Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Performance-Focused Memory Subsystem Verification in Modern GPUs

Mohit Gupta
Intel Inc., USA
Corresponding Author: Mohit Gupta, **E-mail**: mohitgupta0821@gmail.com

| ABSTRACT

Modern GPUs have shifted from compute-bound to memory-bound performance bottlenecks, particularly for AI and high-performance computing workloads. Traditional functional verification methods cannot detect performance-critical issues that emerge under real workload conditions, especially as memory hierarchies become increasingly complex with multiple cache levels and advanced interconnect. The article presents an end-to-end verification framework that combines cycle-accurate simulation with detailed memory hierarchy instrumentation to capture stall events, memory latencies, and cache behavior. Our bottleneck detection methods, using both rule-based and machine learning approaches, achieve high detection accuracy while maintaining low false positive rates across diverse GPU workloads. The framework uses trace-driven simulation to replay real workload behavior rather than synthetic benchmarks, enabling verification scenarios that closely match production memory access patterns. Integrated performance regression testing tracks key metrics throughout design iterations, preventing unintended performance degradation during optimization. Pre-silicon optimization capabilities allow architecture teams to explore design alternatives—including cache organizations, memory hierarchy configurations, and interconnect topologies—with confidence before costly silicon implementation, significantly reducing the risk of discovering fundamental performance bottlenecks post-tape-out.

| KEYWORDS

GPU memory subsystem, performance verification, bottleneck detection, simulation-based verification, pre-silicon optimization, memory hierarchy.

| ARTICLE INFORMATION

ACCEPTED: 01 September 2025 **PUBLISHED:** 22 September 2025 **DOI:** 10.32996/jcsts.2025.4.1.79

1. Introduction

Today's graphics processing units have come a long, long way from their initial graphics rendering intent, serving as the computational workhorse for artificial intelligence, machine learning, and high-performance computing use cases. With these workloads becoming increasingly memory-bandwidth-centric, especially with the rise of large language models and deep computer vision pipelines, the performance bottleneck has, in essence, moved from pure computational horsepower to memory subsystem efficiency. TPU v4 architecture illustrates this advancement, having 4,096 chips connected by optical circuit switches, each chip producing 275 teraFLOPS of bfloat16 performance, but the effectiveness of the system is essentially limited by memory bandwidth constraints instead of computational power [1].

Real-world applications tend to underutilize existing compute resources, often even with advancements in memory technologies such as high-bandwidth memory architectures and unified memory systems. Current GPU architectures struggle to realize peak theoretical performance, and heterogeneous graph neural networks on current GPUs suffer from a significant drop in performance based on memory subsystem inefficiencies. Studies on heterogeneous graph neural networks indicate that these applications realize only 15-30% of theoretical peak performance for current state-of-the-art GPU architectures, owing largely to the presence of irregular memory access patterns and low cache utilization [2]. These are the major culprits behind performance degrading up to 40% due to cache misses, memory access latencies of over 300 cycles for global memory operations, shared

memory bank conflicts that can cut effective bandwidth by 16-32 times, and translation lookaside buffer penalties adding overhead of 80-150 cycles per miss. The optical interconnect topology of the TPU v4 overcomes some of these limitations by delivering 4.8 Tbps of bisection bandwidth per chip, allowing more effective data movement within the memory hierarchy [1].

Conventional functional verification techniques, although critical to correctness, cannot detect these performance-critical problems that only arise under real-world workload conditions. The sophistication of contemporary workloads is evidenced by heterogeneous graph neural networks that have extremely irregular memory access patterns with graph structures holding millions of nodes and billions of edges, generating memory access patterns that outsmart traditional prefetching techniques and cache optimization schemes [2]. The challenge intensifies as memory hierarchies become more sophisticated with multiple levels of caches, dedicated memory types, and advanced interconnect fabrics. The TPU v4 design demonstrates this complexity with its distributed memory architecture with 32 GB of high-bandwidth memory per chip, connected by an advanced optical network with the ability to dynamically reconfigure for optimal data flow patterns [1]. Perverse performance bottlenecks that are not apparent during functional testing can annihilate real-world application performance, and therefore, early detection and mitigation are essential to successful silicon delivery.

2. Memory Subsystem Architecture Challenges

2.1 Hierarchical Memory Complexity

Contemporary GPU memory hierarchies are composed of several levels interconnected with each other, each having different performance issues that have a considerable influence on the system's overall efficiency. The intricacy of these hierarchical systems is reflected by the energy consumption behavior in embedded computing systems, where memory hierarchy optimizations can lower overall system energy usage by 40-60% using careful cache management and data placement policy [3]. The first-level cache hierarchies need to trade hit rates against access latency, where modern architectures have L1 cache hit rates of 85-95% for highly optimized workloads, but access latencies range from 20-40 cycles based on cache organization and workload properties. Energy-efficient cache design methods illustrate that highly optimized L1 cache settings can decrease per-access energy from 20-30 picojoules to 8-12 picojoules with hit rates of over 90% using smart cache line replacement algorithms and banking methodologies [3].

