Towards a Manageable Intra-Host Network

Xinhao Kong
Duke University
Durham, NC, USA

Jiaqi Lou
University of Illinois
Urbana-Champaign, IL, USA

Wei Bai
Microsoft Research
Redmond, WA, USA

Nam Sung Kim
University of Illinois
Urbana-Champaign, IL, USA

Danyang Zhuo
Duke University
Durham, NC, USA

ABSTRACT

Intra-host networks, including heterogeneous devices and interconnect fabrics, have become increasingly complex and crucial. However, intra-host networks today do not provide sufficient manageability. This prevents data center operators from running a reliable and efficient end-to-end network, especially for multi-tenant clouds. In this paper, we analyze the main manageability deficiencies of intra-host networks and argue that a systematic solution should be implemented to bridge this function gap. We propose two key building blocks for a manageable intra-host network: a fine-grained monitoring system and a holistic resource manager. We discuss the research questions associated with realizing these two building blocks.

CCS CONCEPTS

• Networks → Network manageability; **• Computer systems organization → Architectures**.

1 INTRODUCTION

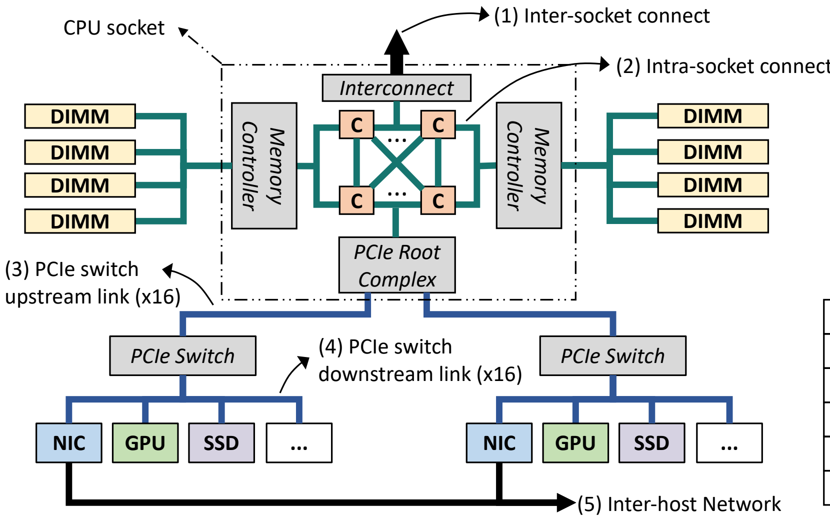
Recent years have witnessed the increasing complexity of servers. Numerous heterogeneous hardware accelerators have been adopted in commodity infrastructure for various purposes. A state-of-the-art commodity server (e.g., NVIDIA DGX [45]) can be equipped with eight high-speed Infiniband network adapters and eight GPUs to achieve superior computation and communication capabilities. These hardware accelerators, as well as the CPU sockets, DRAMs, and storage devices (e.g., NVMe SSD), have formed a complicated *intra-host network* with a comparable complexity as a data center rack.

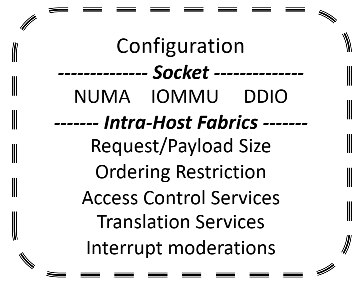
Conventional wisdom believes that the intra-host network fabric (such as the PCIe and memory bus) delivers stable and high performance. For inter-host communication, the latency introduced by the intra-host network is usually sub-microsecond, or at most a few microseconds. This latency is negligible compared to the milliseconds introduced by the Ethernet fabric. However, a line of recent works has demonstrated a paradigm shift in today's data centers [2, 43]. Fast inter-host communication technologies like RDMA and DPDK have significantly improved the inter-host network performance [10, 18, 19, 25]. The microsecond-level intra-host latency thus can become a main contributor to the end-to-end latency, and can even be the bottleneck for end-to-end distributed systems [40]. Besides, high-performance heterogeneous accelerators demand substantial intra-host network resources (e.g., bandwidth). The intra-host fabric, therefore, becomes unprecedentedly congested and causes performance jitters and even anomalies [2, 17, 31–33, 37].

