A Survey on Model Training RDMA Network Topology Congestion Control Load Balancing AI Infrastructure Distributed Computing and Data Center Optimization

www.surveyx.cn

Abstract

This survey paper provides a comprehensive examination of the complex processes and technologies integral to the optimization of AI systems and data centers. Key areas explored include model training, Remote Direct Memory Access (RDMA), network topology, congestion control, load balancing, AI infrastructure, distributed computing, and data center optimization. The paper highlights the significance of scalable AI frameworks in addressing computational demands and the role of digital twin technology in resource allocation. It delves into challenges and advancements in training large-scale models, emphasizing distributed computing's role in enhancing training efficiency. RDMA's benefits for efficient data transfer and its deployment challenges are analyzed, alongside innovations in network topology design for AI applications. The survey also examines congestion control mechanisms, highlighting machine learning approaches, and explores load balancing strategies, including adaptive and energy-efficient techniques. AI and machine learning's impact on data center optimization is discussed, focusing on energy efficiency and environmental sustainability. The paper concludes by synthesizing insights and suggesting future research directions to further enhance AI system and data center performance. These findings underscore the interconnectedness of the discussed technologies and processes, offering a foundation for continued innovation in optimizing modern computing environments.

1 Introduction

1.1 Significance of AI Systems and Data Centers

AI systems and data centers are foundational to the evolution of modern technology, underpinning diverse industries through scalable and distributed frameworks that enhance the efficiency of deep learning applications. These infrastructures address the significant computational demands of AI, facilitating the development of advanced models and optimizing task allocation and performance management across sectors such as healthcare, finance, and manufacturing [1, 2].

The rapid proliferation of the Internet of Things (IoT), with forecasts suggesting 125 billion connected devices by 2030, amplifies the need for robust AI systems capable of managing vast data volumes and ensuring efficient data processing [3]. The integration of small cell Base Stations (SBSs) within 5G networks is essential for handling this data surge and achieving ultra-low latency, highlighting the critical role of these infrastructures in contemporary technology landscapes [4].

Effective resource management at the network edge is vital, particularly in federated learning, which decentralizes data processing and enhances resource allocation, thereby improving operational efficiency and addressing latency and privacy concerns in AI applications [5].

Generated by SurveyX.cn.

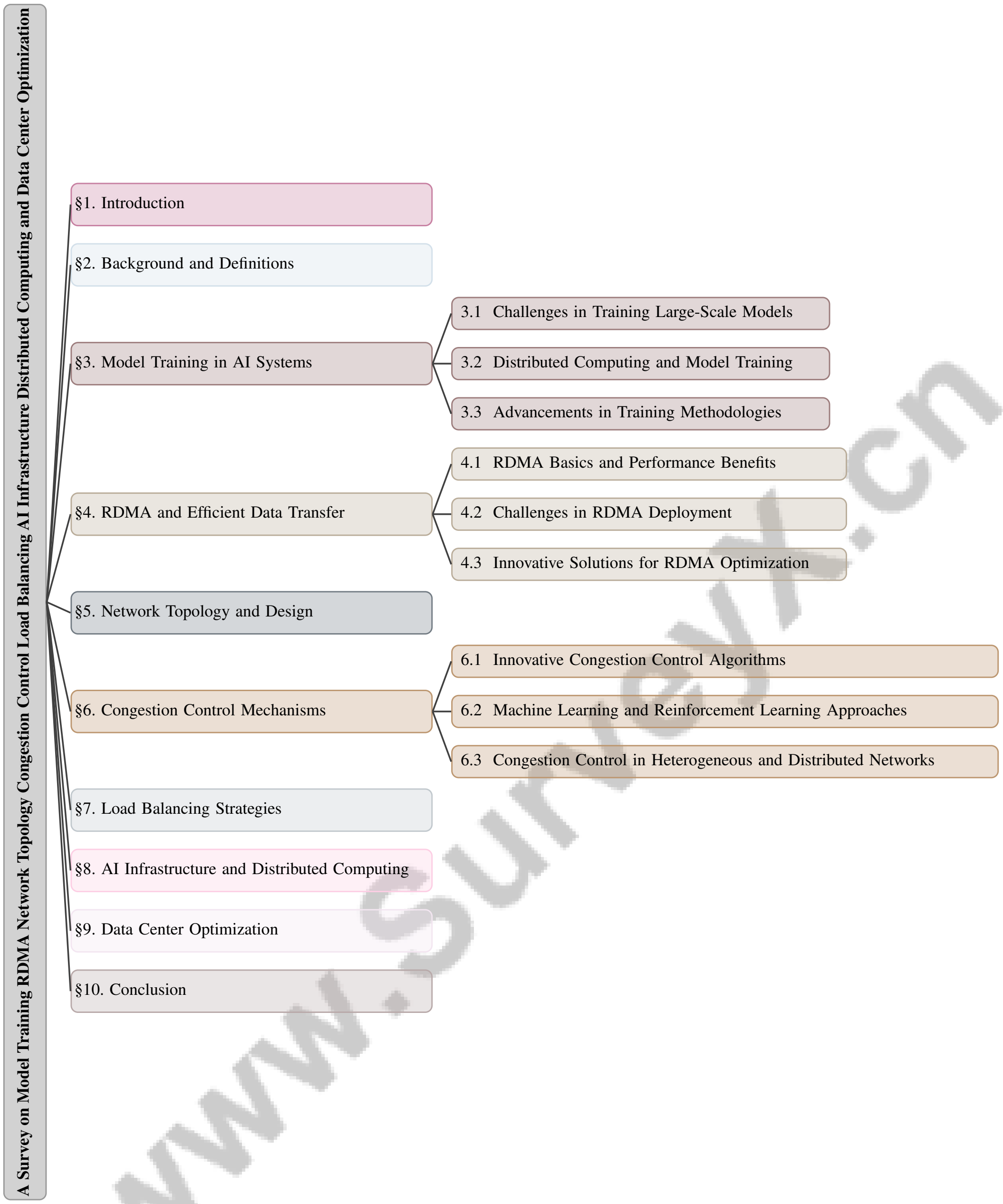

A Survey on Model Training RDMA Network Topology Congestion Control Load Balancing AI Infrastructure Distributed Computing and Data Center Optimization
§1. Introduction
§2. Background and Definitions
§3. Model Training in AI Systems
3.1 Challenges in Training Large-Scale Models
3.2 Distributed Computing and Model Training
3.3 Advancements in Training Methodologies
§4. RDMA and Efficient Data Transfer
4.1 RDMA Basics and Performance Benefits
4.2 Challenges in RDMA Deployment
4.3 Innovative Solutions for RDMA Optimization
§5. Network Topology and Design
§6. Congestion Control Mechanisms
6.1 Innovative Congestion Control Algorithms
6.2 Machine Learning and Reinforcement Learning Approaches
6.3 Congestion Control in Heterogeneous and Distributed Networks
§7. Load Balancing Strategies
§8. AI Infrastructure and Distributed Computing
§9. Data Center Optimization
§10. Conclusion

Figure 1: chapter structure

1.2 Role of Digital Twin Technology

Digital Twin technology significantly enhances AI systems and data centers by creating virtual representations of physical entities, which optimize resource allocation and boost network efficiency. In Industrial Internet of Things (IIoT) networks, Digital Twins facilitate the simulation and analysis of complex systems, enabling effective resource management and predictive maintenance strategies [6]. This technology supports real-time data integration and analysis, essential for improving operational efficiency and minimizing downtime in data centers.

The interplay between Digital Twin technology and edge computing is crucial for addressing the demands of high-volume data processing and low-latency services. Edge computing decentralizes data processing for applications like smart traffic systems and wind farms, thereby reducing latency and enhancing AI responsiveness [3]. By leveraging Digital Twins, data centers can simulate various

AI-generated, for reference only.

operational scenarios, optimizing infrastructure to meet the dynamic requirements of AI applications and improving performance and predictive capabilities.

1.3 Structure of the Survey

This survey is structured to provide an in-depth overview of the key components and technologies essential to AI systems and data centers. It begins with an introduction to the significance of these infrastructures, followed by a discussion on the role of Digital Twin technology in enhancing operational capabilities. The subsequent sections outline foundational concepts and definitions, establishing a clear understanding of the core technologies addressed throughout the paper.

The survey further examines model training, focusing on the challenges and advancements in training large-scale AI models, alongside the role of distributed computing in improving training efficiency. It discusses Remote Direct Memory Access (RDMA), emphasizing its advantages in enhancing data transfer efficiency while addressing the complexities of its implementation. RDMA allows direct server memory access, potentially bypassing CPU overhead, but requires significant software redesign, raising safety and efficiency concerns. In contrast, traditional server-reply models are limited in input/output operations per second (IOPS). Innovations like the Remote Fetching Paradigm (RFP) demonstrate that in-bound RDMA-read operations can achieve performance improvements of 160

Subsequent sections address congestion control mechanisms and load balancing strategies, examining their applications in optimizing resource utilization and performance. The integration of AI infrastructure and distributed computing is scrutinized for its role in enhancing AI performance. Finally, the survey discusses data center optimization strategies, focusing on how AI and machine learning contribute to operational efficiency and energy sustainability.

The conclusion synthesizes insights from the survey, emphasizing the intricate relationships among the discussed technologies, particularly the convergence of Edge Computing and Artificial Intelligence (AI) in emerging wireless networks like 6G. This convergence, termed Edge Intelligence, addresses critical challenges such as latency, energy consumption, and privacy concerns, while outlining potential future research directions to enhance the efficiency and security of these systems through innovative design principles and decentralized machine learning models [7, 8]. This structured approach ensures a comprehensive understanding of the complex dynamics involved in optimizing AI systems and data centers.The following sections are organized as shown in Figure 1.

2 Background and Definitions

2.1 Core Concepts in AI and Data Centers

AI and data centers are central to modern computational frameworks, enabling efficient data processing across various applications. Optimizing network structures is essential for managing high-dimensional data and complex topologies, particularly in distributed machine learning where resource allocation and task scheduling are crucial [5]. Reinforcement Learning (RL) significantly influences network protocol development, underscoring its impact on AI systems and data center operations [9].

Training large AI models demands substantial computational resources, highlighting the need for scalable solutions that integrate edge computing to enhance data processing at the network edge, facilitating real-time decision-making without relying solely on centralized cloud resources [10]. Resource management in data centers involves application, virtual machine (VM), and physical machine management, using techniques like task scheduling, VM migration, and resource provisioning to optimize utilization [5].

Traditional tightly coupled compute and memory architectures limit scalability and resource utilization in data centers. This can be alleviated by exploring distributed shared memory databases, offering a scalable resource management approach [11]. Key concepts such as federated learning, edge resources, load balancing, and energy efficiency are vital for understanding AI systems and data center integration and optimization [5].

Machine learning applications in data center networking involve flow prediction, flow classification, load balancing, resource management, routing optimization, and congestion control, enhancing performance and scalability while addressing communication overheads in distributed systems

AI-generated, for reference only.