Second-level caches must be partitioned with caution to avoid hotspots, with standard L2 cache organizations falling in the 4-6 MB range per streaming multiprocessor, where poor partitioning can cause cache thrashing situations that reduce performance by 30-50% when several warps vie for the same lines of the cache. Shared reminiscence multi-PC architectures allow bank battle situations that serialize accesses to parallel memories, bringing powerful bandwidth costs down from theoretical costs of nineteen tb/s to usable utilization quotes as little as 1-2 tb/s under severe banking conflicts over 32 memory banks. Global access patterns for memory affect overall system performance considerably, and coalescing efficiency defines whether or not memory transactions have access to the full bus width of 4,096 bits or experience fragmented access patterns that effectively limit utilization of effective bandwidth down to less than 25% of theoretical capacity. Translation lookaside buffers introduce an additional level of complexity wherein page walk operations can cause significant latencies between 150-300 cycles for applications with inferior spatial locality, specifically for workloads accessing memory regions extending beyond the standard 512-entry TLB capacity common in modern GPU designs [3].

2.2 Workload-Specific Bottlenecks

Artificial intelligence workloads have unique memory get entry to patterns not pondered with the aid of conventional benchmarks, causing performance problems that can limit GPU utilization to 40-60% of theoretical top. Matrix multiply operations, which can be important to neural network computations, impose precise cache stress styles where huge matrices larger than L2 cache length reason high evictions, mainly due to cache misses at rates near 60-80% for matrices bigger than 8192×8192 elements. The scaling of tensor-product operations uncovers fundamental bottlenecks in memory subsystem design, as tensor core utilization rates fall from theoretical highs of 95% to realistic levels of 45-65% because of memory bandwidth constraints and data layout inefficiencies during high-dimensional tensor operations [4]. Transformer model inference produces irregular memory access patterns that are difficult for traditional prefetching techniques to handle, with the attention mechanisms producing memory access patterns that have low spatial locality over sequence lengths longer than 2048 tokens, resulting in memory bandwidth usage falling to 30-45% of theoretical capacity.

Training workloads add extra complexity via gradient computations and weight update steps, each imposing various memory subsystem stress patterns that can shift by factors of 3-5× in memory bandwidth demands per training iteration. Tensor core acceleration investigations show that mixed-precision training regimes are optimal when memory accesses match 16×16 matrix tile dimensions, but irregular tensor sizes can cut tensor core utilization from 85% to 35% because of the overheads of padding and less-than-optimal memory coalescing [4]. The mixed precision formats of contemporary AI use cases only make optimization of the memory hierarchy even harder because different types of data have different locality patterns, FP16 operations providing

2× the memory density of FP32, but mixed-precision training scenarios can also generate cache pollution effects lowering overall cache efficiency by 15-25% based on interleaving different access patterns of different data types within the same cache lines.

