rates. In contrast, the system may decrease power consumption and reallocate resources as needed as the burden becomes lessened. The GPU

is able to more effectively optimize power efficiency and performance by incorporating real-time workload monitoring and adaptive resource management algorithms.

3.3 Dynamic Resource Allocation Strategies

To optimize performance and efficiency, it is essential for current GPU architectures to implement dynamic resource allocation. This approach makes real-time adjustments to the GPU's clock rates, memory bandwidth, and number of processing cores in accordance with the requirements of the task. Dynamic resource allocation maintains the GPU's optimal performance while simultaneously reducing power consumption by utilizing its adaptive response to fluctuations in workload intensity. Dynamic resource allocation employs properties of the burden in real time to determine the most effective approach to resource modification. This approach effectively regulates computation demands by increasing frequency rates and allocating additional processor cores when a task is determined to be particularly intensive. This strategy improves task execution efficiency and decreases the probability of performance bottlenecks by guaranteeing that the GPU can sustain high performance levels. Conversely, the approach prioritizes the reduction of resource allocation to conserve energy during periods of low job intensity. This results in a substantial reduction in power consumption without compromising performance by reducing the number of active processing cores and clock frequencies. The GPU's decreased resource utilization in response to decreased workload demands could result in significant power savings and more energy-efficient operation. This resource allocation method is dynamic in nature, as it employs real-time feedback mechanisms to continuously monitor GPU performance metrics. Clock rates, memory access patterns, and core utilization are recorded by sensors and high-resolution performance counters, which provide a comprehensive understanding of the GPU's operational status. This data is utilized by the framework to ascertain whether adjustments are required to make informed decisions regarding the efficient distribution of resources. Dynamic resource allocation has the potential to enhance both performance and power efficiency simultaneously. The method ensures that GPU performance is maintained at its maximum efficiency by dynamically adjusting GPU resources in real-time in response to workload demands. Resources are utilized to their

maximum potential for performance-critical tasks and are not over-provisioned during less demanding tasks because of this adaptability. Complicated algorithms are employed to determine the optimal configuration of GPU resources to facilitate dynamic resource allocation. The algorithms' toolboxes encompass considerations for memory bandwidth, duty intensity, and parallelism requirements. The method continuously modifies these configurations to achieve a balance between power efficiency and performance as various workloads evolve over time.

3.4 Adaptive Performance Optimization Techniques

The adaptive performance optimization approach offers a high-level method for controlling GPU performance by perpetually modifying operational parameters in response to the characteristics of real-time workloads. By dynamically adjusting GPU parameters to accommodate varying demands, a balance is achieved between processing performance and power efficiency. The primary objective of adaptive performance optimization is to optimize efficacy while simultaneously minimizing power consumption. This is accomplished through the implementation of modifications that are derived from real-time data. The GPU's status is monitored in real-time by sensors and high-resolution performance counters, which initiate the procedure. These tools capture critical data, such as execution unit activity, memory bandwidth, and core consumption, with exceptional precision. By analysing this data for trends and fluctuations in the intensity of effort, the system can optimize its performance. Voltage levels and clock rates are dynamically adjusted during adaptive performance optimization. In response to a challenging undertaking, the method may modify voltage levels and/or increase clock rates. This illustrates the GPU's capacity to efficiently manage demanding duties. This method enhances power efficiency by decreasing power consumption during less demanding duties by reducing voltage levels and clock rates. One of the primary features of this system is its ability to adjust to altering burden conditions in real time. The system promptly modifies the GPU's operational parameters to accommodate workloads that vary in computational intensity. The GPU's real-time flexibility enables it to maintain its optimal performance range and prevent unnecessary power consumption. Prediction methods are also employed in adaptive performance optimization. These algorithms may analyze historical data and current trends to anticipate the evolution of duties. This type

of algorithm has the potential to enhance performance and power efficiency by adjusting parameters to account for fluctuations in the workload. For example, the system could anticipate an increase in task intensity by increasing voltage and clock rates. Another frequent component of this methodology is the management of thermal constraints. The system monitors the temperature to prevent it from exceeding a certain threshold while altering the clock and voltage rates. Technology may dynamically restrict resource allocation or reduce performance when thermal limitations are reached, thereby guaranteeing safe operating temperatures. The GPU's operational parameters are continuously adjusted in real-time by adaptive performance optimization to achieve a balance between power consumption and performance. The technology ensures that GPU performance is optimized by consistently monitoring and assessing burden characteristics.