[9]. Efficiently configuring distributed computing pipelines connecting scientific instruments with computational resources is crucial for effective data processing [10].

2.2 Foundational Setup for AI Applications

Deploying AI applications requires sophisticated infrastructure for managing distributed systems, resource allocation, and energy consumption. Optimizing distributed machine learning frameworks like Federated Learning (FL), Split Learning (SL), and Split Federated Learning (SFL) is critical, considering their distinct trade-offs in training duration, communication overhead, and resource utilization [12].

Effective resource management in cloud environments prevents underutilization or overload through workload forecasting and dynamic resource allocation strategies that adapt to AI applications' evolving demands [13]. Machine learning techniques in Virtual Machine Placement (VMP) further optimize virtual resource allocation [14].

Addressing inefficiencies in distributed communications, particularly related to socket and TCP features, ensures smooth data flow across networks [15]. Network Functions Virtualization (NFV) enhances this by enabling dynamic resource allocation, optimizing traffic flow through network middleboxes, and increasing AI deployment flexibility [16].

Energy efficiency is crucial, especially in green computing initiatives. Implementing scheduling policies that consider renewable energy variability minimizes the carbon footprint of AI applications [17]. Methods like Delay-Optimal Forwarding and Computation Offloading (DOFCO) optimize data forwarding and computation processes, contributing to energy-efficient AI operations [18].

Dynamic load balancing in extended cloud networks with heterogeneous hardware and IoT elements is vital for managing diverse communication loads. Genetic algorithms and other innovative strategies ensure efficient network resource utilization [19]. Real-time processing of large data volumes from scientific instruments is essential for extracting meaningful insights while minimizing resource usage [20].

Memory disaggregation (MD) is pivotal in scalable and elastic data center design, decoupling compute (CPU) from memory to enhance memory utilization and reduce costs [11]. This approach is complemented by managing live migrations in cloud and edge environments, ensuring optimal resource utilization and service quality [21].

The foundational setup for AI applications demands a multifaceted approach, incorporating advanced resource management, robust communication protocols, and energy-efficient strategies through scalable, distributed frameworks leveraging cloud and edge computing. This comprehensive strategy addresses AI's increasing computational demands while optimizing data storage, model training, and deployment, ensuring enhanced performance with minimized latency and energy consumption [7, 8, 1, 22]. This robust infrastructure is essential for supporting AI technologies' growing demands across various sectors.

3 Model Training in AI Systems

The training of models in artificial intelligence is a resource-intensive process marked by various challenges and methodologies. This section delves into the complexities of model training, emphasizing the obstacles encountered in training large-scale AI systems, particularly the computational and resource constraints that significantly affect efficiency and scalability. As illustrated in Figure 2, the hierarchical structure of model training in AI systems highlights the main challenges, distributed computing solutions, and advancements in training methodologies. This figure categorizes the computational and resource constraints, optimization strategies, parallel execution methodologies, and innovative systems that enhance the efficiency and scalability of AI model training, thereby providing a comprehensive overview of the critical factors influencing this intricate process.

3.1 Challenges in Training Large-Scale Models

Training large-scale AI models poses significant challenges, primarily due to computational and resource constraints that impede efficiency and scalability. As illustrated in Figure 3, these challenges

AI-generated, for reference only.

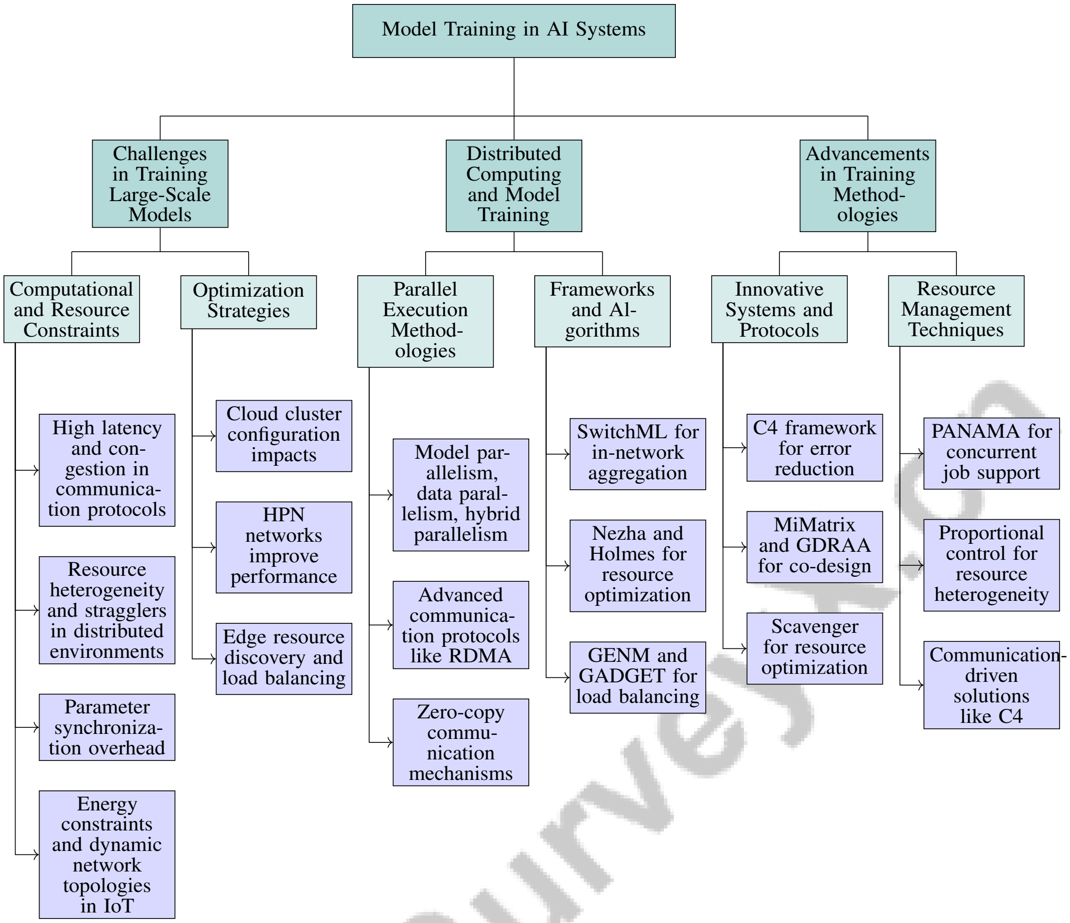

Model Training in AI Systems
Challenges in Training Large-Scale Models
Distributed Computing and Model Training
Advancements in Training Methodologies
Computational and Resource Constraints
Optimization Strategies
Parallel Execution Methodologies
Frameworks and Algorithms
Innovative Systems and Protocols
Resource Management Techniques
High latency and congestion in communication protocols
Resource heterogeneity and stragglers in distributed environments
Parameter synchronization overhead
Energy constraints and dynamic network topologies in IoT
Cloud cluster configuration impacts
HPN networks improve performance
Edge resource discovery and load balancing
Model parallelism, data parallelism, hybrid parallelism
Advanced communication protocols like RDMA
Zero-copy communication mechanisms
SwitchML for in-network aggregation
Nezha and Holmes for resource optimization
GENM and GADGET for load balancing
C4 framework for error reduction
MiMatrix and GDRAA for co-design
Scavenger for resource optimization
PANAMA for concurrent job support
Proportional control for resource heterogeneity
Communication-driven solutions like C4

Figure 2: This figure illustrates the hierarchical structure of model training in AI systems, highlighting the main challenges, distributed computing solutions, and advancements in training methodologies. It categorizes the computational and resource constraints, optimization strategies, parallel execution methodologies, and innovative systems that enhance the efficiency and scalability of AI model training.

can be categorized into three main areas: communication protocols, resource management, and model aggregation. Each category highlights key issues such as high latency in communication, resource heterogeneity in distributed environments, and the complexities of aggregating models from IoT devices using satellite networks. Inefficiencies in current intra-host and inter-host communication protocols result in high latency and congestion during data transfers between processing units and memory [23]. These issues are compounded in distributed machine learning environments, where resource heterogeneity among servers leads to stragglers and performance degradation [24]. Parameter synchronization in distributed deep neural network (DNN) training creates communication overhead that limits scalability, particularly in traditional TCP/IP communication methods, which contribute to high latency in model-serving pipelines, especially in edge computing contexts [25, 26]. Additionally, aggregating models from distributed IoT devices in remote areas using LEO satellites is complicated by energy constraints and dynamic network topologies [27].

Selecting appropriate cloud cluster configurations significantly impacts both training time and costs in distributed machine learning tasks [28]. The dynamic nature of IoT workloads and a lack of centralized control lead to inefficient service placement and load imbalances [2]. HPN networks have enhanced large language model (LLM) training performance by 14.9

AI-generated, for reference only.

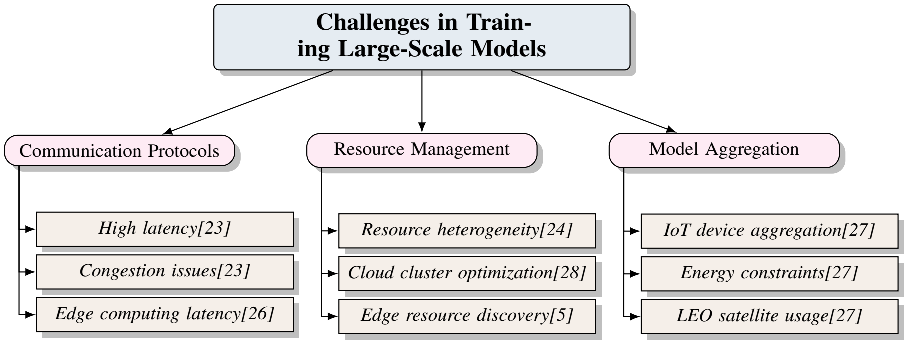

Challenges in Training Large-Scale Models
Communication Protocols
Resource Management
Model Aggregation
High latency[23]
Congestion issues[23]
Edge computing latency[26]
Resource heterogeneity[24]
Cloud cluster optimization[28]
Edge resource discovery[5]
IoT device aggregation[27]
Energy constraints[27]
LEO satellite usage[27]

Figure 3: This figure illustrates the primary challenges in training large-scale AI models, categorized into communication protocols, resource management, and model aggregation. Each category highlights key issues such as high latency in communication, resource heterogeneity in distributed environments, and the complexities of aggregating models from IoT devices using satellite networks.

3.2 Distributed Computing and Model Training

Distributed computing is vital for improving the efficiency and scalability of AI model training by enabling parallel execution of computational tasks across multiple nodes. This paradigm overcomes the limitations of single-node environments, facilitating large-scale model training through methodologies such as model parallelism, data parallelism, and hybrid parallelism [29]. As models and hardware scale, managing communication overhead becomes critical, necessitating innovative solutions [30].

