International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.4 (2025) pp. 9422-9432
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Running Isolated CPU-Pinned, NUMA-Aligned Workloads and Unpinned Workloads on the Same Hypervisor: A Path Toward Intelligent Infrastructure

Binu Kiliamkavunkal Govindan*

Independent Researcher, USA
* **Corresponding Author Email:** binugovindank@gmail.com - **ORCID:** 0000-0002-5247-7990

Article Info:

DOI: 10.22399/ijcesen.4460
Received : 05 October 2025
Revised : 29 November 2025
Accepted : 05 December 2025

Keywords

NUMA Architecture,
Workload Classification,
Artificial Intelligence Optimization,
Container Orchestration,
Performance Isolation

Abstract:

Modern datacenter infrastructure faces significant challenges when hosting diverse workloads on shared computing resources while maintaining performance guarantees and cost efficiency. This article presents a comprehensive framework for running isolated, CPU-pinned, NUMA-aligned workloads alongside unpinned, flexible workloads on shared hypervisor infrastructure through artificial intelligence-driven optimization techniques. The framework addresses fundamental limitations in current virtualization platforms that fail to exploit NUMA topology information for intelligent workload placement, resulting in performance degradation and resource underutilization. Container-based microservices and telecommunications network functions experience substantial performance penalties from cross-NUMA memory access patterns and unpredictable resource allocation decisions. The proposed solution combines systematic workload classification with machine learning algorithms for predictive placement decisions and dynamic rebalancing capabilities. Implementation involves comprehensive hardware topology discovery, resource partitioning strategies, and integration with existing container orchestration platforms. Performance evaluation demonstrates substantial improvements across telecommunications network functions, edge AI systems, high-performance computing applications, multi-tenant cloud platforms, and future 6G network orchestration scenarios. The framework enables organizations to achieve deterministic performance guarantees for critical applications while maximizing infrastructure utilization through intelligent resource sharing, providing economic benefits through reduced hardware requirements and energy consumption.

1. Introduction and Problem Formulation

1.1 Background and Context-Development of Contemporary Cloud and Datacenter Systems

Today's datacenters have gone from conventional single-purpose hardware to virtualized systems running several workloads on common infrastructure. Modern companies use systems that maximize hardware utilization while satisfying demanding performance standards throughout several kinds of applications. This transformation moves away from dedicated appliances toward software-controlled infrastructure that adapts to operational changes.

Enterprise facilities operate as integrated platforms where resource distribution decisions directly affect cost efficiency and system performance. Single servers now run dozens of virtual machines and containers simultaneously, reducing hardware costs and space requirements while simplifying administration. However, this consolidation creates new challenges for maintaining optimal performance and managing resources effectively [1].

1.2 NUMA Architecture Adoption in Enterprise Servers

Modern enterprise servers use Non-Uniform Memory Access designs to overcome symmetric processing limitations. These systems address bottlenecks occurring when memory access speed cannot match increasing processor core counts in traditional memory configurations. Current implementations feature multiple processor sockets

with dedicated memory controllers, creating separate memory zones with distinct performance characteristics based on data location.

These arrangements establish complex memory hierarchies where access speed varies by physical distance between processors and memory components. Despite widespread NUMA adoption, many virtualization platforms fail to utilize topology information when distributing workloads across available resources [1].

1.3 Mixed Workload Orchestrating Obstacles

Running various workload types on shared infrastructure presents operational problems that grow with greater consolidation. Datacenters simultaneously support time-sensitive network operations, data processing tasks, user applications, analytical workloads, and maintenance functions, each having unique performance requirements and resource patterns.

Standard scheduling focuses on CPU usage without examining how resource placement affects application performance. This works for similar workloads but creates problems when combining applications with different response-time requirements. Platform automation often places applications based solely on available resources without considering effects on existing services [2].

1.4 Performance Penalties in Mixed Workload Environments

Current virtualization systems experience performance losses when running mixed workloads because they cannot handle NUMA features and varying application needs. These issues appear as longer response times, lower processing capacity, higher power consumption, and inconsistent behavior complicating planning efforts.