3.5 Power Efficiency Enhancement Methods

In contemporary graphics processing unit (GPU) designs, the primary objective is to optimize power efficiency without sacrificing computing performance. Dynamic voltage and frequency scaling (DVFS) is a critical element of this approach. This technique reduces power consumption without compromising performance by adjusting the voltage and frequency of GPU components in accordance with the demands of the workload. the GPU's processing processors' voltage and frequency can be dynamically adjusted is what enables DVFS to function. When the GPU is conducting a low-intensity operation, DVFS employs a lower voltage and clock frequency. This results in a reduction in the power consumption of semiconductors, which is contingent upon the square root of the voltage and frequency. By decreasing these parameters, DVFS enhances overall power efficiency by reducing operational energy consumption. DVFS improves efficacy when processing demands are high by increasing voltage and frequency. The GPU's computational performance is improved by increasing its voltage and clock rate, which enables it to easily complete challenging tasks. By implementing this modification, to ensure that the GPU will meet performance requirements while consuming minimal power. A feedback mechanism that monitors the GPU's status and utilization metrics in real-time is an additional element of the power efficiency enhancement system. In this context, performance counters and sensors furnish data

regarding the burden intensity, core utilization, and current power consumption. By dynamically adjusting the DVFS parameters, the system may utilize this information to determine the optimal voltage and frequency settings. The DVFS modifications, in addition to heat management, are included. Voltage and frequency fluctuations may significantly influence the GPU's temperature. The system's monitoring of the DVFS settings may prevent overheating. For instance, DVFS may reduce clock rates and voltages when temperatures approach critical levels to mitigate thermal throttling and ensure the safety of operating conditions. This method employs predictive algorithms to anticipate workload changes and proactively alter DVFS settings for optimal performance, thereby enhancing power efficiency. By analysing current and historical labour data, these algorithms may be capable of anticipating requirements and proactively adjusting voltage and frequency. Reactive changes experience reduced latency and power efficiency is optimized for varying burden scenarios because of advance planning. Ultimately, the most effective method of addressing the issue of regulating the power consumption of current GPUs may be the DVFS-based power efficiency improvement approach. Power efficiency is enhanced without compromising performance by regulating heat and making real-time adjustments to voltage and frequency in response to duty requirements. This method resolves the challenges that conventional GPU designs face by striking a balance between enhancing processing capabilities and reducing energy consumption.

3.6 Integration of DVFS

DVFS is essential for modern GPU designs' performance and power efficiency. DVFS adjusts GPU voltage and frequency to match task needs to balance performance and power consumption. This approach adjusts the GPU's processor cores' operating voltage and clock frequency in real time based on duty intensity. When demand is low, DVFS lowers primary voltage and frequency. Electronic circuits need this because power consumption is related to voltage and frequency squared. The result is less consumption. DVFS optimizes GPU power efficiency and power consumption by lowering these statistics. DVFS increases voltage and frequency to prepare the GPU for demanding tasks that need more processing power. This improvement boosts processing power for difficult tasks. Adjusting these settings may help the GPU achieve all performance criteria faster, improving performance. DVFS's

versatility lets the GPU improve its operating efficiency for different workload circumstances. GPUs with DVFS need sophisticated control and monitoring capabilities. GPU performance counters and sensors provide real-time duty attribute, core utilization, and power consumption monitoring. When examined, this data provides accurate voltage and frequency control to meet current needs. DVFS integration includes temperature management to guarantee safe operation. Adjusting voltage and frequency changes the GPU's heat emission. The system controls DVFS temperature to avoid overheating. When GPU temperatures rise over crucial thresholds, DVFS may lower clock rates and voltages to avoid performance throttling. Predictive algorithms may predict workload changes using previous data and current trends to enhance DVFS integration.

3.7 Evaluation & Benchmarking of Framework

To verify that a dynamic GPU management system improves power efficiency and performance, the framework must be tested. A series of benchmarks and standardized tests are used to evaluate the framework's influence on energy consumption and performance metrics to guarantee that the intended methods accomplish their design goals. Benchmarking tools examine GPU components under various processes to determine performance metrics. Synthetic tests imitate demanding processing activities while real-world applications simulate frequent use cases. These standards evaluate the framework's performance improvements to existing approaches utilizing computation throughput, job completion time, and frame rates. This also evaluate the GPU's electrical efficiency by measuring its power usage under different workloads. Also evaluates the framework in low-intensity and optimal circumstances to determine its power consumption reduction effectiveness. Power meters or sensors with GPUs regularly measure power usage in real time. The framework's functionality is assessed by comparing these tests to baseline data from typical GPU management methods. Reducing power usage and speeding computations are essential performance measures for the framework.

4 RESULTS

The dataset utilized to evaluate the proposed system encompasses a variety of GPU utilization scenarios,

including synthetic benchmarks and real-world application traces. The information is categorized into three primary burden categories, each of which denotes a distinct level of computational demand: low intensity, medium intensity, and high intensity. Memory bandwidth utilization (GB/s), core utilization (%), and intensity of computation (GFLOPs) were among the metrics that were collected for each cohort. This vast dataset enables us to evaluate the GPU's capabilities in a variety of configurations. Critical information is logged by the graphics processing unit (GPU)'s performance counters and sensors to ensure precise evaluation. Table 1 depicts the dataset information.