Advanced communication protocols like Remote Direct Memory Access (RDMA) are particularly beneficial in distributed settings, allowing direct GPU communication and bypassing CPU intervention to enhance communication efficiency and reduce latency [31]. Zero-copy communication mechanisms for tensor transfers further improve data transfer speeds compared to traditional RPC-based methods [32]. Frameworks like SwitchML utilize programmable switches for in-network aggregation of model updates, effectively reducing data exchange volumes among workers and accelerating distributed machine learning training [33]. Nezha enhances allreduce performance by coordinating data transfer across multiple network interfaces, ensuring efficient resource utilization [34]. Holmes optimizes task scheduling across heterogeneous NIC environments, ensuring efficient GPU resource utilization [35].

Communication-driven solutions like C4 optimize parallel training by minimizing downtime and enhancing communication efficiency in large-scale AI clusters [36]. The GENM algorithm illustrates the role of distributed computing in optimizing model training processes by improving load balancing and resource allocation in Small Base Stations (SBSs) [4]. Additionally, frameworks like GADGET employ greedy ring-all-reduce techniques to optimize resource scheduling for distributed deep learning jobs [37]. In high-performance distributed systems, methods like DoubleClimb select optimal learning and information nodes while minimizing overall learning costs [38].

The co-designed peta-scale heterogeneous cluster 'Manoa', coupled with the job server framework 'MiMatrix', exemplifies the optimization of resource usage and acceleration of deep learning training [39]. Distributed shared-memory databases (DSM-DB) leverage memory disaggregation to achieve improved performance and scalability, addressing inefficiencies in tightly coupled compute and memory architectures [11]. Scavenger, a cloud service, optimizes training cost and time by searching for optimal configurations in a black-box manner, leveraging performance models [28]. NetDAM provides a programmable architecture for direct memory access over Ethernet, facilitating efficient data processing and communication without the latency of traditional protocols [23].

Distributed computing is crucial for advancing AI model training, offering robust infrastructure to meet the substantial computational requirements of large-scale models. It enhances resource utilization efficiency and reduces energy consumption through optimized parallel and distributed training techniques, including model partitioning, effective communication across computing clusters, and orchestration of hardware accelerators like GPUs. Leveraging cloud computing also facilitates scalable AI frameworks, improving data management, load balancing, and cost-effective deployment, ultimately driving the performance and efficiency of deep learning applications [30, 1]. These advancements ensure AI systems can meet the growing demands of diverse and resource-constrained environments, enhancing the scalability and efficiency of model training processes.

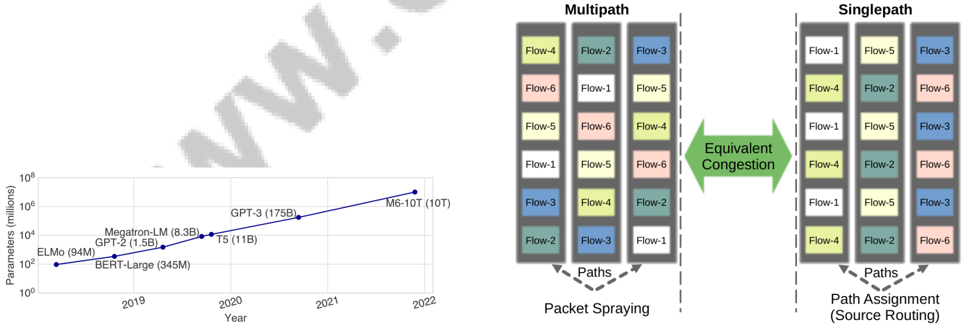
3.3 Advancements in Training Methodologies

Recent advancements in AI model training methodologies have significantly improved scalability, efficiency, and resource management through innovative systems and protocols. The C4 framework, for instance, reduces error-induced overhead by approximately 30

The integration of hardware and software co-design, as exemplified by the MiMatrix framework and GDRAA AllReduce algorithm, marks a significant leap forward, achieving O(1) computation and handshake messages that improve distributed training efficiency by minimizing communication overhead and synchronization delays [39]. The Scavenger cloud service optimizes resource allocation for machine learning training, reducing training times by over 2× and costs by more than 50

Innovative approaches, such as proportional control and PID controller concepts, address resource heterogeneity in distributed clusters by determining stable mini-batch sizes, enabling efficient training on clusters with diverse CPU and GPU resources [24]. These methodologies are essential for optimizing training processes in heterogeneous computing environments, ensuring efficient AI model training despite variations in resource availability.

Recent advancements in AI model training represent a coordinated effort to tackle complex challenges in this field. By integrating innovative methodologies and frameworks, these efforts focus on optimizing resource utilization, enhancing scalability, and improving energy efficiency. Scalable, distributed AI frameworks utilizing cloud computing have significantly boosted deep learning performance while addressing data management and security concerns. The strategic orchestration of thousands of GPUs in large-scale training environments has revealed critical insights into hardware configuration and parallelization strategies essential for maximizing computational efficiency. Communication-driven solutions like C4 have effectively reduced resource wastage and improved runtime performance by streamlining the identification of hardware malfunctions and minimizing network congestion. Collectively, these strategies highlight the importance of a holistic approach to AI infrastructure that combines advanced hardware, optimized software, and robust telemetry to meet the demands of modern AI workloads [30, 1, 36, 22]. These developments are crucial for addressing the increasing demands of AI applications across diverse and resource-constrained environments.



(a) Parameters of Language Models Over Time[29]

(b) Multipath vs. Singlepath: Understanding Path Assignment in Network Design[40]

Figure 4: Examples of Advancements in Training Methodologies

As illustrated in Figure 4, advancements in training methodologies have played a pivotal role in enhancing AI systems' capabilities and efficiency. The first example, "Parameters of Language Models Over Time," highlights the exponential growth in the number of parameters in language models from 2019 to 2022, underscoring the trend toward larger and more complex models, such as ELMo, which have significantly improved natural language processing tasks. The second example, "Multipath vs. Singlepath: Understanding Path Assignment in Network Design," illustrates the comparative analysis of two network design approaches—multipath and singlepath—emphasizing the importance of path assignment in optimizing network performance. Together, these examples

AI-generated, for reference only.

provide insight into the innovative techniques shaping the future of AI model training and network design [29, 40].

4 RDMA and Efficient Data Transfer

4.1 RDMA Basics and Performance Benefits

Remote Direct Memory Access (RDMA) significantly enhances data center operations by facilitating direct memory access between computers without CPU intervention, thereby reducing latency and CPU overhead. This is particularly beneficial in high-performance computing environments requiring rapid data transfer. RDMA, when integrated with hardware-accelerated transport mechanisms, further improves performance by bypassing traditional CPU involvement, offering reduced latency and enhanced throughput compared to conventional TCP-based methods [26]. The effectiveness of RDMA is heightened by protocols like Remote Direct Memory Access over Converged Ethernet (RoCE), which supports high-speed data transfers in distributed training environments. Efficient communication is crucial for distributed deep learning, where minimizing overhead is essential. For example, NetReduce aggregates gradients at the network switch, effectively reducing communication overhead in distributed training [25].

RDMA supports self-modifying programs through its RDMA verbs interface, allowing for complex offloads without hardware changes, thereby enhancing load balancing and performance optimization. The RedN framework utilizes this capability to achieve latency reductions of up to 2.6 times for key-value operations compared to traditional methods, while maintaining performance isolation [41, 42, 43]. Advanced implementations like RDMAbox introduce optimizations that reduce I/O operations, improving workload completion efficiency.

RDMA's integration with container network solutions such as TSoR allows applications using traditional POSIX socket interfaces to benefit from RDMA's performance enhancements without code modifications. Solutions like Virtuoso multi-path RDMA, the Remote Fetching Paradigm, and the Eunomia ordering layer leverage RDMA's capabilities to meet modern computing demands [31, 41, 44, 43]. In congestion control, RDMA's architecture facilitates efficient data transfers by allowing direct server memory access, resulting in higher IOPS compared to traditional methods. Recent advancements like the Remote Fetching Paradigm (RFP) and Remote Direct Cache Access (RDCA) enhance throughput and reduce latency, achieving up to 2.11 times improvement in throughput and up to 86.4

4.2 Challenges in RDMA Deployment

Deploying Remote Direct Memory Access (RDMA) in AI infrastructure presents several challenges, including latency and unpredictability from existing protocols like PCIe and RDMA, which can impact performance in large-scale, cloud-based environments [23]. This unpredictability is exacerbated by networks' inability to manage tail latency effectively, affecting HPC and data center systems [45]. Limited onboard resources in Network Interface Cards (NICs) can lead to bottlenecks and suboptimal performance under high I/O operations [46]. The sensitivity of large language model (LLM) training to single-point failures, particularly in Top-of-Rack (ToR) switches, complicates RDMA deployment [47].

As illustrated in Figure 5, the primary challenges in RDMA deployment can be categorized into three key areas: latency and unpredictability, complexity and overhead, and adaptation and optimization. This figure highlights not only the critical issues but also proposed solutions that address these challenges in a structured manner.

The complexity of the RDMA API, restricted to simple data movement verbs without support for conditionals and loops, poses additional challenges, especially for distributed applications requiring complex operations [42]. Communication strategies like parameter server (PS) and all-reduce introduce significant overhead, complicating RDMA deployment in large-scale distributed systems [25]. Network congestion remains a critical issue, with current all-reduce methods exhibiting inefficiencies that impact performance during data aggregation [48]. Additionally, the lack of generalizability and reproducibility in reinforcement learning (RL) applications, due to inadequate benchmarks for datacenter traffic variability, presents further obstacles [49].

Addressing these challenges requires adaptive congestion control algorithms, advanced virtualization techniques for RDMA in cloud environments, and machine learning-based approaches for optimizing resource management and load balancing. These advancements are essential for overcoming deployment challenges in AI infrastructure, facilitating efficient data transfer and resource utilization, and enhancing AI application performance, particularly in integrating advanced AI models with edge computing technologies [50, 8, 22, 1].

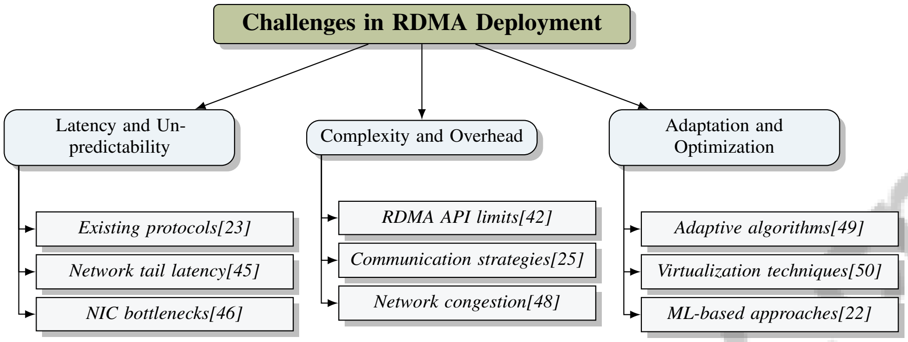


Figure 5: This figure illustrates the primary challenges in RDMA deployment, categorized into latency and unpredictability, complexity and overhead, and adaptation and optimization, highlighting key issues and proposed solutions.

