[BenchCouncil Transactions on Benchmarks, Standards and Evaluations 2 (2022) 100074](https://doi.org/10.1016/j.tbench.2022.100074)



Contents lists available at [ScienceDirect](https://www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/)

BenchCouncil Transactions on Benchmarks,

Standards and Evaluations

journal homepage: [www.keaipublishing.com/en/journals/benchcouncil-transactions-on-](http://www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/) [benchmarks-standards-and-evaluations/](http://www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/)

Research Article

[](http://crossmark.crossref.org/dialog/?doi=10.1016/j.tbench.2022.100074&domain=pdf)Understanding hot interconnects with an extensive benchmark survey

Yuke Li [a](#_bookmark0), Hao Qi [a](#_bookmark0), Gang Lu [b](#_bookmark1), Feng Jin [b](#_bookmark1), Yanfei Guo [c](#_bookmark2), Xiaoyi Lu [a](#_bookmark0),[∗](#_bookmark4)

a *University of California, Merced, 5200 North Lake Rd., Merced, 95343, CA, USA*

b *Tencent, No. 33, Haitian Second Road, Shenzhen, 518054, Guangdong, China*

c *Argonne National Laboratory, 9700 S. Cass Avenue, Lemont, 60439, IL, USA*

A R T I C L E I N F O A B S T R A C T

*Keywords:* Benchmarks Interconnects RDMA

Understanding the designs and performance characterizations of hot interconnects on modern data center and high-performance computing (HPC) clusters is a fruitful research topic in recent years. The rapid and continuous growth of high-bandwidth and low-latency communication requirements for various types of data center and HPC applications (such as big data, deep learning, and microservices) has been pushing the envelope of advanced interconnect designs. We believe this is high time to investigate the performance characterizations of representative hot interconnects with different benchmarks. Hence, this paper presents an extensive survey of state-of-the-art hot interconnects on data center and HPC clusters and the associated representative benchmarks to help the community to better understand modern interconnects. In addition, we characterize these interconnects by the related benchmarks under different application scenarios. We provide our perspectives on benchmarking data center interconnects based on our survey, experiments, and results.

# Introduction

The scales of data center and high-performance computing (HPC) clusters grow rapidly with the increasingly large volume of data and the high demand for distributed computing capabilities [[1](#_bookmark40)]. This trend has led to various designs of modern data center interconnects and made their performance characterizations a rewarding research topic. To continuously improve the performance and scalability of data move- ment or communication across a large number of nodes in modern data center or HPC clusters, different types of advanced interconnects have been designed to meet the requirements of high-bandwidth and low- latency communications in popular data center applications, such as deep learning, big data, microservices, etc.

To upgrade the conventional Ethernet (∼10 Gbps) network and

accelerate the efficiency of data center applications, hardware ven-

dors have demonstrated multiple types of advanced data center in- terconnects. For example, NVIDIA (Mellanox) has produced 200 Gbps InfiniBand (IB) [[2](#_bookmark41)] with well-optimized Remote Direct Memory Ac- cess (RDMA) subsystems to speedup the inter-node communication in applications. Cray has the Slingshot interconnect [[3](#_bookmark42)] and the Aries interconnect [[4](#_bookmark43)] as high-speed interconnects for modern HPC systems. RIKEN (Japanese Institute of Physical and Chemical Research) and Fujitsu developed the Tofu interconnect [[5](#_bookmark44)] family to be equipped on their designed supercomputers. Meanwhile, the Ethernet network speed has improved from 10 Gbps to 100 Gbps [[6](#_bookmark45)] and even above [[7](#_bookmark46)] during the decades of development.

With the trend of hardware evolution and the new interconnects being created, there are several issues that the application developers need to pay attention to. With the hardware upgrading, the developers need to re-evaluate the performance of different generations of hard- ware to design the proper systems software based on the improved data transfer rates. Also, many new interconnects are emerging with the development of novel hardware features. These features may po- tentially impact application performance and need to be systematically investigated.

On the other hand, different types of data center applications repre- sent various performance characterizations, like HPC workloads, deep learning training and inferences, big data analytics, and cloud-based microservice. The impacts of new interconnects on these different workloads should be evaluated separately and carefully. Therefore, we believe this is high time to investigate the performance characteri- zations of modern data centers and HPC interconnects via standard benchmarking experiments under different application scenarios. This observation motivates us to extensively survey hot interconnects on modern data centers and HPC clusters and the associated representative benchmarks to help the community better understand these advanced interconnects.

There exist some surveys to summarize benchmarking experiences with different workloads. For example, Han et al. [[8](#_bookmark47)] surveyed ten big data benchmarks to discuss benchmarking challenges. Zhang et al. [[9](#_bookmark48)]

∗ Corresponding author.

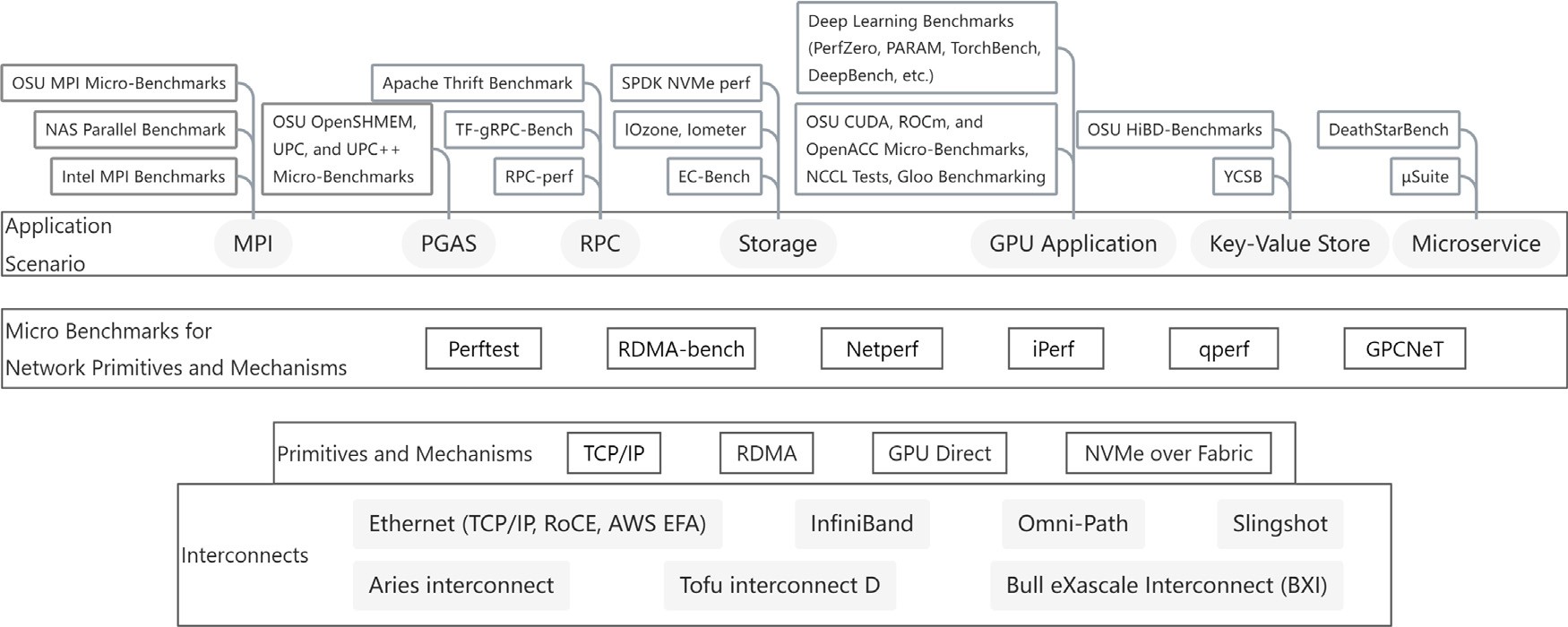
*E-mail addresses:* [yli304@ucmerced.edu](mailto:yli304@ucmerced.edu) (Y. Li), [hqi6@ucmerced.edu](mailto:hqi6@ucmerced.edu) (H. Qi), [gateslu@tencent.com](mailto:gateslu@tencent.com) (G. Lu), [ronyjin@tencent.com](mailto:ronyjin@tencent.com) (F. Jin), [yguo@anl.gov](mailto:yguo@anl.gov) (Y. Guo), [xiaoyi.lu@ucmerced.edu](mailto:xiaoyi.lu@ucmerced.edu) (X. Lu).

<https://doi.org/10.1016/j.tbench.2022.100074>

Received 19 September 2022; Received in revised form 21 October 2022; Accepted 21 October 2022

Available online 28 October 2022

2772-4859/© 2022 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).



**/ig. 1.** An overview of data center interconnects and benchmarks.

investigated fourteen deep learning benchmarks. Zhou et al. [[10](#_bookmark49)] dis- cussed seven microservice benchmarks. Gao et al. [[11](#_bookmark50)] compared fif- teen big data and AI (Artificial Intelligence) benchmarks. However, we did not find such a survey that can extensively cover a broad range of the latest advanced interconnects in modern data centers and the associated representative benchmarks for different application scenar- ios. Therefore, this paper addresses the need to survey different hot interconnects deployed in modern data centers and the corresponding benchmarks to expose their performance characteristics.

[Fig.](#_bookmark5) [1](#_bookmark5) shows an overview of this paper’s surveying scope, including various kinds of hot interconnects and the associated representative and popular benchmarks in the community. The following sections will introduce each component in [Fig.](#_bookmark5) [1](#_bookmark5) with a bottom-up approach. In Section [2](#_bookmark6), we survey the features and characteristics of hot intercon- nects in modern data centers and HPC clusters. In Section [3](#_bookmark9), we survey well-used micro benchmarks for evaluating these hot interconnects with their network primitives and mechanisms. Section [4](#_bookmark10) will sur- vey application-level benchmarks with diverse evaluation granularity, as shown in [Fig.](#_bookmark5) [1](#_bookmark5). In Section [5](#_bookmark22), we choose several representative benchmarks, which include Netperf [[12](#_bookmark51)], Perftest [[13](#_bookmark52)], and OSU Micro- Benchmarks (OMB) [[14](#_bookmark53)] for MPI (Message Passing Interface) [[15](#_bookmark54)] and PGAS (Partitioned Global Address Space) [[16](#_bookmark55)] applications, and inter- connects, which include IB, Omni-Path [[17](#_bookmark56)], and Ethernet to run exper- iments. We present the results to show performance characterizations of these hot interconnects as examples or reference numbers. Section [6](#_bookmark36) will discuss some of our observations and perspectives on benchmark- ing data center interconnects based on our survey, experiments, and results. Section [7](#_bookmark38) discusses more related studies and Section [8](#_bookmark39) concludes the paper.

The main contributions of this paper are as follows:

* We perform an extensive survey on advanced hot interconnects in current-generation and emerging data centers and HPC clusters.
* We also comprehensively survey the associated representative benchmarks from both micro benchmarking and application-level benchmarking perspectives.
* We perform a set of benchmarking experiments on real inter- connects hardware with well-used benchmarks and discuss their performance characterizations.
* We share our observations on improvable aspects of existing benchmarks, such as performance stability, reference number, experimental instructions, etc., to help the community to design better ones.

# Overview of modern interconnects

As an indispensable part of HPC and data center systems, inter- connects play an essential role in achieving higher scalability and

performance for modern clusters. In recent years, the community has witnessed the development of conventional interconnects like Ether- net and InfiniBand, and the birth of proprietary interconnects such as Fugaku Tofu [[5](#_bookmark44)] and BXI (Bull eXascale Interconnect) [[18](#_bookmark57)]. This section will briefly overview some representative state-of-the-art mod- ern interconnects, and their features [[1](#_bookmark40)]. After we go through these interconnects one by one, [Table](#_bookmark7) [1](#_bookmark7) shows a brief comparison of these hot interconnects.

* 1. *Ethernet*

Ethernet is one of the most traditionally utilized interconnects for HPC and data center clusters. At the early stage, 1 Gb/s Ethernet (1-GigE) was widely used. However, with the advancement of CPU per- formance and I/O speed, the 1-GigE has become the bottleneck. With the demand for higher bandwidth and data transfer rate, Ethernet with 10-GigE, 25-GigE, 50-GigE, and even 100-GigE, has been developed. As of June 2022, 25-GigE is the most widely used interconnect in the Top500 list, and the Ethernet interconnect family is the majority in the list, taking up nearly 50% [[19](#_bookmark58)].

