Reexamining Six Common Paradigms Across the Edge-to-Core Spectrum

Chin Fang[1,*], Timothy Stitt[2], Michael J. McManus[3], and Toshio Moriya[4]

***Abstract*—The pursuit of high-performance data transfer often focuses on raw network bandwidth, and international links of 100 Gbps or higher speed are frequently considered the primary enabler. While necessary, this network-centric view is incomplete, equating provisioned link speeds with practical, sustainable data movement capabilities across the entire edge-to-core spectrum. This paper investigates six common paradigms, from the often-cited constraints of network latency and TCP congestion control algorithms to host-side factors such as CPU performance and virtualization that critically impact data movement workflows. We validated our findings using a latency-emulation-capable testbed for high-speed WAN performance prediction and through extensive production measurements from resource-constrained edge environments to a 100 Gbps operational link connecting Switzerland and California, U.S. These results show that the principal bottlenecks often reside outside the network core, and that holistic hardware-software co-design ensures consistent performance, whether moving data at 1 Gbps or 100 Gbps and faster. This approach effectively closes the fidelity gap between benchmark results and diverse and complex production environments.**

***Index Terms*—End-to-End Data Movement, Data Movement Appliances, Hardware-Software Co-design, Burst Buffer Architecture, Cross-Domain Data Flow, System-level Optimization**

I. Introduction:

The efficient movement of data from edge locations to core data centers and cloud resources is a foundational challenge for modern data-intensive enterprises and scientific collaborations. So far, high-speed long-distance networks are rightly celebrated for their raw bandwidth. For example, ESnet6 provides high-capacity transcontinental and international connectivity essential for global data-intensive scientific collaborations, with speeds from 400 Gbps to 1.2 Tbps [1]. Nevertheless, a predominant focus on raw network bandwidth often obscures a more critical truth: end-to-end data movement is constrained by the full environment—including storage, host architectures, software design, and implemented security measures—rather than by the network alone. Sustainable, end-to-end data movement is constrained less by the network itself than by the full systems environment across the entire data path.

Drawing on over a decade of operational experience—spanning extreme scientific workloads at Linac Coherent Light Source II (LCLS-II) [2], sustained petabyte transfers [3] and [4], international deployments [5], and resource-constrained edge environments, all with consistent near-line-rate performance—our work demonstrates that once a "fast" link is in place, storage subsystems and host configurations become the primary determinants of performance for file-based transfers. This is evidenced by achieving near-line-rate performance on a "high-speed" 100 Gbps transcontinental link (~84 Gbps in a pharmaceutical production environment) while also enabling high-efficiency transfers at 1-10 Gbps rates at the edge. In this paper, we define a 'fast' network as 10 Gbps and faster, and a 'high-speed' network as 100 Gbps and faster.

A key architectural component for achieving predictable performance is the use of high-speed burst buffers [6] as staging areas, enabling reliable, near-line-rate movement when the end-to-end path is properly co-designed. Our prior work, including the Overall Winner at the 2019 Supercomputing Asia Inaugural Data Mover Challenge (SCA19 DMC) [7] and ESnet official evaluation [8], underscores that these factors, not the network, set the true performance limits of data movement, regardless of the network's nominal speed.

The core contribution of this paper is to identify and empirically validate a set of widely held engineering assumptions—six specific paradigms—that continue to shape expectations and investments in production end-to-end data movement. This work examines why these paradigms fall short across the performance spectrum. We demonstrate that the path to reliable performance lies not in isolated component optimization, but in holistic, co-designed and highly-integrated hardware and software systems.

The six paradigms we re-examine are: (1) the critical impact of network latency, (2) the prevalence and impact of packet loss and the criticality of TCP congestion control algorithms, (3) the necessity of dedicated private lines for high-speed testing, (4) the direct relationship between increased network bandwidth and transfer rates, (5) the essential need for powerful CPUs, and (6) the universal utility of virtualization and cloud environments. Throughout this paper, all referenced systems are assumed to run Red Hat Enterprise Linux (RHEL) 9.6 [9] or free rebuilds such as Rocky Linux 9.6 [10].

II. Background / Related Work

Research programs such as the Global Research Platform (GRP) [11] have successfully promoted the orchestration of high-speed networks and dynamic path provisioning. A recent workshop [12] and similar community efforts have significantly increased the visibility and understanding of programmable 400 Gbps-class connectivity. However, a predominant focus on the network can lead to treating the data mover software and storage as auxiliary considerations. This work provides a re-balanced perspective from the angle of moving data at scale and speed.

*Corresponding author: fangchin@zettar.com
[1]Zettar Inc., USA
[2]F. Hoffmann-La Roche Ltd., Switzerland
[3]Intel Corporation (Retired), USA
[4]Institute of Materials Structure Science, High Energy Accelerator Research Organization (KEK), Japan

A. *Burst Buffers and Data Staging*

We will now introduce the burst buffer [6]—a high-speed intermediate storage layer that serves as a staging area between production storage and the network. Originally pioneered in supercomputing to absorb data for compute nodes at extreme velocities, we generalize this concept for moving data at scale and speed. A vendor-agnostic data movement appliance design employs Zettar `zx` [13] and burst buffers to create a predictable performance envelope. This enables sustained near-line-rate data movement between burst buffers over fast WAN links. Critically, the unified data mover `zx` was developed as the foundation for the holistic co-design principle central to this work. Its comprehensive and efficient architecture integrates all necessary functions— including fast WAN transfer and internal data staging (between production storage and burst buffers), supporting diverse protocols (file and object), and handling both bulk and fast streaming transfers. This single, highly concurrent, and scale-out `zx` engine manages the complete data placement workflow, from source storage through transit to destination storage.

B. *Bulk and Streaming Transfers*

The data movement workloads supported by `zx` fall into two primary types:

- **Bulk Transfer**: the movement of a static, pre-existing dataset where the complete data is at rest in the source storage before the transfer initiates. This is typical for migrating archives, databases, or completed experimental runs.
- **Streaming Transfer**: the movement of a dynamic, actively growing dataset where data is transferred concurrently as it is being generated or written to the source storage. This type is essential for real-time workloads, such as data acquisition from scientific instruments like LCLS-II [2] and large-scale bioscience imaging operations (e.g., Cryo-EM, virtual staining) that, due to their inherent data rates, will require such concurrent, line-rate data movement to achieve efficient production workflows."

This distinction is critical because streaming transfers demand high storage throughput and low storage latency. The data mover must be of high concurrency.

C. *4.3 Integrated Appliances vs Software-Centric Approaches*

The network-centric approach discussed previously in **Abstract** and **Introduction**, while valuable, does not fully address the processes required for achieving lasting production value after initial deployment. Empirical evidence, including our own experience, also indicates that providing software alone—even with proven performance in validation trials [3]–[5], and [7]—can create significant operational challenges. Integration, tuning, and maintenance require deep, multi-disciplinary expertise, presenting a substantial total cost of ownership that many organizations' IT departments are not equipped to bear. This is a common limitation of traditional software-centric models.

This paper demonstrates that the end-to-end data movement problem, despite its variations across workflows, can be addressed through a singular co-design engineering principle. It is this principle that forms the basis for our reexamination of the six paradigms, providing the holistic framework that reveals why these isolated assumptions are insufficient and how they can be overcome. We present a unified appliance architecture that embodies this principle across the entire data movement workflow spectrum. This co-design principle is applied across a range of hardware form-factors, from compact, low-power nodes and specialized Data Processing Units (DPUs) to certified enterprise servers from tier-1 vendors, supporting data rates from 1 Gbps up to 100 Gbps and beyond. These appliances encapsulate the necessary hardware/software co-design and tuning that was previously a manual, expert-driven process.

The overall approach is grounded in technology developed over the past decade to meet the ambitious data movement requirements of the premier U.S. DOE Exascale Computing Preparation Project, LCLS-II [2]. The core challenges and performance targets defined by the project directly motivated the architecture. Furthermore, the ongoing dialogue with LCLS-II and other tier-1 scientific entities continues to advance and validate the evolution of this holistic approach to data movement systems. The core software's performance has been independently validated on long-distance 100 Gbps links, encompassing petabyte-scale transfers with ESnet [3], [4], and [8], SCA19 DMC [7], and transcontinental production trials [5].

The outcomes documented by these independent institutions [3],[4], and [7] are fully reproducible. Nevertheless, the most consistent and reliable path to reproducing these results in production is not through a software download and a custom integration project—which our experience shows can lead to significant performance variability—but through the use of pre-validated, integrated systems. This co-design principle has been implemented in commercially available appliances, with implementations ranging from regular server-based appliances from tier-1 vendors such as HPE to more specialized form factors, demonstrating that reliable fast data movement is most effectively achieved through complete systems engineering [14] and [15]. The underlying principle is akin to a high-performance electric vehicle: engineered for extreme speed and efficiency, yet perfectly capable of simple, reliable operation for everyday tasks.

Despite the recognized challenges of end-to-end data movement, the prevailing industry and research approach continues to favor complex tuning methodologies and reliance on prohibitively expensive commercial testbeds or bespoke hardware. This complexity is particularly acute at the network edge (the "headwaters" of the data drainage basin, **Fig. 3**), where resource-constrained environments like hospitals, clinics, and remote laboratories typically utilize 1–10 Gbps links. Deploying reliable, high-efficiency data transfer in these settings becomes cost-prohibitive when conventional solutions require specialized IT staff and expensive hardware.

The architecture proposed in this paper directly addresses these barriers by validating a different principle: that architectural co-design can inherently reduce both operational complexity and capital expenditure across the entire data spectrum. This principle is practically realized in a mini-appliance designed for resource constrained edge sites. The hardware costs approximately **$2,000** (**Fig. 2 Mini Appliance**). This demonstrates that high-efficiency 1–10 Gbps data transfer is achievable simply and affordably, without complex system integration or the utilization of inappropriate virtualization options (section **5.6**). This cost efficiency and simplicity are a direct consequence of the holistic approach detailed in the following sections.

III. Six Common Paradigms

A. *The 'Latency Killer' paradigm*

A prevalent assumption in data-intensive computing is that *network latency is the ultimate data movement killer.* While this thinking is common among experienced IT professionals, our empirical testing and architectural analysis demonstrate that this paradigm requires re-examination. The perceived severity of latency is fundamentally governed by TCP's Bandwidth-Delay Product (BDP), which dictates the required window size for full bandwidth utilization.

This paradigm can present significant challenges for distributed, data-intensive engineering and scientific endeavors, an impact particularly relevant in the bio pharma and life sciences sector—the top global data generator, with data volume projected to grow at a Compound Annual Growth Rate (CAGR) of 36% through 2025 [16]. The challenges of moving data at this scale are a primary concern for industry leaders, including organizations such as tier-1 biopharma businesses and enterprise computing providers like HPE. Inefficient data movement, if not properly addressed, can negatively impact drug discovery, precision medicine development, and research timelines. However, as we demonstrate, these challenges can be significantly mitigated through architectural and software approaches that reduce sensitivity to latency.

For years, established resources such as the U.S. DOE ESnet's fasterdata knowledge base [17] have documented methods to mitigate latency's impact on data rates. While ESnet provides valuable guidance as a network service provider, a comprehensive solution requires consideration of the entire environment, not just the network—a point we expand in Section **5.3**. The reproducible test results summarized in **FIG. 1** and the appliance design in **Fig. 2** and **Fig. 3** together exemplify this holistic approach.

The test utilized `iperf3`, a benchmark tool maintained by the U.S. DOE ESnet [18]. Beginning with version 3.16 (released November 30, 2023 [19]), the tool's architecture was re-architected to be genuinely multithreaded, making it significantly easier to tune for high-latency, high-speed links (e.g., ~160 Gbps memory-to-memory over a 200 Gbps ESnet testbed data path) [20].

Section **5.3** is to provide descriptions of a testbed design by us. Among many uses of the testbed, it enables the testing of different latency values automatically (**Fig. 1**. Such a testbed was first constructed in the former Intel Swindon lab, in Swindon, U.K. Note that RTT, as reported by tools like ping [21], is approximately 2 × one-way latency.

For reproducibility, the employed HPE server-based appliance, including network adapter settings, tuned Linux kernel parameters, and simulated latency values are described below. Related reference materials are publicly available in a GitHub repository [22].

Resources like the ESnet fasterdata knowledge base provide invaluable guidance. However, practitioners sometimes apply published tuning parameters as static prescriptions rather than adaptable guidelines. This approach can lead to suboptimal performance when it doesn't account for interactions between specific data mover software, hardware stacks, and application workloads. Effective high-performance data movement requires understanding and adaptation beyond copy and paste. Note that the results in **Fig. 1** and the modest parameter set in **Table III** demonstrate how an architectural approach also can reduce complexity and potential for misconfiguration.

B. *Universal Packet Loss and TCP Congestion Control Algorithms*

This section provides a systematic examination of several interconnected paradigms. We begin by quantifying the data volumes supported by common transfer rates, then introduce a conceptual model—the "Drainage Basin Pattern"—that frames the entire spectrum of data movement workflows. Subsequently, we present empirical evidence that challenges the conventional view regarding packet loss and the criticality of TCP congestion control algorithm selection in high-speed environments. A real-world case study, drawn from a Wall Street Journal report, illustrates the practical consequences of these paradigms. To contextualize the scale of these workflows, **Table V** quantifies the daily data volume achievable at common network speeds. Finally, we discuss why conflating different classes of workflows often leads to suboptimal outcomes.