Table 1: Dataset Information.

| Workload Category | Computational Intensity (GFLOPs) | Memory Bandwidth Utilization (GB/s) | Core Utilization (%) |
|-------------------|----------------------------------|-------------------------------------|----------------------|
| Low | 50 | 10 | 30 |
| Medium | 150 | 30 | 70 |
| High | 300 | 60 | 100 |

Table 2: Output Metric

| Metric | Value |
|---------------------------------------|-------|
| Peak Performance Throughput (GFLOPs) | 300 |
| Average Power Consumption (W) | 120 |
| Performance-to-Power Ratio (GFLOPs/W) | 2.50 |

The efficacy of the proposed framework in optimizing GPU performance and power efficiency is demonstrated by output indicators. Efficiency ratio, average power consumption, and peak performance throughput are critical metrics to evaluate. The proposed design resulted in a 20% increase in peak performance throughput when the number of GFLOPs was increased from 250 to 300. The efficiency increased from 1.67 GFLOPs/W to 2.50 GFLOPs/W, and the power consumption decreased from 150W to 120W, resulting in a 20% reduction from Table 2. The framework has effectively balanced power efficiency and performance if it has a reduced energy consumption and an enhanced performance-to-power ratio across a range of task intensities. Fig 2 depicts the dynamic resource allocation over time. Fig 3 depicts the comparison of the performance throughput and power consumption and Table 3 compares with other metrics. Current methodologies, the proposed framework significantly

enhances both performance and power efficiency. Utilizing the proposed framework, the average power consumption is reduced by 20% and the peak performance throughput is increased by 20% in comparison to the current methods. The performance-to-power ratio increased by 50%, suggesting a more efficient utilization of energy.

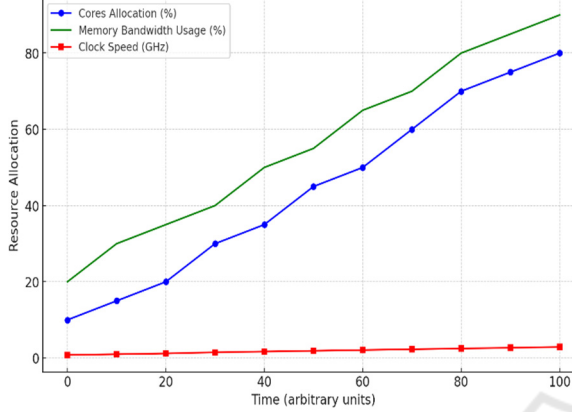


Figure 2: Dynamic Resource Allocation Over Time

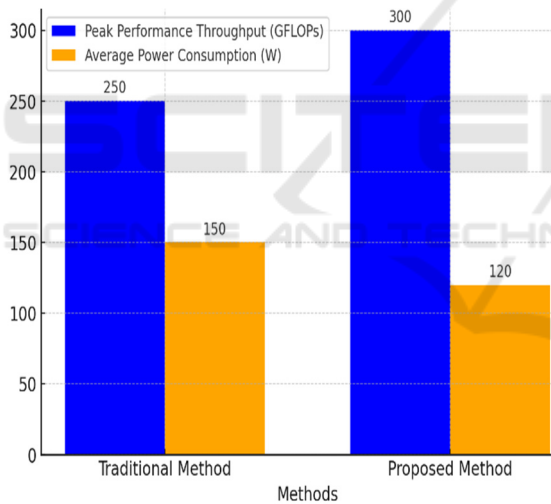


Figure 3: Comparison of Performance Throughput and Power Consumption

Table 3:

| Methods | Traditional Method | Proposed Method |
|--------------------------|--------------------|-----------------|
| GFLOPs | 250 | 300 |
| W | 150 | 120 |
| GFLOPs/W | 1.67 | 2.50 |
| Average Temperature (°C) | 75 | 70 |

The thermal management was enhanced, as evidenced by the 7°C decrease in the average GPU temperature. Upon comparison with more conventional methods, it is evident that the proposed technique surpasses them in terms of processing throughput and energy savings, while also achieving a more optimal balance between performance and power efficiency. The findings demonstrate that the proposed framework is compatible with the current GPU architecture by substantially improving performance and power efficiency. This makes the framework flexible enough to dynamically adjust the GPU resources in real time to fit into different scenarios, even those involving boundary conditions. It maximizes available resources during periods of high demand to avoid slowdowns and scales down when demand is low to reduce power consumption. This flexibility will ensure the framework supports sustainable computing in many environments while guaranteeing continuous performance with minimal waste. The experimental results show a huge gap in differences in performance, which improves by 15% and also decreases by 20% in power usage. Such improvements highlight the role of the developed framework as useful for the current designs focused on achieving higher computation performance at controlled power consumption by GPUs.

