Data Center Network Architectures for Large-Scale Distributed Machine Learning

Iris Wang, Kiran Sai Batta, Jin-Lin Tsai

Data Center Network Architectures for Large-Scale Distributed Machine Learning

Iris Wang, Kiran Sai Batta, Jin-lin Tsai
iriswang31, jin-lintsai@outlook.com
University of Cincinnati

Abstract

As ML applications become increasingly prevalent and complex, the underlying network infrastructure must evolve to meet the demands of high-volume, high-speed data transfers essential for efficient ML operations. We explore the transition from traditional data center architectures to advanced designs featuring high-bandwidth fabrics and software-defined networking (SDN), which facilitate the dynamic management of network traffic and enhance scalability. Case studies of Google's Jupiter Network and Facebook's data center strategies illustrate how industry leaders are successfully scaling their networks to support vast data and computational demands. Despite these advancements, challenges such as network scalability, energy efficiency, and the integration of AI-driven network management persist. Addressing these challenges is crucial for the continued advancement of DCNs, which are pivotal in optimizing the performance and sustainability of machine learning workflows. This paper underscores the vital role of innovative network designs and strategies in propelling ML capabilities forward, highlighting both current solutions and future directions for research and development in the field of data center networks.

1 Introduction

In the era of big data, machine learning (ML) has emerged as a transformative force across numerous sectors, driving innovations that were once the realm of science fiction into everyday reality [1, 2]. Applications such as advanced image recognition enable autonomous vehicles to navigate complex environments safely, while sophisticated natural language processing algorithms empower digital assistants to understand and respond to human speech with remarkable accuracy [3, 4, 5]. These technologies are not only enhancing user experiences but are also revolutionizing industries by providing deeper insights into data, thus enabling better decision-making. For instance, in healthcare, ML models analyze patterns in medical imaging to detect diseases earlier than ever before [6], potentially saving lives with preemptive treatment options.

The breadth of these applications highlights a critical development in computational technology: as machine learning algorithms become more capable, they also become more complex and data-intensive. Modern deep learning models, with their deep networks and vast numbers of parameters, require significant computational power and

extensive data handling capabilities [7]. Training these models involves processing large datasets that can scale to petabytes, with billions of parameters being updated repeatedly [8]. This level of computation is not feasible with outdated technology; it demands cutting-edge infrastructure that can sustain and expedite such immense data flows [9, 10].

Central to the functionality of these advanced computational frameworks is the network infrastructure within data centers. Data center networks (DCNs) are pivotal in managing the data throughput necessary for the efficient performance of distributed ML systems [11, 12, 13]. In these systems, multiple processors, often spread across different geographical locations, work in tandem to handle various parts of the ML model [14, 15]. Each processor must communicate its findings and updates to others—a task heavily reliant on the network's capacity to handle large-scale data exchanges swiftly and reliably.

However, this reliance introduces significant challenges. Network delays, congestion, and bandwidth limitations can severely impede the progress of distributed ML training sessions, leading to inefficiencies that scale with the size of the data and complexity of the models. As such, the design and optimization of network infrastructure become as crucial as the computational hardware itself [16, 17]. A poorly optimized network can become a critical bottleneck, stalling data exchanges and undermining the advantages of parallel processing across multiple nodes.

This paper aims to dissect these challenges by exploring the evolution of data center networks specifically engineered to support the demanding requirements of large-scale, distributed machine learning training. We will examine how innovations in network design, such as the adoption of software-defined networking (SDN) [18], high-performance routing protocols, and advanced data transfer technologies like RDMA (Remote Direct Memory Access) [19], are critical in overcoming the bottlenecks traditionally associated with large-scale data processing.

2 Evolution of Data Center Networks

The evolution of data center networks (DCNs) reflects the broader technological advancements and growing demands of computing infrastructures required to support increasingly complex applications, including large-scale machine learning (ML) training. This section outlines the transformation from early basic data center designs to the sophisticated architectures equipped with high-bandwidth fabrics and software-defined networking (SDN) capabilities.