The end-to-end movement of data is generally perceived as a transparent operation by application users and many stakeholders. Nevertheless, the reality is not so simple. The "Drainage Basin Pattern" (**Fig. 3**) illustrates the entire spectrum of data movement workflows.

While most users experience only the 'source of the river,' where consumer-grade networks may indeed have packet loss, the 'high-speed backbone'—the deep main river channel—is an entirely different environment. This work is based on extensive experience since 2015 with 100 Gbps and faster connections, gained through collaboration with the U.S. DOE ESnet and various tier-1 organizations. Consequently, our observations confirm that in the well-engineered 100 Gbps+ Research and Education (R&E) networks discussed here—principally ESnet and Internet2 [24]—packet loss is practically nonexistent during normal operation. This observed reliability is also a stated characteristic of peer R&E networks such as GÉANT [25], AARNet [26], and SingAREN [27].

*1) **The Packet Loss Paradigm**:* While most users experience only the 'source of the river,' where consumer-grade networks may indeed have packet loss, the 'high-speed backbone'—the deep main river channel—is an entirely different environment. This work is based on extensive experience since 2015 with 100 Gbps and faster connections, gained through collaboration with the U.S. DOE ESnet and various tier-1 organizations.

In fact, the nature of data transmission at high speeds differs fundamentally in both engineering and measurement

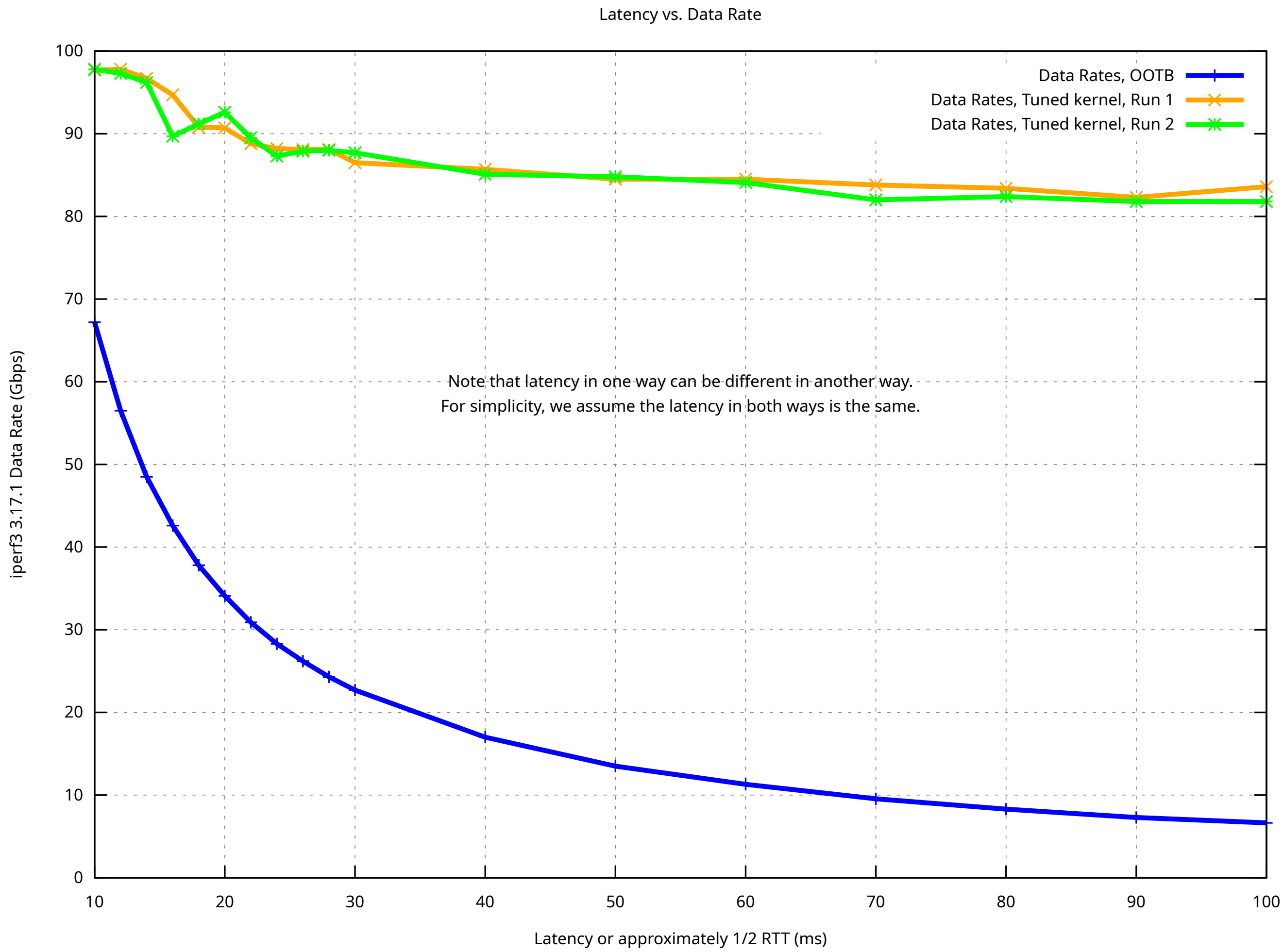


Fig. 1. **iperf3** latency sweep results obtained using two HPE DL380 Gen 11 server-based appliances (**FIG. 2**) on a latency simulation-capable testbed (max network speed 100 Gbps) established in the former Intel Swindon Lab, Swindon, U.K., in 2024. While default kernel networking settings (OOTB) show severe performance degradation under high latency, proper kernel tuning substantially reduces this penalty. See **Table II** for the Ethernet adapter's ring buffer settings; **Table III** for the settings of "Tuned kernel."

TABLE I
KEY FUNCTIONAL CAPABILITIES AND CORE FEATURES OF THE ZETTAR ZX DATA MOVEMENT SOFTWARE.

Capability	**Notes**
Support bulk and streaming transfers	Utilizes TCP for transport
Sustained performance independent of	Encryption, compression, latency and file sizes
Cluster Architecture with Linear Scaling	
Single Application Supports	File, object, locally and over distance, on-prem and cloud
Symmetric Concurrent Send/Receive	rsync and bbcp alike
Embeddable on NVIDIA and Intel DPUs	
Built-in QoS Support	
Integration with typical IAM	E.g., Active Directory. Also, OpenID Connect (OIDC) support

TABLE II
RX AND TX RING BUFFER VALUES FOR INTEL® ETHERNET NETWORK ADAPTER E810-2CQDA2.

Parameter	**Value**
rx_value	8160
tx_value	8160

from casual, end-user level data transfers. For high-speed networks, "packets" often vary in size. For example, for nodes connected to 10 Gbps and faster networks, it is common to employ "jumbo frame" Maximum Transmission Unit (MTU) of **9,014** or **9,018** bytes, rather than a constant 1,500-byte frame. Consequently, high-speed network device vendors typically specify performance using Bit Error Rates (BER) [28] rather than packet loss. While operators historically reported BERs more explicitly, modern practitioners typically must navigate

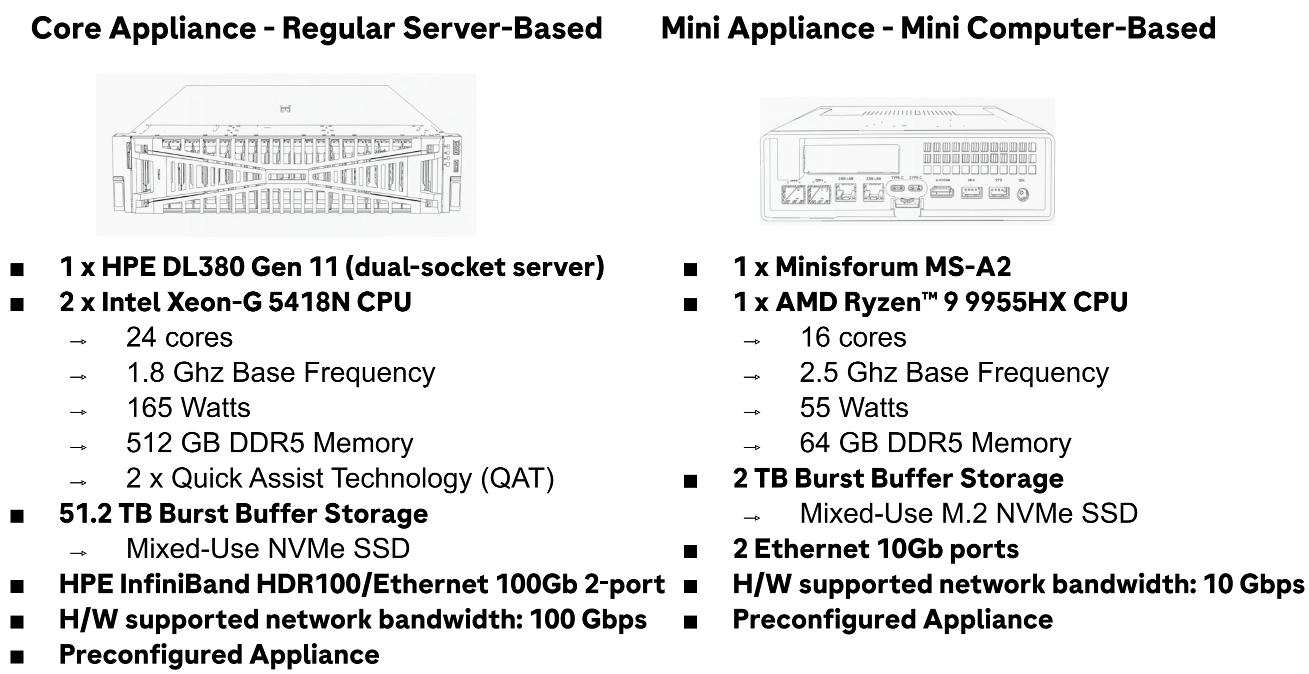


Fig. 2. Bill of Materials (BOM) and component details for the Core (HPE DL380 Gen 11) and Mini (Minisforum MS A2) appliances, demonstrating the vendor- and form-factor-agnostic unified data movement appliance design which achieves consistent performance across implementations.

TABLE III
Tuned Linux kernel parameters. This specific set was empirically determined to be optimal for the zx data mover's concurrency model, whereas other data movers may require more extensive tuning. For example, see ESnet fasterdata, Linux Tuning [23].

Parameter	Value
net.core.rmem_max	2147483647
net.core.wmem_max	2147483647
net.ipv4.tcp_rmem	4096 67108864 1073741824
net.ipv4.tcp_wmem	4096 67108864 1073741824
net.ipv4.tcp_mtu_probing	1
net.core.default_qdisc	fq_codel
net.ipv4.tcp_congestion_control	cubic
net.core.netdev_max_backlog	8192

TABLE IV
Some common latency ranges. 10 ms, 50 ms, and 100 ms were used for latency simulation.

Distance category	Latency range (ms)
Metropolitan Distance (same city or nearby cities)	1 – 10
Interstate Distance (within the same country)	10 – 50
Cross Continent	50 – 100

TABLE V
Daily data volume achievable at three common network speeds.

Data rate (Gbps)	Data moved (TB)/day	Notes
1	10	5G speed $\leq$ 0.5 Gbps
10	100	
100	1000	aka 1 PetaByte

complex monitoring dashboards such as the perfSONAR lookup service [29] to access such information.