4.3 Innovative Solutions for RDMA Optimization

Recent advancements in RDMA have led to innovative solutions that enhance performance in AI and distributed computing. RDMA Direct Cache Access (RDCA) by Jet utilizes a cache-resident buffer pool to alleviate memory bandwidth bottlenecks and improve data handling efficiency [51]. Eunomia introduces an ordering layer for RDMA NICs to manage packet reordering, enhancing network flexibility and performance through a hybrid-dynamic bitmap that adjusts based on reordering degree [44]. Virtuoso decouples path selection and load balancing from hardware, offering flexibility in traffic management and easier integration into existing systems [41].

In traffic management, isolating mice and elephant flows in different queues allows for optimized congestion control [52]. MLTCP improves network efficiency by interleaving communication phases of competing DNN training jobs, reducing training time [53]. RDMAbox optimizes I/O operations and CPU usage through request merging, chaining, and Adaptive Polling [46]. The Alibaba High Performance Network (HPN) architecture addresses hash polarization and single-point failures, enhancing RDMA network reliability and performance [47]. GPUDirect RDMA facilitates direct memory access between GPUs and NICs, eliminating unnecessary data copies and improving transfer efficiency [26].

These innovations, including the Remote Fetching Paradigm (RFP) and advanced RDMA over Converged Ethernet (RoCE) networks, amplify RDMA's performance, positioning it as a vital technology in high-performance computing and AI infrastructure. These advancements optimize IOPS, enhance network reliability, and manage congestion in large-scale distributed AI training, ensuring RDMA meets the evolving demands of modern computational workloads [31, 43].

5 Network Topology and Design

5.1 Taxonomy and Classification of Network Topologies

Classifying network topologies is crucial for optimizing data center scalability, reliability, and efficiency. Traditional Fat-tree topologies, prevalent in data centers, struggle to scale with the demands of distributed deep neural network (DNN) workloads, necessitating alternative frameworks for AI applications [54]. Leaf-spine topologies have gained traction in data-parallel distributed training due to their uniform latency and bandwidth capabilities, especially in 256-node configurations, making them ideal for high-throughput environments [40]. Hybrid and complex topologies, as explored by Sriram et al., improve performance by integrating diverse design principles to manage large-scale data centers [55]. Low-diameter networks like HyperX and Dragonfly, with reduced hop counts and enhanced path diversity, minimize latency and enhance data flow in extensive deployments [56].

To illustrate this classification, Figure 6 presents a comprehensive overview of network topologies, categorizing them into three main groups: Traditional Topologies, Low-diameter Topologies, and Server-centric Topologies. Traditional Topologies encompass structures such as Fat-tree, Leaf-spine, and Hybrid configurations, which are commonly employed in data centers. In contrast, Low-diameter Topologies, including HyperX and Dragonfly, are favored for their reduced hop counts and enhanced path diversity. Lastly, Server-centric Topologies, exemplified by Dual-port and Stellar transformation designs, provide innovative solutions aimed at improving adaptability and scalability in network architectures.

However, centralized algorithms often lack efficiency and scalability in dynamic networks, highlighting the need for decentralized approaches that adapt without global state awareness [57]. Server-centric DCN topologies, especially those with dual-port designs, are limited in variety, restricting design flexibility and performance enhancements [58]. Innovative interconnection strategies are necessary to overcome these challenges and enhance adaptability and scalability.

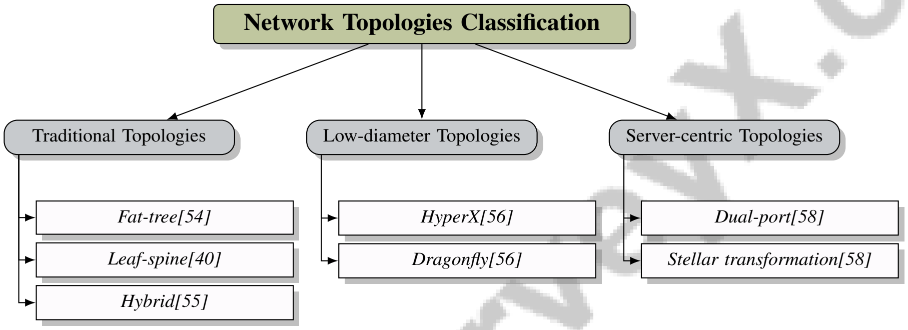


Figure 6: This figure illustrates the classification of network topologies into three main categories: Traditional Topologies, Low-diameter Topologies, and Server-centric Topologies. Traditional Topologies include Fat-tree, Leaf-spine, and Hybrid structures, which are commonly used in data centers. Low-diameter Topologies such as HyperX and Dragonfly are adopted for their reduced hop counts and enhanced path diversity. Server-centric Topologies, including Dual-port and Stellar transformation, offer innovative designs for enhanced adaptability and scalability in network architectures.

5.2 Innovations in Network Topology for AI and Data Centers

Recent advancements in network topology design have significantly improved AI and data center efficiency and scalability. The TOPOOPT framework exemplifies these advancements by co-optimizing network topology and parallelization strategies, dynamically adjusting configurations in response to traffic patterns, crucial for optimizing large-scale AI training performance [59]. This adaptability is vital given the complexity of AI workloads, necessitating infrastructures like Meta's RoCE and decentralized load balancing algorithms such as REPS for optimal resource utilization and rapid failure recovery [31, 60, 61]. Hybrid topologies developed using the Klemm-Eguiliz (KE) method enhance robustness and scalability by integrating diverse design principles, supporting modern AI applications [55]. Buffer management policy innovations have improved low-diameter networks like HyperX and Dragonfly, ensuring efficient data flow and latency reduction [56]. Server-centric DCN architectures require extensive analysis and experimentation for performance optimization. Novel routing algorithms generate paths up to 16

5.3 Hybrid and Server-Centric Network Topologies

Hybrid and server-centric network topologies are essential for optimizing data center performance and resource allocation. Hybrid topologies, combining Small World (SW) and Scale-Free (SF) networks, enhance scalability and robustness, supporting the dynamic demands of modern AI applications [55]. Advancements in server-centric Data Center Networks (DCNs), particularly through the stellar transformation method, expand the design space for dual-port server-centric DCNs, transforming various interconnection networks into efficient architectures, improving resource allocation and network management [58]. Innovations in routing algorithms further optimize server-centric topologies, with new classifications and algorithms producing paths up to 16

6 Congestion Control Mechanisms

6.1 Innovative Congestion Control Algorithms

Innovative congestion control algorithms are pivotal in optimizing data flow and resource utilization in high-performance computing (HPC) environments. SLINGSHOT is notable for its low latency, high throughput, and effective congestion control, making it ideal for both HPC and data center applications [45]. Algorithms like GEMINI and FlashPass advance congestion control by integrating reactive and proactive strategies, dynamically adjusting to network conditions to enhance throughput and reduce latency in inter-datacenter networks (IDNs) [62]. Machine learning approaches significantly augment traditional congestion control methods, adapting to network changes and optimizing bandwidth utilization [9]. Reinforcement learning (RL) algorithms, evaluated via the Iroko benchmark, provide a standardized platform for innovation in data center environments [49]. LALARPL employs learning automata to enhance routing protocols, offering dynamic load balancing and improved energy efficiency, particularly in IoT networks [63]. Proactive management techniques, like Canary, maintain performance resilience against congestion [48]. RDMAbox optimizes RDMA deployments by reducing I/O operations and latency, enhancing CPU efficiency in high-demand applications [46]. These algorithms are crucial for improving HPC environments' efficiency and scalability, enabling data centers to manage increasing data volumes while ensuring performance through advanced task allocation and management techniques [64, 2, 14].

6.2 Machine Learning and Reinforcement Learning Approaches

The integration of machine learning (ML) and reinforcement learning (RL) into congestion control strategies marks a significant advancement in managing data flow within data centers and distributed environments. These methodologies enable adaptive solutions that respond dynamically to real-time conditions, enhancing network performance. A major challenge is the heterogeneous nature of network environments, where existing mechanisms often inadequately address congestion across data center networks (DCN) and wide area networks (WAN) [62]. Deep RL can learn optimal network policies but faces issues with instability and overfitting, limiting its applicability [49]. Frameworks like RL-CC simplify neural network policies into efficient decision trees, reducing inference time while maintaining robust performance. ML algorithms applied to Active Queue Management (AQM) enable adaptive decision-making that optimizes bandwidth and efficiency [65]. RL in routing algorithms, as seen in Yajadda et al.'s work, optimizes path selection and congestion control based on real-time conditions [66]. Differentiating between mice and elephant flows in mixed traffic scenarios improves latency and throughput, highlighting the need for tailored congestion control mechanisms [52]. Integrating ML with existing algorithms allows DNN training jobs to interleave communication phases, optimizing network utilization and reducing training time [53]. Future research should focus on lightweight ML models, explore dataflow computing with GaAs technology, and investigate new ML techniques to enhance performance across diverse environments [9]. Stochastic evaluation models, such as Monte Carlo simulations, provide deeper insights into algorithm performance beyond deterministic models [67]. The integration of ML and RL into congestion control signifies a pivotal advancement in network management, providing dynamic solutions to modern data center challenges. These approaches leverage historical data and real-time telemetry to enhance throughput and fairness, particularly in handling diverse traffic patterns. Algorithms like Fair Datacenter Congestion Control (FDCC) utilize deep RL techniques, such as LSTM, to optimize resource allocation and minimize communication overhead during distributed machine learning tasks [68, 60, 69].

6.3 Congestion Control in Heterogeneous and Distributed Networks

Implementing congestion control in heterogeneous and distributed networks presents challenges due to diverse infrastructures and traffic patterns. Traditional methods depend on precise feedback from routers, which can be sensitive to rapid changes, affecting performance and reliability [70]. This necessitates versatile solutions that adapt to varying conditions. Existing schemes have proven inadequate for enhancing performance in distributed training workloads, highlighting the need for optimized mechanisms tailored to these environments [71].

The FASTFLOW framework exemplifies significant advancement, achieving performance improvements while maintaining fairness among competing flows [72]. Aurora, a machine learning-based algorithm, showcases adaptability to varying conditions, outperforming traditional methods [73]. GEMINI integrates Explicit Congestion Notification (ECN) and delay signals to maintain high throughput and low latency across heterogeneous environments [62].

Despite these advancements, ML-based schemes face challenges such as computational complexity and memory consumption, along with difficulties in achieving fair resource allocation [9]. The integration of ML in diverse networks raises questions, particularly regarding transfer learning and multi-agent systems in AQM [65]. Future research should emphasize lightweight models and improved real-time decision-making capabilities [60].