Taken the advantages of RDMA, RDMA over Converged Ethernet (RoCE) [[20](#_bookmark59)] is developed, which is a network protocol that allows RDMA to operate over Ethernet networks. RoCE is designed to support RDMA over Ethernet on layer 2 networks, and its extended version RoCE v2 enables transportation on layer 3 networks. Traditionally, Eth- ernet has left the congestion control to the TCP (Transmission Control Protocol) layer. With the development, the first algorithm proposed for the Ethernet network is pause frame [[21](#_bookmark60)] in 1996. Congestion control on RoCE uses an extension to the TCP/IP protocol called ECN (Explicit Congestion Notification) [[22](#_bookmark61)]. Other techniques, such as the QCN (Quantized Congestion Notification) [[23](#_bookmark62)], were developed afterward. Both traditional Ethernet and RoCE are available for vari- ous interconnect topologies. In 2019, Amazon announced EFA (Elastic Fabric Adapter) [[24](#_bookmark63)] for its EC2 (Elastic Compute Cloud) instance. The libfabric [[25](#_bookmark64)] interface on EFA provides up to 100 Gbps speed and reduces overhead with techniques like operating system bypass.

* 1. *InfiniBand*

Provided by NVIDIA, InfiniBand (IB) is an industry-standard switch fabric and the second most popular interconnect family in the Top500 list [[19](#_bookmark58)]. As of June 2022, 32.4% of the Top500 clusters are intercon- nected by IB, especially for Top10 clusters such as Summit [[26](#_bookmark65)] and Sierra [[27](#_bookmark66)]. Besides the higher bandwidth (up to 400 Gbps) and lower

latency (<1 μs), IB also supports advanced features like RDMA, which

allows the software to read/write data from/to the memory in remote

Comparison of interconnects.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Ethernet |  |  | InfiniBand | Omni-Path | Slingshot | Aries | TofuD | BXI |
|  | 25–100 Gbps | 200–400 Gbps | RoCE |  |  |  |  |  |  |
| Manufacturer | Many | Many | Many | NVIDIA/Mellanox | Intel/Cornelis | Cray | Cray | Fujitsu | Atos |
| Commodity | Public | Public | Public | Public | Public | Proprietary | Proprietary | Proprietary | Proprietary |
| Unidirectional Bandwidth (Gbps) | 25–100 [[37](#_bookmark76)] | 200–400 [[37](#_bookmark76)] | 100 [[38](#_bookmark77)] | 400 [[39](#_bookmark78)] | 100 [[37](#_bookmark76)] | 200 [[37](#_bookmark76)] | 40 [[4](#_bookmark43)] | 56 [[40](#_bookmark79)] | 100 [[37](#_bookmark76)] |
| End to End Latency (μs) | 10–30 | N/A | ∼1 [[37](#_bookmark76)] | <1 [[37](#_bookmark76)] | <1 [[37](#_bookmark76)] | <2 [[37](#_bookmark76)] | ∼1 [[4](#_bookmark43)] | 0.5–1 [[5](#_bookmark44)] | <1 [[37](#_bookmark76)] |
| Congestion Control | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Topology | Various | Various | Various | Fat-tree, Dragonfly+ | Fat-tree | Dragonfly | Dragonfly | Torus | Various |
| RDMA | No | No | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Year | 2014 | 2017 | 2010 | 1999 | 2015 | 2019 | 2012 | 2018 | 2015 |

nodes without any CPU involvement from the remote side. IB pro- vides reliable or unreliable, and connected or datagram data transport types [[28](#_bookmark67)]. The reliable transport can guarantee the packet delivery in order but it spends extra time to wait for the acknowledgment from the receiving side. The unreliable transport cannot ensure the packet is received, but it does not need extra time for waiting the acknowl- edgment. The queue pairs for connected transports are connected in the one-to-one mapping, while the queue pairs for datagram transports are connected in the one-to-any mapping. The connected transport is more suitable for applications with a small number of connections. The datagram transport usually performs better in large-scale appli- cations because fewer connection contexts need to be maintained in memory [[29](#_bookmark68)].

Specifically, RDMA-capable networks (like InfiniBand) typically support four types of transport modes: Reliable Connection (RC), Re- liable Datagram (RD), Unreliable Connection (UC), and Unreliable Datagram (UD). SEND and RECV operations are supported by all modes, while the RDMA WRITE operation is unsupported by UD, and the RDMA READ operation is unsupported by UD and UC. The most commonly used network topology for IB is fat-tree [[30](#_bookmark69)], but it also sup- ports other topologies like dragonfly+ [[31](#_bookmark70)]. The IB standard includes a congestion control mechanism to detect and resolve congestion by using two relay messages: FECN (Forward Explicit Congestion Notifica- tion) [[32](#_bookmark71)] and BECN (Backward Explicit Congestion Notification) [[33](#_bookmark72)]. When applying IB to GPU, CUDA 5.0 first introduced GDR (GPUDirect RDMA) [[34](#_bookmark73)]. GDR allows IB adapters to directly access the GPU memory while also bypassing the host. GDR can significantly increase data communication performance among GPUs, which further benefits the increasing number of redesigned classical HPC and machine/deep learning applications.

* 1. *Omni-Path*

Omni-Path was first released by Intel in 2015 as a part of Intel’s Scalable System Framework with the purpose of increasing HPC work- load scalability and aiming for low communication latency, low power consumption, and high throughput [[17](#_bookmark56)]. Omni-Path mainly includes the network card, switch, and network manager components. It is built on Intel technology with multiple features, such as traffic flow optimization and packet integrity protection. It is mainly designed to support fat-tree topology, and its CCA (Congestion Control Archi- tecture) has been updated continuously since its first release. The first generation Omni-Path delivers 100 Gbps bandwidth per port and is integrated into some CPU architectures like Skylake and Knights Landing (KNL) [[35](#_bookmark74)]. Although Intel stopped the development of the second-generation Omni-Path in 2019, it still takes 7.8% of Top500 clusters as of June 2022 [[19](#_bookmark58)]. In late 2020, Intel announced its spin-off to Cornelis Networks [[36](#_bookmark75)] to continue the business as a successor to the Omni-Path product.

* 1. *Slingshot*

In 2019, Cray launched its new generation of HPC interconnect technology called Slingshot [[3](#_bookmark42)]. Slingshot uses protocols on standard

Ethernet while also being compatible with proprietary HPC networks when needed. It offers key features like adaptive routing, quality of service guarantee, and advanced congestion control fully implemented in hardware. The slingshot switch is equipped with 64 ports, and each port is running at 200 Gbps. Slingshot also supports multiple interconnect topologies such as fat-tree and dragonfly [[41](#_bookmark80)]. As Cray’s eighth major high-performance interconnection network technology, Slingshot is deployed on a variety of clusters like pre-exascale cluster Perlmutter [[42](#_bookmark81)] and exascale cluster Frontier [[43](#_bookmark82)], which is currently the top 1 supercomputer in the world. Slingshot is also planned to be deployed on upcoming exascale clusters like Aurora [[44](#_bookmark83)] and EI Capitan [[45](#_bookmark84)]. Slingshot is taking up 4.8% of the clusters in Top500 list as of June 2022 [[19](#_bookmark58)].

* 1. *Aries interconnect*

As Cray’s third-generation interconnect architecture, Aries was in- troduced as part of the Cray XC system with the dragonfly topology, and it has been widely used in the HPC field [[4](#_bookmark43)]. A single Aries device with four NICs (Network Interface Card) and a 48-port tiled router can provide a network connection for all four nodes on a Cray XC blade. The NIC and switch in Aries are closely coupled in the dragonfly network to provide cost-effective and scalable global bandwidth. The system is configurable according to users’ global bandwidth require- ment, and its optical connection number can be adjusted according to the cost constraint. It also provides technologies such as adaptive rout- ing, communication mechanisms, and synchronization mechanisms. Aries adopts the dragonfly topology and achieves congestion control by implementing Valiant’s routing algorithm [[46](#_bookmark85)]. As of June 2022, 5% of the Top500 clusters use Aries, including Piz Daint [[47](#_bookmark86)] and Cori [[48](#_bookmark87)].

* 1. *Tofu interconnect D*

As one of the representatives of proprietary interconnects, Tofu [[49](#_bookmark88)] is an interconnect family developed by RIKEN and Fujitsu that is used for the K computer [[50](#_bookmark89)]. In 2018, TofuD (Tofu Interconnect D) was in- troduced as a new member of the Tofu family. Its main features are just as indicated by the name. The Tofu represents ‘‘torus fusion’’ and the letter D stands for high ‘‘density’’ node and ‘‘dynamic’’ packet slicing for ‘‘dual-rail’’ transfer [[5](#_bookmark44)]. TofuD is a proprietary torus-based [[5](#_bookmark44)] six- dimensional network, and it mainly supports congestion control with the family’s virtual channel scheduling algorithm. Compared to previ- ous Tofu and Tofu2 [[51](#_bookmark90)], TofuD has a much higher communication resource density, such as 48 cores per node. It also introduced dynamic packet slicing for the dual-rail transfer technique to solve the latency and fault tolerance issue in Tofu2. TofuD is adopted by the Fugaku [[52](#_bookmark91)], which was the top 1 cluster in the Top500 list at the time built in 2020 and ranked 2nd as of June 2022 [[19](#_bookmark58)].

* 1. *Bull eXascale interconnect (BXI)*

In 2015, Atos designed BXI as a new interconnect for HPC [[18](#_bookmark57)]. BXI is based on the scalable and reliable Portals4 [[53](#_bookmark92)] network pro- gramming interface and decouples computation and communication

**Table 2**

The summary of micro-benchmarks.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Perftest [[13](#_bookmark52)] | RDMA-bench [[56](#_bookmark95)] | NetPerf [[12](#_bookmark51)] | iPerf [[57](#_bookmark96)–[60](#_bookmark99)] | qperf [[61](#_bookmark100)] | GPCNeT [[62](#_bookmark101)] |
| Link layer | IB, Eth (RoCE) | IB, Eth (RoCE) | Eth | Eth | IB, Eth (RoCE) | IB, Eth (RoCE), etc. |
| Programming | RDMA | RDMA | Socket | Socket | RDMA/Socket | MPI |
| Models  Transport | RC/UC/UD | RC/UC/UD | TCP, UDP, SCTP | TCP, UDP, SCTP | TCP, UDP, SCTP, | Any protocol that |
| Protocols | DCT, SRD |  |  |  | SDP  RC/UC/UD, RDS | can be used by MPI |
| Main metrics | Throughput, | Throughput, | Throughput, | Throughput, | Throughput, | Throughput, |
|  | Average latency, Tail latency | Average latency, Tail latency,  WQE cache misses | Average latency | Average latency, Tail latency | Average latency | Average latency, Tail latency,  Congestion Impact |
| Language | C | C | C | C | C | C |
| Thread Model | Single-thread | Single-thread | Single-thread | Multi-thread | Single-thread | Multi-process (MPI) |

Communication pattern

P2P P2P P2P P2P; multicast P2P P2P, collective

communication

Real scenario N Y (w/ real

N N N Y

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | applications) |  | | | |
| Real workload | N | N | N | N | N | Y |
| or trace |  |  |  |  |  |  |
| Parameters of | N | N | N | Y | N | N |
| protocol internals |  |  |  |  |  |  |
| Year of last update | 2022 | 2018 | 2021 | 2022 | 2018 | 2021 |

by hardware offloading. It consists of two ASIC (Application-Specific Integrated Circuit)-based components: BXI NIC and BXI switch. The BXI NIC provides functions like OS bypass, communication offload, and reliability. Each BXI switch is equipped with forty-eight 100 Gbps ports and provides power saving and network performance monitoring functions. BXI supports multiple network topologies such as fat-tree, butterfly [[54](#_bookmark93)], and torus. BXI implements efficient fine-grain adaptive routing on each port basis to minimize the possibility of congestion. It also provides reliability and stability guarantees by some optimizations like deadlock-avoidance and load-balancing mechanisms. As of June 2022, Tera-1000-2 adopts BXI 1.2 and it is ranked 45th in the Top500 list [[19](#_bookmark58)].

* 1. *Summary*

We survey the above hot interconnects because of their popularity for clusters in the Top500 list. [Table](#_bookmark7) [1](#_bookmark7) summarizes a brief comparison of these hot interconnects. They show a huge diversity in aspects, such as bandwidth, latency, congestion control mechanism, and net- work topology, which motivates us to investigate their performance characteristics. Due to the lack of access to proprietary interconnects, we mainly focus on evaluating Ethernet, RoCE, IB, and Omni-Path in this paper. InfiniBand is an essential interconnect for native RDMA designs. 10/25 Gbps Ethernet networks are the majority (27.2%) of interconnects used in data center and HPC clusters. RoCE and TCP/IP can be deployed on 10/25 Gbps Ethernet [[55](#_bookmark94)]. We evaluate these different interconnects and show the results in Section [5](#_bookmark22).

# Survey of micro-benchmarks

The community has designed many benchmarks to evaluate various types of interconnects. In this section, we survey six micro-benchmarks designed to measure low-level performance metrics such as latency and bandwidth. We introduce their features and discuss their pros and cons. As shown in [Table](#_bookmark8) [2](#_bookmark8), six publicly-available micro-benchmarks are included for comparison. The rest of this section discusses these micro-benchmarks one by one.