To place the current argument in context, we cite a 2009 ESnet reference to show the historical lower bounds of key performance metrics. Subsequently, we present a practical 2018 case study to demonstrate that packet loss is not the dominant

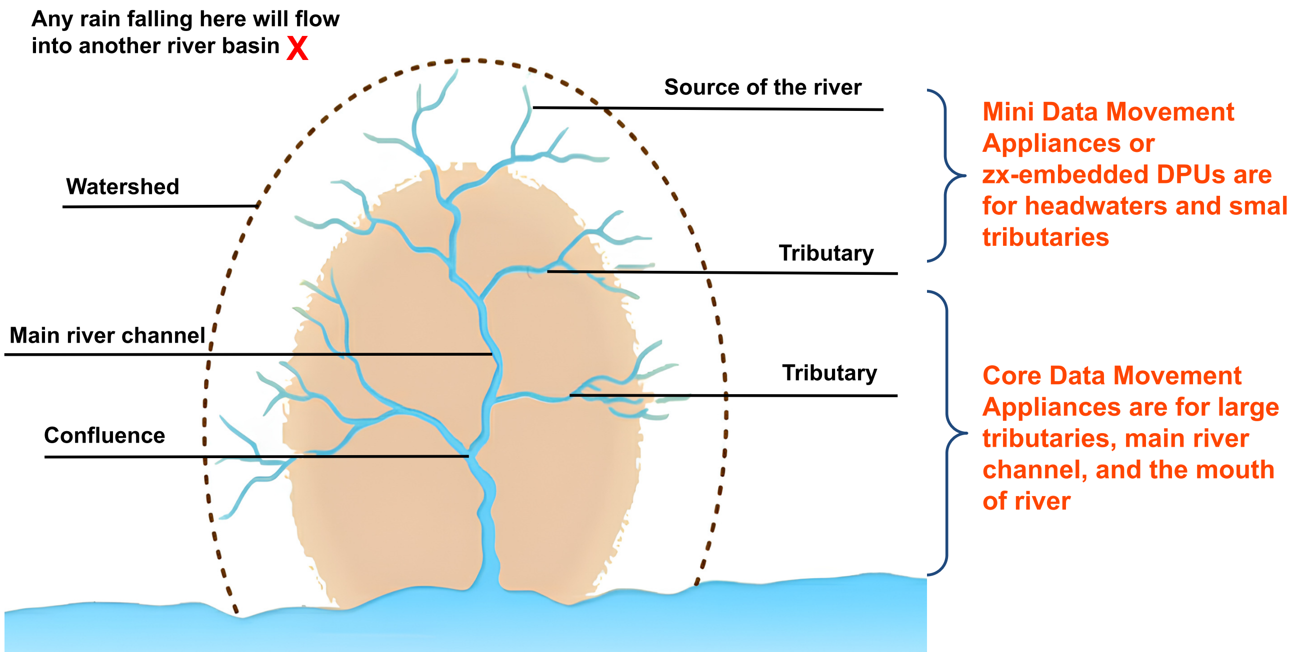


Fig. 3. The experience of moving data for most practitioners is typically limited to the "source of the river," which corresponds to end-user activities, such as transferring photos and videos from mobile phones and moving spreadsheets from one folder to another. At such low data rates, operations appear simple. This may explain why efficient data transfer has historically received limited attention.

constraint for high-volume, high-speed data transfer. ESnet's Services and Service Level Descriptions (SLD) (**Appendix A** in [30]) specifies guaranteed network performance via its **Loss Thresholds** Table. This table typically lists a Frame Loss Rate (FLR) range of 10^{-7} to -10^{-10} and includes a formula for converting the FLR to BER. Furthermore, an accompanying ESnet Fasterdata Knowledge Base (KB) article on Packet Loss provides an approximate translation of BER to end-to-end packet loss, citing a typical scenario of 1 packet out of 22,000 packets, or 0.0046%. Converting this packet loss ratio to a BER illustrates the exceptional quality of these networks:

- Packet Loss Rate = 0.0046% (4.6×10^{-5})
- Assuming a standard 1500-byte frame: BER $\approx (4.6 \times 10^{-5})/(1500 \text{ bytes/frame} \times 8 \text{ bits/byte}) \approx 4 \times 10^{-9}$

This result—a BER of 4×10^{-9}, meaning roughly 4 erroneous bits in every billion—confirms that packet loss is negligible in well-engineered backbones.

This theoretical upper bound of network quality is corroborated by practical, large-scale experience. In September 2018, using the setup shown on (page 19, **Appendix 6.1** of [8]), a production trial run was carried out for **1 PB transfer** in **29 hours** over a 5,000-mile 100 Gbps WAN loop. The setup was designed and implemented at SLAC National Accelerator Laboratory, using equipment that was, at the time of deployment, three years past its initial deployment. The trial was reported in [4]. Of the hardware-limited **80 Gbps** bandwidth, with full encryption and checksumming, an average utilization of **76.63 Gbps** was achieved. This production-scale result provides practical evidence that packet loss did not constrain throughput.

*2) **The TCP Congestion Control Algorithm Paradigm**:* The empirical evidence presented previously established the irrelevance of packet loss on well-engineered network backbones. Next, we examine the related paradigm that the choice of TCP Congestion Control Algorithm (CCA) is critical for performance. The prevalence of this view is evident from the abstract of Google Networking Research's publication "BBR: Congestion-Based Congestion Control" [31], which states: "Physics and climate researchers need to exchange petabytes of data with global collaborators but find their carefully engineered multi-Gbps infrastructure often delivers at only a few Mbps over intercontinental distances."

It is a telling coincidence that during the very period this paradigm was being articulated, a production reality was demonstrated. ESnet publicly reported our 1 PB transfer, which achieved **76.63 Gbps** average utilization [4]. This result was accomplished not with a novel CCA, but with the default CUBIC algorithm in the standard CentOS 7.4 distribution, which did not yet include BBR. This demonstrates that the fundamental bottleneck was not the congestion control algorithm, but the lack of a holistically co-designed system.

Before proceeding, we establish the context regarding high-speed TCP congestion control algorithms (CCAs) by citing relevant Internet Engineering Task Force (IETF) Request for Comments (RFCs) and referring to prior work [4]. Starting in 2018, the suitability of CUBIC for Fast Long-Distance Networks was already addressed by RFC8312 [32]. Coincidentally, earlier work [4] described a production trial conducted on a

5000-mile, 100 Gbps loop, achieving over 96% bandwidth utilization—practically, line rate. The systems used in this trial ran CentOS 7.5-1804, which utilized CUBIC as its default CCA. We note that RFC9438 [33], titled "CUBIC for Fast and Long-Distance Networks", has since obsoleted RFC8312. More recently, RHEL 9.6 and its free rebuilds now include BBR as the default CCA, as demonstrated by the simple command output presented below:

```
$ sysctl net.ipv4.tcp_available_congestion_control
net.ipv4.tcp_available_congestion_control = reno cubic bbr
```

It is noteworthy that while Google researchers introduced the model-based BBR CCA in 2018—the same year CUBIC was already the default in widely used operating systems like CentOS 7—BBR has since progressed to version 3 [34]. However, the BBR implementation that ships with RHEL 9.6 remains version 1, which is notably more aggressive than later versions. Here, we compared the performance of BBR, CUBIC, and Reno (a classical TCP CCA algorithm emerged in the 1990s) and provided the results in **Figs 5-7**. The following definitions and methodology are established before presenting the three figures.

- Hyperscale Data Set**:** A synthetic test data set containing files of a uniform size, where the total number of files is 2^{20} (1,048,576) or the aggregate size is 1 TiB, or both.
- Data transfer sweep:
  - Bulk Transfer Sweep: File sizes range from 1 KiB to 1 TiB, with sizes incremented by powers of two, resulting in 31 distinct datasets.
  - Streaming transfer: File size ranges from 4 MiB to 1 TiB; sizes are incremented in power of two, so there are 19 datasets

  Each sweep iteration processed datasets sequentially from the smallest to the largest file size. This process was repeated for multiple iterations to gather statistical significance.

Note that statistical analysis for each complete sweep (bulk and streaming) included the calculation of the mean, median, and standard deviation. The mean values are plotted automatically using a custom gnuplot command file [35] the three figures.

Based on the demonstrated functional equivalence of high-speed performance, we conclude that the choice among competing high-speed CCAs is not a primary concern for high-volume, high-throughput data transfer. CUBIC is empirically validated, using the appliances in **Fig. 2**, as a safe and effective default.

It is worth noting that Reno, characterized by slower window size adjustments than both CUBIC and BBRv1, is highly susceptible to the non-offloaded userspace overheads of TLS. The implementation of kTLS effectively mitigates this susceptibility, allowing Reno to achieve throughput parity with the other CCAs by removing the userspace processing bottleneck.

*3) **Practical Implications and Real-World Consequences**:* The engineering realities discussed thus far have significant, though often invisible, operational and economic impacts. The tangible consequences of suboptimal data transfer are starkly illustrated by recent extreme measures reported by the Wall Street Journal (WSJ) [37]. Given the exponential growth in scientific data, the mobility of multiple petabytes (PBs) is now a common requirement.

A June 12th article [37] titled "Chinese AI Companies Dodge U.S. Chip Curbs by Flying Suitcases of Hard Drives Abroad" detailed how four engineers transported 4.8 PB of training data by flying sixty hard drives from Beijing to Kuala Lumpur. This volume of data, sufficient for training large language models (LLMs) such as OpenAI's ChatGPT, highlights the severe real-world constraints imposed by inefficient electronic data transfer.

A related article from Tom's Hardware [38] clarified that this physical transport was a "meticulously planned operation and took several months of preparation." In sharp contrast, the performance results illustrated in **Figs. 4-6**, combined with the entries in **Table V**, demonstrate that transferring this volume of data electronically would require *at most one week*.

This dichotomy recalls the hierarchy of solutions presented in **Figs. 2-3**, whose importance can be illustrated by a simple metaphor: providing water. Watering plants on a balcony requires a watering can. Watering a garden demands a hose. However, supplying a metropolitan area from a reservoir necessitates a massively engineered system like the Hetch Hetchy Aqueduct [39].

This hierarchy directly parallels data movement. Transferring data at rates of ~1 Gbps is akin to using a watering can. Rates of ~10 Gbps correspond to the garden hose. But when approaching 100 Gbps and beyond, the endeavor demands the engineering equivalence of a metropolitan water transport system—a holistic, co-designed infrastructure where ad-hoc solutions fail.

C. Dedicated Private Lines Are Essential for High-Speed Testing

A common paradigm asserts that validating high-speed data transfer requires a dedicated, high-bandwidth WAN link. This belief creates a significant barrier, as such links are costly and complex to provision, requiring specialists to operate well. We confronted this directly in 2023 while developing data movement appliances with Intel Corp. Neither organization had a 100 Gbps WAN, raising a critical question: how could we rigorously verify performance without a production-grade link?

From 2015-2019, the `zx` R&D was carried out on a testbed at SLAC National Accelerator Laboratory, using a 5,000-mile, 100 Gbps loop provisioned by ESnet. After 2019, collaboration with ESnet continued [8]. Nevertheless, accessing their 100G SDN Testbed required a formal application process [40], hindering agile development. This need for a fully controlled, on-demand test environment led us to evaluate commercial WAN simulators, which we found to be prohibitively expensive and functionally limited.

A turning point was the discovery of a 2012 technical report, "Validating Linux Network Emulation" [41], which outlined a method using Linux's built-in traffic control (`tc` [42]) and network emulation (`tc-netem` [43]) tools. Recognizing the potential of this software-based approach to create a high-fidelity, 100 Gbps-capable testbed at a fraction of the cost,

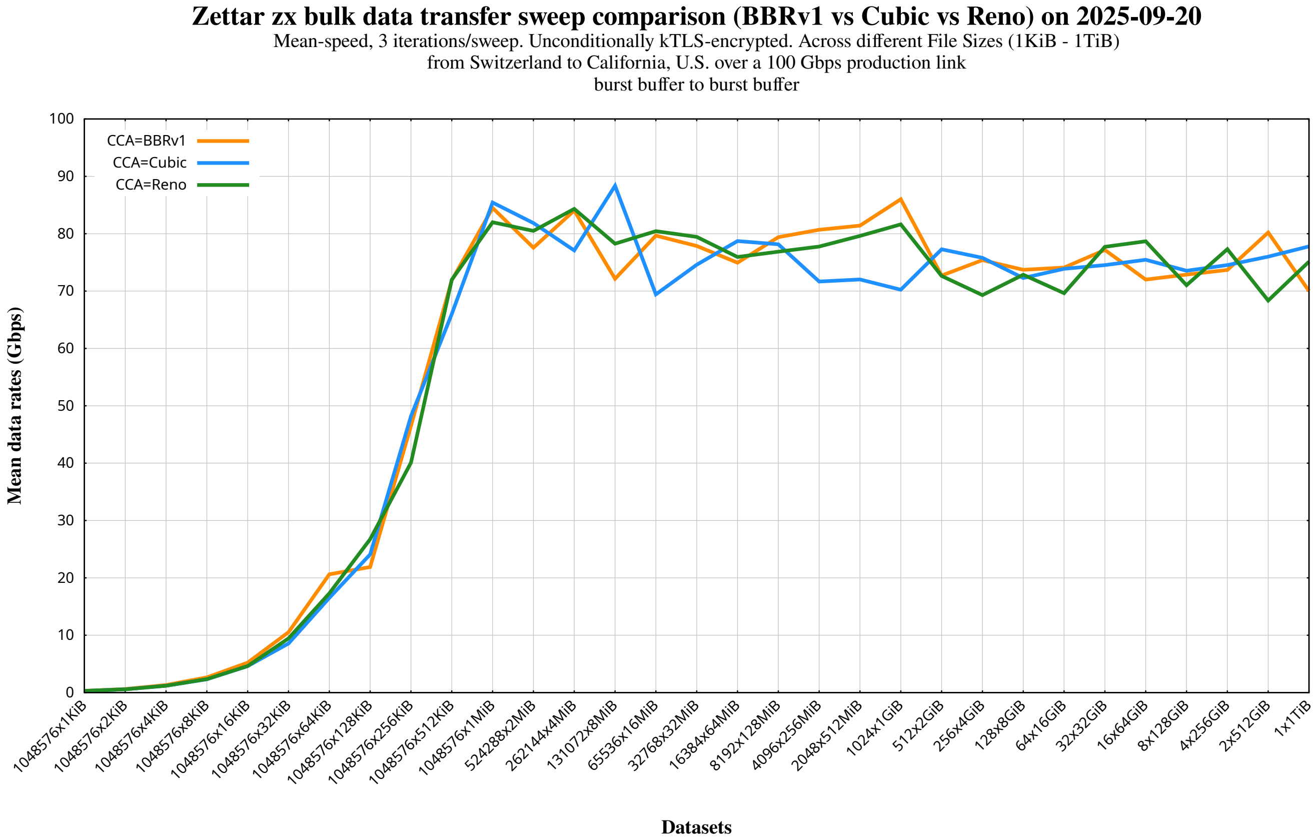


Fig. 4. A bulk transfer sweep leveraging kTLS offload in RHEL 9.6 [36] to evaluate the three default congestion control algorithms (CCAs). BBRv1 did not demonstrate a clear performance benefit over CUBIC or Reno. Incidentally, kTLS did not yield any transfer rate improvement in this case.

we implemented an enhanced version. The resulting testbed (**Fig. 7**) —built at the former Intel Swindon Lab in collaboration with Intel and HPE—consists of two HPE DL380 Gen 11 servers (configured as the appliances under test) connected via a switch to two Intel "latency servers." These servers use `tc` and `tc-netem` to impose precise delay, accurately simulating transcontinental links in an automated, remotely manageable setup.