The roles of the intra-host network and the inter-host network have become equally important. However, the capability to effectively monitor and control the intra-host networks is far less mature than that for inter-host networks. We call this capability the *manageability* of the network. Inter-host network manageability is a well-studied area. For example, many works have been proposed to improve inter-host network manageability from a variety of aspects, including network telemetry [22, 39, 54, 55], traffic engineering [3, 4, 41], load balancing [6, 26, 57], network troubleshooting [9, 23, 52], quality-of-service (QoS) guarantee and performance isolation [8, 11, 12, 24, 30, 47, 48, 50]. Unfortunately, almost none of these functionalities are supported in today's intra-host networks. The insufficient manageability of intra-host networks prevents data center operators from building a high-performance and highly reliable end-to-end network, where the intra-host network serves as the last hop. We believe that two key features are missing. First, the lack of telemetry and monitoring systems leads to poor *observability*, which prevents data center operators from conducting efficient failure analyses and troubleshooting. Besides, the lack of *resource*

HOTOS '23, June 22–24, 2023, Providence, RI, USA

ACM ISBN 979-8-4007-0195-5/23/06…$15.00
https://doi.org/10.1145/3593856.3595890





Item	Capacity	Basic Latency
(1)	20-72 GBps	130-220ns
(2)	100-200 GBps	2-110ns
(3)	~256 Gbps	30-120ns
(4)	~256 Gbps	30-120ns
(5)	~200 Gbps	< 2us

Figure 1: An example topology of a commodity server. The highlighted links depict the complex intra-host network within a modern server. The performance numbers describe the capacity and basic latency of common devices in commodity data center servers, such as Intel Cascade Lake and AMD EPYC CPUs, and PCIe 4.0 buses.

management leads to performance interference. Similar to inter-host networks, multiple users can share the same intra-host fabrics (e.g., PCIe and memory bus) simultaneously. Without careful resource management, one buggy or malicious user may exhaust the resources of some intra-host fabric (e.g, bandwidth) and cause other users to experience poor performance. This interference impairs delivering predictable application performance and can even cause security issues under multi-tenant scenarios.

In this paper, we argue that it is time to rethink the design of intra-host networks for manageability. To this end, we draw inspiration from the design of inter-host networks and propose two key building blocks for a manageable intra-host network: a fine-grained monitoring system and a holistic resource manager. These two building blocks will bridge the manageability gap between the intra-host and the inter-host network. For these two building blocks, we will discuss their key utilities, as well as the new research challenges raised in realizing these building blocks.

2 MANAGEABILITY ISSUES IN INTRA-HOST NETWORKS

Data center applications are constantly striving for more computational resources and better performance. To meet this growing demand, modern data center servers are equipped with advanced CPUs, and many heterogeneous hardware accelerators are being incorporated into these servers to further accelerate computation (e.g., GPU, FPGA), communication (e.g., RDMA NIC), and improve storage performance (e.g., NVMe SSD). These hardware components are connected through various types of intra-host fabrics, including PCIe, intra-socket connects (e.g., memory buses) and inter-socket connects (e.g., Intel QPI and AMD Infinity). We name these fabrics and the end node devices together as the intra-host network.

The performance of intra-host networks is crucial in today's data centers. The primary reason is that the performance of the inter-host network has been advancing rapidly. 200 Gbps Ethernet NICs and switches have already been widely used in data centers, and their latency is within a single-digit microsecond [44, 46]. This trend makes the performance overhead (such as latency) incurred by the intra-host network no longer negligible. We present an example intra-host network and the key performance numbers in Figure 1. The topology and the numbers are mostly based on previous measurement works [38, 43, 49, 53, 56]. The specific numbers depend on the specific device type and the technology. For example, the intra-socket connect latency varies depending on interconnect architecture (e.g., core mesh and the NUMA topology) and the type of memory access (e.g., cache/DRAM and local/remote). The intra-socket capacity (e.g., memory bandwidth) is also determined by the number and the type of memory channels enabled on the memory controller. Therefore, we present general order of magnitude ranges for commodity hardware, which should be sufficient to provide a sense of the intra-host network performance. The fabric latency includes the processing delay of the corresponding component (e.g., PCIe switch). The sum latency of end-to-end access, such as a remote RDMA access traversing all the (1) to (5), can make the intra-host network become the potential bottleneck for the entire end-to-end networked system. A line of research has been proposed to improve the intra-host network performance by tuning the host topology, scheduling the applications, and even introducing new protocols and host interconnects [5, 7, 21, 31, 35, 49]. For example, Compute Express Link (CXL) [49] exposes memory in devices as remote memory in a NUMA system, and it enables devices to directly access host local memory through a cache coherence interface. These features provide a more flexible memory model and reduce the overhead (e.g., with a latency of ~150ns from device to host memory [49]) for intra-host access.

As the performance of the intra-host and the inter-host start to match, we observe two irritating problems.