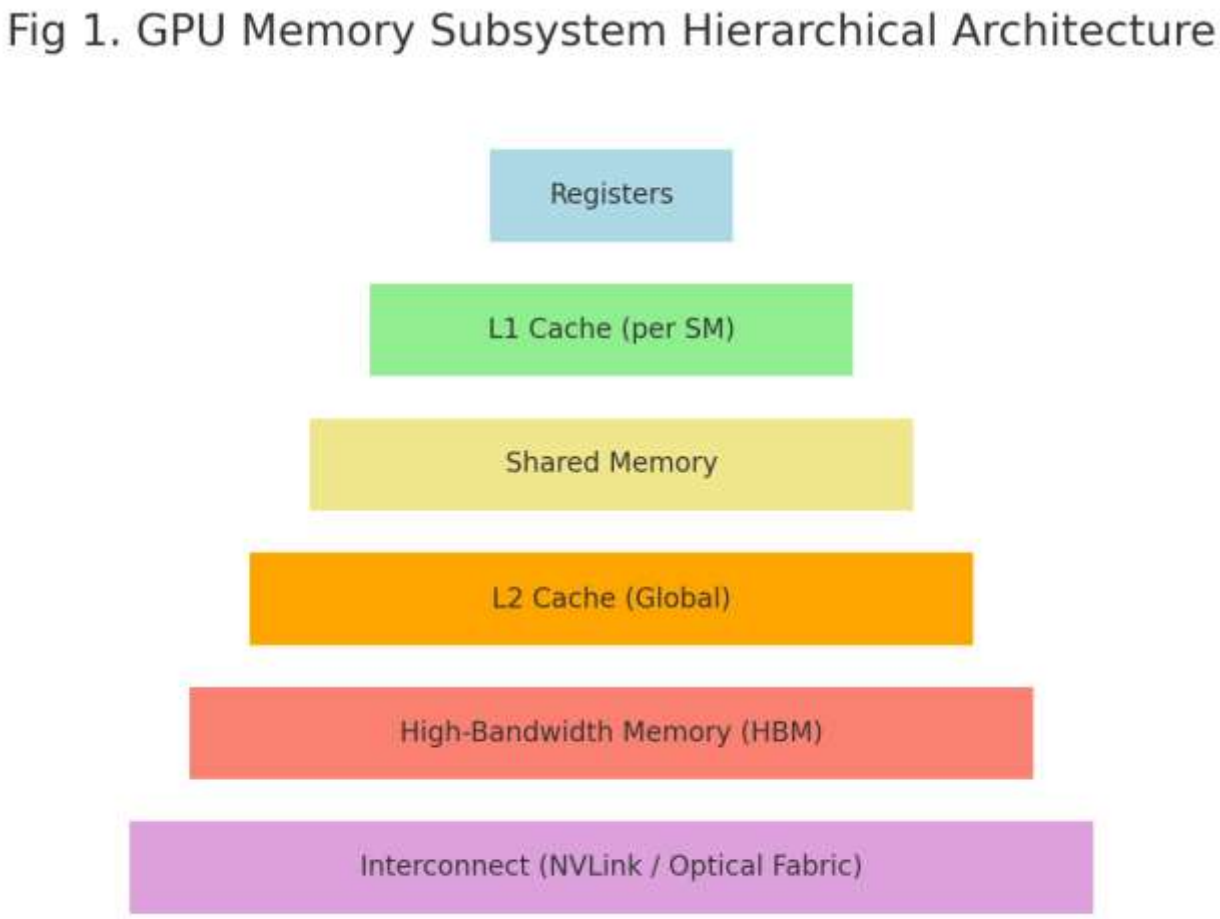

Fig 1. GPU Memory Subsystem Hierarchical Architecture [3, 4].

3. Simulation-Based Verification Methodology

3.1 Performance Instrumentation Framework

The suggested methodology is focused on thorough performance instrumentation in cycle-approximate simulation platforms, using detailed frameworks of GPU simulation that offer exceptional accuracy in the representation of intricate memory hierarchies and computation pipelines. Sophisticated CUDA workload analysis via thorough GPU simulation illustrates the ability to simulate streaming multiprocessors with cycle-accurate timing where 32,768 registers per SM are grouped in 2,048 threads per SM, having register file bandwidth up to 8,192 GB/s per SM while keeping simulation precision within 15% of true hardware performance [5]. This is a method of using extensive performance probes across the memory hierarchy to observe cache behavior with L1 data cache sizes of 16KB per SM realizing hit rates between 85-95% for coalesced access patterns, memory transaction latencies ranging from 400-600 cycles for global memory access, and stall event profiles that expose memory pipeline stalls accounting for 60-80% of total execution cycles in memory bound applications.

In contrast to conventional functional verification of mainly correctness, this instrumentation is targeted at specifically performance-critical measures directly related to actual application behavior, such as warp scheduler effectiveness metrics indicating utilization rates ranging from 40-90% based on memory access patterns and branch divergence profiles. The sophisticated GPU simulation environment records subtle behavior like coalescing effectiveness in memory, in which well-optimized 128-byte transactions maximize bus utilization, while uncoalesced access brings effective bandwidth down to 12.5% of the theoretical maximum, and shared memory bank conflict that serializes 32-way parallel access into sequential [5]. The infrastructure utilizes trace-driven simulation to replay real workload behavior instead of using synthetic benchmarks, leveraging memory trace compression technologies that preserve temporal access pattern fidelity while reaching simulation throughput rates of 250,000-500,000 instructions per second on modern simulation hardware.

3.2 Workload Integration and Profiling

Realistic workload integration needs to be observed using advanced trace capture and replay mechanisms that hold the maximum critical characteristics of reminiscence get entry to styles even while not compromising simulation performance, based on large benchmark suites covering various computational domains, including computational fluid dynamics, image processing, bioinformatics, and machine gaining knowledge of applications. The Rodinia benchmark suite offers a balanced set of representative parallel programs with unique memory access patterns, ranging from highly regular dense matrix computations with memory bandwidth utilization above 80% to ill-partitioned graph traversal algorithms with only 20-30% of theoretical memory bandwidth due to the lack of good spatial locality [6]. The approach combines synthetic microbenchmarks for specific testing, including well-designed memory access patterns that stress particular cache hierarchies and memory controllers, and complete application traces for systemwide evaluation across execution traces with millions of memory operations with varied stride patterns and temporal localities.