The results, shown in **Fig. 8** (bulk transfers) and **Fig. 9** (streaming transfers) demonstrate the testbed's capability to sustain high data rates under simulated latencies of **10ms**, **50ms**, and **100ms**. The critical validation comes from comparing these results with those from the production 100 Gbps link (**Fig. 5** & **6**). The achieved data rates and performance profiles are strikingly similar, proving the emulated testbed's predictive accuracy.

The true significance of this Linux-based emulation approach is that it transforms the high-speed development environment from a logistical burden into a strategic, versatile engineering platform. This approach provides a compact, low-TCO platform capable of accurate performance prediction across numerous critical functions:

- **Systems Focus:** Unlike commercial WAN simulators or network testbeds that isolate network performance (often using RAM-to-RAM methods), this platform is uniquely suited to measure true burst buffer-to-burst buffer data transfer rates. This is essential because, as established in Section 5.1, persistent I/O is the dominant constraint for petascale transfers.
- **Agile Development:** The platform supports rapid software iteration through automated regression testing for CI/CD and accelerated development of new components, such as AI Model Context Protocol (MCP) with guardrails for connecting `zx` to a natural language input for improved usability, without risking a live production network.
- **Operational Utility:** Its reproducibility enables critical regulation validation for compliance-driven sectors, supports WAN planning through latency simulation, facilitates training/education for new systems engineers, and provides an efficient demo showcase environment.

While this method validates core performance metrics under controlled conditions, it is important to note that it does not replace the necessity of an active SDN testbed for validating complex control-plane, security, or multi-flow operational scenarios typical of a fully instrumented network like ES-net. However, for core high-throughput systems performance engineering, the software-defined WAN emulation offers a cost-effective, high-fidelity alternative.

The full testbed benefits and implementation details, together with expanding it to *speed* $\geq$ 100 Gbps, warrant a separate, in-depth tutorial. See also **8 Conclusions**.

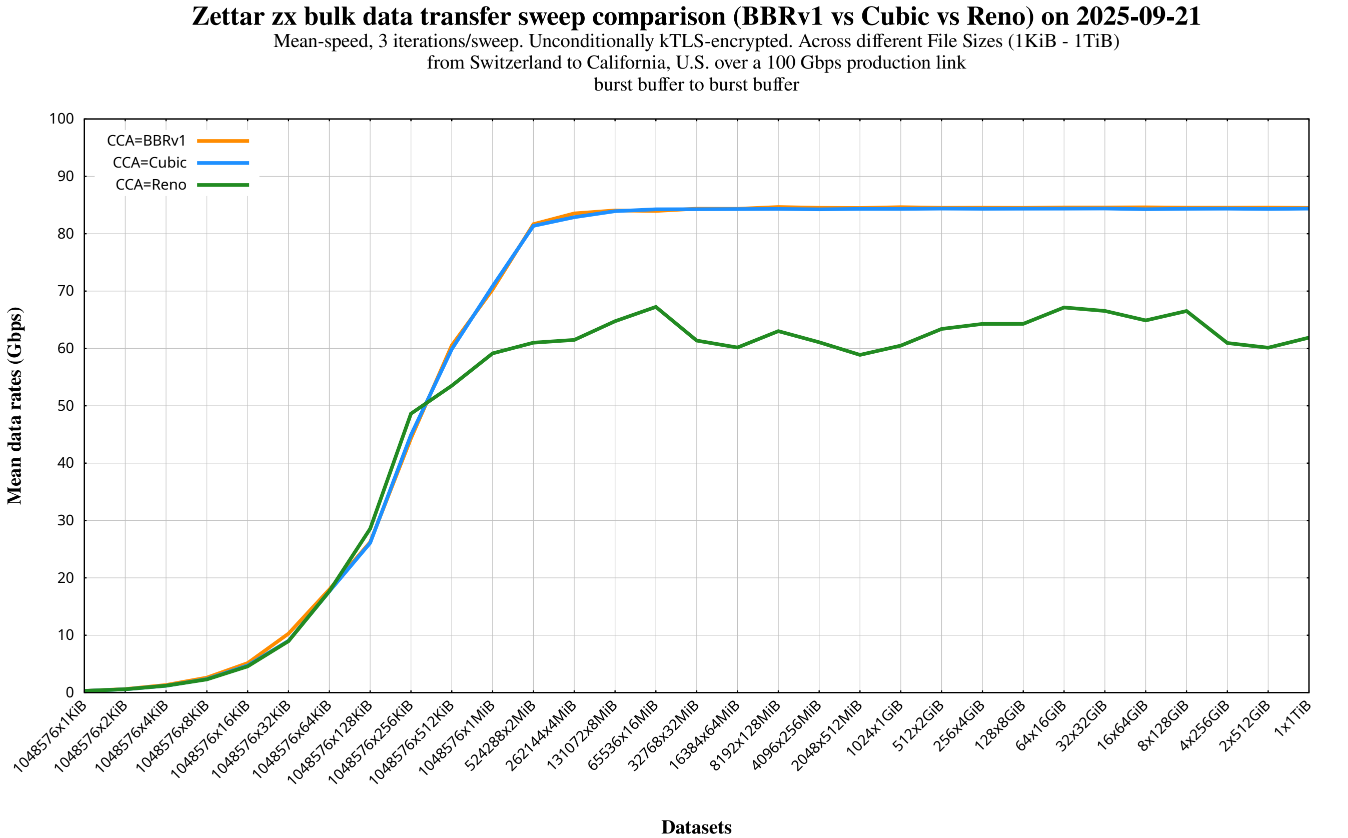


Fig. 5. Subsequent bulk transfer sweeps were performed without kTLS due to the prior degradation. Using the default RHEL 9.6 CCAs, BBRv1 and CUBIC exhibited identical throughput across 4 MiB–1 TiB. Reno showed degradation consistent with older congestion control paradigms.

D. Increasing network bandwidth increases transfer rates

In essence, this paradigm fails because the rest of the infrastructure—primarily storage and compute, combined with the parallelism and concurrency of the software data mover—must be in balance with the available network bandwidth. A chain is only as strong as its weakest link; once the network ceases to be that link, introducing more bandwidth yields no benefit.

Foremost among these limiting factors, and often underestimated, is storage I/O. Once network bandwidth exceeds the system's ability to read from or write to persistent storage, further increases are futile. The bottleneck most frequently occurs during write operations, as virtually all storage media deliver lower write than read performance.

Other system-level factors also contribute: the processing of TCP/IP stacks, filesystem metadata operations, and encryption/checksumming all consume finite CPU cycles that could otherwise be allocated to sustaining storage I/O. As indicated in **Fig. 10**, to attain high and sustainable data transfer performance, the storage service must meet two criteria: 1) high enough storage IOPs/throughput 2) low latency. Both are elaborated below.

In **Fig. 10**, the term “high enough” means the overall aggregated storage IOPs/throughput > the target transfer rate. “Low latency” is particularly crucial because, for highly concurrent data movers, storage latency effectively sabotages concurrency—a scenario often associated with random I/O. What appears as a software bottleneck is frequently a storage latency problem in disguise.

Furthermore, as shown in **Fig. 1**, without a system (including kernel tuning) holistically co-designed to match the data mover's profile, network latency will cripple performance regardless of the raw bandwidth available.

The preceding analysis and evidence support a re-evaluation of this paradigm.

E. Powerful CPUs are essential for high transfer rates

A common consensus in high-performance computing is that powerful CPUs are essential for high transfer rates. During 2023, while discussing with our industry collaborators regarding CPU selection for data movement appliances, the prevailing feedback was to select high-end, high-core-count models. We held reservations about this approach. Furthermore, we also had skepticism about suggestions to utilize CPUs with embedded hardware acceleration such as Quick Assist Technology (QAT) [44], based on the observation that such acceleration can tie software to specific vendor hardware and drivers, increasing complexity and potential "software bloat."

This skepticism was grounded in years of deployment experience, which demonstrated that `zx` consistently achieves high performance without requiring a high core count; typically, 12-24 cores are sufficient. While resources like ESnet's fasterdata

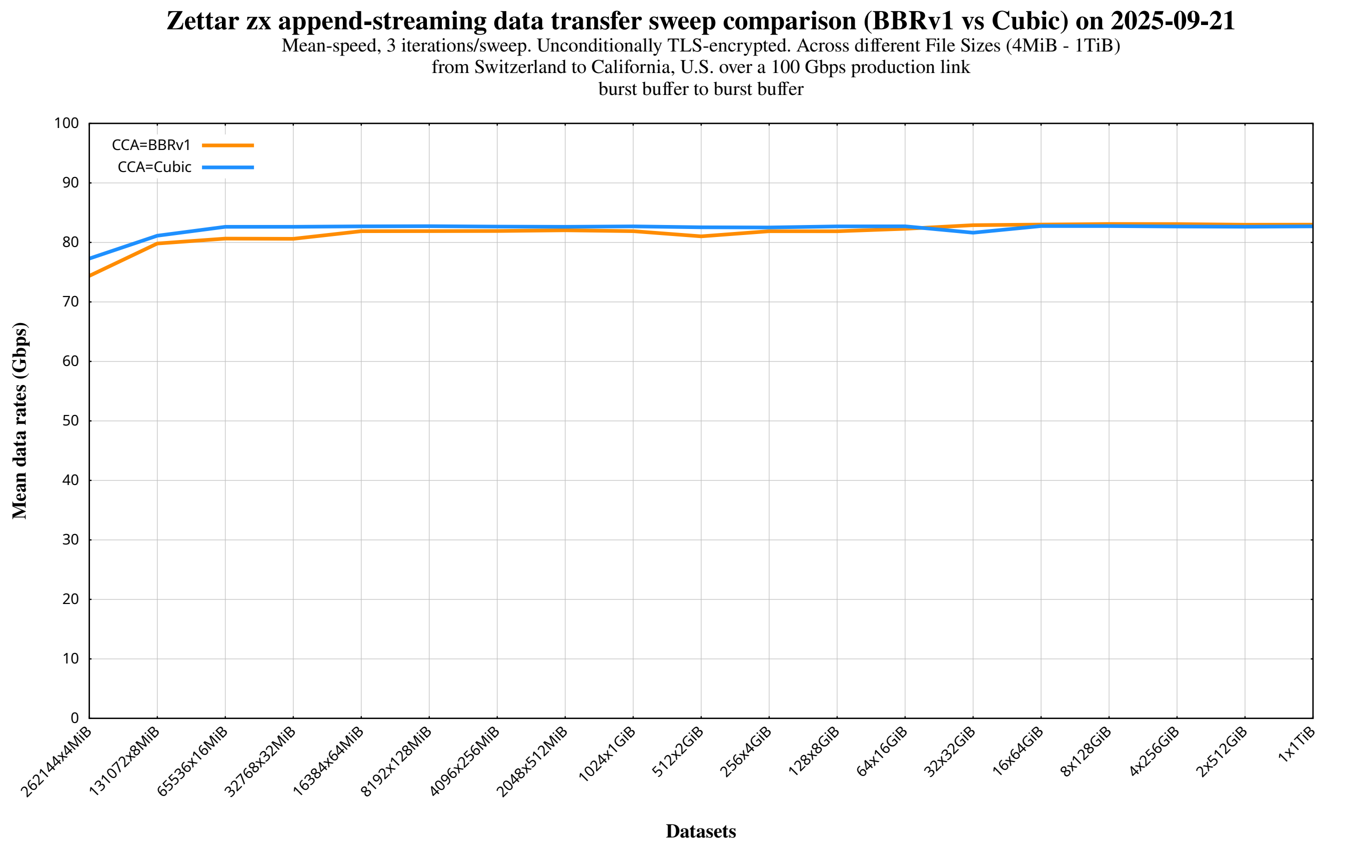
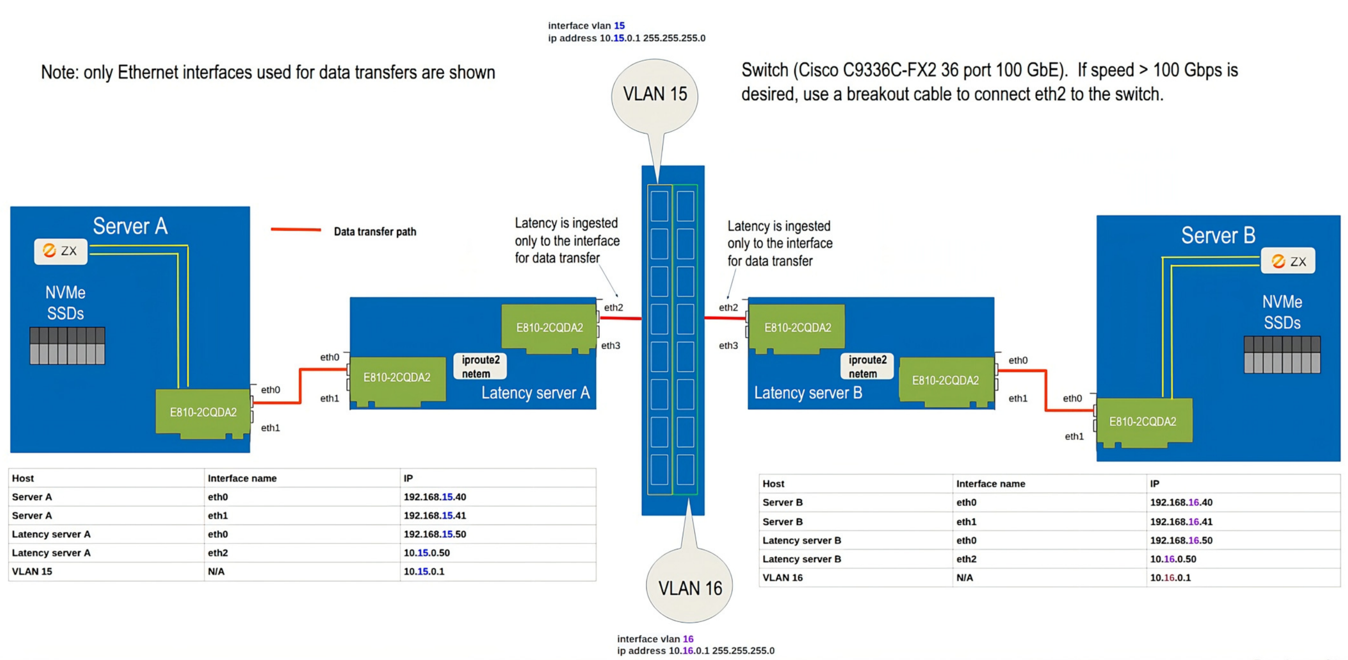