Problems worsen in dense deployments where workloads compete for shared resources, creating interference patterns hard to predict. Without workload-aware scheduling, critical applications suffer from background processes, leading to service violations and forcing excess infrastructure purchases. Performance becomes unpredictable when systems move workloads between NUMA zones during scaling operations [2].

1.5 Underexploitation of NUMA Topology

Although NUMA architectures are standard in enterprise servers, virtualization platforms rarely use topology information for workload positioning. This affects application performance and operational efficiency significantly. Current algorithms treat processor cores identically, making placement choices based on overall usage rather than memory proximity and NUMA relationships influencing performance.

Poor NUMA utilization extends beyond CPU and memory to device allocation and network interface distribution. Modern servers include multiple adapters and controllers across different NUMA zones, yet platforms seldom consider device location for deployment decisions [2].

1.6 Resource Utilization vs Performance Predictability Trade-offs

Organizations face difficult choices between maximizing infrastructure usage to reduce costs and ensuring predictable performance for service commitments. Traditional methods force decisions between excess hardware provisioning or accepting reduced performance for cost-effective sharing. High utilization creates unpredictable patterns, making consistent service quality difficult for applications needing reliable response times.

1.7 Best Deployment Plans for Mixed Workloads

The primary objective is to develop deployment methods enabling shared workloads on general infrastructure to coexist with isolated, CPU-dedicated, NUMA-optimized ones. This includes developing frameworks and testing across workload types representing actual deployment situations. The study establishes performance standards demonstrating how workload classification and NUMA-aware allocation compare to conventional scheduling.

Research elements include building identification systems that categorize applications based on performance needs and resource characteristics. Advanced resource allocation methods make sure constant performance for vital applications while effectively using leftover capacity for flexible workloads.

1.8 AI-Driven Optimization Union

This area concentrates on building artificial intelligence methods that dynamically adjust resource allocation based on current conditions and observed workload patterns. Combining machine learning with NUMA-aware scheduling improves traditional fixed allocation methods used in production systems.

Algorithms examine application behavior, predict resource needs accurately, and optimize placement to reduce conflicts while maximizing efficiency. The research explores reinforcement learning

methods that enhance optimization decisions based on observed results without extensive retraining.

1.9 Performance Characterization and Benefits Quantification

The research develops measurement methods assessing NUMA-aware optimization benefits across application types and infrastructure setups. Testing includes focused benchmarks isolating specific effects and comprehensive benchmarks showing complete system improvements under realistic conditions.

The study measures economic advantages including cost savings through better resource usage, energy reductions through intelligent power management, and efficiency gains through automated optimization.

1.10 Hypervisor-Level Resource Management Focus

This research targets optimization methods implementable within existing virtualization platforms without major application changes or infrastructure replacement. The hypervisor approach ensures benefits remain invisible to applications while maintaining compatibility with current procedures and management systems.

This method enables systematic optimization across datacenter deployments without extensive coordination between development teams or major software modifications.

1.11 Deterministic Performance with Efficient Utilization

A key contribution shows intelligent workload classification combined with NUMA-aware allocation can simultaneously provide consistent performance guarantees for critical applications and efficient resource usage for flexible workloads. This addresses traditional compromises forcing organizations to choose between performance reliability and cost efficiency.

Systematic resource partitioning based on workload characteristics offers stronger isolation than traditional quota methods while enabling higher utilization through intelligent sharing.

1.12 Practical Implementation Framework

The research provides implementation frameworks enabling organizations to deploy NUMA-aware optimization within existing environments through structured approaches. This includes guidance for infrastructure evaluation, workload classification methods, resource allocation setup, and performance monitoring systems adapting to organizational requirements.

Deployment guidance covers phased implementation strategies allowing gradual optimization adoption without disrupting production workloads, including pilot programs, validation procedures, and migration planning, minimizing risks while showing measurable benefits.

2. Current Challenges and Performance Analysis

2.1 Container Flexibility vs Performance Requirements

Container-based microservices provide excellent flexibility for development teams. Applications are simple to switch between settings, scale independently, and deploy swiftly. This freedom, however, runs counter to performance-critical systems needing consistent response times, such as real-time trading systems.