The computational overhead of continuous route updates in RL-based algorithms poses challenges, especially in large networks [66]. Nevertheless, RL methods like ADPG demonstrate robustness in multi-agent scenarios and generalization across configurations, outperforming traditional methods [74].

Innovative mechanisms, such as the Ladder mechanism with Virtual Channel (VC) reuse, address challenges in low-diameter networks, enhancing stability and reducing head-of-line blocking [56]. Delay-aware algorithms proposed by Skowron et al. achieve near-optimal performance with lower communication overhead compared to centralized methods, offering efficient solutions for load balancing in heterogeneous networks [75].

As illustrated in Figure 7, which depicts the hierarchical structure of congestion control in networks, the landscape is characterized by traditional challenges, machine learning-based solutions, and innovative mechanisms. Addressing congestion control challenges in heterogeneous and distributed networks requires a multifaceted approach incorporating advanced ML techniques, adaptive frameworks, and innovative routing strategies. These solutions are essential for enhancing network performance and optimizing resource utilization, particularly in modern networks involving virtualized middleboxes, energy-aware strategies in content delivery networks, and federated learning at the network edge. By tackling complexities in traffic management, server placement, and load balancing, these approaches facilitate more efficient handling of bandwidth and processing resources, improving service availability and energy savings in dynamic environments [76, 16, 77, 5].

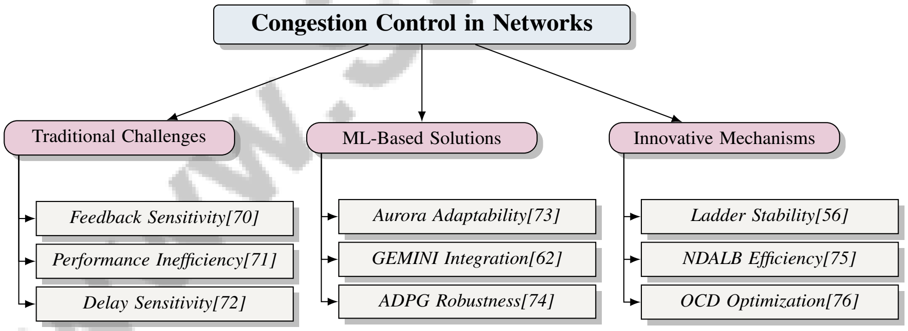


Figure 7: This figure illustrates the hierarchical structure of congestion control in networks, highlighting traditional challenges, machine learning-based solutions, and innovative mechanisms.

7 Load Balancing Strategies

7.1 Innovative Load Balancing Frameworks

Innovative load balancing frameworks are pivotal for optimizing resource utilization and enhancing performance in data centers, particularly with the increasing demands from AI and distributed computing applications. Buffer management policies in low-diameter networks significantly bolster network stability and throughput, ensuring efficient data flow and reduced latency in large-scale deployments [56]. Integrating energy-aware policies with traditional load balancing techniques facilitates dynamic power management and efficient server utilization, enhancing energy savings in content delivery networks (CDNs) [78, 77]. In federated learning environments, effective load balancing frameworks are essential for optimal resource management across distributed nodes, ensuring high performance in complex network settings [5].

Lifelong learning frameworks for privacy-aware reinforcement learning (RL) agents utilize transfer learning to adapt to environmental changes, enhancing performance while maintaining privacy [79]. Modeling load balancing, collocation, and scaling as an integrated optimization problem using Mixed-Integer Linear Programming (MILP) allows for simultaneous optimization, improving decision-making and resource allocation [80]. Additionally, biologically inspired algorithms, such as those adapting immune system strategies, dynamically adjust routing based on traffic load and congestion, enhancing overall system performance [81].

These frameworks underscore the necessity for flexible and adaptive load balancing solutions to optimize data center operations and meet AI and distributed computing demands. By integrating advanced methodologies, including scalable distributed AI systems, hardware-accelerated communication technologies like RDMA, and adaptive load-balancing techniques, these innovations significantly enhance resource utilization, scalability, and performance in diverse, resource-constrained environments like cloud and heterogeneous computing systems [1, 14, 26, 82, 43].

7.2 Adaptive and Dynamic Load Balancing Techniques

Method Name	Adaptability	Methodological Approaches	Resource Optimization
ALBIC[80]	Dynamically Adjusts Allocations	Biologically Inspired Algorithms	Intelligent Workload Distribution
TL-PARL[79]	Real-time Performance	Lifelong Learning, Transfer	Enhance Scheduling Efficiency
HSQ[83]	Dynamic Adjustments	Machine Learning Techniques	Enhancing Scheduling Efficiency
ILB[81]	Dynamically Adjust Based	Biological Metaphors Application	Improved Resource Allocation

Table 1: Overview of adaptive and dynamic load balancing methods, highlighting their adaptability, methodological approaches, and resource optimization capabilities. The table presents a comparative analysis of four prominent techniques: ALBIC, TL-PARL, HSQ, and ILB, each demonstrating unique strategies for enhancing system efficiency and performance in varying computational environments.

Adaptive and dynamic load balancing techniques are essential for optimizing resource utilization and maintaining system performance amidst fluctuating workloads. These techniques employ advanced methodologies to adjust in real-time to network conditions, ensuring efficient load distribution. The ALBIC (Autonomic Load Balancing with Integrated Collocation) method exemplifies this adaptability by optimizing collocations while balancing loads across the system, thus enhancing overall performance [80]. In Fog computing, the TL-PARL method integrates lifelong learning and transfer learning to optimize load balancing, enabling RL agents to adapt to significant environmental changes while preserving privacy [79].

Utilizing historical data in job distribution improves decision-making by considering both stale information and job assignment history, enhancing the accuracy and efficiency of load balancing decisions [83]. Biologically inspired algorithms, particularly those based on immunological approaches, dynamically allocate computational resources based on real-time conditions, akin to immune responses, boosting the system's ability to adapt to changing workloads [81].

Recent studies emphasize the necessity for flexible and responsive adaptive and dynamic load balancing techniques in managing network traffic and resource allocation. Table 1 provides a comparative analysis of various adaptive and dynamic load balancing methods, illustrating their adaptability, methodological approaches, and resource optimization strategies. These methods, including dynamic resource allocation in cloud data centers and homogenization in distributed environments, address challenges posed by heterogeneous hardware and varying application demands. By intelligently distributing workloads and continuously analyzing resource requirements—such as CPU, memory, and bandwidth—these techniques enhance scheduling efficiency and throughput while minimizing resource wastage and improving system performance [84, 83, 85, 86].

7.3 Energy-Efficient Load Balancing

Energy-efficient load balancing is crucial for optimizing resource utilization and minimizing the environmental impact of data center operations. Advanced methodologies focus on reducing energy consumption while maintaining high performance. The Prepartition paradigm exemplifies a proactive approach by planning virtual machine (VM) migrations in advance, enhancing load balancing and resource utilization compared to reactive methods [87]. The integration of artificial intelligence (AI) in load balancing presents significant opportunities for enhancing energy efficiency. Future research should focus on developing hybrid AI-based solutions that address both energy efficiency and scalability in large-scale network environments [88].

Self-learning, threshold-based load balancing approaches offer another pathway to energy-efficient operations by maintaining high performance with minimal communication overhead, requiring only a few messages per task [89]. The TWF framework optimizes job dispatching based on queue information, significantly reducing server overload likelihood and improving overall system performance and energy efficiency [90]. The A2WS (Adaptive Asynchronous Work Stealing) method enhances task distribution and reduces runtime in large-scale environments, contributing to reduced energy consumption [82].

The CCM-LB algorithm demonstrates up to 2.3x speedups in load balancing tasks compared to traditional methods, confirming its effectiveness in parallel computing applications [91]. CAFT utilizes complete congestion information from multiple paths, leading to improved load balancing and performance across both symmetric and asymmetric scenarios [92]. Empirical studies on dynamic load balancing techniques highlight potential enhancements in energy efficiency [86], underscoring the importance of continuous innovation in load balancing strategies to achieve optimal energy savings and environmental sustainability.

Energy-efficient load balancing strategies are essential for optimizing data center operations, facilitating effective resource utilization across services like IaaS, PaaS, and SaaS, while minimizing environmental impact by reducing energy consumption, particularly during low workload periods. By concentrating workloads on fewer servers and transitioning idle servers to low-power states, these strategies help maintain service quality, lower operational costs, and enhance overall energy efficiency [93, 78, 77, 14]. Leveraging advanced methodologies and AI-driven solutions provides a sustainable framework for managing the growing demands of modern computing environments.

8 AI Infrastructure and Distributed Computing

8.1 Role of Distributed Computing

Distributed computing is pivotal in bolstering AI infrastructure by facilitating parallel processing and resource sharing, essential for meeting the computational needs of modern AI applications. This paradigm enhances data processing and memory management, significantly boosting AI systems' performance and scalability. By embedding distributed computing within AI frameworks, task allocation and performance management are optimized, leading to superior system efficiency [2].

Innovative frameworks underscore distributed computing's effectiveness in optimizing resource utilization. The ALBIC framework, for instance, enhances performance in Parallel Stream Processing Engines (PSPEs) through load balancing and horizontal scaling [80]. Similarly, the Scavenger cloud service optimizes training performance by modeling inefficiencies in Stochastic Gradient Descent (SGD) and parallel efficiencies, enabling precise training outcome predictions [28].

In edge computing, effective management of edge resources elevates federated learning task deployment, reinforcing distributed computing capabilities [5]. Hierarchical learning systems further this by integrating space-ground computing for collaborative training on IoT devices [27].

Dynamic Batching methods improve distributed computing by adjusting mini-batch sizes based on throughput, optimizing resource allocation [24]. Frameworks like PANAMA, which integrate hardware accelerators for gradient aggregation and congestion control, enhance performance beyond existing methods [94].

The CVM framework advances AI infrastructure by supporting various frontends and backends, promoting code reuse and optimization [10]. These advancements in distributed computing facilitate efficient parallel processing and resource sharing, meeting the demands of diverse, resource-constrained environments. Scalable AI frameworks and edge computing integration are enhancing AI applications' scalability and efficiency, leveraging cloud computing to meet deep learning's computational demands and optimize data management. Edge Intelligence further enables effective AI model training and inference at the network edge, addressing mobile devices' data processing needs. These innovations collectively drive AI application performance improvements, shaping future technological advancements [50, 8, 1].

8.2 Integration of Computing Resources

Integrating diverse computing resources is essential for optimizing AI performance, particularly in distributed computing environments where efficient resource allocation and task scheduling are vital. The Distributed Load Management (DLM) approach exemplifies leveraging local information and iterative optimization to achieve convergence without global network knowledge [57]. This method enhances system performance by dynamically adapting to network conditions.

Frameworks like GADGET harness the submodular structure of scheduling problems to optimize resource allocation, maintaining competitive performance across distributed nodes [37]. The integration of metaheuristic algorithms, Neural Networks, and Federated Learning into dynamic models advances resource integration, allowing AI systems to adapt to workload demands and optimize performance [95].