The intra-host network becomes a constant source of performance anomalies, but debugging a complex intra-host network is difficult. The complexity of intra-host networks in data centers is steadily increasing, and congestion in the intra-host network causes application-level performance anomalies [2, 33, 37, 43]. For example, an RDMA loopback traffic can exhaust the PCIe bandwidth and causes the application to suffer from PCIe congestion [31]. Pinpointing problems in the intra-host network is notoriously difficult due to the lack of observability in such a complex network. As shown in Figure 1, advanced CPUs have complicated internal microarchitectures. A single CPU socket today can have tens of CPU cores and support multiple PCIe root ports and multiple memory controllers, which creates complex interconnects among cores and between sockets. This also allows the incorporation of various types of devices. For example, multiple PCIe root ports can connect a single CPU socket to more than ten PCIe devices, with a complex multi-level PCIe fabric (e.g, root ports and several PCIe switches). Furthermore, such incorporation leads to a greater number of possible configurations. The dashed box in Figure 1 shows part of the possible configurations, which also heavily impact the performance of intra-host connections. For example, Intel Data Direct I/O technology (DDIO) [27] enables I/O devices to transfer data to the last level cache directly, thus the configuration of DDIO determines the specific communication pathway between I/O devices and the CPU. The trend of hardware offloading will bring more heterogeneous I/O devices into the intra-host networks, and the emerging protocols (e.g., CXL) will also enable more flexible communication patterns. These trends only make this debugging problem predictably worse.

The hardware devices connected by the intra-host network have certain observability features. They usually expose some performance and diagnostic counters. For example, Intel provides a Performance Counter Monitor (PCM) [28] and Resource Director Technology (RDT) [29] that collect hardware counters and reports overall statistics of PCIe, inter-socket connects, memory buses, and CPU cache. Modern Ethernet adapters also provide the received/transmitted bytes/packets counters [42, 44]. However, monitoring these statistics only provides the limited debugging ability. For example, data center operators can use these counters to detect congestion, but identifying the root cause of the congestion and detecting other complex performance anomalies remains challenging.

The intra-host network is a source of performance interference. As the number of hardware accelerators integrated into the host increases, the intra-host network carries more heterogeneous traffic and becomes increasingly congested. Applications that use these high-throughput hardware accelerators can consume a large amount of intra-host network bandwidth, causing intra-host network congestion. When multiple applications are running on the server, sharing bottlenecked intra-host network resources leads to performance interference and degraded quality of service (QoS). For instance, a remote key-value store client and a machine learning application may be co-located on the same host. The machine learning application may have a substantial workload for CPU-GPU communication (e.g., loading training data) and heavily utilize the bandwidth of the PCIe fabric and the memory bus. The key-value store application seems to have no interference with the machine learning application since it does not use GPU at all. However, the traffic of the remote key-value store application may traverse the same PCIe root port and the memory bus and therefore suffer from high latency and poor application performance due to the high utilization of these intra-host fabrics. The situation becomes even worse in multi-tenant scenarios, where tenants may maliciously exhaust intra-host network fabric resources and impair others.

Another type of interference originates from the tight coupling of the intra-host network and other components, including the inter-host network and other host devices. Besides the normal I/O requests (e.g., DMA for payload) that consume the intra-host network resources, there are many unintended resource consumption caused by various reasons. For example, with DDIO enabled, high-bandwidth PCIe devices such as high-performance NIC and RAID SSDs can directly write to the dedicated last-level cache (LLC) ways. However, due to the limited cache spaces and the high throughput direct write, these two devices can cause cache thrashing and the data are evicted from the cache before being consumed by the applications. This cache thrashing ultimately leads to more consumption of the intra-host network resources (e.g., memory bus bandwidth). Similar thrashing also happens to devices with an on-chip cache, such as RDMA NICs and NVMe SSDs.

There are some hardware features that mitigate intra-host network performance interference. For example, Intel RDT technology supports allocating memory bandwidth to different tenants (e.g, VMs), which mitigates the performance interference from the memory bus. Unfortunately, these features only provide limited point solutions that mitigate interference from specific components in a coarse-grained way. They therefore cannot provide an efficient and integrated solution that eliminates the end-to-end performance interference from the intra-host network.

The root cause for the abovementioned two problems is that *modern intra-host networks do not have adequate manageability.* This makes operating an efficient and reliable end-to-end network difficult.

3 TOWARDS A MANAGEABLE INTRA-HOST NETWORK

If we want an intra-host network to be manageable, what features should the intra-host network have? This is an open question without a definite answer to date. However, one area we can draw inspiration from is the design of today's inter-host network, which has been managed by data center operators for several decades. Many manageability features have been integrated into the inter-host network over the years, allowing observability and control of inter-host traffic. In this section, we aim to map these manageability features to the intra-host setting and think about (1) what utilities they will bring to the intra-host network, and (2) what are the new research questions associated with realizing these features in the intra-host setting.

3.1 Fine-grained Monitoring System

Data center operators have implemented many fine-grained monitoring systems to improve observability in inter-host networks. These systems provide informative inter-host network usage statistics, enabling network operators to conduct a detailed analysis of the network status, such as failure or performance anomaly detection. This helps to reduce network downtime, improve availability, and potentially improve resource efficiency. Similarly, we imagine that there should be a fine-grained monitoring system in future intra-host networks to provide observability, consisting of the following components.