Container platforms prioritize overall resource usage over individual application needs. A financial trading system might share CPU cores with data analysis jobs, causing costly transaction delays. Organizations must choose between performance predictability or cost-effective resource sharing [3].

2.2 Scheduling Impact on Critical Applications

Container schedulers use basic metrics - CPU usage, memory consumption, storage availability - for placement decisions. They ignore memory locality, cache efficiency, and processor topology that directly affect performance. This oversight particularly impacts real-time applications where memory access patterns determine response times.

Operations teams struggle with performance issues where traditional monitoring shows normal resource usage but applications run slowly. The problem lies in resource placement rather than shortage, making troubleshooting challenging [3].

2.3 Hardware Abstraction Penalties

Container platforms hide hardware details for portability benefits. Although applications run anywhere without changes, this abstraction restricts high-performance applications from using hardware-specific optimizations like NUMA topology awareness or specialized accelerators. For programs that might gain from hardware-specific capabilities, the abstraction level becomes a performance bottleneck.

2.4 NUMA Architecture Performance Issues Memory Access Penalties

Enterprise servers have NUMA designs that make memory access speed dependent on location. Cross-socket access takes longer owing to interconnection latency; local memory access within the same socket is quick. These timing differences create significant performance variations based solely on workload placement.

Applications with large memory requirements suffer most from poor NUMA placement. Database systems, scientific simulations, and analytics platforms experience major slowdowns when data spans multiple NUMA domains [4].

2.5 Hidden Performance Problems

NUMA misalignment creates performance issues undetectable by standard monitoring tools. Traditional metrics show normal CPU and memory usage while applications run slower than expected. This "silent degradation" accumulates across applications, creating system-wide slowdowns that appear as general performance problems.

Operations teams face troubleshooting scenarios where identical workloads perform differently on various servers depending on NUMA placement. Performance variations correlate with system load and scheduling decisions in ways that confuse traditional diagnostic approaches [4].

2.6 Workload Sensitivity Patterns

Different applications show varying NUMA sensitivity. Memory-intensive applications like databases experience severe degradation from misalignment. CPU-intensive applications with small memory footprints show minimal impact. Network processing functions fall between these extremes.

Scientific computing demonstrates particular sensitivity due to large datasets benefiting from consistent memory access speeds. These applications often exceed single NUMA domain capacity, requiring sophisticated placement strategies. Stateless web applications and batch jobs show minimal sensitivity, making them suitable for remaining capacity.

2.7 Utilization vs Performance Conflicts

Hypervisor schedulers give system usage first rank so as to maximize efficiency and keep costs low. To provide high utilization, they distribute loads across hosts by moving workloads. This conflicts with applications needing consistent, predictable resource access.

Critical applications compete with batch jobs for shared resources - memory bandwidth, processor caches, network interfaces. Schedulers treat all workloads equally, not recognizing performance requirements differences. Service agreements suffer when mission-critical applications experience unpredictable performance.

2.8 Migration-Induced Disruption

Live migration enables load balancing but introduces performance unpredictability. Migration processes consume substantial resources and cause temporary degradation for both migrating and co-located applications. Performance-sensitive applications experience sudden slowdowns when migrations consume network bandwidth or storage capacity.

Applications moved to different hosts encounter varying memory topologies, network configurations, and storage characteristics that alter behavior and complicate performance prediction.

2.9 Consolidation Trade-offs

Infrastructure consolidation reduces costs but achieving high ratios without compromising performance requires sophisticated resource management that current platforms lack. Traditional approaches focus on aggregate utilization without considering workload interactions creating performance interference.

High-performance applications needing dedicated resources may degrade when consolidated with best-effort workloads. Current platforms offer only basic controls that cannot address complex mixed environment requirements.

2.10 Performance Measurement Results

Performance testing reveals substantial differences between default scheduling and NUMA-aligned configurations. Default approaches ignore memory locality and topology, resulting in cross-NUMA access patterns that degrade performance. NUMA-aligned configurations show significant improvements through intelligent placement minimizing memory latency.