Scavenger cloud service fine-tunes resource allocations through performance-model guided searches, ensuring AI applications meet desired performance levels within user constraints [28]. TensorFlow's dynamic batching mechanism enhances resource integration by enabling low-cost training on heterogeneous cloud servers, ensuring efficient resource utilization [24].

These methodologies highlight the importance of integrating diverse computing resources to optimize AI performance. By employing advanced algorithms and dynamic models, AI systems can efficiently manage resources, adapt to changing conditions, and achieve optimal performance across distributed environments. These advancements address key challenges such as latency, energy consumption, and privacy concerns, supporting AI deployment across various platforms and applications [50, 8, 1, 7].

8.3 Enhancements in AI Infrastructure

Recent advancements in AI infrastructure have markedly improved system performance and scalability through innovations in hardware and software. The TPUv4 supercomputer, with a 99.98

Future research in virtual machine management aims to refine queuing-based migration models and explore lightweight migration designs, optimizing resource allocation and enhancing cloud data centers' flexibility [96]. Integrating renewable energy sources into cloud infrastructures offers a promising path for reducing data centers' environmental impact while maintaining performance.

Self-adaptive algorithms are a key research focus, particularly in distributed load balancing, aiming to integrate multiple resource types while minimizing communication costs [97]. These algorithms improve AI systems' efficiency and responsiveness by dynamically adjusting to network conditions.

Network protocol optimizations, as demonstrated by the NetReduce framework, improve communication efficiency in distributed environments [25]. Applying these optimizations across platforms could further enhance AI infrastructure's scalability and performance.

Advancements in AI infrastructure emphasize innovative hardware and software design approaches, ensuring AI systems meet modern applications' demands. Developments in scalable AI frameworks, edge intelligence, and integrated AI pipelines enhance system performance, scalability, and energy efficiency, establishing a robust foundation for AI technologies' ongoing development and deployment across various applications and industries. These innovations optimize resource allocation and facilitate seamless integration into cloud and edge environments, driving intelligent systems' evolution [7, 8, 1, 50].

9 Data Center Optimization

In the realm of data center optimization, adopting advanced technologies is crucial for enhancing both operational efficiency and sustainability. As reliance on cloud services grows, innovative solutions become essential. This section explores the transformative role of artificial intelligence (AI) and machine learning (ML) in data center operations, focusing on their application in optimizing resource management and enhancing performance. The subsequent subsection will delve into specific AI and

AI-generated, for reference only.

ML techniques that are revolutionizing data center optimization, underscoring their effectiveness and potential for future progress.

9.1 AI and Machine Learning in Data Center Optimization

AI and ML are instrumental in optimizing data center operations by improving resource management and operational efficiency. These technologies enhance task allocation and performance management, significantly elevating data center efficiency [2]. AI-driven strategies enable dynamic adaptation to fluctuating network conditions, ensuring effective resource allocation and improved user experiences, especially in edge computing environments [3].

The application of AI methodologies, such as deep reinforcement learning (DRL) for scheduling, has led to notable advancements in data center management. Frameworks like DRLIS optimize multiple objectives, reducing the execution costs of IoT applications and demonstrating AI's capacity to enhance performance through optimized load distribution [98]. In federated learning scenarios, energy-efficient strategies are crucial for managing distributed training tasks while maintaining optimal performance [5].

AI techniques also facilitate sustainable operations. Carbon-intensity-based scheduling can reduce cumulative carbon emissions by up to 54

Implementing integrated migration management strategies that consider both computing and networking resources is vital for optimizing migration scheduling [21]. Proactive demand response strategies enhance the understanding of interactions between smart grids and data centers, improving reliability and cost efficiency [99].

The Scavenger cloud service exemplifies performance-model guided searches for optimizing training performance, with future research focusing on refining these models and extending support to frameworks like PyTorch [28]. The TAEER algorithm also demonstrates superior performance in reducing energy consumption and outage probability in satellite networks, further illustrating AI's role in optimizing data center operations [27].

AI and ML represent a transformative shift in data center optimization, providing dynamic, adaptive solutions that enhance operational efficiency and meet modern computing demands. These advancements in task allocation, performance management, and energy optimization are essential for ensuring peak efficiency while minimizing energy consumption and maintaining high service availability, thereby supporting sustainable operations in the evolving digital landscape [77, 100, 64, 2, 14].

9.2 Energy Efficiency and Environmental Impact

Improving energy efficiency in data centers is imperative due to the substantial operational costs associated with power consumption. The increasing demand for cloud services necessitates efficient energy management strategies, as highlighted by recent surveys emphasizing energy efficiency's importance in cloud data centers [13]. The integration of Software Defined Networking (SDN) offers a promising avenue for enhancing energy efficiency, exemplified by the MERSDN framework, which employs machine learning for effective energy management [101].

Innovative approaches to reducing power consumption, such as using Wide Area Network (WAN) accelerators during virtual machine (VM) migrations, significantly cut power usage, potentially reducing consumption to one-tenth of its usual level while maintaining performance [102]. These solutions are crucial for maintaining system performance and minimizing environmental impact.

The environmental impact of AI model inference also requires attention, as significant variations in carbon and water footprints exist across regions [103]. Addressing these discrepancies necessitates a comprehensive understanding of the environmental costs associated with AI operations and strategies to mitigate these impacts, including optimizing the geographical distribution of data centers for balanced environmental and operational efficiency.

Challenges remain in accurately modeling server states and workload characteristics, essential for effective load balancing and energy management [93]. Developing sophisticated modeling techniques to predict energy usage patterns and optimize resource allocation is necessary to overcome these challenges.

9.3 Emerging Technologies and Future Directions

Emerging technologies in data center optimization are poised to reshape distributed computing, fostering more efficient and sustainable operations. A key focus is enhancing load balancing techniques. Future research should refine homogenization techniques to support Distributed Artificial Intelligence (DAI) within Distributed Intelligent Systems (DIS), potentially improving performance in complex networked environments [84]. Exploring hybrid network topologies, such as those developed using the Klemm-Eguiliz (KE) model, is critical for optimizing performance in larger data center environments, with further refinements promising enhanced scalability and efficiency [55].

Integrating Big Data and machine learning into load balancing frameworks offers another promising research direction. These technologies can improve prioritization and resource allocation, leading to efficient load distribution and enhanced data center performance [104]. Additionally, bolstering the robustness of distributed algorithms in dynamic networks and exploring their applications in other optimization challenges could yield valuable insights for improving data center operations [57].

In scheduling, extending algorithms to accommodate arbitrary numbers of data localities and assessing their performance in job-based systems are vital avenues for exploration. Such advancements could yield more flexible and efficient scheduling solutions, optimizing resource utilization across diverse environments [105].

The CVM framework signifies a notable advancement in virtual machine management, with future work aiming to expand it with additional frontends and hardware platforms, potentially enhancing data center optimization, flexibility, and scalability of cloud infrastructures [10].

The integration of emerging technologies, such as machine learning and edge computing, alongside innovative research directions in task allocation and performance management, underscores the necessity for ongoing advancements in data center optimization to effectively tackle increasing workloads, resource utilization, and network performance challenges [2, 52, 64, 8, 106]. By leveraging advanced methodologies and exploring new frontiers in distributed computing, these efforts aim to drive efficiency, scalability, and sustainability in modern computing environments.

10 Conclusion

10.1 Future Directions and Research Opportunities

Advancing AI and data center optimization requires a strategic focus on enhancing efficiency, scalability, and sustainability. Key areas for future exploration include the development of dynamic sparse matrix compression techniques and in-memory computing optimizations, particularly through the integration of CXL-based FPGA and CPU architectures. These innovations hold the potential to significantly improve data processing capabilities and reduce latency in distributed environments. Additionally, refining dynamic batching mechanisms, with a focus on throughput estimation and synchronization methods, is crucial for optimizing resource allocation and performance in diverse computing settings.

In the realm of federated learning, robust resource management solutions are essential to balance communication, energy efficiency, and latency, thereby enhancing the scalability and responsiveness of distributed AI systems. Moreover, the development of energy-aware load balancing algorithms is imperative to improve server load predictions and optimize transitions between active and sleep states, which can substantially enhance system responsiveness and energy efficiency.

Optimizing congestion control algorithms remains a critical research area, with a particular emphasis on improving collision management strategies and expanding frameworks like Canary to support more collective operations. These improvements can significantly boost data transfer efficiency and network performance in high-demand computing environments. Additionally, future research should prioritize optimizing memory usage, enhancing interoperability among various transport mechanisms, and exploring the trade-offs between latency and resource sharing to further augment communication efficiency in AI infrastructure.

These research directions underscore the importance of continuous innovation in AI and data center optimization. By leveraging advanced methodologies and exploring new frontiers in distributed computing, these initiatives aim to foster a more efficient, scalable, and sustainable future for contemporary computing environments.

AI-generated, for reference only.

References

[1] Neelesh Mungoli. Scalable, distributed ai frameworks: Leveraging cloud computing for enhanced deep learning performance and efficiency, 2023.

[2] Nidhika Chauhan, Navneet Kaur, Kamaljit Singh Saini, Sahil Verma, Abdulatif Alabdulatif, Ruba Abu Khurma, Maribel Garcia-Arenas, and Pedro A. Castillo. A systematic literature review on task allocation and performance management techniques in cloud data center, 2024.

[3] Zeinab Nezami, Kamran Zamanifar, Karim Djemame, and Evangelos Pournaras. Decentralized edge-to-cloud load-balancing: Service placement for the internet of things, 2021.

[4] Thembelihle Dlamini and Sifiso Vilakati. Lstm-based traffic load balancing and resource allocation for an edge system, 2020.

[5] Silvana Trindade, Luiz F. Bittencourt, and Nelson L. S. da Fonseca. Management of resource at the network edge for federated learning, 2022.

[6] Yueyue Dai, Ke Zhang, Sabita Maharjan, and Yan Zhang. Deep reinforcement learning for stochastic computation offloading in digital twin networks, 2020.

[7] Khaled B. Letaief, Yuanming Shi, Jianmin Lu, and Jianhua Lu. Edge artificial intelligence for 6g: Vision, enabling technologies, and applications, 2021.

[8] Shuiguang Deng, Hailiang Zhao, Weijia Fang, Jianwei Yin, Schahram Dustdar, and Albert Y. Zomaya. Edge intelligence: The confluence of edge computing and artificial intelligence, 2020.

[9] Zhilbert Tafa and Veljko Milutinovic. Machine learning in congestion control: A survey on selected algorithms and a new roadmap to their implementation, 2021.

[10] Ingo Müller, Renato Marroquín, Dimitrios Koutsoukos, Mike Wawrzoniak, Sabir Akhadov, and Gustavo Alonso. The collection virtual machine: An abstraction for multi-frontend multi-backend data analysis, 2020.

[11] Ruihong Wang, Jianguo Wang, Stratos Idreos, M. Tamer Özsu, and Walid G. Aref. The case for distributed shared-memory databases with rdma-enabled memory disaggregation, 2022.