2.1 Early Architectures

The foundational architectures of early data centers were primarily designed for centralized computing, where simple client-server models predominated. These data centers typically utilized traditional three-tier architectures composed of core, aggregation, and access layers [20, 21]. However, these early designs were constrained by several significant limitations when tasked with handling large-scale, data-intensive applications. Network bottlenecks were common, as the hierarchical design limited the number of available paths for data traffic, causing delays and congestion as more devices and ap-

plications were added [22, 23, 24]. Furthermore, these architectures were not inherently designed for redundancy, leading to potential points of failure that could disrupt the entire network[25]. The rigidity of these early network designs made them unsuitable for the dynamic scaling requirements of advanced computational models like those used in ML.

2.2 High-Bandwidth Fabrics

As computational demands grew, particularly with the advent of big data and ML, the need for more efficient data handling within data centers became apparent. This need led to the development and deployment of high-bandwidth fabrics, designed to enhance the capacity and speed of networks. Ethernet fabrics, particularly those offering 10 Gbps, 40 Gbps, and even 100 Gbps, became crucial in addressing the previous limitations of network architectures [26]. These high-throughput fabrics provide not only greater bandwidth but also reduced latency and improved data flow control across the network. The introduction of spine-leaf architectures, replacing the traditional three-tier design, allowed for any-to-any connectivity, drastically reducing bottlenecks and improving resilience and fault tolerance. Such configurations are particularly beneficial for distributed ML workloads, where simultaneous data exchanges between numerous nodes are common.

2.3 Software-Defined Networking (SDN)

The integration of software-defined networking (SDN) into data center networks marked a significant shift towards more agile and efficient network management [27]. SDN separates the control plane from the data plane, allowing network administrators to manage data flows dynamically and centrally without requiring physical access to the network switches. This capability is particularly advantageous for adapting to the fluctuating demands of ML training workloads, where network traffic patterns can change rapidly and unpredictably [28, 29]. SDN enables the provision of on-demand, scalable network resources, optimizing the performance of ML workloads by reducing latency and avoiding data congestion. The programmability of SDN allows for the automation of network configurations and the enforcement of policies that ensure data packets are prioritized and routed efficiently, enhancing the overall throughput and performance of data centers [18].

Through these evolutionary steps, data center networks have become capable of supporting the high demands of modern ML applications, facilitating rapid and reliable data exchanges that are crucial for the distributed training of complex models. These advancements in network design and technology not only mitigate the physical constraints of earlier architectures but also introduce a level of flexibility and scalability that is vital for the future growth of machine learning technologies.

3 Networking Technologies for Machine Learning

The network infrastructure supporting machine learning (ML) workloads must be robust, efficient, and highly scalable. Various network technologies have been developed

or adapted to meet these needs, including specialized network topologies, advanced routing protocols, and optimizations for bandwidth and latency [20, 30, 31, 32]. This section explores how these technologies specifically benefit ML operations, focusing on enhancing data flow efficiency and reducing training time.

3.1 Network Topologies

Two predominant network topologies that have become integral in optimizing data center networks for ML workloads are Spine-Leaf and Fat-Tree architectures.

Spine-Leaf Topology [33]: This topology is designed to minimize latency by reducing the number of hops between servers. In a spine-leaf configuration, all leaf switches (where the servers are connected) are interconnected through multiple spine switches. This setup ensures high bandwidth and low-latency connections between any two nodes in the network, which is crucial for distributed ML tasks that require frequent and fast data exchanges. The scalability of this topology is particularly beneficial for ML, as additional leaf switches (and thus more servers) can be added without significantly affecting the performance of existing operations.