* 1. *Perftest*

Perftest [[13](#_bookmark52)] was developed by Mellanox and has been well main- tained since 2005. The RDMA community widely uses it for latency and bandwidth performance evaluation on InfiniBand and RoCE networks. The included micro benchmarks adopt a single-thread and ping-pong

communication pattern to evaluate the throughput and latency of basic RDMA operations. We can also use it to compare different transports by specifying the transport as RC, UC, UD, Raw Ethernet, and even Mellanox DCT (Dynamic Connected Transport) [[63](#_bookmark102)] and AWS SRD (Scalable Reliable Datagram) [[64](#_bookmark103)] transports that are not specified in the standard IB specification [[65](#_bookmark104)]. Besides the basic operations and transports, Perftest also supports the GPUDirect feature for direct inter- GPU communication through GPUDirect RDMA and the AESXTS [[66](#_bookmark105)] feature for data encryption and decryption scenarios using RDMA. Perftest is designed without emulating any real application traffic or traffic probability distribution. It does not allow users to choose the traffic pattern but only with a parameter to specify the message size in each test. These tests are mainly helpful for hardware or software tuning as well as for functional testing.

* 1. *RDMA-bench*

RDMA-bench [[56](#_bookmark95)] was developed by Carnegie Mellon University in 2016. Unlike Perftest, RDMA-bench is a new benchmark suite used to understand the RDMA performance in a few scenarios extracted from real applications. With the guidelines obtained from running RDMA-bench, the authors of RDMA-bench succeeded in developing a networked sequencer and a key–value store far superior to others [[67](#_bookmark106)]. The benchmarks in RDMA-bench can be classified into several categories: (1) application benchmarks which include HERD [[68](#_bookmark107)] and MICA [[69](#_bookmark108)] as RDMA-based key–value store systems, and DrTM-KV [[70](#_bookmark109)] as an RDMA-based in-memory transaction processing system; (2) micro- benchmarks which measure the throughput of outbound and inbound RDMA operations; (3) micro-benchmarks that emulate an echo server, in which users can choose different RDMA operations for the re- quests and responses; (4) micro-benchmarks which emulate an RPC (Remote Procedure Calls) based sequencer server using different RDMA transports and operations; (5) micro-benchmarks which emulate a com- plex communication scheme with configurable thread-QP ratios to the scalability evaluation; (6) micro-benchmarks which help understand low-level factors that affect RDMA performance, such as WQE cache

misses of outbound READs and WRITEs, etc.

* 1. *Netperf*

Netperf [[12](#_bookmark51)] was developed by Hewlett-Packard in 2005. It is widely used to measure the performance of BSD Sockets [[71](#_bookmark110)] for TCP, UDP, or SCTP (Stream Control Transmission Protocol) [[72](#_bookmark111)] using IPv4 and IPv6, Unix domain sockets [[73](#_bookmark112)], and DLPI (Data Link Provider

Interface) [[74](#_bookmark113)]. Netperf adopts a simple client–server model without multi-threading support. The main parameters include the socket buffer size, the message size, the TCP\_NODELAY option, and the test mode. There are two test modes supported in Netperf: (1) the STREAM mode, which transfers bulk data through a TCP or UDP socket; (2) the RR (Request/Response) mode, which emulates iterative requester–response transactions between the client and server. The data transmitted is synthetic. Neither different probability distributions nor real-world data trace is supported. Hewlett-Packard made a plan of version 4.x of Netperf, which aimed to support synchronized and multi-threaded benchmarking.

* 1. *iPerf*

iPerf [[57](#_bookmark96),[58](#_bookmark97)] is used to evaluate the performance of TCP, UDP, and SCTP traffic with IPv4 and IPv6. It provides abundant features [[57](#_bookmark96),[58](#_bookmark97)]:

(1) iPerf adopts a multi-threaded design that can scale with the number of CPUs within a system; (2) iPerf supports tuning of various parameters that are rarely supported in Netperf, such as timing, buffers, and most importantly, the internal parameters of the protocols; (3) iPerf supports multicast tests and bidirectional tests; (4) iPerf can run on many platforms which include Linux and Windows; (5) users can get various forms of outputs in iPerf; (6) iPerf provides the libiperf library, which is an straightforward way to use and customized the functionality of iPerf.

iPerf has evolved into two incompatible active branches. One branch is iPerf2 [[57](#_bookmark96)] which is the newer version of the original iPerf. The other branch is iPerf3 [[58](#_bookmark97)] which is a redesign of the original iPerf and was now principally developed by ESnet and Lawrence Berkeley National Laboratory. Either of them contains several options and functions that are not present in the other. Generally, for TCP and UDP in Ethernet, iPerf2 and iPerf3 are about the same if running with the default configuration. However, users should check the detailed comparison in [[59](#_bookmark98),[60](#_bookmark99)] to avoid misuse.

* 1. *qperf*

qperf [[61](#_bookmark100)] was initially developed by QLogic in 2007 and then maintained by the Linux community. qperf can measure the bandwidth and latency between two hosts using TCP, UDP, SCTP, RDMA, SDP (Sockets Direct Protocol), and RDS (Reliable Datagram Sockets). It adopts a single-threaded client–server model similar to Netperf. For RDMA, we can test the bandwidth and latency of RC, UC, and UD transports. All the operations can be measured for each transport in the tests. Compared to Perftest, qperf supports fewer transports and features from the perspective of evaluating RDMA performance. For non-RDMA protocols, the option of qperf can only change the message size. Evaluations of the internal features of the protocols cannot be done by using qperf. Even though qperf only reports average latency and fails to perform precise tail latency measurements, it is still popular as it is a handy and tool. The release of qperf is stable and the light-weight update was four years ago.

* 1. *GPCNeT*

The Global Performance and Congestion Network Test (GPCNeT) [[62](#_bookmark101),[75](#_bookmark114)] was developed by Cray in 2019 to evaluate the network per- formance of MPI-based systems with the MPI-3.0 specification [[76](#_bookmark115)]. GPCNeT is compromised of two benchmarks: *network\_test* and *net- work\_load\_test*.

*network\_test* characterize the latency and bandwidth of an MPI application when it runs without network congestion. It builds the natural ring and random ring pattern such that all communication occurs over the network rather than within local groups. The commu- nication patterns include two-sided peer-to-peer (8 bytes latency and 128K bytes bandwidth, natural and random rings), one-sided remote

memory access (8 bytes latency and 128K bytes bandwidth, random ring), allreduce (8 bytes latency, random ring), and alltoall (128 bytes bandwidth, random ring).

*network\_load\_test* measure the performance of an MPI application with network congestion. This simulates the scenario when running on multi-tenant HPC networks. Each congestor has a unique random ring, and the communication patterns include Point-to-point Incast, All-to-all, One-sided RMA Incast, and One-sided RMA Broadcast. Two measurements execute in the random ring infrastructure: Point-to-point Latency measurement by sending and receiving 8 bytes messages from and to two sides, Point-to-point Bandwidth with Synchronization by sending and receiving eight 128K bytes messages from two sides.

The default settings are intended to be utilized in general production scenarios. It reports the mean and 99th percentile latencies as well as the bandwidth per rank. With congestors, it also reports the Congestion Impact metric, which is defined as the ratio of congested latency or bandwidth divided by the uncongested latency or bandwidth. The Con- gestion Impact metric is an indicator to study the impact of congestion across systems with different networks.

* 1. *Summary*

The above micro-benchmarks are surveyed because of their pop- ularity in the community. In [Table](#_bookmark8) [2](#_bookmark8), we show a summary of these micro-benchmarks. Among the six micro-benchmarks, we will test the interconnects with Perftest and NetPerf in this survey. They are both widely used and well maintained since their first release. Besides, Perftest is provided by Mellanox, the most popular manufacturer of In- finiBand. Hence, we believe Perftest and NetPerf can represent defacto standard benchmarks for RDMA-based and socket-based programming models on various interconnects, respectively. We show the related results in Section [5](#_bookmark22).

# Survey of application-level benchmarks

There are diverse types of workloads running across machines in a data center, from parallel computing to microservice, from GPU applications for deep learning workloads to Key–Value Store for big data workloads. The same issue these workloads share is that they all need efficient data communication through the interconnects. As mentioned above, different interconnects may show different charac- terizations on the same application. Therefore, researchers need to use benchmarks to characterize the application that runs on a specific interconnect. This section surveyed application-level benchmarks with diverse evaluation granularity for different application scenarios in data centers that involve cross-node communication via interconnects. To save space, we put detailed descriptions of these benchmarks in tables.

* 1. *MPI benchmarks*

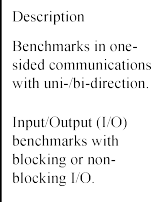
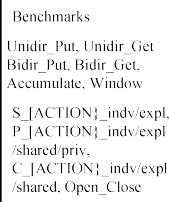
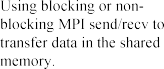
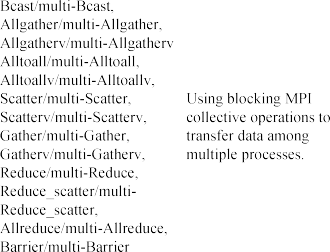
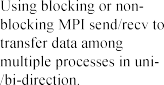
MPI [[15](#_bookmark54)] is a message-passing standard and widely used in HPC where many processes or cores are organized to run parallel program simultaneously for acceleration. Using a benchmark to characterize MPI libraries on different interconnects can help developers understand the characteristics of interconnects and design applications in efficient ways.

We surveyed three popular MPI benchmarks. The OSU MPI Micro- Benchmarks [[14](#_bookmark53)] provided by Ohio State University (OSU) consist of point-to-point MPI operations, blocking/non-blocking collective MPI operations, and one-sided MPI operations. [Table](#_bookmark11) [3](#_bookmark11) shows the descrip- tion details. The NAS Parallel Benchmarks (NPB) [[77](#_bookmark116)], provided by NASA, are derived from CFD (computational fluid dynamics) applica- tions and are designed with MPI programming. Its description details

by Intel perform MPI 1.0 ∼ 3.0 measurements for communication are shown in [Table](#_bookmark13) [4](#_bookmark13). The Intel MPI Benchmarks (IMB) [[78](#_bookmark117)] provided

operations for a range of message sizes, which are shown in [Table](#_bookmark12) [5](#_bookmark12).

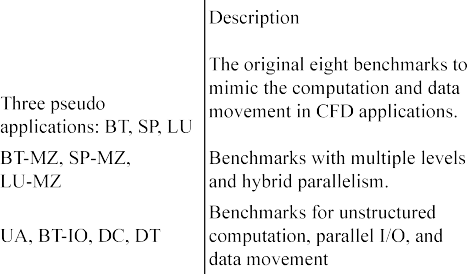
The details of OSU MPI Micro-Benchmarks. The details of Intel MPI Benchmarks.



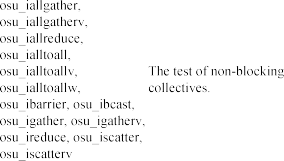
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |

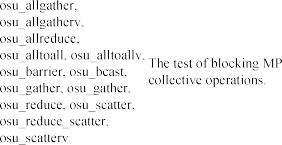
**Table 4**

The details of NAS Parallel Benchmarks (NPB).



* 1. *PGAS benchmarks*

PGAS (Partitioned Global Address Space) is a parallel programming model in the HPC community. PGAS is defined by communications on a shared memory space that every Processing Element (PE) can access without permission issues. Many programming languages and libraries are designed from the PGAS model, e.g., Unified Parallel C (UPC) [[79](#_bookmark118)] and OpenSHMEM [[80](#_bookmark119)]. Communication happens when the processes transfer data from the global memory or to the global memory space, including within and across a node. OSU Micro-Benchmarks also pro- vides benchmarks on PGAS model: OpenSHMEM benchmark is shown in [Table](#_bookmark14) [6](#_bookmark14); UPC and UPC++ benchmarks with point-to-point (*put* and *get* ) and collective communications.



* 1. *RPC benchmarks*

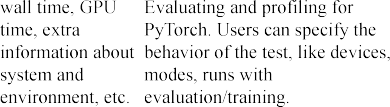
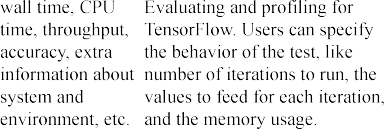
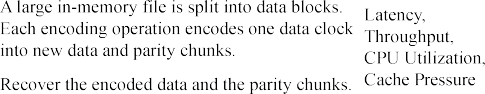
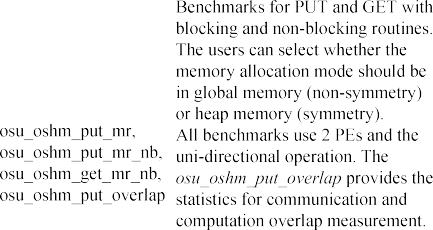
RPC is a method when a process on a machine calls procedures on other machines where the execution of the procedure happens [[81](#_bookmark120)]. It is a client–server interaction where data is transferred frequently over the interconnects to call (from the client) and respond (from the server) procedures. Therefore, the characteristics of the interconnect can have a direct impact on the RPC performance.