[12] Yansong Gao, Minki Kim, Chandra Thapa, Sharif Abuadbba, Zhi Zhang, Seyit A. Camtepe, Hyoungshick Kim, and Surya Nepal. Evaluation and optimization of distributed machine learning techniques for internet of things, 2021.

[13] Deepika Saxena and Ashutosh Kumar Singh. workload forecasting and resource management models based on machine learning for cloud computing environments, 2021.

[14] Smruti Rekha Swain, Ashutosh Kumar Singh, and Chung Nan Lee. Efficient resource management in cloud environment, 2022.

[15] Ioannis Argyroulis. Recent advancements in distributed system communications, 2021.

[16] Moses Charikar, Yonatan Naamad, Jennifer Rexford, and X. Kelvin Zou. Multi-commodity flow with in-network processing, 2018.

[17] Chien-Sheng Yang, Chien-Chun Huang-Fu, and I-Kang Fu. Carbon-neutralized task scheduling for green computing networks, 2022.

[18] Jinkun Zhang, Yuezhou Liu, and Edmund Yeh. Delay-optimal forwarding and computation offloading for service chain tasks, 2024.

[19] Marek Bolanowski, Alicja Gerka, Andrzej Paszkiewicz, Maria Ganzha, and Marcin Paprzycki. Application of genetic algorithm to load balancing in networks with a homogeneous traffic flow, 2023.

AI-generated, for reference only.

[20] Rafael Vescovi, Ryan Chard, Nickolaus Saint, Ben Blaiszik, Jim Pruyne, Tekin Bicer, Alex Lavens, Zhengchun Liu, Michael E. Papka, Suresh Narayanan, Nicholas Schwarz, Kyle Chard, and Ian Foster. Linking scientific instruments and hpc: Patterns, technologies, experiences, 2022.

[21] TianZhang He and Rajkumar Buyya. A taxonomy of live migration management in cloud computing, 2021.

[22] Talia Gershon, Seetharami Seelam, Brian Belgodere, Milton Bonilla, Lan Hoang, Danny Barnett, I-Hsin Chung, Apoorve Mohan, Ming-Hung Chen, Lixiang Luo, Robert Walkup, Constantinos Evangelinos, Shweta Salaria, Marc Dombrowa, Yoonho Park, Apo Kayi, Liran Schour, Alim Alim, Ali Sydney, Pavlos Maniotis, Laurent Schares, Bernard Metzler, Bengi Karacali-Akyamac, Sophia Wen, Tatsuhiro Chiba, Sunyanan Choochotkaew, Takeshi Yoshimura, Claudia Misale, Tonia Elengikal, Kevin O Connor, Zhuoran Liu, Richard Molina, Lars Schneidenbach, James Caden, Christopher Laibinis, Carlos Fonseca, Vasily Tarasov, Swaminathan Sundararaman, Frank Schmuck, Scott Guthridge, Jeremy Cohn, Marc Eshel, Paul Muench, Runyu Liu, William Pointer, Drew Wyskida, Bob Krull, Ray Rose, Brent Wolfe, William Cornejo, John Walter, Colm Malone, Clifford Perucci, Frank Franco, Nigel Hinds, Bob Calio, Pavel Druyan, Robert Kilduff, John Kienle, Connor McStay, Andrew Figueroa, Matthew Connolly, Edie Fost, Gina Roma, Jake Fonseca, Ido Levy, Michele Payne, Ryan Schenkel, Amir Malki, Lion Schneider, Aniruddha Narkhede, Shekeba Moshref, Alexandra Kisin, Olga Dodin, Bill Rippon, Henry Wrieth, John Ganci, Johnny Colino, Donna Habeger-Rose, Rakesh Pandey, Aditya Gidh, Aditya Gaur, Dennis Patterson, Samsuddin Salmani, Rambilas Varma, Rumana Rumana, Shubham Sharma, Aditya Gaur, Mayank Mishra, Rameswar Panda, Aditya Prasad, Matt Stallone, Gaoyuan Zhang, Yikang Shen, David Cox, Ruchir Puri, Dakshi Agrawal, Drew Thorstensen, Joel Belog, Brent Tang, Saurabh Kumar Gupta, Amitabha Biswas, Anup Maheshwari, Eran Gampel, Jason Van Patten, Matthew Runion, Sai Kaki, Yigal Bogin, Brian Reitz, Steve Pritko, Shahan Najam, Surya Nambala, Radhika Chirra, Rick Welp, Frank DiMitri, Felipe Telles, Amilcar Arvelo, King Chu, Ed Seminaro, Andrew Schram, Felix Eickhoff, William Hanson, Eric Mckeever, Michael Light, Dinakaran Joseph, Piyush Chaudhary, Piyush Shivam, Puneet Chaudhary, Wesley Jones, Robert Guthrie, Chris Bostic, Rezaul Islam, Steve Duersch, Wayne Sawdon, John Lewars, Matthew Klos, Michael Spriggs, Bill McMillan, George Gao, Ashish Kamra, Gaurav Singh, Marc Curry, Tushar Katarki, Joe Talerico, Zenghui Shi, Sai Sindhur Malleni, and Erwan Gallen. The infrastructure powering ibm's gen ai model development, 2025.

[23] Kevin Fang and David Peng. Netdam: Network direct attached memory with programmable in-memory computing isa, 2021.

[24] Sahil Tyagi and Prateek Sharma. Taming resource heterogeneity in distributed ml training with dynamic batching, 2023.

[25] Shuo Liu, Qiaoling Wang, Junyi Zhang, Qinliang Lin, Yao Liu, Meng Xu, Ray CC Chueng, and Jianfei He. Netreduce: Rdma-compatible in-network reduction for distributed dnn training acceleration. *arXiv preprint arXiv:2009.09736*, 2020.

[26] Walid A. Hanafy, Limin Wang, Hyunseok Chang, Sarit Mukherjee, T. V. Lakshman, and Prashant Shenoy. Understanding the benefits of hardware-accelerated communication in model-serving applications, 2023.

[27] Jingyang Zhu, Yuanming Shi, Yong Zhou, Chunxiao Jiang, and Linling Kuang. Hierarchical learning and computing over space-ground integrated networks, 2024.

[28] Sahil Tyagi and Prateek Sharma. Scavenger: A cloud service for optimizing cost and performance of ml training, 2023.

[29] Kabir Nagrecha. Systems for parallel and distributed large-model deep learning training, 2023.

[30] Jared Fernandez, Luca Wehrstedt, Leonid Shamis, Mostafa Elhoushi, Kalyan Saladi, Yonatan Bisk, Emma Strubell, and Jacob Kahn. Hardware scaling trends and diminishing returns in large-scale distributed training, 2024.

[31] Adithya Gangidi, Rui Miao, Shengbao Zheng, Sai Jayesh Bondu, Guilherme Goes, Hany Morsy, Rohit Puri, Mohammad Riftadi, Ashmitha Jeevaraj Shetty, Jingyi Yang, et al. Rdma over ethernet for distributed training at meta scale. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pages 57–70, 2024.

[32] Jilong Xue, Youshan Miao, Cheng Chen, Ming Wu, Lintao Zhang, and Lidong Zhou. Rpc considered harmful: Fast distributed deep learning on rdma, 2018.

[33] Amedeo Sapio, Marco Canini, Chen-Yu Ho, Jacob Nelson, Panos Kalnis, Changhoon Kim, Arvind Krishnamurthy, Masoud Moshref, Dan Ports, and Peter Richtárik. Scaling distributed machine learning with {In-Network} aggregation. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 785–808, 2021.

[34] Enda Yu, Dezun Dong, and Xiangke Liao. Full-stack allreduce on multi-rail networks, 2024.

[35] Fei Yang, Shuang Peng, Ning Sun, Fangyu Wang, Yuanyuan Wang, Fu Wu, Jiezhong Qiu, and Aimin Pan. Holmes: Towards distributed training across clusters with heterogeneous nic environment, 2024.

[36] Jianbo Dong, Bin Luo, Jun Zhang, Pengcheng Zhang, Fei Feng, Yikai Zhu, Ang Liu, Zian Chen, Yi Shi, Hairong Jiao, et al. Boosting large-scale parallel training efficiency with c4: A communication-driven approach. *arXiv preprint arXiv:2406.04594*, 2024.

[37] Menglu Yu, Ye Tian, Bo Ji, Chuan Wu, Hridesh Rajan, and Jia Liu. Gadget: Online resource optimization for scheduling ring-all-reduce learning jobs, 2022.

[38] Francesco Malandrino, Carla Fabiana Chiasserini, Nuria Molner, and Antonio De La Oliva. Network support for high-performance distributed machine learning, 2022.

[39] Xin Chen, Hua Zhou, Yuxiang Gao, and Yu Zhu. A novel co-design peta-scale heterogeneous cluster for deep learning training, 2018.

[40] Vamsi Addanki, Prateesh Goyal, Ilias Marinos, and Stefan Schmid. Ethereal: Divide and conquer network load balancing in large-scale distributed training, 2025.

[41] Feng Tian, Wendi Feng, Yang Zhang, and Zhi-Li Zhang. A novel software-based multi-path rdma solutionfor data center networks, 2020.

[42] Waleed Reda, Marco Canini, Dejan Kostić, and Simon Peter. Rdma is turing complete, we just did not know it yet!, 2021.

[43] Maomeng Su, Mingxing Zhang, Kang Chen, Yongwei Wu, and Guoliang Li. Rfp: A remote fetching paradigm for rdma-accelerated systems, 2015.

[44] Sana Mahmood, Jinqi Lu, and Soudeh Ghorbani. Orderly management of packets in rdma by eunomia, 2024.

[45] Daniele De Sensi, Salvatore Di Girolamo, Kim H. McMahon, Duncan Roweth, and Torsten Hoefler. An in-depth analysis of the slingshot interconnect, 2020.

[46] Juhyun Bae, Ling Liu, Yanzhao Wu, Gong Su, and Arun Iyengar. Rdmabox : Optimizing rdma for memory intensive workloads, 2021.

[47] Kun Qian, Yongqing Xi, Jiamin Cao, Jiaqi Gao, Yichi Xu, Yu Guan, Binzhang Fu, Xuemei Shi, Fangbo Zhu, Rui Miao, et al. Alibaba hpn: A data center network for large language model training. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pages 691–706, 2024.

[48] Daniele De Sensi, Edgar Costa Molero, Salvatore Di Girolamo, Laurent Vanbever, and Torsten Hoefler. Canary: Congestion-aware in-network allreduce using dynamic trees, 2023.

[49] Fabian Ruffy, Michael Przystupa, and Ivan Beschastnikh. Iroko: A framework to prototype reinforcement learning for data center traffic control, 2018.

AI-generated, for reference only.