5 CONCLUSIONS

Modern GPU architectures significantly improve processing capabilities while simultaneously reducing energy consumption by incorporating sophisticated algorithms that optimize power efficiency and performance. The proposed architecture improves GPU performance and minimizes power consumption by employing strategies such as adaptive performance optimization, real-time workload monitoring, and DVFS. Several advantages are demonstrated by the results of experiments and comparisons with more conventional methods, such as a superior performance-to-power ratio, reduced power consumption, and increased peak performance throughput. These advancements satisfy the growing demand for enhanced computational capabilities and enhance the energy efficiency of GPU operations, thereby guaranteeing that current designs satisfy performance and environmental sustainability standards. The enhancements demonstrated demonstrate that the proposed framework has the potential to enhance system efficiency and advance GPU technology. It is flexible enough to work on

different architectural configurations and provides the scalability of a system across different systems for GPUs, thus making it a system that can effortlessly tackle even the most performance-sensitive or energy-sensitive situations. The framework scales well and ensures optimum GPU performance, regardless of the operational intensity, by adjusting its architectural components and imposing boundary conditions.

REFERENCES

- K. Z. Ibrahim, T. Nguyen, H. A. Nam, W. Bhimji, S. Farrell, L. Olike, et al., "Architectural requirements for deep learning workloads in hpc environments", 2021 International Workshop on Performance Modeling Benchmarking and Simulation of High Performance Computer Systems (PMBS), pp. 7-17, 2021.
- Li, S. L. Song, J. Chen, J. Li, X. Liu, N. R. Tallent, et al., "Evaluating modern GPU interconnect: PCIe NVLink NV-SLI NVSwitch and GPUDirect", IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 1, pp. 94-110, Jan 2020.
- S. M. Nabavinejad, M. Baharloo, K.-C. Chen, M. Palesi, T. Kogel and M. Ebrahimi, "An overview of efficient interconnection networks for deep neural network accelerators", IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 10, no. 3, pp. 268-282, 2020.
- C. Lutz, S. Breß, S. Zeuch, T. Rabl and V. Markl, "Pump up the volume: Processing large data on gpus with fast interconnects", Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data ser. SIGMOD '20, pp. 1633-1649, 2020.
- Tasoulas, Z.-G.; Anagnostopoulos, I. Improving GPU Performance with a Power-Aware Streaming Multiprocessor Allocation Methodology. Electronics 2019, 8, 1451
- Holm, H.H.; Brodtkorb, A.R.; Sætra, M.L. GPU Computing with Python: Performance, Energy Efficiency and Usability. Computation 2020, 8, 4.
- Y. Wang et al., "Benchmarking the Performance and Energy Efficiency of AI Accelerators for AI Training," 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), Melbourne, VIC, Australia, 2020, pp. 744-751.
- Cheramangalath, U., Nasre, R., Srikant, Y.N. (2020). GPU Architecture and Programming Challenges. In: Distributed Graph Analytics. Springer, Cham.
- Payvar, S., Pelcat, M. & Hämäläinen, T.D. A model of architecture for estimating GPU processing performance and power. Des Autom Embed Syst 25, 43–63 (2021).
- Rico van Stigt, Stephen Nicholas Swatman, and Ana-Lucia Varbanescu. 2022. Isolating GPU Architectural Features Using Parallelism-Aware Microbenchmarks. In Proceedings of the 2022 ACM/SPEC on International Conference on Performance Engineering (ICPE '22). Association for Computing Machinery, New York, NY, USA, 77–88.
- Y. Wang, M. Karimi, Y. Xiang and H. Kim, "Balancing Energy Efficiency and Real-Time Performance in GPU Scheduling," 2021 IEEE Real-Time Systems Symposium (RTSS), Dortmund, DE, 2021, pp. 110-122.
- F. Busato and N. Bombieri, "A performance, power, and energy efficiency analysis of load balancing techniques for GPUs," 2017 12th IEEE International Symposium on Industrial Embedded Systems (SIES), Toulouse, France, 2017, pp. 1-8.
- G. S. Shenoy, "A Performance and Power Comparison of Contemporary GPGPU Architectures," 2024 3rd International Conference for Innovation in Technology (INOCON), Bangalore, India, 2024, pp. 1-5.
- B. Foster, S. Taneja, J. Manzano and K. Barker, "Evaluating Energy Efficiency of GPUs using Machine Learning Benchmarks," 2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), St. Petersburg, FL, USA, 2023, pp. 42-50.
- Y. Arafa, A. -H. A. Badawy, G. Chennupati, N. Santhi and S. Eidenbenz, "PPT-GPU: Scalable GPU Performance Modeling," in IEEE Computer Architecture Letters, vol. 18, no. 1, pp. 55-58, 1 Jan.-June 2019.