A monitor for intra-host network configuration and resources. The state of an inter-host network is usually collected periodically by a centralized service [51] to allow for centralized monitoring and control of network traffic. Similarly, a manageable intra-host network should monitor configurations and resource usage on all the links in the intra-host network. The resources include the bandwidth usage of the memory bus and the PCIe bus. Further, the resource usage information should include total and per-tenant (e.g., VM, container) usage statistics of the various types of resources. This allows resource management discussed in §3.2.

A platform for anomaly detection. We believe future intra-host networks should contain a platform to analyze monitoring results holistically, enabling device failure, misconfiguration, and performance anomaly detection. A motivating case is that a hardware failure occurring on the PCIe switch may silently cause the connected PCIe device to suffer performance degradation. Applications thereby suffer from poor communication performance. This cannot be easily detected using performance counters only and will take network operators a long time to debug. This can be addressed by having devices on the intra-host network periodically send "heartbeats" to each other, similar to works like Pingmesh [23] for inter-host networks.

Diagnostic tools for symptom analysis and automatic troubleshooting. A manageable intra-host network should provide a set of diagnostic tools for debugging purposes, such as `ping`, `traceroute`, `iperf`, and `wireshark` in inter-host networks. When a performance issue occurs, data center operators can manually or automatically use these tools to profile the behaviors and performance of the intra-host network. This enables them to pinpoint the root cause of the performance issues efficiently.

The above functions are crucial to increasing the intra-host network observability and hence improving the debugging efficiency and thus the network availability. However, achieving them raises many interesting open questions.

Q1. Informative data and where to find them? The monitoring data can be collected in various ways, such as from hardware device counters and software module interception. However, what will be the best data source for future intra-host network monitoring remains an open question. Software interception is relatively more flexible but may be less informative without visibility into the hardware components. Hardware counters can provide more detailed information but in a coarse-grained way. For example, almost none of today's hardware counters supports accurate per-tenant monitoring, and the access frequency (e.g., data points per second) is usually limited. Future generations of hardware may support more monitoring abilities, but hardware vendors may prioritize utilizing the same amount of hardware resources to further improve the performance over exposing more diagnostic counters.

Q2. The dilemma of storage and processing. The monitor can generate a large amount of data given the complexity of the intra-host networks, and the data needs to be processed efficiently to provide accurate real-time monitoring. This makes where to store and process this data an interesting question. Processing the data locally may consume on-device computation resources, which are usually very limited. However, sending the collected data to other host devices may consume substantial intra-host communication resources. For example, storing data in memory consumes memory bandwidth (and may consume PCIe bandwidth if the source of monitoring data comes from a PCIe device). Additionally, how to achieve real-time monitoring is also a challenging question. Given the ultra-low latency of modern intra-host networks, even micro-second level latency overheads in the monitoring loop can be significant.

Q3. Advanced diagnostic capabilities. Monitoring systems in inter-host networks bring many opportunities to build advanced solutions. For example, it is increasingly a trend to use machine learning for inter-host network failure localization and troubleshooting [1, 9, 14, 15]. Inter-host network links are homogeneous, and so are the collected data. Usually, data center operators monitor transmitted/received bytes per second, packets per second, and packet loss per second on Ethernet links. Intra-host networks are more heterogeneous, so the collected data will have more modalities (e.g., DDIO cache usage, and PCIe bandwidth consumption). This means using machine learning may be more essential in order to leverage these high-modality data for diagnosis than that in inter-host networks.

3.2 Holistic Resource Management

Holistic resource management is one of the most important success factors for inter-host networks to eliminate performance interference and deliver predictable end-to-end performance [8, 13, 30, 34, 47, 48]. For example, inter-host network controllers can allocate a fixed amount of resources (e.g., bandwidth) based on resource model like hose model [16], and enforce the allocation through many resource managers within the inter-host network, including the virtual switch on the end host and the routers in fabrics. We imagine that future intra-host networks should provide similar holistic resource management for manageability, including the following parts.

Virtualized intra-host network abstraction. The manageable intra-host networks for multi-tenant scenarios should provide an independent, virtualized view of the intra-host network for different tenants, similar to the virtual network abstraction provided in inter-host networks [8]. Each tenant should see a dedicated isolated virtual intra-host network. For example, if a tenant is only allocated half of the PCIe bandwidth to an I/O device, from the tenant's perspective, it should see an illusion that the allocated bandwidth is the corresponding PCIe capacity. This provides a clean and simple abstraction for tenant applications. Similar to inter-host networks, such a virtualized abstraction should improve resource efficiency and flexibility for both infrastructure providers and tenant applications. For instance, this abstraction should enable tenants to easily migrate their VMs or containers without reconfiguring their own intra-host networks.