Memory-intensive applications benefit most from NUMA alignment. Databases achieve faster responses and higher throughput. Scientific applications complete computations quicker with improved bandwidth utilization [3].

2.11 Latency and Throughput Gains

NUMA-aware optimization particularly benefits tail latency characteristics critical for latency-sensitive applications. Default configurations exhibit high tail latency due to cross-NUMA access and resource contention. NUMA-aligned deployments demonstrate improved latency distributions with reduced variability.

Applications achieve higher sustained processing rates through better memory efficiency and reduced contention. Network functions show increased packet processing while databases demonstrate improved transaction rates [4].

2.12 Efficiency Enhancements in Energy Use

NUMA-aware strategies provide power consumption advantages through improved utilization and reduced cross-socket communication. Default configurations result in inefficient power usage due to frequent transfers and suboptimal processor utilization. NUMA-aligned deployments enable better power management by consolidating workloads within domains, allowing unused components to enter low-power states.

These improvements prove valuable in large deployments where power represents substantial expense. NUMA optimization achieves equivalent performance with reduced energy consumption, translating to lower costs and better sustainability metrics.

3. NUMA-Aligned Architecture and AI-Driven Optimization Framework

3.1 Pinned Workloads: Dedicated Resource Strategies

Mission-critical applications require dedicated computing resources through processor core isolation and memory allocation within designated NUMA zones. This strategy proves most effective for systems that demand unwavering performance stability - network communication functions in telecom operations, rapid-response trading platforms, and computational research applications needing reliable processing speeds.

These restricted workloads obtain exclusive access to processing units, memory banks, and peripheral devices within assigned NUMA territories. Such arrangement establishes separated operating spaces that achieve hardware-level performance characteristics. The methodology encompasses processor assignment plus network adapter coordination and storage component positioning, guaranteeing all operational resources function within identical locality boundaries [5].

3.2 Unpinned Workloads: Flexible Resource Sharing

Systems that accommodate performance variations employ adaptive scheduling throughout accessible computing capacity. Data processing operations, system maintenance procedures, web-based services, and testing environments represent this workload category. These applications adjust to fluctuating resource conditions while preserving core operational capabilities.

Non-restricted workloads can extend across several NUMA territories when required for maximum utilization. They gain an advantage from expandable assignment methods that adjust resources spontaneously according to usage requirements. Platform orchestration tools inherently handle these workloads using conventional scheduling methods, while intelligent algorithms improve positioning choices through past usage information [5].

3.3 Resource Topology Management

Successful NUMA enhancement demands thorough hardware structure identification and strategic zone arrangement. This requires documenting connections among processing units, memory management systems, and peripheral components to comprehend performance qualities and connection arrangements.

Structure management establishes separate resource areas according to NUMA properties. Individual locality zones house dedicated processors, memory units, and equipment functioning within identical nodes for peak performance. The framework observes usage trends and executes rebalancing to sustain effective assignment while avoiding resource division [6].

3.4 AI-Driven Scheduling Architecture Real-time Workload Analysis

Artificial intelligence scheduling frameworks constantly examine monitoring information to comprehend application patterns and forecast resource demands precisely. Observation encompasses processor usage, memory interaction sequences, network activity, storage operations, plus application-focused measurements that characterize workload profiles.

Computational learning systems process collected data to recognize trends, allowing precise categorization of incoming workloads according to resemblance with previous implementations. This method adjusts automatically to evolving application patterns without manual setup

modifications. Multi-factor examination evaluates resource accessibility together with performance enhancement possibilities [7].

3.5 Reinforcement Learning for Placement

Advanced scheduling employs comprehensive reinforcement learning to establish ideal positioning approaches through environmental engagement and performance monitoring. Learning systems investigate various positioning methods and obtain responses according to performance indicators, usage effectiveness, and service contract adherence. These systems create guidelines enhancing placement throughout several goals - performance separation, resource effectiveness, power usage, plus operational limitations. This approach manages complicated enhancement challenges with conflicting objectives that traditional rule-oriented systems cannot handle successfully. Methods adjust constantly to evolving workload trends without major retraining [7].