Fig. 6. As a unified software data mover [13], zx has built-in streaming capability, motivated by [2]. BBRv1 shows no advantage over CUBIC, even for streaming transfers. kTLS was not used.



Host	Interface name	IP
Server A	eth0	192.168.15.40
Server A	eth1	192.168.15.41
Latency server A	eth0	192.168.15.50
Latency server A	eth2	10.15.0.50
VLAN 15	N/A	10.15.0.1

Host	Interface name	IP
Server B	eth0	192.168.16.40
Server B	eth1	192.168.16.41
Latency server B	eth0	192.168.16.50
Latency server B	eth2	10.16.0.50
VLAN 16	N/A	10.16.0.1

Fig. 7. Intel Corp. arranged its former Swindon Lab in Swindon, U.K., to collaborate with Zettar using an enhanced version of the approach created by Dr. Ezra Kissel. The essential components are labeled explicitly.

recommend high clock rates forilllgood encryption performance [45], our observations suggested an alternative path. We hypothesized that fewer cores with CPU-built-in encryption acceleration and moderate clock rates would reduce context switches (improving software efficiency) and lower energy consumption, thereby reducing the total cost of ownership.

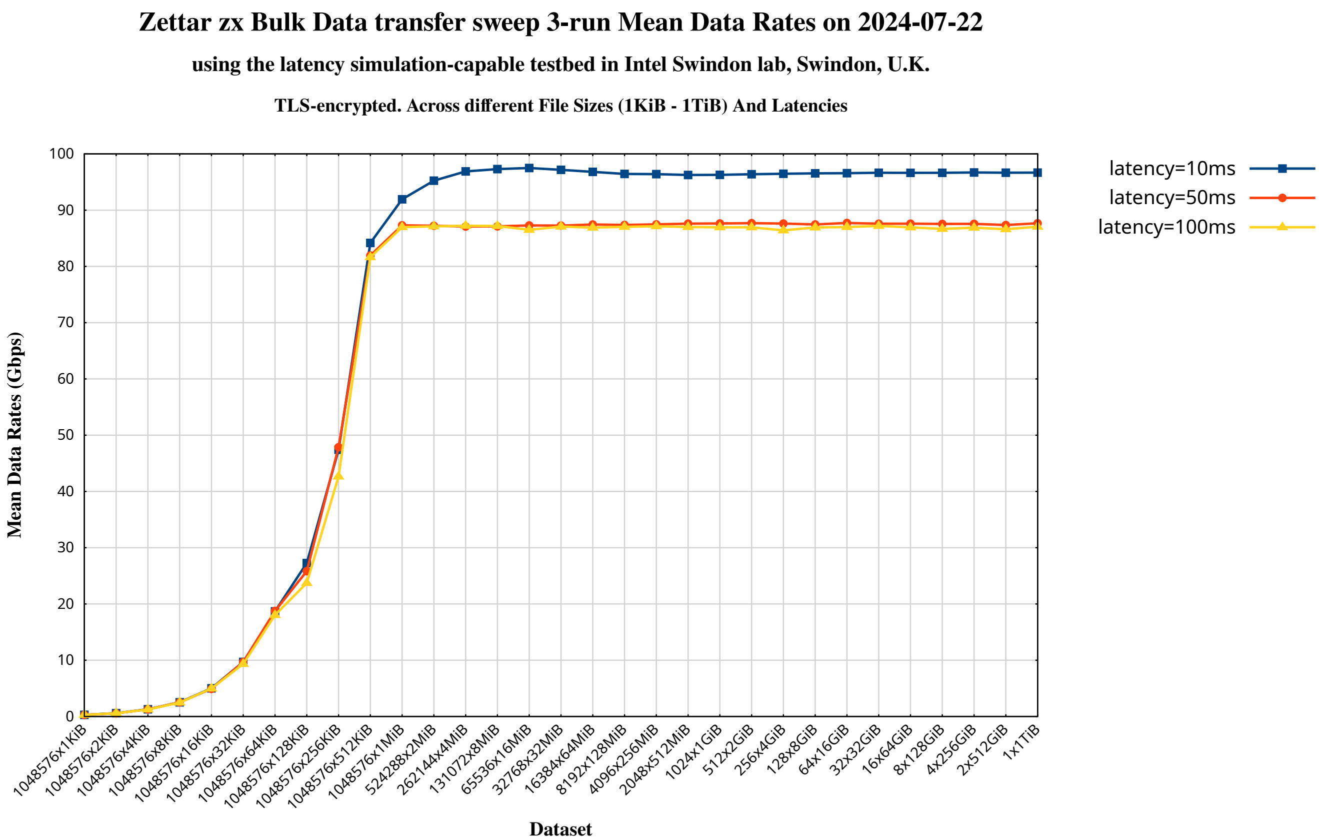


Fig. 8. Bulk transfer sweeps vs three simulated latency values: 10 ms, 50 ms, and 100 ms, corresponding respectively to Los Angeles to San Francisco, California, U.S.; Singapore to Alaska, U.S.; and Singapore to Atlanta, Georgia, U.S.

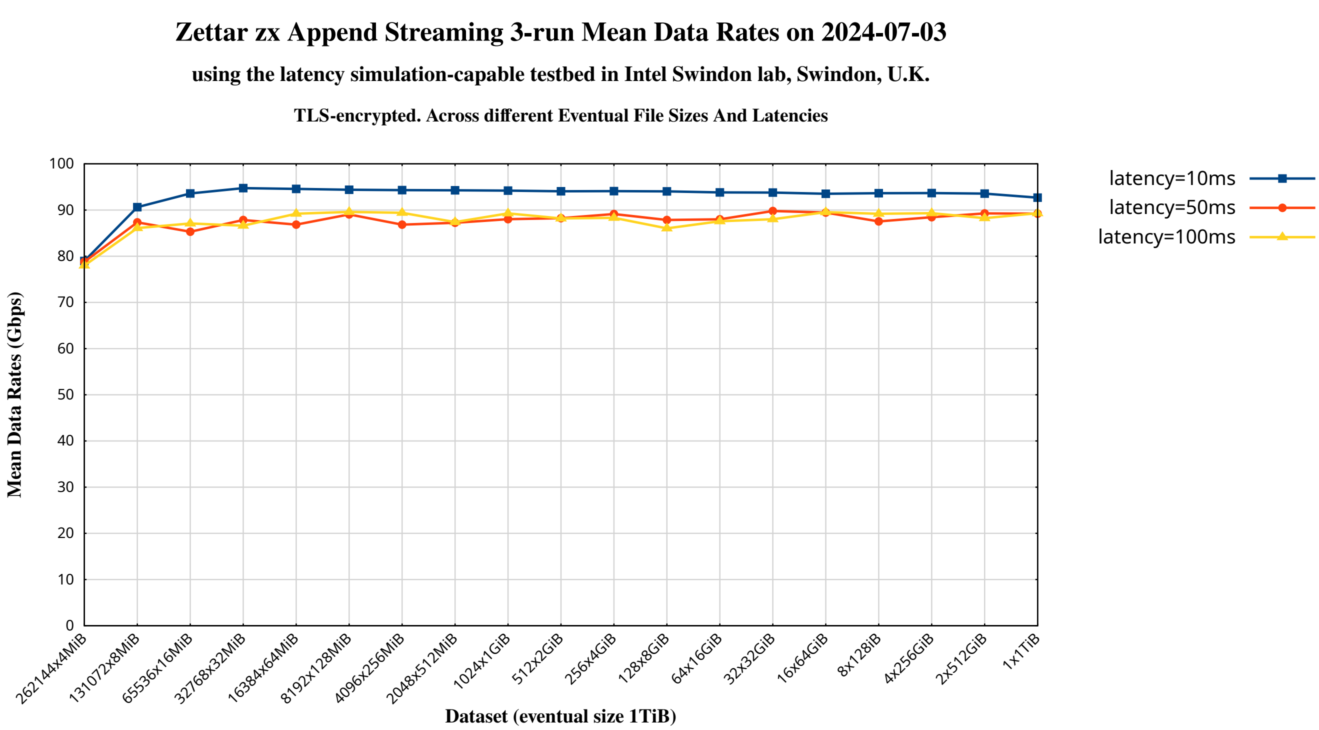


Fig. 9. Streaming transfer sweeps vs three simulated latency values, 10ms, 50ms, and 100ms. Note that the data rate levels are quite close to their counterparts from the bulk transfer sweeps shown in **Fig. 6**.

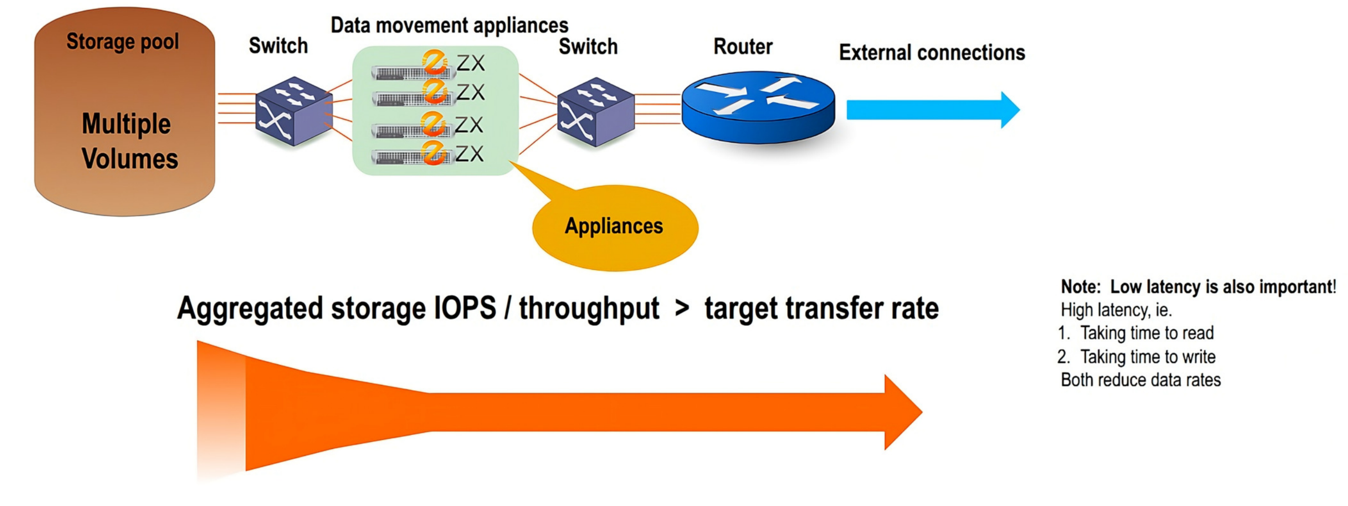


Fig. 10. Illustrates the essence discussed so far.

This led to the selection of the Intel Xeon 5418N, a mid-range model [46].

In 2024, more extensive testing with the two HPE DL380 Gen11 server-based appliances confirmed that even with full encryption, QAT was unnecessary. The CPU's native instruction set extensions were sufficient. The results shown in **Figs. 4-6**, **8-9** were all achieved without hardware acceleration and using this modest CPU.