Performance targets interpreter . The manageable intra-host network needs to "interpret" the application intent (i.e., performance targets) into a set of low-level requirements based on a resource model. For example, an application may desire 20 Gbps end-to-end bandwidth for its distributed GPU communication. To meet this target, resources of memory buses and several PCIe links all need to be allocated. The interpreter needs to generate the requirements in a holistic way, enabling different components to collaboratively provide end-to-end allocation. The interpreter should also be general and flexible because the intra-host network topology and capacities may vary on different hosts.

Topology-aware resource scheduler. The intra-host network should schedule the resources based on the topology and current usage. The resource requirements generated by the compiler might be achieved differently. For example, there can be several GPU-SSD pathways within an intra-host network that can support the same amount of bandwidth. The scheduler needs to carefully choose one of the pathways based on topology and usage information to maximize overall resource efficiency when satisfying the requirements.

Dynamic resource arbiter. The manageable intra-host networks should be able to dynamically arbitrate resources at run time. This enables data center operators to enforce any resource schedule plan generated by the scheduler. The arbiter should dynamically adjust the allocation promptly when applications come and go to avoid interference and poor resource utilization.

The compile-schedule-arbitrate scheme as well as the virtualized abstraction allows the intra-host networks to eliminate performance interference and deliver predictable performance based on the applications' intent. We identify several intriguing open questions about such resource management for future intra-host networks.

Q1. What resource model to apply for intra-host networks? Intra-host networks are usually more heterogeneous than inter-host networks, with more types of hardware components and resources. What resource model (e.g., pipe and hose [16]) best fits the intra-host network becomes an interesting question. If work-conserving should or can be supported also remains unknown when multiple types of resources exist in a complex, heterogeneous network. The resource scheduler may need to maintain different models for different components accordingly.

Q2. Where to implement the resource arbiter? Inter-host networks can control the traffics both on the end hosts and in fabrics. However, many devices in the intra-host networks may lack sufficient programmability. For example, it is currently challenging to implement complex resource allocation functions on a PCIe switch. The next generation of hardware may provide opportunities to address this, which however could be a significant hardware challenge. Another possibility is that there may be a unified software shim layer for future intra-host networks that arbitrates all the intra-host network operations, including various I/O requests.

Q3. How to reduce the overhead of resource management? Modern intra-host networks provide ultra-low latency. Therefore, the overhead of such a resource management system should only introduce negligible overhead. For example, the schedule and arbitration may need to be finished in microsecond level in order to achieve efficient and accurate resource management. Reducing the overhead of resource management needs careful design and much optimization efforts, which remains an interesting open question.

4 RELATED WORK

Understanding intra-host networks. There are many works that focus on understanding intra-host networks. For example, Agarwal et. al. [2] study the impact of IO memory management units and memory bus contention on the host interconnect performance. Zambre et. al. [56] use a PCIe analyzer to conduct a latency breakdown for message transmission and analyze the overhead introduced by different components, including the intra-host network. Neugebauer et. al. [43] propose a theoretical model for PCIe subsystem, and characterize and benchmark different PCIe platforms to evaluate their impacts on the host networking performance. Li et. al. [36] assess the performance of modern GPU interconnects. Recently, Hostping [40] is proposed to diagnose intra-host bottlenecks in RDMA networks. This line of literature indicates the increasing importance of intra-host networks.

Improving intra-host networks performance. Many efforts have been spent to improve the intra-host network performance. Emerging hardware protocols such as CXL [49] and the systems leveraging the new protocols [20] reduce the intra-host communication overhead and provide more flexible communication patterns. Besides, many systems are built on top of commodity hardware to improve the intra-host networks. For example, Lamda [37] focuses on the impact of memory bus congestion on the end-to-end network and proposes a mitigation solution by leveraging the DDIO technology. BytePS [31] schedules the machine learning workload to reduce PCIe contention and improve communication among GPU workers. These works primarily focus on performance optimization rather than manageability improvement for intra-host networks. Additionally, they currently only focus on parts of the intra-host networks and lack a holistic view.

5 CONCLUSION

The performance of inter-host networks has substantially improved in the past decade, making the intra-host network resources bottlenecks for networked systems. In this paper, we argue that an intra-host network should have similar manageability features as those in inter-host networks. This will allow operators to better debug and monitor the intra-host network. In addition, intra-host network resources can be better utilized and shared across applications. We present two key building blocks to achieve this vision.

ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for their helpful feedback. This work is partially supported by grants from NSF (CNS-2238665) and DARPA/SRC JUMP 2.0. This work is also partially supported by gifts from Adobe, Amazon, IBM, Meta, and Intel. Nam Sung Kim has a financial interest in Samsung Electronics and NeuroRealityVision.