We surveyed three benchmarks for RPC applications in data centers: Apache Thrift Benchmarks (ATB), TF-gRPC-Bench, and RPC-perf. ATB is proposed in [[82](#_bookmark121)], which evaluates the Apache Thrift [[83](#_bookmark122)] based

RPC performance and consists of three categories: the RPC latency evaluation benchmark, the RPC throughput evaluation benchmark, and the mixed RPC latency and throughput evaluation benchmark. [Table](#_bookmark15) [7](#_bookmark15) shows the details of TF-gRPC-Bench, which evaluates the communi- cation performance between parameter server and worker process. Twitter maintains RPC-perf [[84](#_bookmark123)]. It is designed to evaluate the RPC’s performance for caching systems regarding latency and message rate.

* 1. *Storage benchmarks*

With the development of hardware technology, much new storage hardware is produced, such as the NVMe SSD [[85](#_bookmark124)]. Storage systems rely on different drivers and libraries in data center, with interactions between the processors and the storage devices via interconnects. Intel SPDK [[86](#_bookmark125)] provides NVMe perf [[87](#_bookmark126)] as an NVMe SSDs benchmarking tool with minimal overhead in benchmarking. NVMe perf provides several runtime options to support the most common workload. Users

The details of OSU OpenSHMEM Micro-Benchmarks.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |  |  |

**Table 7**

The details of TF-gRPC-Bench.

|  |  |  |
| --- | --- | --- |
|  |  | |
|  |  |  |
|  |
|  |  |

**Table 8**

The details of IOzone and Iometer.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

can configure the NVMe perf in many aspects like the workload charac- terizations (e.g., the percentage of Read/Write, with/without random Read/Write), the data movement protocols (e.g., PCIe, RDMA, TCP), and the execution time [[87](#_bookmark126)].

Besides the hardware, modern data centers also have different storage systems. Two surveyed benchmarks for storage systems are shown in [Table](#_bookmark16) [8](#_bookmark16). IOzone [[88](#_bookmark127)] is a filesystem benchmark to measure the file operations in storage systems. Iometer [[89](#_bookmark128)] is an I/O subsystem measurement tool for single and clustered systems. And one bench- mark for EC (Erasure Coding) coder on distributed storage systems. EC-Bench [[90](#_bookmark129)] is an erasure coding scheme benchmark for storage architectures with description details in [Table](#_bookmark17) [9](#_bookmark17).

* 1. *GPU applications benchmarks*

GPU has been becoming incredibly popular for compute-intensive workloads in data centers and HPC clusters in recent years. Two popular deep learning frameworks, TensorFlow [[91](#_bookmark130)] and PyTorch [[92](#_bookmark131)], provide benchmarks to evaluate deep learning models, such as Per- fZero [[93](#_bookmark132)] and TorchBench [[94](#_bookmark133)]. PerfZero is a benchmark framework

The details of EC-Bench.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |

**Table 10**

The details of PerfZero and TorchBench.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

for debugging and tracking the TensorFlow performance regression and change. TorchBench includes a collection of open-source bench- marks to evaluate models and workloads with PyTorch. More details about PerfZero and TorchBench are shown in [Table](#_bookmark18) [10](#_bookmark18). The PARAM benchmark [[95](#_bookmark134)] from Meta Platforms (Facebook formerly) can both evaluate the performance of communication components in the Py- Torch deep learning framework, and evaluate the application-level workloads, like deep learning recommendation models [[96](#_bookmark135),[97](#_bookmark136)]. Deep- Bench [[98](#_bookmark137)] produced by BaiduResearch is another benchmark for eval- uating deep learning operations on different platforms. NCCL (NVIDIA Collective Communications Library) [[99](#_bookmark138)] and Gloo [[100](#_bookmark139)] provide their benchmarks on collective communication libraries, which are NCCL Tests [[101](#_bookmark140)] and Gloo Benchmarking [[102](#_bookmark141)] to evaluate the performance on collective operations.

OSU Micro-Benchmarks also provide several extensions for GPU programming models and libraries, such as CUDA [[103](#_bookmark142)], ROCm [[104](#_bookmark143)], and OpenACC [[105](#_bookmark144)] extensions by configuring with --enable-cuda,

--enable-rocm, and --enable-openacc in the runtime [[14](#_bookmark53)].

* 1. *Key-Value Store benchmarks*

Key–Value Store holds a data storage model that stores associa- tions between keys and values. Keys are primitives, and values can be primitive or complex. It is popular in the big data community and widely used in NoSQL databases in data centers because of its high efficiency and scalability. We surveyed two benchmarks for Key– Value Store. YCSB [[106](#_bookmark145)] (Yahoo! Cloud Serving Benchmark) is used for evaluating the performance of key–value and cloud serving stores. YCSB provides five workloads with different percentages of database operations and evaluates three metrics of performance: the latency of requests, the database performance when increasing machines, and the database performance with increasing machines while the sys- tem is running. OSU HiBD-Benchmarks [[107](#_bookmark146)] provide benchmarks for evaluating Memcached and HBase based Key–Value Store.

* 1. *Microservice benchmarks*

Microservice is a type of cloud service architecture. Unlike tradi- tional monolithic applications, microservice consists of multiple ser- vices working together to finish a workload. Therefore, communi- cations happen frequently among services via interconnects in data centers. We surveyed two benchmarks for microservice workloads.

**Table 11**

The details of DeathStarBench.

**Table 13**

The details of the testbeds in the experiments. PADSYS and Pinnacles [[110](#_bookmark149)] clusters are used in Section [5.2](#_bookmark24) and Section [5.3](#_bookmark28), Bebop [[111](#_bookmark150)] and JLSE [[112](#_bookmark151)] clusters are used in Section [5.4](#_bookmark34).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |

Testbed (Nodes)

Interconnect (Gbps)

Intel Xeon CPU

RAM (GB)

Communication Subsystem





**Table 12**

The details of *𝜇*Suite.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

DeathStarBench [[108](#_bookmark147)] is an open-source benchmark suite for microser- vices on cloud and edge systems, and the details are shown in [Table](#_bookmark19) [11](#_bookmark19).

*𝜇*Suite [[109](#_bookmark148)] can be used for evaluating the influence of OS and

network on microservices, and the details are shown in [Table](#_bookmark21) [12](#_bookmark21).

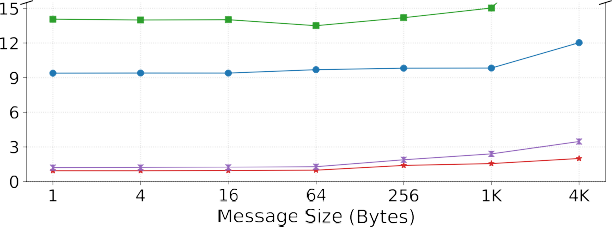
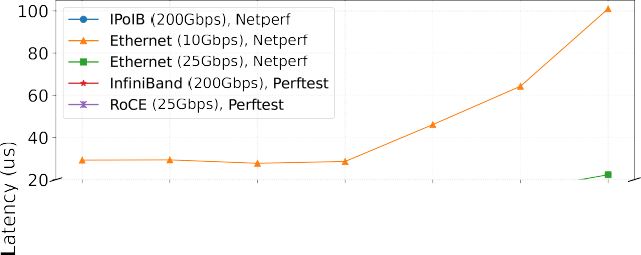
* 1. *Summary*

The above-mentioned application-level benchmarks can represent a broad range of data center applications, including HPC, big data, AI, and cloud computing. In this survey, we choose MPI and PGAS based benchmarks as application examples and run them on different interconnects. MPI and PGAS based benchmarks have been designed and maintained for many years with a lot of contributed optimizations from the community. Our experience also reveals that they are easy to deploy and convenient to run. The experiment results are shown in Section [5](#_bookmark22).

# Experiment

This section presents performance characterizations with the se- lected benchmarks on various hot interconnects.

**/ig. 2.** The latency of *Perftest on 200 Gbps InfiniBand*, *Perftest on RoCE (25 Gbps Ethernet)*, *Netperf on 10/25 Gbps Ethernet*, and *Netperf on IPoIB (200 Gbps InfiniBand)*.



* 1. *Benchmarking setup*

We run benchmarks on different clusters with various interconnects and [Table](#_bookmark20) [13](#_bookmark20) shows the details of each cluster. We try to keep the comparisons of the experiment results as fair as possible by: (1) allocat- ing nodes in the same rack across different experiments; (2) tuning the number of iterations (Perftest) or time duration (Netperf) of benchmark options until getting the relatively stable results.

* 1. *Micro-benchmark evaluation*

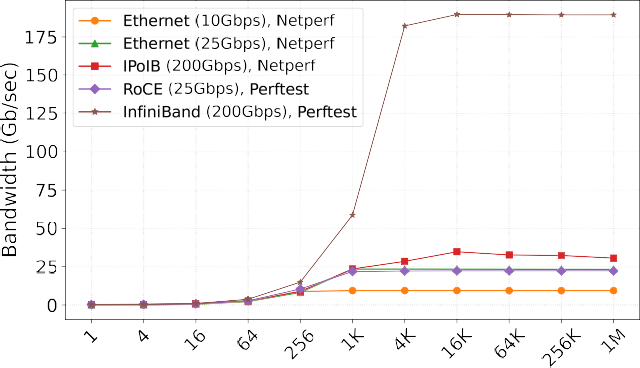
We organize the following experiments using two programming models, RDMA and socket, with two Perftest and Netperf micro- benchmarks on three 10/25 Gbps Ethernet and 200 Gbps InfiniBand interconnects. We discuss the experiment results in three aspects:

(1) the latency comparison; (2) the bandwidth usage comparison;

(3) the impact on the performance using two different InfiniBand interconnects.

* + 1. *Latency*

[Fig.](#_bookmark23) [2](#_bookmark23) shows the latency of benchmarks based on different network. The Perftest benchmark on 200 Gbps InfiniBand, the fastest intercon- nect in our experiments, has the lowest latency because of the nature of kernel-bypass and high-performance protocol in RDMA. Although RoCE also supports RDMA, the hardware it uses is 25 Gbps Ethernet on our testbed which is lower than 200 Gbps InfiniBand, so the Perftest on RoCE are slower than those on InfiniBand. The Netperf [[12](#_bookmark51)] is a TCP benchmark running on IPoIB and 10/25 Gbps Ethernet. Due to the well- known heavy overhead of TCP [[29](#_bookmark68),[115](#_bookmark154)], the latency numbers of these three are far slower than the native RDMA designs, and the latency becomes larger with the decrease of the network bandwidth.



**/ig. 3.** The bandwidth of *Perftest on 200 Gbps InfiniBand*, *Perftest on RoCE (25 Gbps Ethernet)*, *Netperf on 10/25 Gbps Ethernet*, and *Netperf on IPoIB (200 Gbps InfiniBand)*.

* + 1. *Bandwidth*

[Fig.](#_bookmark25) [3](#_bookmark25) shows the bandwidth comparisons using different benchmarks on different interconnects. The RDMA benchmark, Perftest, runs on 200 Gbps InfiniBand and can achieve the highest bandwidth of Infini- Band when the message size is large enough (4K bytes) because of the 4K bytes MTU setting on InfiniBand. The same benchmark running on 25 Gbps Ethernet, shown as RoCE in the figure, shows the same behav- ior but its MTU is 1K bytes so RDMA RoCE saturates the bandwidth earlier than the one on RDMA IB. The Netperf benchmark running on 10/25 Gbps Ethernet shows a similar behavior when the message size is larger than one MTU (1K bytes) but does not show the same on 200 Gbps InfiniBand. The reason is the TCP protocol stack overhead by deploying IPoIB on InfiniBand. We also observed the unstable results on Netperf benchmark evaluations and the reason could come from the performance fluctuation nature of TCP. Therefore, we ran the experiment five times for each one and took the average results to show in the figure.

* + 1. *InfiniBand EDR VS. HDR*

200 Gbps InfiniBand (HDR) is emerging as a replacement of the widely-used 100 Gbps InfiniBand (EDR) in data center and HPC clus- ters. Therefore, it is high time to use benchmarks to compare the performance characteristics between EDR and HDR. In this experiment, we run the same benchmark, Perftest, on these two kinds of InfiniBand interconnects. [Fig.](#_bookmark29) [4](#_bookmark29) shows the bandwidth comparison and [Fig.](#_bookmark30) [5](#_bookmark30) shows the throughput comparison. As we expect, the saturated bandwidth (when message size is larger than 4K bytes, which equals to one MTU) of IB HDR is around two times that of IB EDR. We observe that the throughput of IB EDR is lower than the throughput of IB HDR all the time, and the IB HDR throughput numbers are 1.5X–2X times of the EDR numbers, which corresponds to the bandwidths differences between IB EDR and HDR. For *read* in 4K bytes message size, its performance is poorer than *send* and *write*. The reason is the *read* requests need to be maintained with more context overhead to wait the responses arrive [[68](#_bookmark107)].