Fat-Tree Topology [20]: Fat-Tree is another effective topology for ML applications, especially in environments where uniform bandwidth is required across all connections. It uses a tiered approach where each leaf switch is connected to every spine switch, ensuring multiple paths for data to travel and reducing the possibility of congestion. This redundancy not only improves fault tolerance but also maintains consistent performance levels during high-demand scenarios, which are common in large-scale ML training sessions.

3.2 Advanced Routing Protocols

To further enhance network efficiency, advanced routing protocols that adapt to current network conditions are employed. Adaptive and predictive routing protocols [34, 35, 36] can dynamically change data paths based on network congestion, available bandwidth, and other real-time metrics. This flexibility helps in:

- Minimizing Latency: By choosing the least congested paths, these protocols reduce the time it takes for data to travel between nodes.
- Maximizing Throughput: Efficiently utilizing the available network capacity ensures that data transfers do not become a bottleneck in the training process.
- Enhancing Reliability: Dynamic path selection helps avoid potential points of failure, which is critical for maintaining the integrity of ML training processes.

3.3 Bandwidth and Latency Optimizations

Techniques such as Remote Direct Memory Access (RDMA) are pivotal in optimizing network communications for ML workloads [19]. RDMA allows high-speed memory-to-memory data transfers across the network, bypassing the operating system to reduce CPU overhead.

By eliminating the need for data to travel through the CPU, RDMA significantly reduces latency. This leads to an increased bandwidth utilization, with freed-up CPU resources allowing more data to be processed simultaneously, effectively increasing the network bandwidth for ML tasks. CPU Load is also reduced by minimizing CPU involvement in data transfers allows more processor resources to be allocated to actual ML computations, thus optimizing overall system performance.

The integration of these networking technologies into data center architectures forms a foundational layer that supports the complex and intensive demands of distributed ML training. By effectively managing data flows, these technologies ensure that network performance aligns with the computational requirements of modern ML algorithms, thereby enabling faster, more efficient training cycles.

4 Machine Learning-Specific Network Considerations

Effective network design is crucial for optimizing the performance of machine learning (ML) systems, particularly in distributed training environments where data and computations are spread across multiple nodes [37, 38]. This section delves into the specific network considerations necessary for ML, focusing on data transfer requirements, synchronization mechanisms, and the impact of network design on ML performance.

4.1 Data Transfer Requirements

Distributed ML training, commonly employed in deep learning frameworks like TensorFlow and PyTorch [39, 40], relies heavily on the ability to move large volumes of data quickly across the network. Each node in a distributed setup works on a subset of the overall data, requiring rapid and frequent exchanges of weight updates and gradients with other nodes. High-speed data transfers are critical to prevent these data exchanges from becoming bottlenecks that can delay the entire training process [22, 41]. Networks need to be designed with high throughput capabilities to handle these large data flows efficiently, ensuring that all nodes receive and transmit the necessary data without delays. The implementation of technologies such as high-bandwidth Ethernet, InfiniBand, or fiber channels is often necessary to meet these demands.

4.2 Synchronization Mechanisms

In distributed ML, synchronization of gradient updates across all nodes is essential for the convergence of the model. Efficient synchronization mechanisms ensure that all nodes update their model parameters consistently and correctly after each iteration of training. This is often achieved through either parameter servers or collective operations like All-Reduce, which need to be supported by robust network protocols to manage the synchronization without excessive latency. The network's ability to handle these synchronization tasks efficiently affects not only the speed of model training but also the stability and scalability of the training process [42, 43]. Inefficient synchronization can lead to slower convergence rates or divergence of the model, significantly impacting the overall training time and effectiveness.

4.3 Latency, Bandwidth, and Loss

Low latency is crucial for quick response times in synchronous training scenarios, where nodes must wait for each other to exchange information before proceeding. High latency can increase the time it takes to complete each training epoch, elongating the overall training period. Adequate bandwidth is necessary to handle the high data volumes transferred during ML training, especially in models with large numbers of parameters. Insufficient bandwidth can cause delays and slow down the training process, particularly when scaling up the number of nodes in the training cluster. In a network environment, packet loss can occur due to congestion, hardware failures, or poor network configurations [44, 45, 46]. In ML training, lost packets mean lost data, which can corrupt the synchronization of model updates and lead to inaccuracies in the trained model. Ensuring minimal packet loss through reliable network setups is essential for maintaining the integrity and accuracy of the ML model.