[50] Miguel de Prado, Jing Su, Rabia Saeed, Lorenzo Keller, Noelia Vallez, Andrew Anderson, David Gregg, Luca Benini, Tim Llewellynn, Nabil Ouerhani, Rozenn Dahyot, , and Nuria Pazos. Bonseyes ai pipeline – bringing ai to you. end-to-end integration of data, algorithms and deployment tools, 2020.

[51] Qiang Li, Qiao Xiang, Derui Liu, Yuxin Wang, Haonan Qiu, Xiaoliang Wang, Jie Zhang, Ridi Wen, Haohao Song, Gexiao Tian, Chenyang Huang, Lulu Chen, Shaozong Liu, Yaohui Wu, Zhiwu Wu, Zicheng Luo, Yuchao Shao, Chao Han, Zhongjie Wu, Jianbo Dong, Zheng Cao, Jinbo Wu, Jiwu Shu, and Jiesheng Wu. From rdma to rdca: Toward high-speed last mile of data center networks using remote direct cache access, 2023.

[52] Wenxue Cheng, Fengyuan Ren, Wanchun Jiang, Kun Qian, Tong Zhang, and Ran Shu. Isolating mice and elephant in data centers, 2016.

[53] Sudarsanan Rajasekaran, Sanjoli Narang, Anton A Zabreyko, and Manya Ghobadi. Mltcp: Congestion control for dnn training. *arXiv preprint arXiv:2402.09589*, 2024.

[54] Weiyang Wang, Moein Khazraee, Zhizhen Zhong, Manya Ghobadi, Zhihao Jia, Dheevatsa Mudigere, Ying Zhang, and Anthony Kewitsch. Topoopt: Co-optimizing network topology and parallelization strategy for distributed training jobs, 2022.

[55] Ilango Sriram and Dave Cliff. Hybrid complex network topologies are preferred for component-subscription in large-scale data-centres, 2011.

[56] Alejandro Cano, Cristóbal Camarero, Carmen Martínez, and Ramón Beivide. Analysing mechanisms for virtual channel management in low-diameter networks, 2024.

[57] Wuqiong Luo, Wee Peng Tay, Peng Sun, and Yonggang Wen. On distributed algorithms for cost-efficient data center placement in cloud computing, 2018.

[58] Alejandro Erickson, , Iain A. Stewart, Javier Navaridas, and Abbas E. Kiasari. The stellar transformation: From interconnection networks to datacenter networks, 2016.

[59] Weiyang Wang, Moein Khazraee, Zhizhen Zhong, Manya Ghobadi, Zhihao Jia, Dheevatsa Mudigere, Ying Zhang, and Anthony Kewitsch. {TopoOpt}: Co-optimizing network topology and parallelization strategy for distributed training jobs. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 739–767, 2023.

[60] Huiling Jiang, Qing Li, Yong Jiang, Gengbiao Shen, Richard Sinnott, Chen Tian, and Mingwei Xu. When machine learning meets congestion control: A survey and comparison, 2020.

[61] Tommaso Bonato, Abdul Kabbani, Ahmad Ghalayini, Mohammad Dohadwala, Michael Papamichael, Daniele De Sensi, and Torsten Hoefler. Arcane: Adaptive routing with caching and network exploration. *arXiv preprint arXiv:2407.21625*, 2024.

[62] Gaoxiong Zeng. Congestion control mechanisms for inter-datacenter networks, 2022.

[63] Mohammadhossein Homaei. Learning automata-based enhancements to rpl: Pioneering load-balancing and traffic management in iot, 2024.

[64] Bo Li, Ting Wang, Peng Yang, Mingsong Chen, Shui Yu, and Mounir Hamdi. Machine learning empowered intelligent data center networking: A survey, 2022.

[65] Mohammad Parsa Toopchinezhad and Mahmood Ahmadi. Machine learning approaches for active queue management: A survey, taxonomy, and future directions, 2024.

[66] Seyed Hassan Yajadda and Farshad Safaei. A novel reinforcement learning routing algorithm for congestion control in complex networks, 2023.

[67] Aymen Hasan Alawadi, Maiass Zaher, and Sándor Molnár. Methods for predicting behavior of elephant flows in data center networks, 2019.

[68] Jiagui Wu, Yang Qin, Weihong Yang, and Ruonan Li. Rdma based congestion control strategy for dml training optimization in data center networks. 2022.

[69] Kunlin Zhang. An investigation on data center congestion control algorithms. 2023.

[70] R. Jain, K. Ramakrishnan, and D. Chiu. Congestion avoidance in computer networks with a connectionless network layer, 1998.

[71] Tarannum Khan, Saeed Rashidi, Srinivas Sridharan, Pallavi Shurpali, Aditya Akella, and Tushar Krishna. Impact of roce congestion control policies on distributed training of dnns. In *2022 IEEE Symposium on High-Performance Interconnects (HOTI)*, pages 39–48. IEEE, 2022.

[72] Tommaso Bonato, Abdul Kabbani, Daniele De Sensi, Rong Pan, Yanfang Le, Costin Raiciu, Mark Handley, Timo Schneider, Nils Blach, Ahmad Ghalayini, Daniel Alves, Michael Papamichael, Adrian Caulfield, and Torsten Hoefler. Fastflow: Flexible adaptive congestion control for high-performance datacenters, 2024.

[73] Nathan Jay, Noga H. Rotman, P. Brighten Godfrey, Michael Schapira, and Aviv Tamar. Internet congestion control via deep reinforcement learning, 2019.

[74] Chen Tessler, Yuval Shpigelman, Gal Dalal, Amit Mandelbaum, Doron Haritan Kazakov, Benjamin Fuhrer, Gal Chechik, and Shie Mannor. Reinforcement learning for datacenter congestion control, 2022.

[75] Piotr Skowron and Krzysztof Rzadca. Network delay-aware load balancing in selfish and cooperative distributed systems, 2012.

[76] Aran Bergman, Israel Cidon, Isaac Keslassy, Noga Rotman, Michael Schapira, Alex Markuze, and Eyal Zohar. Pied piper: Rethinking internet data delivery, 2018.

[77] Vimal Mathew, Ramesh K. Sitaraman, and Prashant Shenoy. Energy-aware load balancing in content delivery networks, 2011.

[78] Ashkan Paya and Dan C. Marinescu. Energy-aware load balancing policies for the cloud ecosystem, 2014.

[79] Maad Ebrahim, Abdelhakim Senhaji Hafid, and Mohamed Riduan Abid. Lifelong learning for fog load balancing: A transfer learning approach, 2023.

[80] Kasper Grud Skat Madsen, Yongluan Zhou, and Jianneng Cao. Integrative dynamic reconfiguration in a parallel stream processing engine, 2016.

[81] James J. Clark. Immunological approaches to load balancing in mimd systems, 2022.

[82] João B. Fernandes, Ítalo A. S. de Assis, Idalmis M. S. Martins, Tiago Barros, and Samuel Xavier de Souza. Adaptive asynchronous work-stealing for distributed load-balancing in heterogeneous systems, 2024.

[83] Shafinaz Islam. Network load balancing methods: Experimental comparisons and improvement, 2017.

[84] M. Shahriar Hossain, M. Muztaba Fuad, Debzani Deb, Kazi Muhammad Najmul Hasan Khan, and Md. Mahbubul Alam Joarder. Load balancing in a networked environment through homogenization, 2011.

[85] Sakshi Chhabra and Ashutosh Kumar Singh. Dynamic resource allocation method for load balance scheduling over cloud data center networks, 2022.

[86] Ardhendu Mandal and Subhas Chandra Pal. An empirical study and analysis of the dynamic load balancing techniques used in parallel computing systems, 2011.

[87] Minxian Xu, Guangchun Luo, Ling Tian, Aiguo Chen, Yaqiu Jiang, Guozhong Li, and Wenhong Tian. Prepartition: Paradigm for the load balance of virtual machine allocation in data centers, 2015.

[88] Ahmed Hazim Alhilali and Ahmadreza Montazerolghaem. Artificial intelligence based load balancing in sdn: A comprehensive survey, 2023.

[89] Diego Goldsztajn, Sem C. Borst, Johan S. H. van Leeuwaarden, Debankur Mukherjee, and Philip A. Whiting. Self-learning threshold-based load balancing, 2023.

[90] Distributed dispatching in the parallel server model.

[91] Jonathan Lifflander, Philippe P. Pebay, Nicole L. Slattengren, Pierre L. Pebay, Robert A. Pfeiffer, Joseph D. Kotulski, and Sean T. McGovern. A communication- and memory-aware model for load balancing tasks, 2024.

[92] Sultan Alanazi and Bechir Hamdaoui. Caft: Congestion-aware fault-tolerant load balancing for three-tier clos data centers, 2020.

[93] Freek van den Berg, Björn F. Postema, and Boudewijn R. Haverkort. Evaluating load balancing policies for performance and energy-efficiency, 2016.

[94] Nadeen Gebara, Manya Ghobadi, and Paolo Costa. In-network aggregation for shared machine learning clusters. *Proceedings of Machine Learning and Systems*, 3:829–844, 2021.

[95] Saimin Chen Zhang. Hierarchical optimization of metaheuristic algorithms and federated learning for enhanced capacity management and load balancing in hetnets, 2023.

[96] Misbah Liaqat, Shalini Ninoriya, Junaid Shuja, Raja Wasim Ahmad, and Abdullah Gani. Virtual machine migration enabled cloud resource management: A challenging task, 2016.

[97] Minxian Xu, Wenhong Tian, and Rajkumar Buyya. A survey on load balancing algorithms for vm placement in cloud computing, 2017.

[98] Zhiyu Wang, Mohammad Goudarzi, Mingming Gong, and Rajkumar Buyya. Deep reinforcement learning-based scheduling for optimizing system load and response time in edge and fog computing environments, 2023.

[99] Hao Wang, Jianwei Huang, Xiaojun Lin, and Hamed Mohsenian-Rad. Proactive demand response for data centers: A win-win solution, 2015.

[100] Mohammad Noormohammadpour and Cauligi S. Raghavendra. Datacenter traffic control: Understanding techniques and trade-offs, 2017.

[101] Beakal Gizachew Assefa and Oznur Ozkasap. Mer-sdn: Machine learning framework for traffic aware energy efficient routing in sdn, 2021.

[102] Shin ichi Kuribayashi. Improving quality of service and reducing power consumption with wan accelerator in cloud computing environments, 2013.

[103] Pengfei Li, Jianyi Yang, Adam Wierman, and Shaolei Ren. Towards environmentally equitable ai via geographical load balancing, 2024.

[104] Ricardo N. Boing, Hugo Vaz Sampaio, Fernando Koch, Rene N. S. Cruz, and Carlos B. Westphall. Distributed load orchestration for vision computing in multi-access edge computing, 2022.

[105] Amir Moaddeli, Iman Nabati Ahmadi, and Negin Abhar. The power of d choices in scheduling for data centers with heterogeneous servers, 2019.

[106] Sakshi Chhabra and Ashutosh Kumar Singh. A comprehensive vision on cloud computing environment: Emerging challenges and future research directions, 2022.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.