As such, we can conclude that CPU raw computing power matters, but software efficiency and storage architecture matter more. The right software makes adequate CPUs perform exceptionally; the wrong software cannot be saved by the most powerful CPUs.

F. Virtualization and cloud are universally useful

We recently received an inquiry about evaluating `zx` for a lab in South Africa needing to transfer data to Europe at higher rates. The lab's proposal to use a virtual machine (VM) immediately raised a red flag, illustrating a fundamental mismatch between virtualized environments and data movement performance.

Virtualization technologies have been a mainstay for over two decades [47]. A critical determinant of their performance for data-intensive workloads is the level of system control accessible to the user, summarized in **Table VI**.

This hierarchy of control directly leads to four critical performance implications for high-speed data transfer:

1. **Interrupt Overhead:** Even with SR-IOV [48], interrupt handling in VMs incurs significant overhead due to interception and remapping by the hypervisor.
2. **I/O Performance Degradation:** High-speed I/O operations, which generate a high volume of interrupts, consequently achieve lower performance in virtualized environments.
3. **Host-Level Dependency:** If the host kernel is not tuned to mitigate latency (as established in Section 5.1), adjustments within the guest OS are largely ineffective.

Therefore, achieving consistent, latency-insensitive data transfer performance within a VM is intrinsically constrained by the architecture. Cloud providers, as massive consumers of virtualization, inherit these limitations. Coordinating the kernel parameter adjustments necessary for maximum VM performance is application-dependent (see **Table III**) and practically infeasible for a heterogeneous customer base.

*1) **The Architectural Cost of Cloud Abstraction**:* While virtualization and cloud abstraction deliver significant benefits in manageability and elasticity, they impose substantial performance penalties on data movement. The canonical cloud data path—encumbered by layers of overheads, such as hypervisor, virtualized storage, and HTTP/REST APIs—incurs a substantial performance inefficiency compared to a tuned bare-metal environment. This inefficiency is empirically observed in our tests to routinely around 30-50%. For instance, on 10 Gbps EC2 instances in 2019, we achieved a maximum of ~6 Gbps per VM, a 40% loss from line rate. More recently, as detailed in **Fig. 11**, the native cloud tooling (`aws-cli`) performed at a small fraction of the available capacity, while our co-designed appliance, even when partially firewalled, consistently attained a significantly higher percentage of the physical link's potential.

The standard cloud approach to mitigating this inherent limitation of the current abstraction model often introduces significant architectural complexity. Workarounds such as AWS multipart uploads [49] transform a conceptually simple file transfer into a distributed coordination problem. What should be a single, robust streaming operation is fragmented into a multi-step process of part uploads, ETag collections, and finalization requests—effectively creating a Rube Goldberg machine [50] for data movement, which is then presented as a necessary feature.

*2) **A Critical Examination of Cloud Provider Metrics**:* The prevailing paradigm of general-purpose cloud data paths, regardless of provider, introduces inherent architectural conflicts for high-performance data movement workflows. This is a universal architectural challenge that becomes acute at

TABLE VI
LINUX KERNEL PARAMETER SCOPE: VIRTUAL MACHINE VS. CONTAINER.

Setting location	Target Platform	Scope of sysctrl Change
Inside VM (Guest OS)	Virtual Machine (e.g., KVM)	Affects *only* the Guest OS kernel
Inside Container (default)	Container (Docker/Podman)	Affects only a small, namespaced subset of settings for the container. Cannot change host-level settings.
Inside Container (privileged)	Container (Docker/Podman)	Can change host-level settings, affecting both the host and all running containers.
On Host	VM Host or Container Host	Affects the underlying physical server and, in the case of containers, the default for all running containers.

petascale. An analysis of the three most established U.S. cloud provider communications reveals a pronounced emphasis on theoretical network bandwidth figures, often decoupled from the systemic limitations of storage I/O and end-to-end data path performance. The following tables, compiled from official provider documentation, demonstrate this focus on network potential, which frequently exceeds the sustainable throughput capabilities of attached storage subsystems.

a) The Disconnect Between Theoretical Bandwidth and Practical Throughput: Providers often highlight maximum network bandwidth figures—such as 1,000 Gbps or 400 Gbps—that are theoretically impressive but often unrealizable in practice. This focus on network potential overlooks a critical constraint: if the storage subsystem cannot match this throughput, the data path cannot be fully utilized. This inherent limitation exemplifies the systemic disconnect that our co-design principle seeks to resolve, where individual component specifications are prioritized over end-to-end workflow performance.

Google Cloud Platform (GCP):
Amazon Web Services (AWS):
Microsoft Azure (Azure):

b) Focusing only on computing, ignoring data availability, and I/O: Premises are evident:

- Without data, compute is useless.
- Data is predominantly produced at the edge (e.g., LCLS-II [2]).
- Efficiently moving this data to the cloud remains unsolved.
- Therefore, regardless of computational speed, data transport imposes a fundamental and protracted delay.
- Ignoring this latency [37, 38] deserves scrutiny.

c) 5.6.2.3 Lack of reproducibility: The results presented-primarily in marketing form and omit key engineering details required for reproducibility. This omission of specifics on storage, network, and system configuration prevents rigorous validation

*3) **The Antidote: A Co-Designed Data Path for the Cloud**:* The solution to this performance limitation is not to work within the constraints of the generalized API stack, but to architect a parallel data path that coexists with it. By embedding the `zx` data mover directly in DPUs [51][52], we create a high-performance data path that operates alongside—yet architecturally bypasses—the cloud's general-purpose software stacks. `zx` has a tiny footprint, depends only on the standard libraries in a RHEL Minimal Install, and requires no DPU vendor firmware or SDKs. This approach provides:

- **Native Transfer Semantics:** Direct file-to-object transfer, avoiding complex HTTP chunking.
- **Line-Rate Performance:** Latency-insensitive, encrypted movement at full line rate.
- **Co-existence:** Crucially, this requires zero changes to existing REST APIs, enabling a fundamental performance improvement without disrupting existing services.

This architecture does more than just mitigate the cloud's data movement penalties; it has the potential to enable cloud providers to evolve their role by equipping them with true, high-performance data mobility as a fundamental infrastructure capability, enabling previously impossible workflows and establishing an efficient, dynamic data circulation system.

*4) **Empirical Evidence of the Cloud Data Movement Limitations**:* The performance penalties imposed by typical cloud data paths are measurable. Using the GoToCloud platform on AWS [53], a comparative analysis conducted by KEK (High Energy Accelerator Research Organization, Japan) [54] vividly illustrates this point. As summarized in **Fig. 11**, the transfer of a 1.2 TiB Cryo-EM dataset was measured using the `zx` data mover versus the native aws-cli tool [55].

The results:

- Over a **63 km** distance (KEK to AWS Tokyo Region), `zx` completed the transfer in **22.66 minutes**.
- Over a **10,851 km** trans-Pacific distance (KEK to AWS Northern Virginia Region), `zx` completed the transfer in **39.97 minutes**. The increase in latency and distance resulted in a predictable and manageable **1.76x increase in transfer time**.

This performance differential is notable given the 10 Gbps network connection at KEK's CryoEM lab. While partially constrained by a site firewall, `zx` utilized this bandwidth effectively. In comparison, the `aws-cli` transfer to AWS Northern Virginia required 235.18 minutes, operating at a substantially lower effective throughput.

These measurements suggest two technical observations:

1. **The Cloud's Self-Imposed Bottleneck:** The drastic underperformance of the cloud provider's own tool (`aws-cli`) on its own infrastructure highlights that the bottleneck is not the physical network, but the inefficient software stack and lack of end-to-end control afforded to the cloud user. The environment is a "black box" that prevents effective optimization.
2. **The Architectural Solution:** The performance of `zx` demonstrates that long-distance transfers can and should be latency-insensitive. The fact that a trans-Pacific transfer

TABLE VII
GOOGLE CLOUD PLATFORM (GCP) VIRTUAL INSTANCE PARAMETERS.

Service	Maximum Bandwidth	Key Features & Optimizations
Compute Engine VMs (e.g., A3, G4, C4, H4D)	Up to 1,000 Gbps (A3-highgpu-8g) and 400 Gbps (G4-standard-384)	Tier1 networking is required on selected machine types (like C2D, N2, N2D, C4) to reach 100 Gbps or 200 Gbps per-VM egress. Google Virtual NIC (gVNIC) and Fast Socket are recommended to improve performance for distributed workloads, especially for NCCL on 100 Gbps+ networks.
Dedicated Interconnect	10 Gbps and 100 Gbps port speeds	Provides a direct, private physical connection from your on-premises network to Google's n etwork for consistent, high-throughput hybrid connectivity.

TABLE VIII
AMAZON WEB SERVICES (AWS) VIRTUAL INSTANCE PARAMETERS.

Service	Maximum Bandwidth	Key Features & Optimizations
EC2 Instances (e.g., HPC, GPU, and Network-optimized instance families)	Up to 400 Gbps (for latest instances like P5)	Requires specific, high-end instance types. Enhanced Networking via the Elastic Network Adapter (ENA) is essential for high packet-per-second and low-latency performance. AWS also offers EC2 UltraClusters with specialized low-latency networking for massive-scale HPC and AI/ML workloads.
AWS Direct Connect	1 Gbps to 100 Gbps (Dedicated Connections)	Provides a dedicated network connection from your premises to an AWS Direct Connect location. You can use Link Aggregation Groups (LAGs) to combine multiple 10 Gbps connections or utilize single 100 Gbps dedicated connections.

TABLE IX
AZURE VIRTUAL INSTANCE PARAMETERS.

Service	Maximum Bandwidth	Key Features & Optimizations
Azure VMs (e.g., specific HPC and GPU series)	Varies significantly by instance type and can reach 100 Gbps or more for specialized instances.	Accelerated Networking is a key feature to reduce latency and maximize throughput up to the VM's assigned limit. High-performance computing (HPC) instances often use specialized InfiniBand networking for inter-VM communication within a cluster.
Azure ExpressRoute	Up to 100 Gbps (Dedicated connections)	Provides a private connection from your infrastructure to Azure. ExpressRoute Direct supports 10 Gbps and 100 Gbps dedicated connections, and the ExpressRoute Premium SKU offers higher limits on advertised routes and virtual network links.
Azure Firewall Premium	Up to 100 Gbps throughput	The Premium tier offers significantly higher throughput than the Standard tier, specifically to handle high-volume network traffic with deep packet inspection enabled.

> takes less than twice the time of a local one underscores the efficacy of a co-designed, holistic approach. Embedding this capability directly into DPUs would effectively eliminate this performance delta, bringing cloud data movement performance into parity with optimized, local bare-metal transfers. A detailed examination of this DPU-based architecture will be presented in a future paper.

This case study moves beyond benchmarking to demonstrate a fundamental performance trade-off in the standard cloud model for fast data movement.

G. *Methodology and Reproducibility*

To enable independent validation of the performance claims presented in this paper, we have provided all essential configuration details required to replicate our testing environment. This includes the exact kernel parameters, latency simulation values, and visualization commands, all of which are publicly available in a GitHub repository [22]. The `iperf3` benchmark tool used in comparative analysis is also publicly available from its official repositories [21].

The Zettar `zx` data mover and associated deployment tooling represent proprietary commercial technology reserved

TABLE X
Typical performance information published by hyperscalers.

Cloud Provider	Achievement/Workload	Quantitative Metric	Throughput Implication
AWS	Western Digital HPC Simulation	2.5 million simulation tasks on a 1 million vCPU cluster completed in 8 hours.	Demonstrates massive-scale, low-latency, high-bandwidth communication between 1,000,000 virtual CPUs for a single run, reducing the job time from 20 days on-premises to 8 hours in the cloud.
GCP	PGS Seismic Processing	Scaled their workload from 202,000 on-premises cores to a peak of 1.2 million vCPUs in the cloud.	Represents a 6x increase in compute-and-data-transfer capacity for a single user, which—if run continuously—would place it among the world's largest supercomputers.
Azure	University of Bath HPC	Increased available compute resources from 250 on-premises nodes to thousands in Azure (a 900% increase).	Demonstrates the ability to provide on-demand, massive, high-speed burst capacity to avoid researcher wait times, a practical measure of ultra-high data access elasticity.

for customers and partners. The provided materials allow independent verification of our benchmarking methodology without requiring access to our commercial software.

H. Acknowledgment

The authors wish to extend their sincere gratitude to the following colleagues for their invaluable insights and for graciously reviewing early drafts of this manuscript:

- **Chin Guok** (CTO, ESnet) for reviewing the manuscript and providing network expertise.
- **Dr. Ezra Kissel** (Network Research Engineer, ESnet) for his thoughtful review of our testbed methodology and for his foundational 2012 work on Linux network emulation conducted at the University of Delaware.
- **Dr. Amedeo Perazzo** (formerly LCLS-II Controls and Data Systems Director, SLAC National Accelerator Laboratory) for offering insights from large-scale scientific facilities.
- **Dr. Jana Thayer** (LCLS Experimental Data Systems Division Director) for her review and insights on the data system architecture and real-time processing requirements of the LCLS-II, which motivated our work on streaming data movement and holistic system design.
- **Dr. Wilko Kroeger** (LCLS-II, Information Specialist, SLAC National Accelerator Laboratory) for his invaluable insights and review of the data management challenges associated with the Linac Coherent Light Source II (LCLS-II), which greatly informed the architectural perspectives presented in this work.
- **Mark Gray** (Head of Strategic Partnership, ex-Head of Scientific Platforms, Pawsey Supercomputing Research Centre) for contributing HPC and storage perspectives, sharpened by our long-standing discussions on the data movement challenges for the Square Kilometer Array (SKA) since 2018.
- **Sven Breuner** (creator of the BeeGFS parallel file system and the `elbencho` storage benchmarking tool) for his review and insights from a storage architecture and performance perspective.
- **Chih Chuan Shih** (Platform Lead, Genome Institute of Singapore) for providing genomics and data lifecycle insights.
- **Dr. Tsukasa Nakamura (KEK, Institute of Materials Structure Science)** for his assistance in establishing the KEK IT and internal network environment for the `zx` implementation, leveraging his expertise in large-scale computational research and bioinformatics.