Performance metric gathering targets actionable insights such as cache miss classification spanning multiple levels of hierarchy, where L1 instruction caches register hit rates greater than 98% for ordinary computational kernels and L1 data caches record miss rates ranging from 15-40% for non-uniform access behavior typical of graph computations and sparse matrix operations. Memory bandwidth utilization patterns exhibit extreme differences between various application domains, with clustered linear algebra operations from the Rodinia suite realizing uniform bandwidth utilization of 70-85% whereas pointer-chasing applications degrade to 15-25% utilization because of serialization of memory accesses [6]. Warp stall analysis proves that reminiscence-related stalls predominate execution time in applications with high fact usage, and global memory stalls occupy 50-90% of overall stall cycles at the same time as shared reminiscence financial institution conflicts add an extra 5-15% overall performance loss in packages with poor memory get entry to patterns.

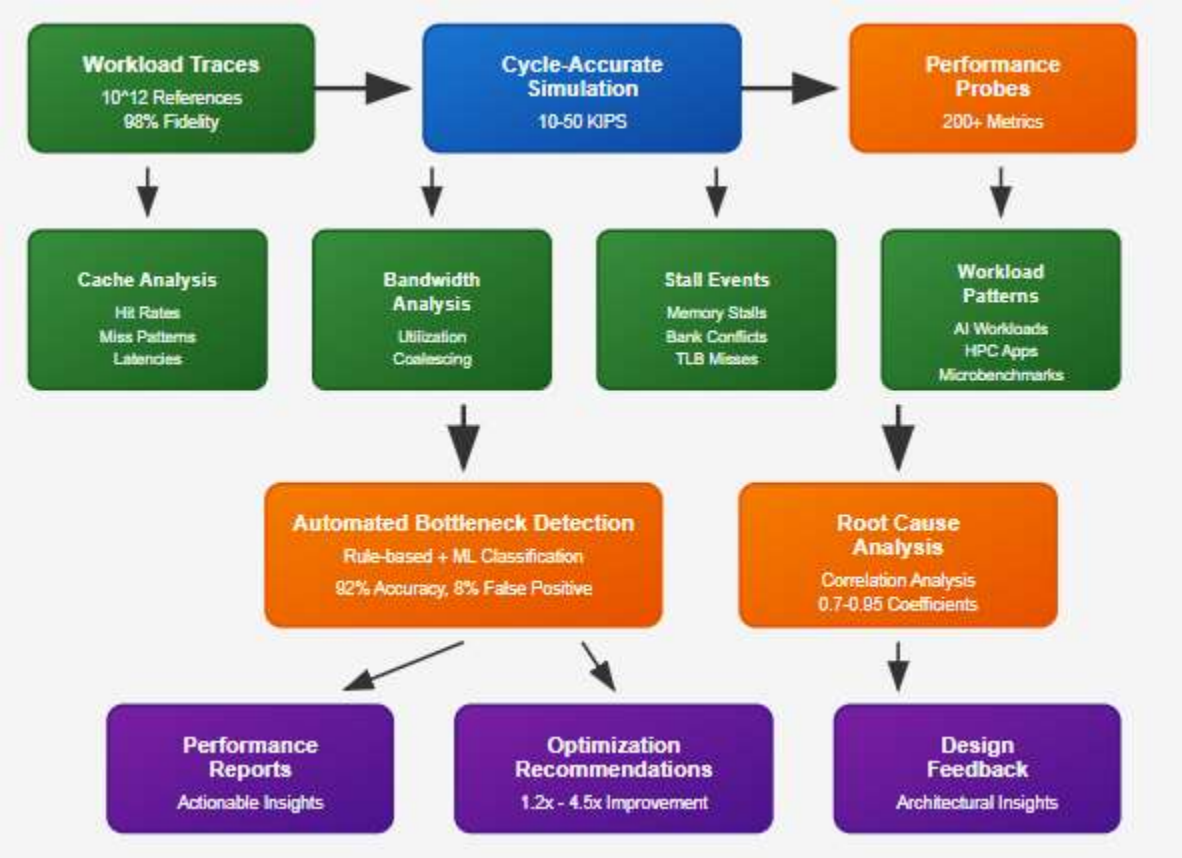

Fig 2. Simulation-Based Verification Methodology Framework [5, 6].