3.6 Dynamic Rebalancing and Optimization

Smart systems observe performance constantly and modify workload positioning automatically to preserve ideal resource assignment. Methods identify performance problems, competition trends, and usage disparities suggesting poor approaches, initiating rebalancing activities without interrupting applications.

Automated enhancement functions continuously behind the scenes, recognizing improvement chances and executing performance-boosting modifications. This procedure includes forecasting abilities expecting future needs and modifying assignment ahead of time. Economic evaluation compares migration advantages against operational expenses and brief performance effects [6].

3.7 Implementation Framework
Hardware Discovery and Resource Division

Implementation begins with complete hardware structure identification documenting processing units, memory managers, and equipment within target systems. Discovery mechanisms employ specialized utilities to generate comprehensive NUMA node diagrams displaying memory interaction properties and equipment connections influencing performance.

Resource separation splits computing resources into separate groups according to structure and organizational needs. This procedure establishes dedicated allocations for essential applications while preserving adaptable groups for expandable workloads. Advanced methods implement layered arrangement allowing detailed management and effective distribution [5].

3.8 Container Platform Integration

The enhancement framework connects smoothly with current container systems without interrupting operational methods or demanding major modifications. Connection employs standard programming interfaces and extension capabilities to incorporate NUMA consciousness into scheduling while preserving implementation procedure compatibility.

Specialized schedulers and managers assess enhancement chances during positioning while honoring resource limits and security requirements. Connection offers workload notation methods allowing operators to indicate needs without application changes. Advanced abilities support specialized resources and operators automating implementation approaches [6].

3.9 Performance Monitoring and Feedback

Complete observation offers continuous enhancement awareness and allows information-based assignment improvement according to measured results. Frameworks gather performance indicators from various sources - performance measurements, application data, network information, plus energy usage providing system effectiveness understanding.

Response methods employ performance information to enhance algorithms and scheduling choices through computational learning adjustment. This procedure examines relationships between assignment choices and results to recognize successful approaches and identify poor strategies. Automated adjustment modifies algorithms according to performance properties without manual involvement [7].

3.10 Optimization Algorithms
Machine Learning for Workload Prediction

Advanced models examine past behavior information to forecast future resource needs allowing proactive enhancement. Forecasting includes usage trends, performance indicators, user interaction information, plus external elements like business patterns affecting workload behavior.

Models employ time sequence examination, neural networks, plus combination methods to understand complicated relationships. The framework executes specialized model categories for various workload types and time periods, supporting immediate

positioning choices and extended capacity planning. Real-time models calculate needs for incoming workloads according to past resemblance [5].

3.11 Decision Trees and Constraint Management

Sophisticated algorithms execute decision structure frameworks and limitation satisfaction balancing several competing goals while ensuring policy adherence. Decision structures encode complicated positioning reasoning considering workload properties, resource accessibility, performance needs, plus enhancement objectives.

Limitation satisfaction resolves complicated enhancement challenges considering performance separation, usage effectiveness, energy reduction, plus operational needs. The framework employs integer programming, genetic methods, plus simulated annealing to recognize near-ideal solutions for multi-objective challenges [6].

3.12 Energy-Aware Consolidation

Smart consolidation executes energy enhancement approaches reducing power usage while preserving performance needs. Energy enhancement considers fixed consumption from inactive resources plus changing consumption varying with workload properties.

The framework combines compatible applications onto minimum hardware while ensuring separation and performance assurances. Power control coordinates with hardware management allowing aggressive conservation through frequency adjustment and power conditions without compromising responsiveness. Forecasting abilities expect demand trends and modify power configurations ahead of time [7].

4. Applications and Use Case Analysis

4.1 Telecommunications Network Functions (5G/6G)

VNF Performance Requirements

Telecommunication operators struggle with a fundamental shift from purpose-built hardware to shared virtualized platforms. Communication functions previously running on custom appliances now must achieve identical speed and reliability while competing for shared computing resources. This transition presents significant obstacles for systems managing voice communications, video distribution, and data transport requiring stable processing capabilities.

Modern virtualized communication functions handle extensive packet loads measured in millions per second while sustaining response intervals counted in microseconds. Core processing elements like User Plane handlers and Session Control modules demand assured resource availability to avoid degradation during traffic surges. Conventional virtualization methods fail to satisfy these demanding timing constraints [8].