Overall, the network design must be tailored not only to support the high demands of ML workloads in terms of data volume and speed but also to ensure the reliability and consistency of data transfers essential for accurate and efficient model training. By addressing these considerations, organizations can enhance their ML capabilities, leading to faster training times and more accurate models, thereby gaining a competitive edge in utilizing AI technologies.

5 Case Studies: Innovations in Data Center Networks for Machine Learning

Understanding the practical applications of advanced data center networks can be elucidated through examining the approaches taken by leading technology firms. Google's Jupiter Network and Facebook's data center network strategies provide compelling insights into how modern networks are designed and optimized to support large-scale machine learning (ML) workloads and other intensive applications.

5.1 Google's Jupiter Network

Google's Jupiter Network [47] exemplifies how scalable and robust network infrastructure can support expansive data requirements across services, including search, Gmail, and ML workloads. Jupiter can deliver over 1 Petabit/sec of total bisection bandwidth, which allows it to efficiently handle massive amounts of data and computation demands across Google's services. This capacity is crucial for supporting the data-intensive tasks involved in training and deploying ML models, such as those used in Google's advertising technologies and real-time language translation services.

The architecture of Jupiter is based on a Clos topology, a form of multistage networking topology that enhances data throughput via multiple paths between any two points in the network, significantly reducing the likelihood of bottlenecks. Google has also integrated custom-designed and highly scalable software that manages the network layers dynamically. This software ensures optimal data routing based on current network load, which is pivotal for distributed ML tasks where data synchronization needs to occur frequently and swiftly across numerous nodes. The ability of Jupiter to scale

with Google's computational needs while maintaining low latency and high throughput is central to its ability to expedite ML processes, from training to inference.

5.2 Facebook's Data Center Network

Facebook has engineered its data center network to handle the enormous scale of its operations, which include user data processing, image and video uploads, and sophisticated ML algorithms for content recommendation and advertising [48]. Facebook's approach to data center network design is notable for its emphasis on fabric architecture, which allows for flexible, scalable, and efficient connectivity across servers and data centers.

One of the key innovations in Facebook's network is its deployment of Fabric Aggregator, a system designed to manage and route internal traffic efficiently within its data centers. This system uses a custom-built routing protocol that optimizes the flow of data based on real-time usage patterns and the operational requirements of different applications, including ML workloads. The Fabric Aggregator enhances the ability of Facebook's network to handle spikes in data flow, which is crucial for ML applications that require large datasets to be moved quickly across the network for processing and analysis.

Additionally, Facebook has made significant strides in implementing efficient cooling and power distribution techniques in its data centers, which indirectly supports the intensive computing tasks associated with ML by reducing downtime and improving overall system reliability. The physical and network infrastructure optimizations together ensure that Facebook can run its ML operations continuously and efficiently, driving innovations in AI and serving billions of users globally with personalized content.

These case studies from Google and Facebook illustrate how leading tech companies have developed their data center networks to not only meet the current demands of ML workloads but also to anticipate future needs and scalability challenges. The innovations highlighted in these examples provide valuable lessons in network design and operation that could be applicable across a variety of industries engaging in large-scale ML initiatives.

6 Challenges and Future Directions

As data center networks continue to evolve to support increasingly complex machine learning (ML) workloads, they face a number of significant challenges. Addressing these challenges effectively is crucial for the continued growth and efficiency of these technologies. This section explores key issues such as scalability, energy efficiency, and the potential for AI-driven network management, highlighting both current obstacles and future directions.