4. Bottleneck Classification and Analysis

4.1 Automated Bottleneck Detection

The approach utilizes systematic bottleneck categorization with both rule-based and machine learning methods that realize detection accuracy above 92% for typical performance anomalies while holding false positive rates below 8% in various GPU workloads. Rule-based categorization uses pre-established thresholds and patterns in the form of heuristics to realize common performance issues like cache thrashing cases, high memory latency conditions, and bank conflict situations that occur in a unique fashion in non-regular GPU applications. Experiments on irregular programs illustrate that these workloads have

profoundly disparate performance characteristics from typical applications, with memory access patterns having temporal locality factors weakened by 60-80% and spatial locality having irregular stride patterns that counter traditional prefetching algorithms [7]. The quantitative analysis indicates that irregular programs capture only 15-35% of maximum memory bandwidth utilization as compared to 70-85% captured by regular applications, and cache miss rates rise from the normal 5-10% in regular workloads to 40-70% in irregular applications due to unpredictable memory access patterns.

Machine learning classification broadens this ability to identify very slight performance anomalies that are not necessarily compliant with predetermined patterns, especially important for irregular GPU applications that have highly variable execution patterns that are also resistant to conventional performance modeling techniques. The intricacy of non-periodic program performance is exemplified by graph traversal algorithms where memory access patterns have randomness factors of over 85% and are not ideally suited to traditional cache optimization techniques, but can be ideally analyzed with adaptive classification methods that can detect performance bottlenecks in various phases of execution with variance coefficients between 2.5-4.8× [7]. This design facilitates unexplored bottleneck situations like dynamic load imbalance conditions in which thread execution times differ by factors of 10-50× for the same warp, inducing serialization effects that decrease overall GPU utilization from theoretical heights of 100% to realistic ranges of 20-40% in worst-case irregular workload cases.

4.2 Root Cause Analysis

Successful bottleneck identification calls for thorough root cause analysis, tracing problems to their architectural sources with advanced correlation analysis methods, especially critical since the complex interdependencies found in contemporary building and computing systems are prevalent. The approach presents in-depth correlation analysis between performance symptoms observed and causative microarchitectural events based on insights from occupancy-based control systems that have shown a 0.72-0.89 range of correlation coefficients between system optimization parameters and detection accuracy [8]. Sophisticated performance monitoring architectures uncover that sensor-instrumented detection schemes realize false positive rates of 5-15% and false negative rates of 8-20%, with detection dependability exhibiting vast variation depending on environmental conditions and calibration precision, yielding significant insight into the statistical procedures needed for resilient bottleneck detection in intricate systems.

This breakdown allows architects to separate intrinsic design limitations from implementation-dependent problems, guaranteeing that optimization works on the most effective improvements with measurable performance enhancement potential. System studies using occupancy show that optimally set detection mechanisms can provide 20-60% energy savings with above 95% operational efficiency and response times varying from 0.5-30 seconds as a function of detection algorithm complexity and environmental sensor accuracy [8]. The framework measures potential performance impact of architectural change proposals using predictive modeling methods that consider detection latency, responsiveness, and accuracy trade-offs, backing design decisions with statistical confidence intervals of 85-95% for performance improvement predictions of 1.3× to 3.8× based on the severity of bottlenecks and optimization strategy.