4.2 Packet Processing Optimization

NUMA-coordinated implementations address packet handling bottlenecks by maintaining processor units, memory banks, and networking hardware within identical physical zones. Typical virtualized arrangements experience memory retrieval delays when data buffers cross multiple NUMA boundaries, producing irregular timing fluctuations during heavy traffic conditions.

Enhanced configurations prevent cross-boundary memory operations by clustering all communication function components within unified NUMA domains. This methodology allows network operations to manage increased traffic loads without extra server hardware, boosting economic efficiency while achieving performance objectives. The technique encompasses both data reception and forwarding within consolidated memory areas [8].

4.3 RAN Implementation Economic Benefits

Virtualized Radio Access implementations showcase notable financial benefits through NUMA enhancement approaches. Conventional radio systems demand individual hardware per function, generating elevated expenses and minimal equipment usage. Virtual deployments permit several radio operations to share computing infrastructure while preserving performance isolation.

Financial evaluation reveals considerable expense reductions via enhanced equipment concentration and power savings versus dedicated hardware approaches. Network operators can combine multiple radio operations onto reduced server counts while maintaining performance commitments, decreasing both hardware acquisition and operational overhead [9].

4.4 Edge AI and Real-Time Inference Systems

Customer Application Response Requirements

Edge computing sites accommodate machine intelligence applications demanding reliable swift responses for satisfactory user interactions. Image

analysis platforms, natural language processing systems, and suggestion engines require reaction times below particular millisecond limits for fluid user engagement. Edge installations encounter resource restrictions compared to centralized cloud infrastructures, making productive utilization vital.

NUMA-aware implementation proves particularly beneficial for edge intelligence applications by keeping model parameters, input datasets, and computational operations within matching memory regions. Edge processing typically employs extensive neural networks demanding considerable memory benefiting from stable access velocities and ideal cache utilization configurations [9].

4.5 Edge Computing Performance Enhancement

Performance improvement in edge settings demands balancing computational effectiveness against resource constraints to optimize concurrent inference operations while maintaining satisfactory reaction speeds. Edge facilities function with limited computational capacity versus central installations, making efficient resource employment essential for supporting multiple simultaneous applications and users.

Enhancement encompasses individual inference activities plus batch processing methods combining several requests for improved effectiveness while preserving rapid responses for separate operations. NUMA coordination enables superior batch processing by guaranteeing computational resources stay within matching locality regions [9].

4.6 Power Efficiency in Limited Deployments

Energy effectiveness constitutes a vital objective for edge computing installations where power usage directly influences operational expenses and implementation feasibility in distant locations with restricted electrical systems. Edge intelligence applications require intensive neural network calculations consuming substantial energy, making enhancement crucial for economically viable deployments.

NUMA-aware enhancement minimizes unnecessary information movement between processor units and memory controllers, removing power consumption from cross-region communication expenses. Resource consolidation via NUMA enhancement achieves comparable performance with reduced active hardware elements, permitting idle processors to access power-conservation modes [9].

4.7 High-Performance Computing Integration Scientific Computing Properties

Research computing applications demonstrate specific properties making them especially appropriate for NUMA enhancement, including extensive memory demands, intensive calculations, and responsiveness to memory access delays influencing total performance. Weather modeling, fluid mechanics, molecular research, and genetic analysis require continuous computing capability with predictable resource access for acceptable completion periods.

Research applications frequently maintain information collections surpassing individual NUMA region capacity, demanding sophisticated memory positioning methods minimizing cross-region access while productively employing computational resources. Numerous research applications utilize parallel processing frameworks enhanced through NUMA-aware thread positioning and memory distribution approaches [8].

4.8 Combined Workload Coexistence

Research computing settings benefit from mixed implementation strategies permitting operational infrastructure to coexist with scientific applications without mutual disruption. Traditional research implementations frequently isolate computational activities from monitoring and management services on separate systems, producing poor total utilization and elevated expenses.