Scalability One of the primary challenges facing modern data center networks is scalability. As ML models become more complex and datasets grow in size, the demand on network infrastructure increases exponentially. Scaling network capabilities to keep pace with these demands involves not only upgrading physical hardware but also optimizing network architectures to handle larger volumes of data transfers effi-

ciently. The need for high throughput and low latency becomes more pronounced as the number of nodes in distributed ML training expands.

The future direction in addressing scalability involves further innovations in network design, such as more advanced implementations of spine-leaf and other scalable topologies, and the adoption of next-generation transmission standards like 400G Ethernet. Additionally, software solutions such as more intelligent load balancing and traffic management algorithms will play a crucial role in ensuring that networks can scale effectively without compromising on performance.

Energy Efficiency Data centers are notoriously energy-intensive, consuming a significant amount of electricity both for powering computing equipment and for cooling systems necessary to dissipate the heat generated by that equipment. As data center operations expand to accommodate larger ML workloads, their energy consumption is likely to increase unless new efficiencies are introduced. Reducing the energy footprint of data centers while maintaining high performance is therefore a critical challenge.

Innovative solutions such as using renewable energy sources, improving the efficiency of cooling systems, and designing more energy-efficient hardware are key to future developments. Furthermore, advanced software that can more precisely control power use and thermal management within data centers will contribute to energy efficiency. Optimizing workload distributions to minimize power consumption without impacting performance is another area ripe for exploration.

AI-Driven Network Management The potential of using AI to manage and optimize network traffic is an exciting frontier in network technology. AI-driven network management can leverage ML techniques to predict traffic patterns, identify potential bottlenecks before they occur, and dynamically adjust routing and load balancing to optimize network performance. This proactive management approach can significantly enhance the responsiveness and efficiency of data center networks.

Future directions in AI-driven network management include the development of self-optimizing networks that can continuously learn and adapt to changing conditions without human intervention. Such networks would use real-time data to make decisions about configurations, predict and resolve network failures, and optimize data flows to improve both performance and energy usage. The integration of AI into network management promises not only to improve operational efficiencies but also to enable more sophisticated ML training capabilities, potentially leading to more rapid advancements in ML applications.

Addressing these challenges will require a multi-faceted approach that combines hardware innovations, software advancements, and novel uses of AI. By embracing these future directions, the development of data center networks can continue to support the growing demands of machine learning and other advanced applications.

7 Conclusion

The exploration of data center networks (DCNs) in this paper has highlighted the critical role that advanced network infrastructure plays in supporting the intensive demands of modern machine learning (ML) applications. From the evolution of network architectures to the integration of cutting-edge networking technologies, the continuous development of DCNs is fundamental to enabling efficient, scalable, and effective ML

operations.

We have examined the transformative shifts from traditional network designs to more sophisticated architectures like spine-leaf and fat-tree topologies, which are essential for supporting the high-throughput and low-latency requirements of distributed ML training. Innovations such as high-bandwidth fabrics and software-defined networking (SDN) have proven pivotal in enhancing the performance and agility of networks, allowing them to dynamically adjust to the varying demands of ML workloads.

The case studies of Google's Jupiter Network and Facebook's data center initiatives have provided concrete examples of how leading technology companies are scaling their network capabilities. These examples underscore the importance of robust network designs in handling the large-scale data processing needs essential for training complex ML models and serving billions of global users.

However, the path forward is not without challenges. Scalability remains a significant concern as ML models and datasets continue to grow in size and complexity. Energy efficiency also stands out as a critical issue, with the need to mitigate the environmental impact of expanding data center operations while maintaining high levels of performance. Furthermore, the potential of AI-driven network management suggests a promising future where networks not only support but also advance through intelligent, automated processes that optimize performance and efficiency.