* 1. *MPI benchmark evaluation*

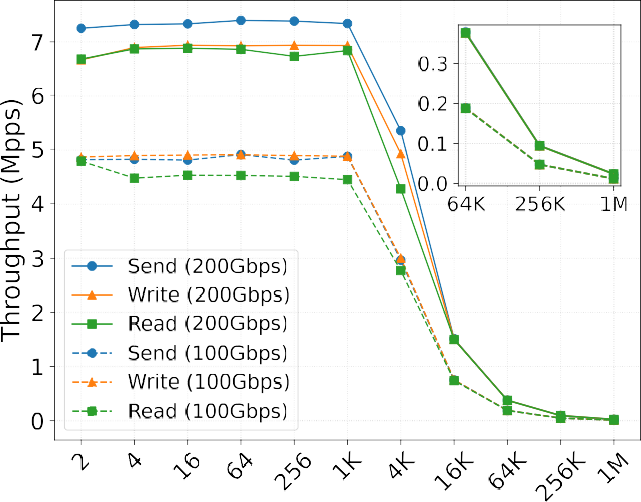
This section gives the evaluation results (latency, bandwidth, and throughput) with OSU MPI Micro-Benchmarks on IB EDR and HDR.

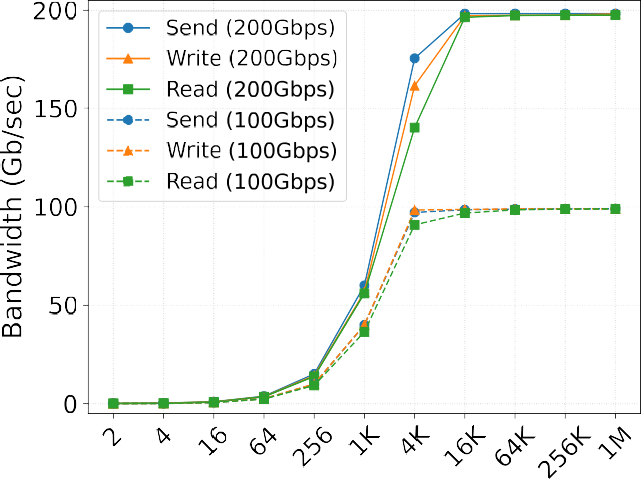
* + 1. *Latency*

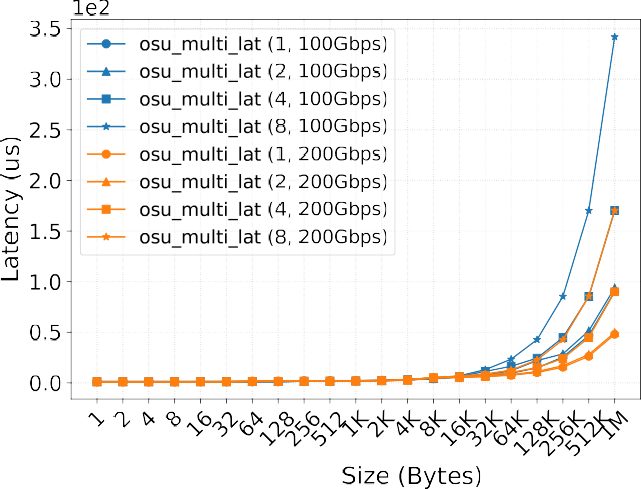
The latency evaluation is shown in [Fig.](#_bookmark31) [6](#_bookmark31). MPI adds an extra software layer over low-level RDMA verbs. Therefore, the latency of MPI is slightly higher than that of Perftest in Section [5.2](#_bookmark24). The first observation is that running MPI on IB HDR has lower latency than on IB EDR, as expected. The second observation is that the average latency will increase with more processes usage, which scenario is closer to the real world because of the more communication overhead.



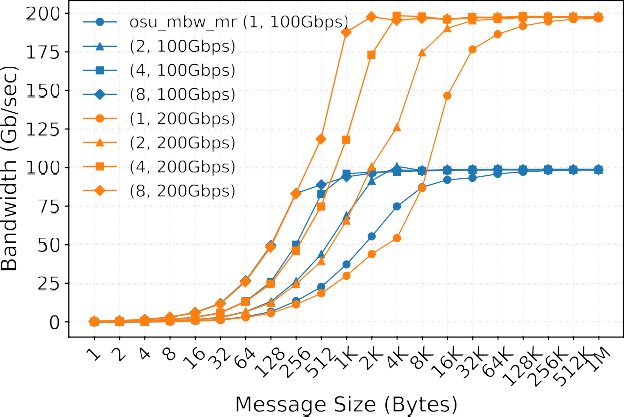
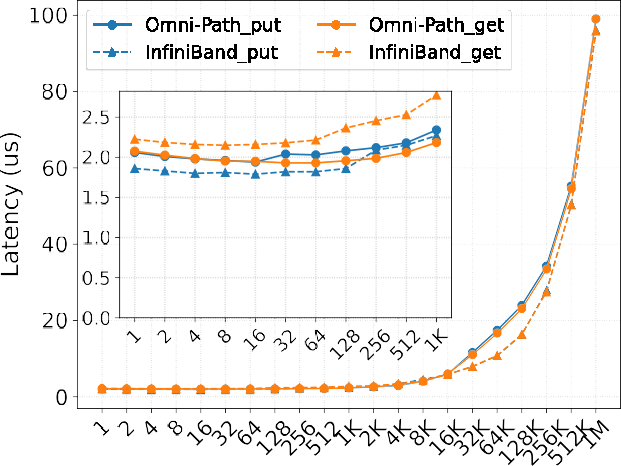
**/ig. 4.** Bandwidth evaluations on IB EDR and HDR.



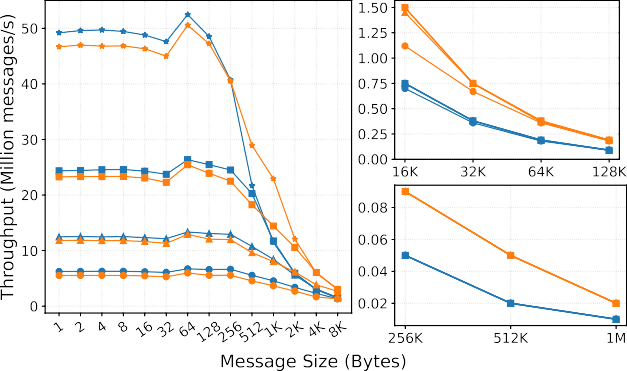
**/ig. 5.** Throughput evaluations on IB EDR and HDR.



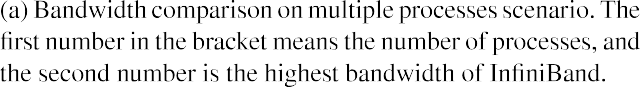
**/ig. 6.** OSU Micro-Benchmarks for MPI latency on multiple process pairs scenario. The first number in the bracket means the number of process pairs, and the second number is the highest bandwidth of InfiniBand.







**/ig. 7.** The OSU Micro-Benchmarks for MPI bandwidth and throughput on multiple processes scenario.

* + 1. *Bandwidth*

We also see the same trend as in Section [5.2.2](#_bookmark26) when the MPI benchmark runs on different IB interconnects in [Fig.](#_bookmark32) [7(a)](#_bookmark32): The saturated bandwidth on IB HDR is two times the saturated bandwidth on IB EDR. The more processes are used, the earlier the bandwidth will be saturated.

* + 1. *Throughput*

Although in Section [5.2.3](#_bookmark27) we can see that when the message size is the same, the throughput of Perftest on IB EDR is lower than that on IB HDR, we do not get the exact same behavior on the throughput of OSU Micro-Benchmarks for MPI which is shown in [Fig.](#_bookmark33) [7(b)](#_bookmark33). The reason comes from different aspects. When the message size is small, MPI cannot saturate the bandwidth, so the throughput is almost the same at that stage. When the message size becomes larger, the through- put decreases rapidly and starts to saturate the bandwidth. We can observe that the throughput of IB HDR is around two times that of IB EDR, corresponding to the bandwidth ratio between two InfiniBand interconnects.

* 1. *PGAS benchmark evaluation*

We use OSU Micro-Benchmarks for OpenSHMEM to characterize the performance of running OpenSHMEM benchmarks on different interconnects. We use Sandia-OpenSHMEM (SOS) [[116](#_bookmark155)] because SOS

**/ig. 8.** OSU Micro-Benchmarks for OpenSHMEM for point-to-point communication on Omni-Path and InfiniBand.

is one of the native OpenSHMEM implementations. To evaluate how interconnect influences the OpenSHMEM performance, we evaluate the latency of point-to-point communication on 2 nodes (1 PE per node) and collective communication on 8 nodes (1 PE per node) on two different interconnects: Omni-Path Fabric and InfiniBand with the same 100 Gbps bandwidth. [Fig.](#_bookmark35) [8](#_bookmark35) shows the latency performance comparison that are divided into two parts, point-to-point operations: *put* and *get*, and [Fig.](#_bookmark37) [9](#_bookmark37) shows the comparison of collective operations: *broadcast* (one-to-all) and *alltoall* (all-to-all).

For the point-to-point communication in [Fig.](#_bookmark35) [8](#_bookmark35), the latency results on Omni-Path and InfiniBand are comparable in most cases. The latency of *get* operation on Omni-Path is slightly better than that on IB with

medium message size (∼1K bytes). When it goes to large messages (16–

512K bytes), the latency of IB is better than that of Omni-Path for both

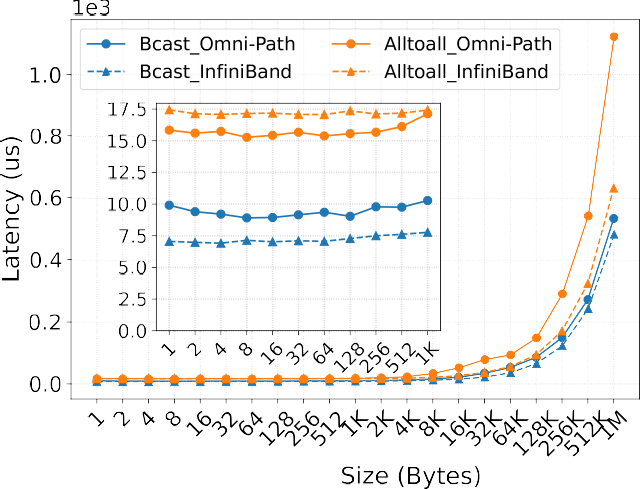
*put* and *get* operations. Although SOS has some specific optimizations on Omni-Path, we do not observe the corresponding optimizations compared with IB for *put* and *get*, which could attribute to the hetero- geneous hardware configurations, like the cache size or CPU, on two clusters.

The collective communication performance is shown in [Fig.](#_bookmark37) [9](#_bookmark37). InfiniBand shows a lower latency number than Omni-Path in most cases, except for *alltoall* operation in small message size. Especially, the latency number of *alltoall* operation on Omni-Path is much slower than the one on InfiniBand. In addition to the heterogeneous hardware configurations, the different network frameworks (Libfabric on Omni- Path and UCX on InfiniBand) could also be why the performance is different.

# Discussion

This section summarizes some improvable aspects of existing bench- marks as we have observed after showing the example experiments and performance characterization results. (1) *It is not always easy to get stable numbers.* As we mentioned earlier, many benchmarks are not stable, and we may need to tune many parameters carefully or take the average of multiple rounds of running to get stable numbers. (2) *Not many benchmarks provide reference numbers.* Some benchmarks do not give reference numbers so that the users cannot evaluate whether their results are reasonable or not. Hence, we encourage our community to publish more reference numbers with the surveyed benchmarks in this paper on various interconnects as guidance. (3) *Some benchmarks lack clear instructions or specifications.* Some benchmarks assume that users are experts and do not provide clear instructions or specifications

Computers) [[136](#_bookmark173)] benchmarks the workloads and shared-memory pro- grams for chip-multiprocessors and contains thirteen programs in dif- ferent areas.



To the best of our knowledge, we are the first to extensively survey the benchmarks for hot interconnects.

# 8. Conclusion

**/ig. 9.** OSU Micro-Benchmarks for OpenSHMEM with Broadcast and Alltoall on Omni-Path and InfiniBand.

about benchmark installation, configuration, and usage information.

(4) *Not many benchmarks provide detailed warnings.* We find that the proper warning messages are helpful for benchmarking. For example, Perftest can warn the users when ‘CPU frequency is not max’. (5) *Benchmarks usually use different ways to calculate and present numbers.* Some benchmarks may use number of iterations to characterize perfor- mance, while some use time duration. Some benchmarks present the best performance numbers, while some benchmarks give the average performance numbers. Therefore, users need to compare the numbers across benchmarks carefully.

# Related work

Besides the related survey studies and benchmarks discussed in Sec- tion [1](#_bookmark3), [3](#_bookmark9), and [4](#_bookmark10), this section summarizes more benchmarking studies.

**Benchmarking distributed storage systems:** The distributed stor- age benchmarks evaluate how the storage system serves requests for reading and writing files and objects. For example, SKB [[117](#_bookmark156)] sup- ports performance benchmarking of 43 distributed storage systems. The Cloud Object Storage Benchmark [[118](#_bookmark157)] is for benchmarking cloud object storage services. Acquaviva et al. [[119](#_bookmark158)] developed a benchmark to evaluate different Cloud Distributed File Systems.