NUMA-aware enhancement enables intelligent workload coexistence via strong performance isolation between essential research calculations and supporting operational systems through dedicated resource distribution within particular locality regions. This permits organizations to accomplish improved total infrastructure usage while preserving performance commitments for critical research applications [9].

4.9 Performance Restoration Approaches

Intelligent NUMA distribution enables research applications to regain substantial performance improvements from current infrastructure without hardware enhancements or application modifications. Numerous research settings experience inadequate resource distribution causing frequent cross-region memory access, creating restrictions limiting computational velocity and extending task completion periods.

Restoration techniques encompass memory allocation strategies for applications with working information exceeding individual region capacity, enabling productive memory usage while reducing remote access costs. Sophisticated distribution algorithms analyze application memory

configurations to establish optimal positioning approaches balancing locality with utilization productivity [9].

4.10 Multi-Tenant Cloud Platform Optimization Customer Separation and Concentration

Multi-tenant cloud frameworks must deliver strong performance isolation between customer processing while maximizing infrastructure employment for competitive pricing and profitability. Traditional isolation techniques employ resource boundaries and priorities offering restricted protection against disruption when multiple customers compete for shared memory and processor resources.

NUMA-aware isolation approaches provide enhanced performance commitments by distributing dedicated regions to individual customers or processing categories, removing memory conflicts and guaranteeing predictable properties regardless of other customer operations. This enables cloud suppliers to increase customer consolidation without compromising service commitments [8].

4.11 Anticipatory Resource Positioning

Anticipatory positioning employs machine learning evaluation of customer usage patterns to enhance resource distribution before processing deployment, minimizing startup delays and improving customer satisfaction. Cloud frameworks gather comprehensive usage information enabling precise prediction of future resource requirements, permitting pre-positioning of resources in ideal configurations before customer application implementation [9].

4.12 Service Agreement Adherence in Dense Implementations

Service commitment compliance in high-concentration situations demands advanced resource management preventing performance deterioration as consolidation increases. NUMA-aware enhancement enables elevated concentrations while sustaining compliance through intelligent placement preventing resource disputes and guaranteeing predictable properties.

The framework executes continuous observation and automatic modification identifying potential violations before affecting customers, activating rebalancing sustaining compliance without manual involvement [9].

4.13 6G Network Orchestration Requirements AI-Integrated Network Implementation

Sixth-generation wireless networks incorporate artificial intelligence as fundamental architecture rather than supplementary service, demanding infrastructure enhancement supporting both traditional network operations and intelligence-controlled services with reliable performance. Implementation requires sophisticated coordination considering computational requirements of learning algorithms alongside traditional processing operations [8].

4.14 Distributed Microservice Coordination

6G structures rely on distributed microservice implementations across numerous geographic positions requiring coordinated enhancement for reliable service quality and productive utilization. Geographic distribution introduces complexity as network delays and bandwidth restrictions must be evaluated alongside local distribution choices [9].

4.15 Communication-Computing Joint Optimization

6G networks require integrated approaches, simultaneously evaluating communication infrastructure and computing resource distribution for ideal system-wide performance. Traditional enhancement concentrates separately on communication and computing elements, but convergence demands comprehensive methods evaluating interdependencies between network performance and computational distribution [9].

Table 1: *NUMA Architecture Performance Impact Compariso*

Workload Category	NUMA-Misaligned Performance	NUMA-Aligned Performance
Memory-Intensive Applications	Severe degradation with cross-socket access	Optimal performance with local memory access
CPU-Intensive Applications	Minimal performance impact	Consistent processing speeds

Network Processing Functions	Variable performance based on packet volumes	Predictable packet processing latency

n.