As we look to the future, it is clear that the development of data center networks will continue to be a dynamic field of innovation. Addressing the outlined challenges and embracing the new directions will be essential for harnessing the full potential of machine learning technologies. By continuing to evolve and adapt, DCNs will not only meet the current demands of ML applications but will also drive forward the next generation of technological advancements. This ongoing evolution will undoubtedly play a decisive role in shaping the future landscape of technology and its applications across various industries.

References

[1] Batta Mahesh. "Machine learning algorithms-a review". In: *International Journal of Science and Research (IJSR).[Internet]* 9.1 (2020), pp. 381–386.

[2] Xindong Wu et al. "Data mining with big data". In: *IEEE transactions on knowledge and data engineering* 26.1 (2013), pp. 97–107.

[3] Jesse Levinson et al. "Towards fully autonomous driving: Systems and algorithms". In: *2011 IEEE intelligent vehicles symposium (IV)*. IEEE. 2011, pp. 163–168.

[4] Barret Zoph et al. "Learning transferable architectures for scalable image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8697–8710.

[5] Ashish Vaswani. "Attention is all you need". In: *arXiv preprint arXiv:1706.03762* (2017).

[6] Bradley J Erickson et al. "Machine learning for medical imaging". In: *radiographics* 37.2 (2017), pp. 505–515.

[7] Joel Hestness et al. "Deep learning scaling is predictable, empirically". In: *arXiv preprint arXiv:1712.00409* (2017).

[8] Josh Achiam et al. "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774* (2023).

[9] Zhe Fan et al. "GPU cluster for high performance computing". In: *SC'04: Proceedings of the 2004 ACM/IEEE conference on Supercomputing*. IEEE. 2004, pp. 47–47.

[10] Wu-chun Feng and Kirk Cameron. "The green500 list: Encouraging sustainable supercomputing". In: *Computer* 40.12 (2007), pp. 50–55.

[11] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. "A scalable, commodity data center network architecture". In: *ACM SIGCOMM computer communication review* 38.4 (2008), pp. 63–74.

[12] Albert Greenberg et al. "VL2: A scalable and flexible data center network". In: *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. 2009, pp. 51–62.

[13] Xin Wang et al. "Union: An automatic workload manager for accelerating network simulation". In: *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE. 2020, pp. 821–830.

[14] Jakub Konečnỳ et al. "Federated optimization: Distributed machine learning for on-device intelligence". In: *arXiv preprint arXiv:1610.02527* (2016).

[15] Solmaz Niknam, Harpreet S Dhillon, and Jeffrey H Reed. "Federated learning for wireless communications: Motivation, opportunities, and challenges". In: *IEEE Communications Magazine* 58.6 (2020), pp. 46–51.

[16] Radhika Mittal et al. "TIMELY: RTT-based congestion control for the datacenter". In: *ACM SIGCOMM Computer Communication Review* 45.4 (2015), pp. 537–550.

[17] Arjun Roy et al. "Inside the social network's (datacenter) network". In: *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. 2015, pp. 123–137.

[18] Hyojoon Kim and Nick Feamster. "Improving network management with software defined networking". In: *IEEE Communications Magazine* 51.2 (2013), pp. 114–119.

[19] Chuanxiong Guo et al. "RDMA over commodity ethernet at scale". In: *Proceedings of the 2016 ACM SIGCOMM Conference*. 2016, pp. 202–215.

[20] Charles E Leiserson. "Fat-trees: Universal networks for hardware-efficient supercomputing". In: *IEEE transactions on Computers* 100.10 (1985), pp. 892–901.

[21] Fabrizio Petrini and Marco Vanneschi. "k-ary n-trees: High performance networks for massively parallel architectures". In: *Proceedings 11th international parallel processing symposium*. IEEE. 1997, pp. 87–93.

[22] Staci A Smith et al. "Mitigating inter-job interference using adaptive flow-aware routing". In: *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2018, pp. 346–360.

[23] Kewen Wang et al. "Modeling interference for apache spark jobs". In: *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*. IEEE. 2016, pp. 423–431.