**Benchmarking big data systems:** Many benchmarks are proposed to evaluate the big data systems with the big data boom. HiBench [[120](#_bookmark159)] and MRBench [[121](#_bookmark160)] are designed for evaluating MapReduce systems. TextBenDS [[122](#_bookmark161)] is applied to evaluate the performance of Hive, Spark, and MongoDB on a textual corpus. The TPC [[123](#_bookmark162)] organiza- tion designed benchmark standards what were data-centric benchmark and disseminated verifiable data to the industry. Wang et al. [[124](#_bookmark163)] discussed the challenges of using the widely-used benchmarks (TPC-C and YCSB) for systems evaluation. DCQCN [[125](#_bookmark164)] and DSCP-BASEDPFC

[[126](#_bookmark165)] introduce how to benchmark and monitor the RDMA traffic on data centers with RoCEv2 networks.

**Benchmarking AI systems:** Both AIBench [[127](#_bookmark166)–[129](#_bookmark167)] and MLPerf [[130](#_bookmark168)–[132](#_bookmark169)] cover a broad diversity of scenarios to evaluate the AI systems. DataPerf [[133](#_bookmark170)] benchmarks the datasets in machine learning and the algorithms in processing these datasets. HPC AI500 [[134](#_bookmark171)] is a benchmark suite to evaluate HPC systems that run real-world workloads.

**Benchmarking computing systems:** SPEC (Standard Performance Evaluation Corporation) [[135](#_bookmark172)] designed standardized benchmarks and tools to evaluate performance and energy efficiency for computing sys- tems. PARSEC (Princeton Application Repository for Shared-Memory

This paper presents an extensive survey on hot interconnects in modern data centers and HPC clusters and associated benchmarks to help the community understand these advanced interconnects bet- ter. After introducing some representative modern interconnects and their features, we survey some commonly used micro-benchmarks and application-level benchmarks that can be used on these interconnects to measure their performance. Based on the micro-/application-level benchmarks survey, we conduct experiments on some kinds of real in- terconnects with the corresponding benchmarks, illustrate performance characteristics of these interconnects, and provide our interpretation of the experiment results. Considering the continuous evolution of data center interconnects and benchmarks in the future, we also discuss existing benchmarks’ improvable aspects and our insights for future benchmark design and development.

# Declaration of competing interest

The authors declare that they have no known competing finan- cial interests or personal relationships that could have appeared to influence the work reported in this paper.

# Acknowledgments

This work was supported in part by the NSF research grant CCF #2132049. Part of this research was supported by the Exascale Comput- ing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Admin- istration, and by the U.S. Department of Energy, Office of Science, under Contract DE-AC02-06CH11357. We gratefully acknowledge the computing resources provided on Bebop, an HPC cluster operated by the Laboratory Computing Resource Center at Argonne National Labo- ratory, USA. Part of this research was conducted using Pinnacles (NSF MRI, #2019144) at the Cyberinfrastructure and Research Technologies (CIRT) at University of California, Merced.

# References

1. [D.K. Panda, X. Lu, D. Shankar, High-Performance Big Data Computing, The MIT](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb1) [Press, 2022.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb1)
2. Mellanox, Introducing 200G HDR InfiniBand Solutions, 2022, [https:](https://network.nvidia.com/sites/default/files/pdf/whitepapers/WP_Introducing_200G_HDR_InfiniBand_Solutions.pdf)

[//network.nvidia.com/sites/default/files/pdf/whitepapers/WP\_Introducing\_](https://network.nvidia.com/sites/default/files/pdf/whitepapers/WP_Introducing_200G_HDR_InfiniBand_Solutions.pdf) [200G\_HDR\_InfiniBand\_Solutions.pdf](https://network.nvidia.com/sites/default/files/pdf/whitepapers/WP_Introducing_200G_HDR_InfiniBand_Solutions.pdf).

1. Slingshot, 2022, [https://www.hpe.com/us/en/compute/hpc/slingshot-](https://www.hpe.com/us/en/compute/hpc/slingshot-interconnect.html) [interconnect.html](https://www.hpe.com/us/en/compute/hpc/slingshot-interconnect.html).
2. [B. Alverson, E. Froese, L. Kaplan, D. Roweth, Cray XC Series Network, 2012,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb4) [Cray Inc., White Paper WP-Aries01-1112.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb4)
3. [Y. Ajima, T. Kawashima, T. Okamoto, N. Shida, K. Hirai, T. Shimizu, S.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb5) [Hiramoto, Y. Ikeda, T. Yoshikawa, K. Uchida, et al., The Tofu Interconnect](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb5) [D, in: 2018 IEEE International Conference on Cluster Computing, CLUSTER,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb5) [IEEE, 2018, pp. 646–654.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb5)
4. 100 Gigabit Ethernet, 2022, [https://en.wikipedia.org/wiki/100\_Gigabit\_](https://en.wikipedia.org/wiki/100_Gigabit_Ethernet) [Ethernet](https://en.wikipedia.org/wiki/100_Gigabit_Ethernet).
5. Terabit Ethernet, 2022, <https://en.wikipedia.org/wiki/Terabit_Ethernet>.
6. [R. Han, X. Lu, J. Xu, On Big Data Benchmarking, in: Workshop on Big Data](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb8) [Benchmarks, Performance Optimization, and Emerging Hardware, Springer,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb8) [2014, pp. 3–18.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb8)
7. [Q. Zhang, L. Zha, J. Lin, D. Tu, M. Li, F. Liang, R. Wu, X. Lu, A Survey](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb9) [on Deep Learning Benchmarks: Do We Still Need New Ones? in: International](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb9) [Symposium on Benchmarking, Measuring and Optimization, Springer, 2018, pp.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb9) [36–49.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb9)
8. [X. Zhou, X. Peng, T. Xie, J. Sun, C. Xu, C. Ji, W. Zhao, Poster: Bench-](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb10) [marking Microservice Systems for Software Engineering Research, in: 2018](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb10) [IEEE/ACM 40th International Conference on Software Engineering: Companion](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb10) [(ICSE-Companion), IEEE, 2018, pp. 323–324.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb10)
9. W. Gao, J. Zhan, L. Wang, C. Luo, D. Zheng, X. Wen, R. Ren, C. Zheng, X. He,

H. Ye, et al., BigDataBench: A Scalable and Unified Big Data and AI Benchmark Suite, 2018, arXiv preprint [arXiv:1802.08254](http://arxiv.org/abs/1802.08254).

1. Netperf, 2022, <https://github.com/HewlettPackard/netperf>.
2. Open Fabrics Enterprise Distribution (OFED) Performance Tests, 2022, [https:](https://github.com/linux-rdma/perftest)

[//github.com/linux-rdma/perftest](https://github.com/linux-rdma/perftest).

1. D.K. Panda, H. Subramoni, C.-H. Chu, M. Bayatpour, The MVAPICH project: Transforming Research into High-performance MPI Library for HPC Community, J. Comput. Sci. 52 (2021) 101208, [http://dx.doi.org/10.](http://dx.doi.org/10.1016/j.jocs.2020.101208) [1016/j.jocs.2020.101208](http://dx.doi.org/10.1016/j.jocs.2020.101208), URL [https://www.sciencedirect.com/science/article/](https://www.sciencedirect.com/science/article/pii/S1877750320305093) [pii/S1877750320305093](https://www.sciencedirect.com/science/article/pii/S1877750320305093). Case Studies in Translational Computer Science.
2. M.P.I. Forum, MPI: A Message-Passing Interface Standard Version 4.0, 2021, URL <https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf>.
3. G. Almasi, PGAS (Partitioned Global Address Space) Languages, in: D. Padua (Ed.), Encyclopedia of Parallel Computing, Springer US, Boston, MA, 2011, pp. 1539–1545, <http://dx.doi.org/10.1007/978-0-387-09766-4_210>.
4. OmniPath, 2022, <https://www.cornelisnetworks.com/products/>.
5. [S. Derradji, T. Palfer-Sollier, J.-P. Panziera, A. Poudes, F.W. Atos, The](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb18) [BXI Interconnect Architecture, in: 2015 IEEE 23rd Annual Symposium on](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb18) [High-Performance Interconnects, IEEE, 2015, pp. 18–25.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb18)
6. TOP 500 Supercomputer Sites, 2022, [http://www.top500.org](http://www.top500.org/).
7. [I.T. Association, et al., Supplement to InfiniBand Architecture Specification,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb20) [Release 1 (2) (2010) 1.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb20)
8. [IEEE std 802.3x-1997 and IEEE std 802.3y-1997 (supplement to ISO/IEC 8802-](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb21) [3: 1996, in: ANSI/IEEE Std 802.3, 1996 ed., IEEE Standards for Local and](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb21) [Metropolitan Area Networks: Supplements to Carrier Sense Multiple Access with](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb21) [Collision Detection (CSMA/CD).](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb21)
9. [S. Floyd, TCP and Explicit Congestion Notification, ACM SIGCOMM Comput.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb22) [Commun. Rev. 24 (5) (1994) 8–23.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb22)
10. [M. Alizadeh, B. Atikoglu, A. Kabbani, A. Lakshmikantha, R. Pan, B. Prabhakar,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb23)

[M. Seaman, Data Center Transport Mechanisms: Congestion Control Theory](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb23) [and IEEE Standardization, in: 2008 46th Annual Allerton Conference on](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb23) [Communication, Control, and Computing, IEEE, 2008, pp. 1270–1277.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb23)

1. Elastic Fabric Adapter - Amazon Elastic Compute Cloud, 2022, [https://docs.](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa.html) [aws.amazon.com/AWSEC2/latest/UserGuide/efa.html](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa.html).
2. OpenFabrics Interfaces Working Group, Libfabric OpenFabrics, 2022, URL [https:](https://ofiwg.github.io/libfabric/)

[//ofiwg.github.io/libfabric/](https://ofiwg.github.io/libfabric/).

1. Summit, 2022, <https://www.olcf.ornl.gov/summit/>.
2. Sierra, 2022, <https://hpc.llnl.gov/hardware/compute-platforms/sierra>.
3. InfiniBand Trade Association, InfiniBand Architecture Specification : Release 1.2.1, 2007, <https://www.afs.enea.it/asantoro/V1r1_2_1.Release_12062007.pdf>.
4. X. Lu, D. Shankar, S. Gugnani, H. Subramoni, D.K. Panda, Impact of HPC Cloud Networking Technologies on Accelerating Hadoop RPC and HBase, in: 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), IEEE Computer Society, Los Alamitos, CA, USA, 2016,

pp. 310–317, <http://dx.doi.org/10.1109/CloudCom.2016.0057>, URL [https://](https://doi.ieeecomputersociety.org/10.1109/CloudCom.2016.0057) [doi.ieeecomputersociety.org/10.1109/CloudCom.2016.0057](https://doi.ieeecomputersociety.org/10.1109/CloudCom.2016.0057).

1. [M. Al-Fares, A. Loukissas, A. Vahdat, A Scalable, Commodity Data Center](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb30) [Network Architecture, ACM SIGCOMM Comput. Commun. Rev. 38 (4) (2008)](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb30) [63–74.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb30)
2. [A. Shpiner, Z. Haramaty, S. Eliad, V. Zdornov, B. Gafni, E. Zahavi, Dragonfly+:](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb31) [Low Cost Topology for Scaling Datacenters, in: 2017 IEEE 3rd International](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb31) [Workshop on High-Performance Interconnection Networks in the Exascale and](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb31) [Big-Data Era (HiPINEB), IEEE, 2017, pp. 1–8.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb31)
3. [J. Jiang, R. Jain, C. So-In, An Explicit Rate Control Framework for Lossless Eth-](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb32) [ernet Operation, in: 2008 IEEE International Conference on Communications,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb32) [IEEE, 2008, pp. 5914–5918.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb32)
4. [P. Newman, Backward Explicit Congestion Notification for ATM Local Area](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb33) [Networks, in: Proceedings of GLOBECOM’93. IEEE Global Telecommunications](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb33) [Conference, IEEE, 1993, pp. 719–723.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb33)
5. GPUDirect RDMA, 2022, [https://docs.nvidia.com/cuda/gpudirect-rdma/index.](https://docs.nvidia.com/cuda/gpudirect-rdma/index.html) [html](https://docs.nvidia.com/cuda/gpudirect-rdma/index.html).
6. [A. Sodani, Knights Landing (KNL): 2nd Generation Intel® Xeon Phi Processor,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb35)

[in: 2015 IEEE Hot Chips 27 Symposium, HCS, IEEE, 2015, pp. 1–24.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb35)

1. Cornelis Networks, 2022, <https://www.cornelisnetworks.com/>.
2. A. Tekin, A. Tuncer Durak, C. Piechurski, D. Kaliszan, F. Aylin Sungur, F. Robertsén, P. Gschwandtner, State-of-The-Art and Trends for Computing and Interconnect Network Solutions for HPC and AI, 2021, Partnership for Advanced Computing in Europe, Available Online At [www.Praceri.Eu](http://www.Praceri.Eu/).
3. QSFP28 Adapter, 2022, [https://www.ibm.com/docs/en/power9?topic=ad-](https://www.ibm.com/docs/en/power9?topic=ad-pcie3-2-port-100-gbe-nic-roce-qsfp28-adapter-fc-ec3l-ec3m-ccin-2cec) [pcie3-2-port-100-gbe-nic-roce-qsfp28-adapter-fc-ec3l-ec3m-ccin-2cec](https://www.ibm.com/docs/en/power9?topic=ad-pcie3-2-port-100-gbe-nic-roce-qsfp28-adapter-fc-ec3l-ec3m-ccin-2cec).
4. NVIDIA Mellanox InfiniBand NDR 400G Architecture, 2022, [https:](https://applieddatasystems.com/wp-content/uploads/2020/11/br-ndr-architecture-brochure.pdf)

[//applieddatasystems.com/wp-content/uploads/2020/11/br-ndr-architecture-](https://applieddatasystems.com/wp-content/uploads/2020/11/br-ndr-architecture-brochure.pdf) [brochure.pdf](https://applieddatasystems.com/wp-content/uploads/2020/11/br-ndr-architecture-brochure.pdf).