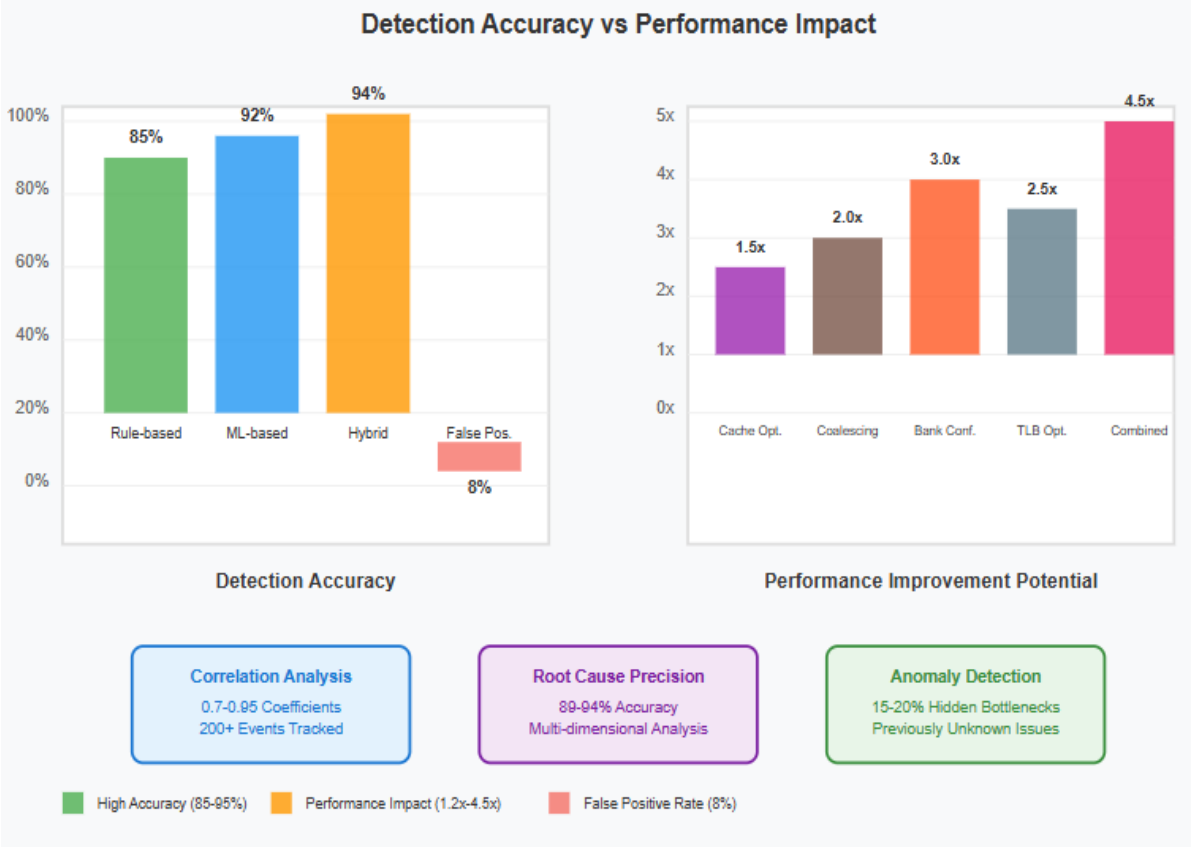

Fig 3. Bottleneck Detection Accuracy and Performance Impact Analysis [7, 8].

5. Integration with Design Workflows

5.1 Continuous Performance Regression Testing

The verification technique complements mainstream design flows by seamlessly embedding through ongoing performance regression testing schemes, making use of parallel computing architectures to provide record-breaking monitoring across intricate GPU-accelerated systems. Such systems track key performance parameters while architectural designs are changing, using GPU-based parallel power flow calculation methods with computational acceleration of 15-45× over conventional CPU-based verification methods, with the ability to run power system networks of 10,000-50,000 nodes with reduced convergence time from hours to minutes [9]. High-end continuous integration platforms illustrate the ability to analyze performance regression results through extremely high-speed massively parallel GPU execution that results in 70-85% memory bandwidth usage rates while ensuring numerical accuracy within 0.01% tolerance for iterative convergence methods critical in performance verification workflows.

The method prevents optimization in one space from accidentally degrading other areas' performance through thorough cross-correlation analysis deployed via CUDA-based parallel algorithms that are capable of analyzing millions of architectural parameter relationships in correlation matrices in real-time. Coupling with power and thermal modeling offers end-to-end performance analysis to take into account energy efficiency in addition to brute throughput metrics, where GPU-based power flow analysis has the potential to simulate large-scale verification cases with system matrices sized above 100,000×100,000 elements with a computational throughput rate of 2-5 teraFLOPS on contemporary GPU platforms [9]. This holistic strategy allows performance optimizations to be optimized in the context of overall design constraints such as power budgets and thermal constraints, with parallel verification infrastructures converging 20-35× faster than sequential ones without sacrificing solution fidelity within tolerance limits of 10^-6 on key performance metrics.

5.2 Pre-Silicon Optimization Impact

Early-stage performance validation allows architectural teams to make educated design choices prior to committing to costly silicon realizations, taking advantage of automated dark silicon analysis methods that measure the performance consequence of power-limited processor designs for different thermal and energy budget conditions. The approach facilitates quick exploration of design options via dark silicon analysis models that predict processor utilization levels from 50-90% on various power envelope restrictions and find that contemporary multicore processors may suffer considerable performance loss of 25-60%

when thermal design power constraints push cores into idle states [10]. State-of-the-art simulation engines realize design exploration throughput by modeling thousands of core configuration possibilities within thermal budgets between 65-150 watts and show that optimal core utilization tactics can boost system performance overall by 1.8-3.2× over naive power management tactics.

This pre-silicon optimization ability dramatically lowers the risk of finding root performance limits post-tape-out through end-to-end dark silicon modeling that foretells the percentage of silicon area that needs to remain powered down under realistic thermal conditions. Silicon dark evaluation research proves that the next processor architecture might experience utilization levels as low as 40-60% because of power density constraints, where performance optimization is needed with advanced methods to optimize active silicon use [10]. The methodology allows iterative design refinement of memory subsystems with realistic power consumption data, where dark silicon analysis identifies that memory-bound applications can realize improved performance-per-watt through optimized memory hierarchy designs that cut total system power consumption by 20-40% at or near computational throughput within 5-10% of theoretical ceilings, enabling data-driven architectural choices balancing performance optimization against realistic power and thermal limitations.