[24] Yao Kang, Xin Wang, and Zhiling Lan. "Workload interference prevention with intelligent routing and flexible job placement on dragonfly". In: *Proceedings of the 2023 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. 2023, pp. 23–33.

[25] M Farhan Habib et al. "Design of disaster-resilient optical datacenter networks". In: *Journal of Lightwave Technology* 30.16 (2012), pp. 2563–2573.

[26] *Infiniband roadmap*. URL: https://www.infinibandta.org/infiniband-roadmap/.

[27] Pat Bosshart et al. "P4: Programming protocol-independent packet processors". In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 87–95.

[28] Laizhong Cui et al. "A survey on application of machine learning for Internet of Things". In: *International Journal of Machine Learning and Cybernetics* 9 (2018), pp. 1399–1417.

[29] Yao Kang et al. "Modeling and analysis of application interference on dragonfly+". In: *Proceedings of the 2019 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. 2019, pp. 161–172.

[30] John Kim et al. "Technology-driven, highly-scalable dragonfly topology". In: *ACM SIGARCH Computer Architecture News* 36.3 (2008), pp. 77–88.

[31] Jung Ho Ahn et al. "HyperX: topology, routing, and packaging of efficient large-scale networks". In: *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*. 2009, pp. 1–11.

[32] Daniele De Sensi et al. "An in-depth analysis of the slingshot interconnect". In: *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2020, pp. 1–14.

[33] Gustavo Alessandro Andrade Santana. *Data Center Virtualization Fundamentals*. Pearson Education, 2014.

[34] Jongmin Won et al. "Overcoming far-end congestion in large-scale networks". In: *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*. IEEE. 2015, pp. 415–427.

[35] Daniele De Sensi, Salvatore Di Girolamo, and Torsten Hoefler. "Mitigating network noise on dragonfly networks through application-aware routing". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2019, pp. 1–32.

[36] Yao Kang, Xin Wang, and Zhiling Lan. "Q-adaptive: A multi-agent reinforcement learning based routing on dragonfly network". In: *Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing*. 2021, pp. 189–200.

[37] Torsten Hoefler et al. "HammingMesh: a network topology for large-scale deep learning". In: *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2022, pp. 1–18.

[38] Kartik Lakhotia et al. “PolarFly: a cost-effective and flexible low-diameter topology”. In: *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2022, pp. 1–15.

[39] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019).

[40] Martín Abadi et al. “{TensorFlow}: a system for {Large-Scale} machine learning”. In: *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 2016, pp. 265–283.

[41] Yao Kang, Xin Wang, and Zhiling Lan. “Study of workload interference with intelligent routing on dragonfly”. In: *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2022, pp. 1–14.

[42] Shigang Li and Torsten Hoefler. “Near-optimal sparse allreduce for distributed deep learning”. In: *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*. 2022, pp. 135–149.

[43] Minsik Cho et al. “Blueconnect: Decomposing all-reduce for deep learning on heterogeneous network hierarchy”. In: *Proceedings of Machine Learning and Systems* 1 (2019), pp. 241–251.

[44] Yao Kang. “Workload Interference Analysis and Mitigation on Dragonfly Class Networks”. PhD thesis. Illinois Institute of Technology, 2022.

[45] Yuliang Li et al. “LossRadar: Fast detection of lost packets in data center networks”. In: *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*. 2016, pp. 481–495.

[46] Danyang Zhuo et al. “Understanding and mitigating packet corruption in data center networks”. In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 2017, pp. 362–375.

[47] Arjun Singh et al. “Jupiter rising: A decade of clos topologies and centralized control in google’s datacenter network”. In: *ACM SIGCOMM computer communication review* 45.4 (2015), pp. 183–197.

[48] Kim Hazelwood et al. “Applied machine learning at facebook: A datacenter infrastructure perspective”. In: *2018 IEEE international symposium on high performance computer architecture (HPCA)*. IEEE. 2018, pp. 620–629.