REFERENCES

[1] Bhavish Agarwal, Ranjita Bhagwan, Tathagata Das, Siddharth Eswaran, Venkata N Padmanabhan, and Geoffrey M Voelker. Netprints: Diagnosing home network misconfigurations using shared knowledge. In *NSDI*, volume 9, pages 349–364, 2009.

[2] Saksham Agarwal, Rachit Agarwal, Behnam Montazeri, Masoud Moshref, Khaled Elmeleegy, Luigi Rizzo, Marc Asher de Kruijf, Gautam Kumar, Sylvia Ratnasamy, David Culler, and Amin Vahdat. Understanding host interconnect congestion. In *Proceedings of the 21st ACM Workshop on Hot Topics in Networks*, HotNets '22, page 198–204, New York, NY, USA, 2022. Association for Computing Machinery.

[3] Ian F Akyildiz, Ahyoung Lee, Pu Wang, Min Luo, and Wu Chou. A roadmap for traffic engineering in sdn-openflow networks. *Computer Networks*, 71:1–30, 2014.

[4] Mohammad Al-Fares, Sivasankar Radhakrishnan, Barath Raghavan, Nelson Huang, Amin Vahdat, et al. Hedera: dynamic flow scheduling for data center networks. In *Nsdi*, volume 10, pages 89–92. San Jose, USA, 2010.

[5] Mohammad Alian and Nam Sung Kim. Netdimm: Low-latency near-memory network interface architecture. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 699–711, 2019.

[6] Mohammad Alizadeh, Tom Edsall, Sarang Dharmapurikar, Ramanan Vaidyanathan, Kevin Chu, Andy Fingerhut, Vinh The Lam, Francis

Matus, Rong Pan, Navindra Yadav, et al. Conga: Distributed congestion-aware load balancing for datacenters. In *Proceedings of the 2014 ACM conference on SIGCOMM*, pages 503–514, 2014.

[7] Marcelo Amaral, Jorda Polo, David Carrera, Seetharami Seelam, and Malgorzata Steinder. Topology-aware gpu scheduling for learning workloads in cloud environments. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12, 2017.

[8] Sebastian Angel, Hitesh Ballani, Thomas Karagiannis, Greg O'Shea, and Eno Thereska. End-to-End Performance Isolation through Virtual Datacenters. In *OSDI*, 2014.

[9] Behnaz Arzani, Selim Ciraci, Boon Thau Loo, Assaf Schuster, and Geoff Outhred. Taking the blame game out of data centers operations with netpoirot. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 440–453, 2016.

[10] Wei Bai, Shanim Sainul Abdeen, Ankit Agrawal, Krishan Kumar Attre, Paramvir Bahl, Ameya Bhagat, Gowri Bhaskara, Tanya Brokhman, Lei Cao, Ahmad Cheema, et al. Empowering azure storage with {RDMA}. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 49–67, 2023.

[11] Hitesh Ballani, Paolo Costa, Thomas Karagiannis, and Ant Rowstron. Towards predictable datacenter networks. In *Proceedings of the ACM SIGCOMM 2011 Conference*, pages 242–253, 2011.

[12] Jingrong Chen, Yongji Wu, Shihan Lin, Yechen Xu, Xinhao Kong, Thomas Anderson, Matthew Lentz, Xiaowei Yang, and Danyang Zhuo. Remote procedure call as a managed system service. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 141–159, 2023.

[13] Jingrong Chen, Hong Zhang, Wei Zhang, Liang Luo, Jeffrey Chase, Ion Stoica, and Danyang Zhuo. {NetHint}:{White-Box} networking for {Multi-Tenant} data centers. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 1327–1343, 2022.

[14] Mike Chen, Alice X Zheng, Jim Lloyd, Michael I Jordan, and Eric Brewer. Failure diagnosis using decision trees. In *International Conference on Autonomic Computing, 2004. Proceedings.*, pages 36–43. IEEE, 2004.

[15] Haiwei Dong, Ali Munir, Hanine Tout, and Yashar Ganjali. Next-generation data center network enabled by machine learning: Review, challenges, and opportunities. *IEEE Access*, 9:136459–136475, 2021.

[16] Nick G Duffield, Pawan Goyal, Albert Greenberg, Partho Mishra, Kadangode K Ramakrishnan, and Jacobus E van der Merive. A flexible model for resource management in virtual private networks. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 95–108, 1999.

[17] Alireza Farshin, Amir Roozbeh, Gerald Q. Maguire, and Dejan Kostić. Reexamining direct cache access to optimize i/o intensive applications for multi-hundred-gigabit networks. In *Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference*, USENIX ATC'20, USA, 2020. USENIX Association.

[18] Daniel Firestone and Madhan Sivakumar. DPDK on Microsoft Azure. In *DPDK Summit*, 2017.