Table 2: *Workload Classification and Resource Allocation Strategies. [5, 6]*

Workload Type	Resource Allocation Strategy	Performance Characteristics
Pinned Workloads	Dedicated CPU cores and memory binding	Near-bare-metal performance with isolation
Unpinned Workloads	Dynamic scheduling across domains	Flexible resource sharing with elasticity
Mixed Deployments	Hierarchical resource organization	Balanced performance and utilization

Table 3: *AI-Driven Optimization Components and Functions. [7]*

Optimization Component	Primary Function	Key Benefits
Real-time Workload Analysis	Telemetry data processing and pattern recognition	Accurate workload classification and prediction
Reinforcement Learning	Predictive placement decision development	Adaptive optimization without manual tuning
Dynamic Rebalancing	Automated performance monitoring and adjustment	Continuous optimization with minimal disruption

Table 4: *Application Domains and Performance Benefits. [9]*

Application Domain	Primary Performance Challenge	NUMA Optimization Benefit
Telecommunications (5G/6G)	Microsecond-level latency requirements	Consistent packet processing and reduced jitter
Edge AI Systems	Resource-constrained inference processing	Improved throughput and energy efficiency
High-Performance Computing	Large memory footprint management	Performance recovery and workload coexistence

5. Conclusions

The deployment of mixed workloads on shared hypervisor infrastructure represents a fundamental challenge that requires sophisticated resource management techniques beyond traditional virtualization approaches. This article demonstrates that an intelligent combination of NUMA-aware resource allocation with AI-driven optimization can simultaneously deliver deterministic performance guarantees for critical applications and efficient resource utilization for elastic workloads. The framework addresses the historical trade-off between performance predictability and cost efficiency through systematic workload classification and machine learning-enhanced scheduling decisions. Performance evaluation across telecommunications network functions, edge AI systems, high-performance computing integration, multi-tenant cloud platforms, and 6G network orchestration validates the practical benefits of NUMA-aware optimization. Implementation results show substantial improvements in application response times, throughput characteristics, and energy efficiency while enabling higher consolidation ratios without service-level agreement violations. The economic benefits include reduced infrastructure requirements, lower operational costs, and improved environmental sustainability metrics that justify implementation investments. Organizations can deploy the framework within existing virtualization environments through structured approaches that minimize operational disruption while providing measurable performance improvements. Future developments in processor architectures and AI algorithms will continue to enhance optimization capabilities, making NUMA-aware scheduling essential for next-generation infrastructure supporting diverse application portfolios with varying performance requirements.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Andi Kleen, SUSE Labs, "An NUMA API for Linux," Novell Inc., 2004. [Online]. Available: https://halobates.de/numaapi3.pdf

[2] GeeksforGeeks, "Locality of Reference," 2025. [Online]. Available: https://www.geeksforgeeks.org/computer-organization-architecture/locality-of-reference-and-cache-operation-in-cache-memory/

[3] Brendan Burns, et al., "Kubernetes: Up and Running, 3rd Edition," O'Reilly Media, 2022. [Online]. Available: https://www.oreilly.com/library/view/kubernetes-up-and/9781098110192/

[4] Mohammad Dashti et al., "Traffic management: a holistic approach to memory placement on NUMA systems," *ACM Digital Library*, 2013. [Online]. Available: https://dl.acm.org/doi/10.1145/2451116.2451157

[5] Kubernetes Documentation, "Utilizing the NUMA-aware Memory Manager," 2024. [Online]. Available: https://kubernetes.io/docs/tasks/administer-cluster/memory-manager/

[6] Yiannis Georgiou, et al., "Topology-aware resource management for HPC applications," ACM Digital Library, 2017. [Online]. Available: https://dl.acm.org/doi/10.1145/3007748.3007768

[7] Xin Li et al., "Topology-Aware Scheduling Framework for Microservice Applications in Cloud," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2022 [Online]. Available: https://cis.temple.edu/~wu/research/publications/Publication_files/Topology-Aware_Scheduling_Framework_for_Microservice_Applications_in_Cloud.pdf

[8] Wen-Ping Lai, Kuan-Chun Chiu, "NUMAP: NUMA-aware Multi-core Pinning and Pairing for Network Slicing at the 5G Mobile Edge," *IEEE Xplore*, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9023272

[9] Gia Khanh Tran et al., "WS 16 - AI-Driven Digital Twin Orchestration for 6G Networks: Paving the Way for Future Connectivity," IEEE PIMRC 2025. [Online]. Available: https://pimrc2025.ieee-pimrc.org/workshop/ws-16-ai-driven-digital-twin-orchestration-6g-networks-paving-way-future-connectivity