Their feedback, drawn from deep expertise across networking, large-scale scientific facilities, high-performance computing, high-performance storage, and genomics, significantly strengthened this work. The views and conclusions presented herein are, of course, solely those of the authors.

The authors also wish to acknowledge the generous in-kind support that enabled the critical empirical validation of the architecture: SLAC National Accelerator Laboratory for providing testbed space and facilities (2015–2019); ESnet for provisioning a dedicated, 5,000-mile 100 Gbps OSCAR loop (2015–2019); Mellanox Technologies (now part of NVIDIA) and Intel Corporation for providing essential hardware components; and AIC for providing the storage servers.

Chin Fang would like to offer special thanks to Dr. Roger Leslie (Les) Anderton Cottrell, who, before his retirement from SLAC in 2017, provided invaluable mentorship and guidance that facilitated our further collaboration with the DOE lab community.

Chin Fang also wishes to acknowledge two Zettar colleagues, Igor Solovyov and Alexander Nazarenko, for their critical engineering contributions to the `zx` software. Their years-long engineering effort has been instrumental in generating the commercial revenue that sustained the company's operation. Mr. Nazarenko, paired with Chin Fang, is a key contributor to the team that earned the Overall Winner title at the SCA19 Data Mover Challenge [7].

Finally, Chin Fang wishes to express his profound gratitude to his wife, whose steadfast support over 34 years provided the essential personal foundation necessary for the intense, concentrated engineering effort documented in this paper. This work would not have been possible without her commitment.

PoC of “Zettar zx”

Rated outbound network bandwidth at KEK: **10 Gbps**
NiR benchmark dataset size: **1.2 TiB**


Toshio Moriya


Tsukasa Nakamura

KEK to AWS S3 in Tokyo region

aws-cli (AWS native)

	Xfer Rate (Gbps)	Xfer Time (mins)
1st run	2.02	75.13
2nd run	1.99	76.33
3rd run	1.99	76.07
Mean	2.00	75.84
SD	0.017	0.63

Zettar zx

	Xfer Rate (Gbps)	Xfer Time (mins)
1st run	6.71	22.67
2nd run	6.72	22.62
3rd run	6.70	22.7
Mean	6.71	22.66
SD	0.01	0.04

3.34x faster xfer speed

172x the distance

KEK to AWS S3 in N. Virginia region

aws-cli (AWS native)

	Xfer Rate (Gbps)	Xfer Time (mins)
1st run	-	238.55
2nd run	-	234.35
3rd run	-	232.65
Mean	-	235.18
SD	-	2.48

Zettar zx

	Xfer Rate (Gbps)	Xfer Time (mins)
1st run	4.10	44.57
2nd run	4.74	38.52
3rd run	4.96	36.83
Mean	4.60	39.97
SD	0.36	3.32

Better!

5.88x faster xfer speed

3.1x longer xfer time → Better! → 1.8x longer xfer time

Fig. 11. A comparison of transferring a 1.2 TiB Cryo-EM dataset from the Cryo-EM facility of KEK (Tsukuba, Ibaraki, Japan) to different AWS Regions using `aws-cli` and zx.

I. Experimental Setup and Author Disclosure

1) Validation Platforms: Testing for this study was performed on multiple platforms, demonstrating the adaptability of the co-design principle: a testbed utilizing two HPE DL380 Gen 11 servers and two Intel servers at the former Intel Swindon Lab (U.K.), and two HPE DL380 Gen 11 server-based appliances located in Switzerland and California, U.S. data centers, connected via a shared production 100 Gbps link carrying other network traffic.

2) Author Role and Technology Disclosure: **Chin Fang** is the Founder and CEO of Zettar Inc. He is the originator of the co-design principle and the “ drainage basin conceptual model” (**Fig. 3**) that form the core of this work. He was the sole architect, systems engineer, and hands-on builder of the data movement appliances and the latency simulation testbed. Furthermore, he personally conducted all empirical validation, data generation, and analysis presented in this paper—including the tests, figures, tables, and the landmark production-scale results cited in [3], [4], [5], and [8]. This entire body of work, from initial concept to final validation and documentation, is the product of his individual technical execution. Zettar's technologies, including the `zx` data mover, are commercially integrated into appliances sold by partners such as Hewlett Packard Enterprise (HPE). Intel Corporation has collaborated with Zettar on related development projects.

Timothy Stitt brings over two decades of cross-disciplinary expertise in strategizing, procuring, and optimizing high-performance computing (HPC) and AI infrastructure for data-intensive scientific workflows. His career spans premier academic and corporate research institutions, where he has led the architectural planning and integration of large-scale, composable compute and storage services. This extensive experience with the full stack of HPC technologies—from application-level tuning to global service design—provides a critical real-world perspective on the infrastructural barriers and requirements for end-to-end data movement. His contributions have been recognized with awards, including “Best Use of HPC in Life Sciences” (SC16) and “Best Practice in IT Infrastructure/HPC” (Bio-IT World 2017). Other than his current position at a tier-1 biopharma business, Dr. Stitt's background includes roles as a Research Assistant Professor at the University of Notre Dame, a Lecturer in Computer Science, and an HPC Application Scientist at the Swiss National Supercomputing Centre (CSCS), underpinned by a Ph.D. in

Computational Science.

Michael J. McManus brings a cross-disciplinary perspective from a career spanning deep science and enterprise IT. With a Ph.D. in synthetic organic chemistry (MIT) and a B.S. in polymer chemistry (UMass Amherst), his experience—from the U.S. Army and Intel to Fujitsu, Kodak, and six scientific software startups—embodies the cross-domain integration this paper advocates as an antidote to siloed optimization. His role as Principal Engineer and Director of Precision Medicine at Intel, coupled with his tenure on the NIH AI Working Group, provides critical context for the discussion on data movement in computational science and AI/ML workflows.

Toshio Moriya is a Project Associate Professor at the High Energy Accelerator Research Organization (KEK) in Japan. He provided an independent scientific validation environment for the current work, specifically through the Cryo-EM data use case, enabling independent scientific replication and domain-specific benchmarking. His research focuses on automating Cryo-EM single-particle analysis to remove human involvement from the workflow and to overcome practical barriers in applying Cryo-EM to compound screening for structure-based drug design (SBDD) and other industrial applications. In line with this objective, he is establishing an IoT-based Cryo-EM network across Japan. Using AWS cloud services as the hub and Zettar `zx` as the main data mover (**Fig. 11**), this system will fully automate data processing between nationwide Cryo-EM facilities and the cloud. This effort aims to rapidly build a large-scale database of compound-bound protein structures and enable future big-data-driven discovery.

*3) **Data Availability and Licensing**:* In the spirit of reproducible research, all elements required to replicate the testing and benchmarking presented in this paper are provided in a public GitHub repository [22]. The `zx` software data mover is a commercially licensed product and is not available for public distribution.

J. Conclusions

The arguments presented in this paper against over-reliance on isolated, formulaic optimizations are rooted in a fundamental understanding of complex systems, a perspective developed during Ph.D. research on the design of composite laminates [56]. As in the arrangement of discrete plies with specific orientations defining the macroscopic properties of a composite plate, the performance of a data movement system emerges from the interplay of discrete, non-linear components: CPU cores, memory hierarchies, storage devices, network interfaces, among others.

The holistic co-design principle central to this work represents the culmination of a sustained engineering philosophy first articulated in our early architectural blueprints [58] and subsequently refined through practical implementation and public discourse [59], [60]. This longitudinal perspective underscores that consistent, high-performance data movement has always been fundamentally a systems engineering challenge rather than an isolated network optimization problem.

This work also introduces a validated path toward democratizing high-performance data movement. The principle of co-design integrates the data mover software, host OS, and hardware stack. Said path eliminates the reliance on complex, manual, and often proprietary system tuning, resulting in a straightforward and cost-effective deployment model that provides predictable outcomes from the resource-constrained edge sites to the high-throughput core. This consistency is evidenced by the use of commonly available hardware valued at approximately $2,000 per device for the 1–10 Gbps mini-appliances. This capital efficiency proves that rich data transfer capabilities are no longer confined to environments with multi-million dollar testbeds or dedicated network engineering staff. The resulting architecture is not merely faster, but fundamentally more accessible to the Research and Education communities at the perimeter.

The enduring lesson is that seeking a simple, elegant formula to govern such systems is a mirage. The true path to performance lies not in mathematical purity applied to a simplified model, but in embracing the inherent complexity through empirical, holistic co-design. This co-design is required on two fundamental levels: optimizing the internal host architecture and ensuring efficient, scale-out coordination among peer appliances. This paper demonstrates that the same systems-thinking principles that govern the design of physical materials are equally critical for architecting the high-performance data systems of the digital age.

The scalability of this architectural approach was demonstrated as early as 2018 in the petabyte-transfer setup [3] and [4] shown in **Fig. 13** (reproduced from [7] **Appendix 6.1**). This scale-out design used multiple older servers with aggregated 10Gbps interfaces, proving that the software architecture could distribute workload across nodes without cluster management overhead.

Modern 2U2N servers, e.g., the SYS-222BT-DNR [57] (**Fig. 12**), enable the same scale-out efficiency within a single chassis. It extends our testbed-based methodology to **400 Gbps** and beyond, using the same architectural principles that successfully moved petabytes in 2018.

References

[1] Energy Sciences Network (ESnet), "About ESnet," ESnet. [Online]. Available: https://www.es.net/about/. [Accessed: Oct. 15, 2025].

[2] J. Thayer and C. Fang, "Big data at the Linac Coherent Light Source," DOE at SC, 2019 Featured Speakers, Nov. 19, 2019. [Online]. Available: https://scdoe.info/featured-talks/. [Accessed: Oct. 13, 2025].

[3] ESnet, "SLAC, AIC and Zettar move petabyte datasets at unprecedented speed via ESnet," LightBytes ESnet, Jul. 26, 2017. [Online]. Available: https://lightbytes.es.net/2017/07/26/slac-aic-and-zetta-move-petabyte-datasets-at-unprecedented-speed-via-esnet/. [Accessed: Oct. 13, 2025].

[4] ESnet, "ESnet's network software helps SLAC researchers in record-setting transfer of 1 petabyte of data," ESnet News, Oct. 17, 2018. [Online]. Available: https://www.es.net/news-and-publications/esnet-news/2018/esnets-network-software-help-slac-researchers-in-record-setting-transfer-of-1-petabyte-of-data/. [Accessed: Oct. 13, 2025].

[5] ICM, "Historical 1st Poland–Singapore data transfer production trial," ICM Blog, Nov. 13, 2019. [Online]. Available: https://icm.edu.pl/en/blog/2019/11/13/poland-singapore-data-transfer-over-new-cae-1-100g-transcontinental-link/. [Accessed: Oct. 13, 2025].

[6] W. Bhimji, D. Bard, M. Romanus, D. Paul, A. Ovsyannikov, B. Friesen, et al., "Accelerating Science with the NERSC Burst Buffer Early User Program," in *Proc. Cray User Group (CUG) Conf.*, 2016, Art. no. 162S2-2. [Online]. Available: https://cug.org/proceedings/cug2016_proceedings/includes/files/pap162s2-file2.pdf. [Accessed: Oct. 13, 2025].