[19] Yixiao Gao, Qiang Li, Lingbo Tang, Yongqing Xi, Pengcheng Zhang, Wenwen Peng, Bo Li, Yaohui Wu, Shaozong Liu, Lei Yan, Fei Feng, Yan Zhuang, Fan Liu, Pan Liu, Xingkui Liu, Zhongjie Wu, Junping Wu, Zheng Cao, Chen Tian, Jinbo Wu, Jiaji Zhu, Haiyong Wang, Dennis Cai, and Jiesheng Wu. When cloud storage meets RDMA. In *NSDI 21*, 2021.

[20] Donghyun Gouk, Miryeong Kwon, Hanyeoreum Bae, Sangwon Lee, and Myoungsoo Jung. Memory pooling with cxl. *IEEE Micro*, pages 1–11, 2023.

[21] Donghyun Gouk, Sangwon Lee, Miryeong Kwon, and Myoungsoo Jung. Direct access,{High-Performance} memory disaggregation with {DirectCXL}. In *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, pages 287–294, 2022.

[22] Albert Greenberg, Dave Maltz, Guohan Lu, Jiaxin Cao, Ratul Mahajan, and Yibo Zhu. Packet-level telemetry in large datacenter networks. In *SIGCOMM'15*, August 2015.

[23] Chuanxiong Guo. Pingmesh: A large-scale system for data center network latency measurement and analysis. In *SIGCOMM*. ACM, August 2015.

[24] Chuanxiong Guo, Guohan Lu, Helen J. Wang, Shuang Yang, Chao Kong, Peng Sun, Wenfei Wu, and Yongguang Zhang. SecondNet: A Data Center Network Virtualization Architecture with Bandwidth Guarantees. In *CoNEXT*, 2010.

[25] Chuanxiong Guo, Haitao Wu, Zhong Deng, Gaurav Soni, Jianxi Ye, Jitu Padhye, and Marina Lipshteyn. RDMA over Commodity Ethernet at Scale. In *SIGCOMM*, 2016.

[26] Christian Hopps. Analysis of an equal-cost multi-path algorithm. Technical report, 2000.

[27] Intel. Intel data direct i/o technology. https://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/data-direct-i-o-technology-brief.pdf.

[28] Intel. Intel performance counter monitor. https://github.com/intel/pcm.

[29] Intel. Intel resource director technology (rdt). https://intel.github.io/cri-resource-manager/stable/docs/policy/rdt.html.

[30] Vimalkumar Jeyakumar, Mohammad Alizadeh, David Mazières, Balaji Prabhakar, Changhoon Kim, and Albert Greenberg. Eyeq: Practical network performance isolation at the edge. *REM*, 1005(A1):A2, 2013.

[31] Yimin Jiang, Yibo Zhu, Chang Lan, Bairen Yi, Yong Cui, and Chuanxiong Guo. A Unified Architecture for Accelerating Distributed DNN Training in Heterogeneous GPU/CPU Clusters. In *OSDI*, 2020.

[32] Xinhao Kong, Jingrong Chen, Wei Bai, Yechen Xu, Mahmoud Elhaddad, Shachar Raindel, Jitendra Padhye, Alvin R Lebeck, and Danyang Zhuo. Understanding {RDMA} microarchitecture resources for performance isolation. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 31–48, 2023.

[33] Xinhao Kong, Yibo Zhu, Huaping Zhou, Zhuo Jiang, Jianxi Ye, Chuanxiong Guo, and Danyang Zhuo. Collie: Finding Performance Anomalies in RDMA Subsystems. In *NSDI*, 2022.

[34] Praveen Kumar, Nandita Dukkipati, Nathan Lewis, Yi Cui, Yaogong Wang, Chonggang Li, Valas Valancius, Jake Adriaens, Steve Gribble, Nate Foster, et al. Picnic: predictable virtualized nic. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 351–366. 2019.

[35] Nikita Lazarev, Shaojie Xiang, Neil Adit, Zhiru Zhang, and Christina Delimitrou. Dagger: efficient and fast rpcs in cloud microservices with near-memory reconfigurable nics. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 36–51, 2021.

[36] Ang Li, Shuaiwen Leon Song, Jieyang Chen, Jiajia Li, Xu Liu, Nathan R. Tallent, and Kevin J. Barker. Evaluating modern GPU interconnect: PCIe, NVLink, NV-SLI, NVSwitch and GPUDirect. *IEEE Transactions on Parallel and Distributed Systems*, 31(1):94–110, jan 2020.

[37] Qiang Li, Qiao Xiang, Derui Liu, Yuxin Wang, Haonan Qiu, Gexiao Tian, Xiaoliang Wang, Lulu Chen, Ridi Wen, Jianbo Dong, Yuanyuan Gong, Yixiao Gao, Haohao Song, Zhiwu Wu, Shaozong Liu, Zicheng Luo, Yuchao Shao, Yaohui Wu, Chao Han, Chen Tian, Zhongjie Wu, Zheng Cao, Jinbo Wu, Jiwu Shu, and Jiesheng Wu. Lamda: The last mile of the datacenter network does matter, 2022.