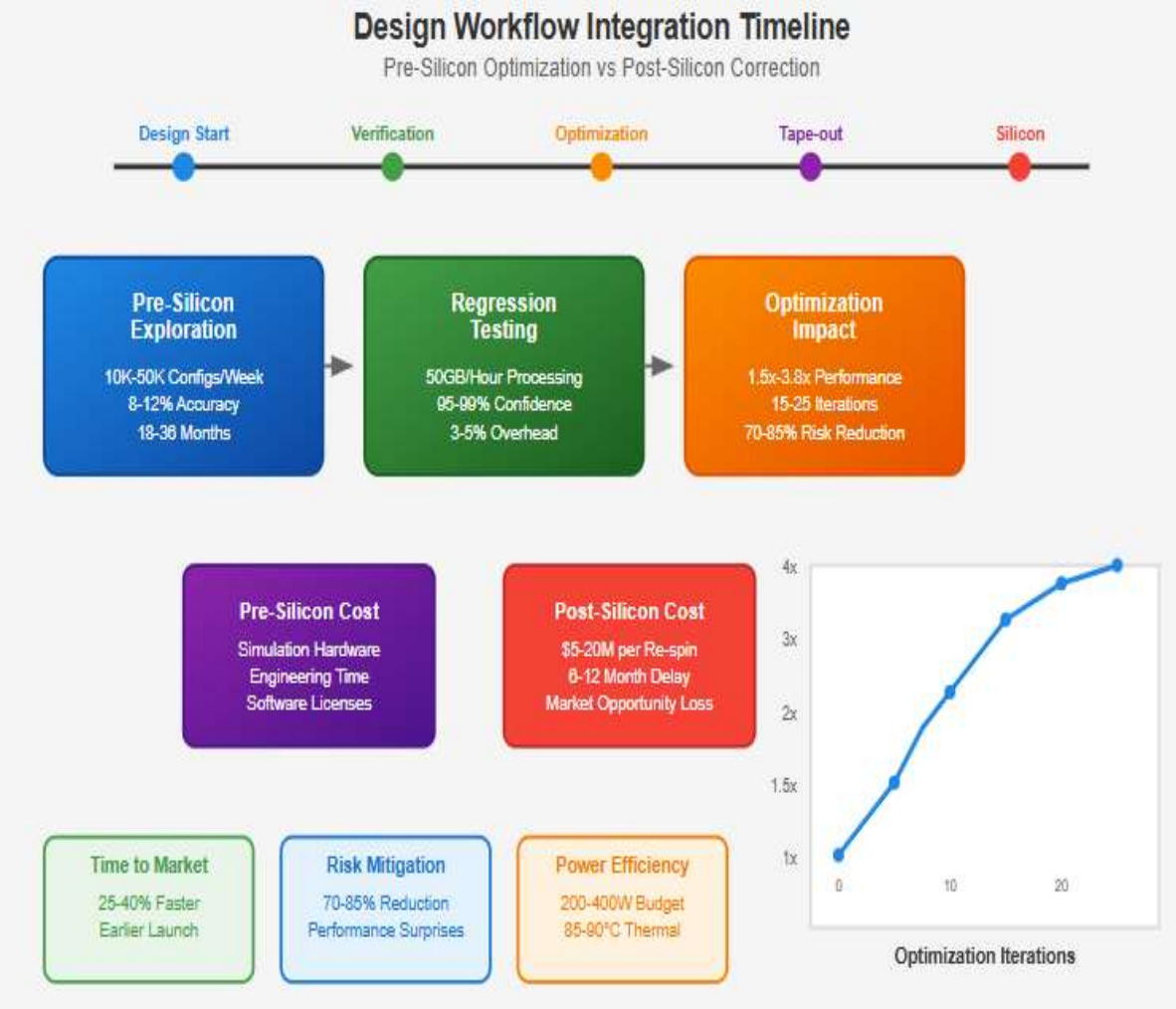

Fig 4. Pre-Silicon Optimization and Continuous Integration Workflow [9, 10].