1. About Fugaku, 2022, <https://www.r-ccs.riken.jp/en/fugaku/about/>.
2. [J. Kim, W.J. Dally, S. Scott, D. Abts, Technology-driven, Highly-scalable Drag-](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb41) [onfly Topology, in: 2008 International Symposium on Computer Architecture,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb41) [IEEE, 2008, pp. 77–88.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb41)
3. Perlmutter, 2022, <https://www.nersc.gov/systems/perlmutter/>.
4. Frontier, 2022, <https://www.olcf.ornl.gov/frontier/>.
5. Aurora, 2022, <https://www.alcf.anl.gov/aurora>.
6. EI Capitan, 2022, [https://www.hpe.com/us/en/compute/hpc/cray/doe-el-](https://www.hpe.com/us/en/compute/hpc/cray/doe-el-capitan-press-release.html) [capitan-press-release.html](https://www.hpe.com/us/en/compute/hpc/cray/doe-el-capitan-press-release.html).
7. [L.G. Valiant, A Scheme for Fast Parallel Communication, SIAM J. Comput. 11](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb46)

[(2) (1982) 350–361.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb46)

1. Piz Daint, 2022, <https://www.cscs.ch/computers/piz-daint/>.
2. Cori, 2022, <https://docs.nersc.gov/systems/cori/>.
3. [Y. Ajima, S. Sumimoto, T. Shimizu, Tofu: A 6D Mesh/Torus Interconnect for](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb49) [Exascale Computers, Computer 42 (11) (2009) 36–40.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb49)
4. [Y. Ajima, T. Inoue, S. Hiramoto, T. Shimizu, Tofu: Interconnect for the K](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb50) [Computer, Fujitsu Sci. Tech. J. 48 (3) (2012) 280–285.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb50)
5. [Y. Ajima, T. Inoue, S. Hiramoto, S. Uno, S. Sumimoto, K. Miura, N. Shida,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb51)

[T. Kawashima, T. Okamoto, O. Moriyama, et al., Tofu Interconnect 2:](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb51) [System-on-chip Integration of High-performance Interconnect, in: International](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb51) [Supercomputing Conference, Springer, 2014, pp. 498–507.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb51)

1. Fugaku, 2022, <https://www.fujitsu.com/global/about/innovation/fugaku/>.
2. [B.W. Barrett, R.B. Brightwell, R.E. Grant, K.S. Hemmert, K.T. Pedretti, K.B.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb53) [Wheeler, K.D. Underwood, R. Riesen, A.B. Maccabe, T. Hudson, The Portals](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb53)

[4.0. 2 Networking Programming Interface, Sandia National Lab.(SNL-NM),](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb53) [Albuquerque, NM (United States); Intel Corp . . . , 2014.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb53)

1. [D.-G. Sun, Z.-H. Weng, Butterfly Interconnection Implementation for an n-bit](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb54) [Pparallel Ripple Carry Full Adder, Appl. Opt. 30 (14) (1991) 1781–1785.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb54)
2. A. Kashyap, X. Lu, NVMe-OAF: Towards adaptive NVMe-of for IO-intensive workloads on HPC cloud, in: Proceedings of the 31st International Sympo- sium on High-Performance Parallel and Distributed Computing, HPDC ’22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 56–70, <http://dx.doi.org/10.1145/3502181.3531476>.
3. RDMA-bench: A Framework to Understand RDMA Performance, 2022, [https:](https://github.com/efficient/rdma_bench)

[//github.com/efficient/rdma\_bench](https://github.com/efficient/rdma_bench).

1. iPerf2 - A Means to Measure Network Responsiveness and Throughput, 2022, <https://sourceforge.net/projects/iperf2/>.
2. iPerf3 - The Ultimate Speed Test Tool for TCP, UDP and SCTP, 2022, [https:](https://iperf.fr/)

[//iperf.fr/](https://iperf.fr/).

1. Comparison Table of iPerf2 and iPerf3, 2022, [https://iperf2.sourceforge.io/](https://iperf2.sourceforge.io/IperfCompare.html) [IperfCompare.html](https://iperf2.sourceforge.io/IperfCompare.html).
2. Change between iPerf 2.0, iPerf 3.0 and iPerf 3.1, 2022, [https://iperf.fr/iperf-](https://iperf.fr/iperf-doc.php) [doc.php](https://iperf.fr/iperf-doc.php).
3. qperf: Measure RDMA and IP performance, 2022, [https://github.com/linux-](https://github.com/linux-rdma/qperf) [rdma/qperf](https://github.com/linux-rdma/qperf).
4. GPCNet, 2022, <https://github.com/netbench/GPCNET>.
5. [H. Subramoni, K. Hamidouche, A. Venkatesh, S. Chakraborty, D.K. Panda,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb63) [Designing MPI Library with Dynamic Connected Transport (DCT) of InfiniBand:](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb63) [Early Experiences, in: International Supercomputing Conference, Springer, 2014,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb63)

[pp. 278–295.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb63)

1. [L. Shalev, H. Ayoub, N. Bshara, E. Sabbag, A Cloud-Optimized Transport](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb64) [Protocol for Elastic and Scalable HPC, IEEE Micro 40 (6) (2020) 67–73.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb64)
2. Transport Modes - RDMA Aware Programming User Manual v1.7, 2022, [https://docs.nvidia.com/networking/display/RDMAAwareProgrammingv17/](https://docs.nvidia.com/networking/display/RDMAAwareProgrammingv17/Transport%2BModes) [Transport+Modes#](https://docs.nvidia.com/networking/display/RDMAAwareProgrammingv17/Transport%2BModes).
3. IEEE Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices, IEEE Std 1619-2007 (2008) 1–40, [http://dx.doi.org/10.1109/IEEESTD.](http://dx.doi.org/10.1109/IEEESTD.2008.4493450)

[2008.4493450](http://dx.doi.org/10.1109/IEEESTD.2008.4493450).

1. [A. Kalia, M. Kaminsky, D.G. Andersen, Design Guidelines for High Performance](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb67) [RDMA systems, in: 2016 USENIX Annual Technical Conference (USENIX ATC](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb67) [16), 2016, pp. 437–450.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb67)
2. A. Kalia, M. Kaminsky, D.G. Andersen, Using RDMA Efficiently for Key-Value Services, in: Proceedings of the 2014 ACM Conference on SIGCOMM, SIGCOMM ’14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 295–306, <http://dx.doi.org/10.1145/2619239.2626299>.
3. [H. Lim, D. Han, D.G. Andersen, M. Kaminsky, MICA: A Holistic Approach to](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb69) [Fast In-Memory Key-Value Storage, in: 11th USENIX Symposium on Networked](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb69) [Systems Design and Implementation (NSDI 14), 2014, pp. 429–444.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb69)
4. [X. Wei, J. Shi, Y. Chen, R. Chen, H. Chen, Fast In-memory Transaction](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb70) [Processing Using RDMA and HTM, in: Proceedings of the 25th Symposium on](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb70) [Operating Systems Principles, 2015, pp. 87–104.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb70)
5. [Wikipedia contributors, Berkeley Sockets — Wikipedia, the free encyclopedia,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb71) [2022.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb71)
6. Wikipedia contributors, Stream Control Transmission Protocol — Wikipedia, the free encyclopedia, 2022, [https://en.wikipedia.org/w/index.php?title=](https://en.wikipedia.org/w/index.php?title=Stream_Control_Transmission_Protocol&oldid=1091520392) [Stream\_Control\_Transmission\_Protocol&oldid=1091520392](https://en.wikipedia.org/w/index.php?title=Stream_Control_Transmission_Protocol&oldid=1091520392) (Online; Accessed 12 September 2022).
7. W. contributors, Unix Domain Socket — Wikipedia, the free encyclopedia, 2022, [https://en.wikipedia.org/w/index.php?title=Unix\_domain\_socket&oldid=](https://en.wikipedia.org/w/index.php?title=Unix_domain_socket&oldid=1100609125)

[1100609125](https://en.wikipedia.org/w/index.php?title=Unix_domain_socket&oldid=1100609125) (Online; Accessed 12 September 2022).

1. The Open Group, Data Link Provider Interface (DLPI), 2022, [https://pubs.](https://pubs.opengroup.org/onlinepubs/009638599/toc.htm) [opengroup.org/onlinepubs/009638599/toc.htm](https://pubs.opengroup.org/onlinepubs/009638599/toc.htm) (Online; accessed 12 September 2022).
2. S. Chunduri, T. Groves, P. Mendygral, B. Austin, J. Balma, K. Kandalla,

K. Kumaran, G. Lockwood, S. Parker, S. Warren, N. Wichmann, N. Wright, GPCNeT: Designing a Benchmark Suite for Inducing and Measuring Contention in HPC Networks, SC ’19, Association for Computing Machinery, New York, NY, USA, 2019, <http://dx.doi.org/10.1145/3295500.3356215>.

1. M.P.I. Forum, MPI: A Message-Passing Interface Standard Version 3.0, 2012, URL <https://www.mpi-forum.org/docs/mpi-3.0/mpi30-report-book.pdf>.
2. NAS Parallel Benchmarks, 2022, <https://www.nas.nasa.gov/software/npb.html>.
3. Intel® MPI Benchmarks, 2022, [https://www.intel.com/content/www/us/en/](https://www.intel.com/content/www/us/en/developer/articles/technical/intel-mpi-benchmarks.html) [developer/articles/technical/intel-mpi-benchmarks.html](https://www.intel.com/content/www/us/en/developer/articles/technical/intel-mpi-benchmarks.html).
4. U. Consortium, D. Bonachea, G. Funck, UPC Language and Library Speci- fications, Version 1.3, 2013, <http://dx.doi.org/10.2172/1134233>, URL [https:](https://www.osti.gov/biblio/1134233)

[//www.osti.gov/biblio/1134233](https://www.osti.gov/biblio/1134233).