[38] Shang Li, Dhiraj Reddy, and Bruce Jacob. A performance & power comparison of modern high-speed dram architectures. In *Proceedings of the International Symposium on Memory Systems*, pages 341–353, 2018.

[39] Yuliang Li, Rui Miao, Changhoon Kim, and Minlan Yu. Flowradar: A better netflow for data centers. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 311–324, 2016.

[40] Kefei Liu, Zhuo Jiang, Jiao Zhang, Haoran Wei, Xiaolong Zhong, Lizhuang Tan, Tian Pan, and Tao Huang. Hostping: Diagnosing intra-host network bottlenecks in {RDMA} servers. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 15–29, 2023.

[41] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*, 38(2):69–74, 2008.

[42] Mellanox. Mellanox neo-host network adapter management software. https://support.mellanox.com/s/productdetails/a2v50000000N2OlAAK/mellanox-neohost.

[43] Rolf Neugebauer, Gianni Antichi, José Fernando Zazo, Yury Audzevich, Sergio López-Buedo, and Andrew W. Moore. Understanding pcie performance for end host networking. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '18, page 327–341, New York, NY, USA, 2018. Association for Computing Machinery.

[44] Nvidia. Nvidia connectx ethernet network adapters. https://www.nvidia.com/en-us/networking/ethernet-adapters/.

[45] Nvidia. Nvidia dgx a100. https://www.nvidia.com/en-us/data-center/dgx-a100/.

[46] Nvidia. Nvidia sn4000 series ethernet switches. https://nvdam.widen.net/s/6269c25wv8/nv-spectrum-sn4000-product-brief.

[47] Lucian Popa, Gautam Kumar, Mosharaf Chowdhury, Arvind Krishnamurthy, Sylvia Ratnasamy, and Ion Stoica. FairCloud: Sharing the Network in Cloud Computing. In *SIGCOMM*, 2012.

[48] Lucian Popa, Praveen Yalagandula, Sujata Banerjee, Jeffrey C Mogul, Yoshio Turner, and Jose Renato Santos. Elasticswitch: Practical Work-Conserving Bandwidth Guarantees for Cloud Computing. In *SIGCOMM*, 2013.

[49] Debendra Das Sharma. Compute express link®: An open industry-standard interconnect enabling heterogeneous data-centric computing. In *2022 IEEE Symposium on High-Performance Interconnects (HOTI)*, pages 5–12. IEEE, 2022.

[50] Alan Shieh, Srikanth Kandula, Albert Greenberg, Changhoon Kim, and Bikas Saha. Sharing the Data Center Network. In *NSDI*, 2011.

[51] Peng Sun, Ratul Mahajan, Jennifer Rexford, Lihua Yuan, Ming Zhang, and Ahsan Arefin. A network-state management service. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, page 563–574, New York, NY, USA, 2014. Association for Computing Machinery.

[52] Cheng Tan, Ze Jin, Chuanxiong Guo, Tianrong Zhang, Haitao Wu, Karl Deng, Dongming Bi, and Dong Xiang. Netbouncer: Active device and link failure localization in data center networks. In *NSDI*, pages 599–614, 2019.

[53] Markus Velten, Robert Schöne, Thomas Ilsche, and Daniel Hackenberg. Memory performance of amd epyc rome and intel cascade lake sp server processors. In *Proceedings of the 2022 ACM/SPEC on International Conference on Performance Engineering*, pages 165–175, 2022.

[54] Tong Yang, Jie Jiang, Peng Liu, Qun Huang, Junzhi Gong, Yang Zhou, Rui Miao, Xiaoming Li, and Steve Uhlig. Elastic sketch: Adaptive and fast network-wide measurements. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 561–575, 2018.

[55] Da Yu, Yibo Zhu, Behnaz Arzani, Rodrigo Fonseca, Tianrong Zhang, Karl Deng, and Lihua Yuan. dshark: a general, easy to program and scalable framework for analyzing in-network packet traces. In *Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation*, pages 207–220, 2019.

[56] Rohit Zambre, Megan Grodowitz, Aparna Chandramowlishwaran, and Pavel Shamis. Breaking band: A breakdown of high-performance communication. In *Proceedings of the 48th International Conference on Parallel Processing*, ICPP '19, New York, NY, USA, 2019. Association for Computing Machinery.

[57] Junlan Zhou, Malveeka Tewari, Min Zhu, Abdul Kabbani, Leon Poutievski, Arjun Singh, and Amin Vahdat. Wcmp: Weighted cost multipathing for improved fairness in data centers. In *Proceedings of the Ninth European Conference on Computer Systems*, pages 1–14, 2014.