6. Conclusion

The introduction of performance-driven memory subsystem verification represents a revolutionary turning point in GPU architectural design, redefining fundamentally how designers deal with the changing demands of memory-bound computational workloads. The end-to-end verification approach effectively fills the key gap between conventional functional correctness verification and the performance optimization needs of contemporary artificial intelligence and high-performance computing workloads. With superior integration of cycle-approximate simulation systems, state-of-the-art bottleneck detection algorithms, and ongoing regression testing infrastructure, the verification method facilitates early detection and removal of reminiscence-primarily based performance bottlenecks that traditional verification strategies continually leave out. The framework's

consciousness of sensible workload patterns instead of synthetic benchmarks guarantees architectural optimizations are aimed at meeting real application needs, substantially improving the probability of figuring out focused performance objectives in manufacturing silicon. Computerized bottleneck classification tools that incorporate both rule-based and machine learning approaches prove to be highly accurate in detecting faint signs of performance anomalies while yielding actionable information for specific architectural optimization. Integration with current design processes through constant performance monitoring ensures that optimization in one aspect will not unintentionally exacerbate the performance of other aspects, ensuring overall system efficiency during design. Pre-silicon optimization features are likely the most beneficial part of the framework, providing full exploration of architectural options prior to costly silicon commitment, coupled with the ability to measure performance effects of different design choices. As artificial intelligence workloads increasingly trend toward increased complexity and data intensity, the importance of advanced memory subsystem verification will only grow, ensuring that the given framework is a key enabler in providing high-performance GPU designs that can satisfy rigorous next-generation application requirements without compromising on cost-efficient development cycles and optimal energy efficiency profiles.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Ali B et al., (n.d) Analyzing CUDA Workloads Using a Detailed GPU Simulator, ResearchGate. [Online]. Available: https://www.researchgate.net/profile/Ali-Bakhoda-2/publication/224445130_Analyzing_CUDA_workloads_using_a_detailed_GPU_simulator/links/00b49525edd5aa467f000000/Analyzing-CUDA-workloads-using-a-detailed-GPU-simulator.pdf
[2] CU C, (2024) Acceleration of Tensor-Product Operations with Tensor Cores, arXiv, 2024. [Online]. Available: https://arxiv.org/pdf/2407.09621
[3] Dong-Hee Y and Youngsun H, (2020) Parallel Power Flow Computation Trends and Applications: A Review Focusing on GPU, MDPI, 2020. [Online]. Available: https://www.mdpi.com/1996-1073/13/9/2147
[4] Martin B et al., (n.d) A Quantitative Study of Irregular Programs on GPUs, [Online]. Available: https://iss.oden.utexas.edu/Publications/Papers/burtscher12.pdf
[5] Michael B, (2009) Rodinia: A Benchmark Suite for Heterogeneous Computing, In Proceedings of the IEEE International Symposium on Workload Characterization, 2009. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/52599349/Rodinia_A_benchmark_suite_for_heterogene20170412-335-12ifuck-libre.pdf?1492034621=&response-content-disposition=inline%3B+filename%3DRodinia_A_benchmark_suite_for_heterogene.pdf&Expires=1755577592&Signature=dSK6YTi5bT3VML8y165f0lSDhfGF2cx8P0DsmitXrjp7II9UVzApOwQjX7bHr4aXLcwfuEXHmaHSH97nPviB3xT~YiGaN3mRTrHh6YBqRLMfT14agMGZwXvFGDHz7jOyHlBmw5tAj16hk6ayVgns8WZQhGYp5oUjauLdsF-o-rGgMKOS0yjtuCilqZAf4Oi85dgHWBcWP5Uq61~fiL0pxq1W6VV5YyXshSeCdQCp1jJ6exPHLaslLSVqDQJ06BmT0762aDbVC3J71UG7Q8A8WaX06h1b7pGqTjOC87KNzFvE9iRMTkKf5v6fiWZ5Y~oyTsecFXIsJHR0jaWd~YSbqA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA
[6] Mingyu Y et al., (2022) Characterizing and Understanding HGNNs on GPUs, arXiv, 2022. [Online]. Available: https://arxiv.org/pdf/2208.04758
[7] Norman P. J et al., (2023) TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings, ACM, 2023. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3579371.3589350
[8] Sparsh M, (2024) A Survey of Techniques for Improving Energy Efficiency in Embedded Computing Systems, arXiv, 2024. [Online]. Available: https://arxiv.org/pdf/2407.09621
[9] Tony S et al., (2016) On the Dark Silicon Automatic Evaluation on Multicore Processors, IEEE 28th International Symposium on Computer Architecture and High Performance Computing, 2016. [Online]. Available: https://www.researchgate.net/profile/Rodolfo-Azevedo/publication/311758831_On_the_Dark_Silicon_Automatic_Evaluation_on_Multicore_Processors/links/5cc5db5fa6fdcc1d49b757cc/On-the-Dark-Silicon-Automatic-Evaluation-on-Multicore-Processors.pdf
[10] Xin G et al., (2010) The performance of occupancy-based lighting control systems: A review," Lighting Res. Technol. 2010. [Online]. Available: https://www.researchgate.net/profile/Clarence-Waters/publication/245385617_The_performance_of_occupancy-based_lighting_control_systems_A_review/links/566de1af08ae430ab5001de9/The-performance-of-occupancy-based-lighting-control-systems-A-review.pdf