1. [OpenSHMEM Application Programming Interface, v1. 5 Final, 2022.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb80)
2. [M. Van Steen, A.S. Tanenbaum, Distributed Systems, Maarten van Steen Leiden,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb81) [The Netherlands, 2017.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb81)
3. T. Li, H. Shi, X. Lu, HatRPC: Hint-Accelerated Thrift RPC over RDMA, in: Proceedings of the International Conference for High Performance Comput- ing, Networking, Storage and Analysis, SC ’21, Association for Computing Machinery, New York, NY, USA, 2021, [http://dx.doi.org/10.1145/3458817.](http://dx.doi.org/10.1145/3458817.3476191) [3476191](http://dx.doi.org/10.1145/3458817.3476191).
4. Apache Thrift, 2022, <https://thrift.apache.org/>.
5. rpc-perf, 2022, <https://github.com/twitter/rpc-perf>.
6. Wikipedia contributors, NVM Express — Wikipedia, the free encyclo- pedia, 2022, [https://en.wikipedia.org/w/index.php?title=NVM\_Express&oldid=](https://en.wikipedia.org/w/index.php?title=NVM_Express&oldid=1105967834) [1105967834](https://en.wikipedia.org/w/index.php?title=NVM_Express&oldid=1105967834) (Online; accessed 13-September-2022).
7. [Z. Yang, J.R. Harris, B. Walker, D. Verkamp, C. Liu, C. Chang, G. Cao, J. Stern,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb86)

[V. Verma, L.E. Paul, SPDK: A Development Kit to Build High Performance Stor-](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb86) [age Applications, in: 2017 IEEE International Conference on Cloud Computing](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb86) [Technology and Science (CloudCom), IEEE, 2017, pp. 154–161.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb86)

1. SPDK: NVMe Driver, 2022, <https://spdk.io/doc/nvme.html>.
2. IOzone Filesystem Benchmark, 2022, <https://www.iozone.org/>.
3. Iometer project, 2022, <http://www.iometer.org/>.
4. [H. Shi, X. Lu, D.K. Panda, EC-Bench: Benchmarking Onload and Offload Erasure](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb90) [Coders on Modern Hardware Architectures, in: C. Zheng, J. Zhan (Eds.),](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb90) [Benchmarking, Measuring, and Optimizing, Springer International Publishing,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb90) [Cham, 2019, pp. 215–230.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb90)
5. [M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb91) [Ghemawat, G. Irving, M. Isard, et al., TensorFlow: A System for Large-Scale](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb91) [Machine Learning, in: 12th USENIX Symposium on Operating Systems Design](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb91) [and Implementation (OSDI 16), 2016, pp. 265–283.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb91)
6. [A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb92)

[Z. Lin, N. Gimelshein, L. Antiga, et al., PyTorch: An Imperative Style,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb92) [High-Performance Deep Learning Library, Adv. Neural Inf. Process. Syst. 32](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb92) [(2019).](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb92)

1. PerfZero, 2022, [https://github.com/tensorflow/benchmarks/tree/master/](https://github.com/tensorflow/benchmarks/tree/master/perfzero) [perfzero](https://github.com/tensorflow/benchmarks/tree/master/perfzero).
2. TorchBench, 2022, <https://github.com/pytorch/benchmark>.
3. Meta, PARAM, 2022, <https://github.com/facebookresearch/param>.
4. M. Naumov, D. Mudigere, H.M. Shi, J. Huang, N. Sundaraman, J. Park,

X. Wang, U. Gupta, C. Wu, A.G. Azzolini, D. Dzhulgakov, A. Mallevich, I. Cherniavskii, Y. Lu, R. Krishnamoorthi, A. Yu, V. Kondratenko, S. Pereira,

X. Chen, W. Chen, V. Rao, B. Jia, L. Xiong, M. Smelyanskiy, Deep Learning Recommendation Model for Personalization and Recommendation Systems, 2019, CoRR [arXiv:1906.00091](http://arxiv.org/abs/1906.00091).

1. Deep Learning Recommendation Model for Personalization and Recommenda- tion Systems, 2022, <https://github.com/facebookresearch/dlrm>.
2. BaiduResearch, DeepBench, 2022, [https://github.com/baidu-research/](https://github.com/baidu-research/DeepBench) [DeepBench](https://github.com/baidu-research/DeepBench).
3. [S. Jeaugey, Nccl 2.0, in: GPU Technology Conference, GTC, vol. 2, 2017.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb99)
4. Gloo, 2022, <https://github.com/facebookincubator/gloo>.
5. NCCL Tests, 2022, <https://github.com/NVIDIA/nccl-tests>.
6. Gloo, 2022, <https://github.com/facebookincubator/gloo>.
7. [D. Kirk, et al., NVIDIA CUDA Software and GPU Parallel Computing](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb103) [Architecture, in: ISMM, vol. 7, 2007, pp. 103–104.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb103)
8. ROCm, 2022, <https://www.amd.com/en/graphics/servers-solutions-rocm>.
9. OpenACC, 2022, <https://www.openacc.org/>.
10. B.F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, R. Sears, Benchmarking Cloud Serving Systems with YCSB, in: Proceedings of the 1st ACM Symposium on Cloud Computing, Association for Computing Machinery, New York, NY, USA, 2010, pp. 143–154, <http://dx.doi.org/10.1145/1807128.1807152>.
11. The High-Performance Big Data Project, 2022, <https://hibd.cse.ohio-state.edu/>.
12. Y. Gan, Y. Zhang, D. Cheng, A. Shetty, P. Rathi, N. Katarki, A. Bruno, J. Hu,

B. Ritchken, B. Jackson, K. Hu, M. Pancholi, Y. He, B. Clancy, C. Colen, F. Wen, C. Leung, S. Wang, L. Zaruvinsky, M. Espinosa, R. Lin, Z. Liu, J. Padilla,

C. Delimitrou, An Open-Source Benchmark Suite for Microservices and Their Hardware-Software Implications for Cloud & Edge Systems, in: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, Association for Computing Machinery, New York, NY, USA, 2019, pp. 3–18, [http://dx.doi.org/10.1145/](http://dx.doi.org/10.1145/3297858.3304013) [3297858.3304013](http://dx.doi.org/10.1145/3297858.3304013).

1. A. Sriraman, T.F. Wenisch, *𝜇*Suite: A benchmark suite for microservices, in:

2018 IEEE International Symposium on Workload Characterization, 2018, pp. 1–12, <http://dx.doi.org/10.1109/IISWC.2018.8573515>.

1. Pinnacles cluster, 2022, [https://ucmerced.github.io/hpc\_docs/#/Pinnacles](https://ucmerced.github.io/hpc_docs/%23/Pinnacles).
2. Bebop, 2022, <https://www.lcrc.anl.gov/systems/resources/bebop/>.
3. Joint Laboratory for System Evaluation (JLSE), 2022, <https://www.jlse.anl.gov/>.
4. The Unified Communication X Library, 2022, [http://www.openucx.org](http://www.openucx.org/).
5. [P. Shamis, M.G. Venkata, M.G. Lopez, M.B. Baker, O. Hernandez, Y. Itigin,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb114)

[M. Dubman, G. Shainer, R.L. Graham, L. Liss, et al., UCX: An Open Source](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb114) [Framework for HPC Network APIs and Beyond, in: 2015 IEEE 23rd Annual](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb114) [Symposium on High-Performance Interconnects, IEEE, 2015, pp. 40–43.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb114)

1. X. Lu, D. Shankar, S. Gugnani, D.K. Panda, High-Performance Design of Apache

Spark with RDMA and Its Benefits on Various Workloads, in: 2016 IEEE International Conference on Big Data (Big Data), 2016, pp. 253–262, [http:](http://dx.doi.org/10.1109/BigData.2016.7840611)

[//dx.doi.org/10.1109/BigData.2016.7840611](http://dx.doi.org/10.1109/BigData.2016.7840611).

1. Sandia-OpenSHMEM/SOS, 2022, <https://github.com/Sandia-OpenSHMEM/SOS>.
2. [K. Munegowda, N. Sanjay Kumar, Design and Implementation of Storage Bench-](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb117) [mark Kit, in: Emerging Research in Computing, Information, Communication](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb117) [and Applications, Springer, 2022, pp. 45–62.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb117)
3. [Q. Zheng, H. Chen, Y. Wang, J. Duan, Z. Huang, COSBench: A Benchmark Tool](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb118)

[for Cloud Object Storage Services, in: 2012 IEEE Fifth International Conference](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb118) [on Cloud Computing, IEEE, 2012, pp. 998–999.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb118)

1. [L. Acquaviva, P. Bellavista, A. Corradi, L. Foschini, L. Gioia, P.C.M. Picone,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb119)

[Cloud Distributed File Systems: A Benchmark of HDFS, Ceph, GlusterFS, and](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb119) [XtremeFS, in: 2018 IEEE Global Communications Conference, IEEE, 2018, pp.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb119) [1–6.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb119)

1. [S. Huang, J. Huang, J. Dai, T. Xie, B. Huang, The HiBench Benchmark Suite:](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb120)

[Characterization of the MapReduce-based Data Analysis, in: 2010 IEEE 26th](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb120) [International Conference on Data Engineering Workshops, ICDEW 2010, IEEE,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb120) [2010, pp. 41–51.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb120)

1. [K. Kim, K. Jeon, H. Han, S.-g. Kim, H. Jung, H.Y. Yeom, MRBench: A Benchmark](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb121)

[for Mapreduce Framework, in: 2008 14th IEEE International Conference on](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb121) [Parallel and Distributed Systems, IEEE, 2008, pp. 11–18.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb121)

1. [C.-O. Truică, E.-S. Apostol, J. Darmont, I. Assent, TextBenDS: a Generic Textual](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb122)

[Data Benchmark for Distributed Systems, Inform. Syst. Front. 23 (1) (2021)](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb122) [81–100.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb122)

1. The Transaction Processing Performance Council (TPC), 2022, [https://www.tpc.](https://www.tpc.org/)

[org/](https://www.tpc.org/).

1. Y. Wang, M. Yu, Y. Hui, F. Zhou, Y. Huang, R. Zhu, X. Ren, T. Li, X. Lu, A Study of Database Performance Sensitivity to Experiment Settings, Proc. VLDB Endow. 15 (7) (2022) 1439–1452, <http://dx.doi.org/10.14778/3523210.3523221>.
2. Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye,

S. Raindel, M.H. Yahia, M. Zhang, Congestion Control for Large-Scale RDMA Deployments, in: Proceedings of the 2015 ACM Conference on Special In- terest Group on Data Communication, Association for Computing Machinery, New York, NY, USA, 2015, pp. 523–536, [http://dx.doi.org/10.1145/2785956.](http://dx.doi.org/10.1145/2785956.2787484) [2787484](http://dx.doi.org/10.1145/2785956.2787484).

1. C. Guo, H. Wu, Z. Deng, G. Soni, J. Ye, J. Padhye, M. Lipshteyn, RDMA over

Commodity Ethernet at Scale, in: Proceedings of the 2016 ACM SIGCOMM Conference, Association for Computing Machinery, New York, NY, USA, 2016,

pp. 202–215, <http://dx.doi.org/10.1145/2934872.2934908>.

1. AIBench, 2022, <https://www.benchcouncil.org/aibench/>.
2. [W. Gao, F. Tang, J. Zhan, X. Wen, L. Wang, Z. Cao, C. Lan, C. Luo, X. Liu,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb128)

[Z. Jiang, Aibench Scenario: Scenario-distilling AI Benchmarking, in: 2021 30th](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb128) [International Conference on Parallel Architectures and Compilation Techniques,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb128) [IEEE, 2021, pp. 142–158.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb128)

1. [F. Tang, W. Gao, J. Zhan, C. Lan, X. Wen, L. Wang, C. Luo, Z. Cao, X. Xiong,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb129)

[Z. Jiang, et al., AIBench Training: Balanced Industry-standard AI Training](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb129) [Benchmarking, in: 2021 IEEE International Symposium on Performance Analysis](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb129) [of Systems and Software, IEEE, 2021, pp. 24–35.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb129)

1. [P. Mattson, V.J. Reddi, C. Cheng, C. Coleman, G. Diamos, D. Kanter, P.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb130)

[Micikevicius, D. Patterson, G. Schmuelling, H. Tang, et al., MLPerf: An Industry](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb130) [Standard Benchmark Suite for Machine Learning Performance, IEEE Micro 40](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb130)

[(2) (2020) 8–16.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb130)

1. [P. Mattson, C. Cheng, G. Diamos, C. Coleman, P. Micikevicius, D. Patterson, H.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb131) [Tang, G.-Y. Wei, P. Bailis, V. Bittorf, et al., MLPerf Training Benchmark, Proc.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb131) [Mach. Learn. Syst. 2 (2020) 336–349.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb131)
2. C. Banbury, V.J. Reddi, P. Torelli, J. Holleman, N. Jeffries, C. Kiraly, P.

Montino, D. Kanter, S. Ahmed, D. Pau, et al., MLPerf tiny benchmark, 2021, arXiv preprint [arXiv:2106.07597](http://arxiv.org/abs/2106.07597).

1. M. Mazumder, C. Banbury, X. Yao, B. Karlaš, W.G. Rojas, S. Diamos, G. Diamos,

L. He, D. Kiela, D. Jurado, D. Kanter, R. Mosquera, J. Ciro, L. Aroyo, B. Acun, S. Eyuboglu, A. Ghorbani, E. Goodman, T. Kane, C.R. Kirkpatrick, T.-S. Kuo, J. Mueller, T. Thrush, J. Vanschoren, M. Warren, A. Williams, S. Yeung,

N. Ardalani, P. Paritosh, C. Zhang, J. Zou, C.-J. Wu, C. Coleman, A. Ng, P. Mattson, V.J. Reddi, DataPerf: Benchmarks for Data-Centric AI Development, 2022, <http://dx.doi.org/10.48550/ARXIV.2207.10062>, URL [https://arxiv.org/](https://arxiv.org/abs/2207.10062) [abs/2207.10062](https://arxiv.org/abs/2207.10062).

1. [Z. Jiang, W. Gao, L. Wang, X. Xiong, Y. Zhang, X. Wen, C. Luo, H. Ye, X. Lu,](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb134)

[Y. Zhang, S. Feng, K. Li, W. Xu, J. Zhan, HPC AI500: A Benchmark Suite for](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb134) [HPC AI Systems, in: C. Zheng, J. Zhan (Eds.), Benchmarking, Measuring, and](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb134) [Optimizing, Springer International Publishing, Cham, 2019, pp. 10–22.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb134)

1. The Standard Performance Evaluation Corporation (SPEC), 2022, [https://www.](https://www.spec.org/)

[spec.org/](https://www.spec.org/).

1. [C. Bienia, Benchmarking Modern Multiprocessors (Ph.D. thesis), Princeton](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb136) [University, 2011.](http://refhub.elsevier.com/S2772-4859(22)00061-8/sb136)