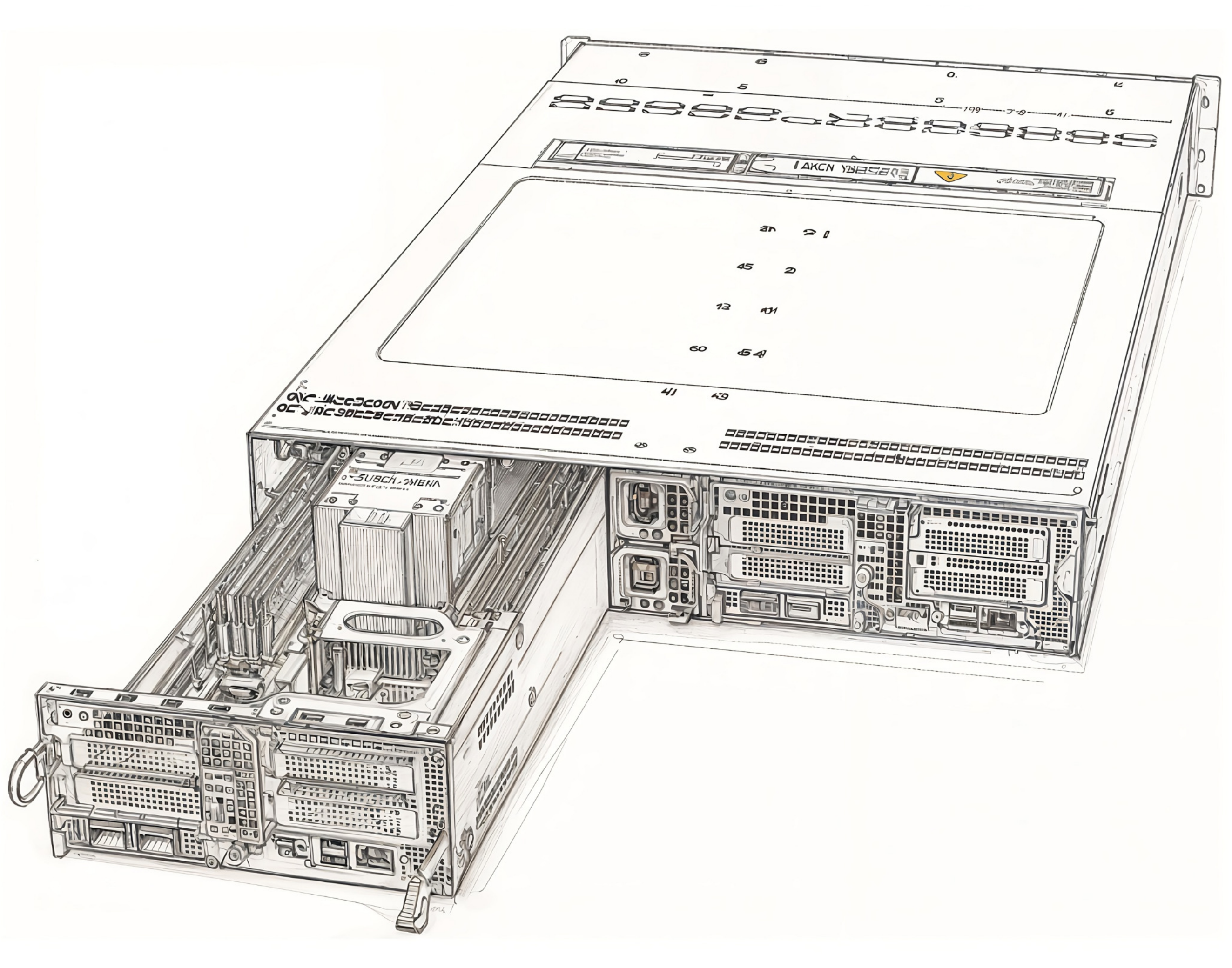


Fig. 12. The Supermicro 2U 2-Node BigTwin with 12 hot-swap 2.5" NVMe drives per node (Diagram by Chin Fang).

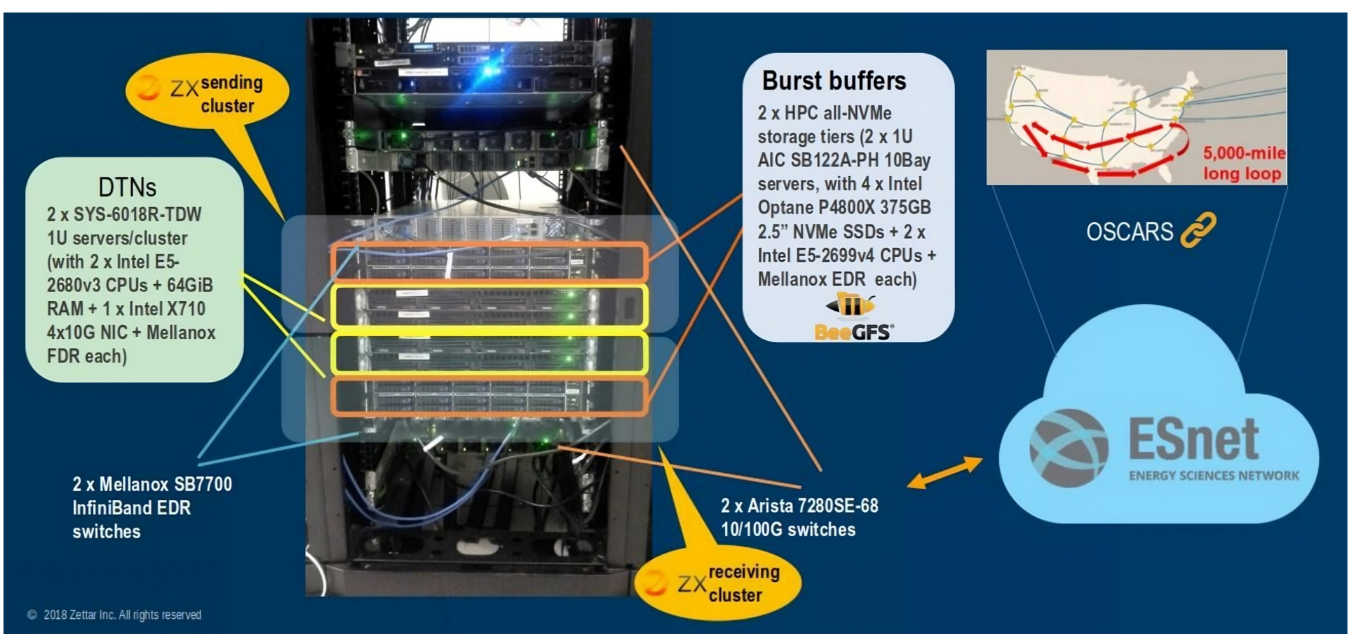


Fig. 13. Zettar's testbed at SLAC from 2015 to 2019. See page 19 of [7] for more details of the configuration. (Copyright Zettar Inc. Reproduced with permission for this publication.)

[7] NSCC, “Overall winner, supercomputing asia 2019 data mover challenge,” SC-Asia, Mar. 13, 2019. [Online]. Available: https://www.nscc.sg/wp-content/uploads/2019/03/Data-Mover-Challenge-Release-13-March-2019-2.pdf. [Accessed: Oct. 13, 2025].

[8] E. Kissel and C. Fang, “Zettar zx evaluation for ESnet DTNs,” U.S. DOE ESnet Technical Report, 2020. [Online]. Available: https://www.es.net/assets/Uploads/zettar-zx-dtn-report.pdf. [Accessed: Oct. 13, 2025].

[9] Red Hat, “Red Hat Enterprise Linux,” Red Hat. [Online]. Available: https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux. [Accessed: Oct. 13, 2025].

[10] Rocky Enterprise Software Foundation, “Enterprise Linux, the community way.” [Online]. Available: https://rockylinux.org/. [Accessed: Oct. 13, 2025].

[11] The Global Research Platform (GRP), “The global research platform (GRP),” [Online]. Available: https://www.theglobalresearchplatform.net/. [Accessed: Oct. 13, 2025].

[12] The Global Research Platform (GRP), “Mini global research platform (GRP) workshop at SCA2025,” Mar. 28, 2025. [Online]. Available: https://www.theglobalresearchplatform.net/meetings/mini-global-research-platform-grp-workshop-at-sca2025/. [Accessed: Oct. 13, 2025].

[13] Zettar Inc., “Our products”, *Zettar – Moving Data at Scale and Speed*. [Online]. Available: https://www.zettar.com/our-products/#products. [Accessed: Oct. 13, 2025].

[14] E. Kissel and C. Fang, “A.6.1 The SLAC/Zettar data transfer setup,” in “Zettar zx evaluation for ESnet DTNs,” U.S. DOE ESnet Technical Report, 2020, p. 19. [Online]. Available: https://www.es.net/assets/Uploads/zettar-zx-dtn-report.pdf. [Accessed: Oct. 13, 2025].

[15] E. Kissel and C. Fang, “2 Testbed resources, 3 Science DMZ, 4. System and software environment,” in “Zettar zx evaluation for ESnet DTNs,” U.S. DOE ESnet Technical Report, 2020, pp. 2–5. [Online]. Available: https://www.es.net/assets/Uploads/zettar-zx-dtn-report.pdf. [Accessed: Oct. 13, 2025].

[16] RBC Capital Markets, “The healthcare data explosion,” RBC Capital Markets. [Online]. Available: https://www.rbccm.com/en/gib/healthcare/episode/the_healthcare_data_explosion. [Accessed: Oct. 13, 2025].

[17] Energy Sciences Network (ESnet), “ESnet fasterdata knowledge base.” [Online]. Available: https://fasterdata.es.net/. [Accessed: Oct. 13, 2025].

[18] ESnet, “`iperf3`: A TCP, UDP, and SCTP network bandwidth measurement tool,” *[Computer Software]*, GitHub Repository. [Online]. Available: https://github.com/esnet/iperf. [Accessed: Oct. 13, 2025].

[19] ESnet, “`iperf-3.16`,” iperf Releases, Nov. 30, 2023. [Online]. Available: https://github.com/esnet/iperf/releases. [Accessed: Oct. 13, 2025].

[20] Energy Sciences Network (ESnet), “iperf3 FAQ.” [Online]. Available: https://software.es.net/iperf/faq.html#faq. [Accessed: Oct. 13, 2025].

[21] man7.org, “`ping` - send ICMP ECHO REQUEST to network hosts,” Linux man pages. [Online]. Available: https://man7.org/linux/man-pages/man8/ping.8.html. [Accessed: Oct. 13, 2025].

[22] C. Fang, *pieee repo*, GitHub. Available: https://github.com/fangchin/pieee. [Online]. [Accessed: Oct. 24, 2025].

[23] ESnet, “Linux tuning,” ESnet Fasterdata Knowledge Base. [Online]. Available: https://fasterdata.es.net/host-tuning/linux/#toc-anchor-2. [Accessed: Oct. 13, 2025].

[24] Internet2, “Internet2”. [Online]. Available: https://internet2.edu/. [Accessed: Oct. 13, 2025].

[25] GANT Association, “GANT is the collaboration of European national research and education networks (NRENs).” [Online]. Available: https://geant.org/. [Accessed: Oct. 13, 2025].

[26] AARNet, “Australia’s national research and education network.” [Online]. Available: https://www.aarnet.edu.au/. [Accessed: Oct. 13, 2025].

[27] SingAREN, “Singapore advanced research and education network.” [Online]. Available: https://www.singaren.net.sg/. [Accessed: Oct. 13, 2025].

[28] National Institute of Standards and Technology, U.S. Dept. of Commerce, “Bit error rate,” Information Technology Laboratory, Computer Security Resource Center Glossary. [Online]. Available: https://csrc.nist.gov/glossary/term/bit_error_rate. [Accessed: Oct. 13, 2025].

[29] perfSONAR, “perfSONAR public.” [Online]. Available: https://stats.perfsonar.net/. [Accessed: Oct. 13, 2025].

[30] U.S. DOE ESnet, “ESnet services and service level descriptions,” 2009. [Online]. Available: https://www.es.net/assets/ESnetServiceLevels-V4.0.pdf. [Accessed: Oct. 13, 2025].

[31] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Newman, J. Zhang, and H. K. J. Chu, “BBR: Congestion-based congestion control,” ACM Queue, vol. 14, Sept.–Oct. 2016, pp. 20–53. [Online]. Available: https://research.google/pubs/bbr-congestion-based-congestion-control-2/. [Accessed: Oct. 13, 2025].

[32] I. Rhee, L. Xu, S. Ha, A. Zimmermann, L. Eggert, and R. Scheffenegger, “CUBIC for Fast Long-Distance Networks,” Request for Comments (RFC) 8312, Internet Engineering Task Force (IETF), Feb. 2018. doi: 10.17487/RFC8312. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8312. [Accessed: Oct. 13, 2025].

[33] L. Xu, S. Ha, I. Rhee, V. Goel, and L. Eggert, Ed., “CUBIC for Fast and Long-Distance Networks,” Request for Comments (RFC) 9438, Internet Engineering Task Force (IETF), Aug. 2023. doi: 10.17487/RFC9438. [Online]. Available: https://www.rfc-editor.org/rfc/rfc9438.html. [Accessed: Oct. 13, 2025].

[34] Google, “bbr: Latest BBR code from Google’s BBR team,” GitHub. [Online]. Available: https://github.com/google/bbr. [Accessed: Oct. 13, 2025].

[35] Gnuplot Development Team, *gnuplot 6.1: An Interactive Plotting Program* (Manual), 2024. [Online]. Available: http://www.gnuplot.info/docs_6.1/Gnuplot_6.pdf. [Accessed: Oct. 13, 2025].

[36] Red Hat, “Is Kernel TLS Offload (kTLS) supported in RHEL?” Solution 7120108, Red Hat Customer Portal. Updated May 20, 2025. [Online]. Available: https://access.redhat.com/solutions/7120108. [Accessed: Oct. 13, 2025].

[37] The Wall Street Journal Staff, “China AI Chip Curb Suits Case,” The Wall Street Journal. June 12, 2025. [Online]. Available: https://www.wsj.com/tech/china-ai-chip-curb-suitcases-7c47dab1. [Accessed: Oct. 13, 2025].

[38] J. Morales, “Chinese AI Outfits Smuggling Suitcases Full of Hard Drives to Evade U.S. Chip Restrictions, Training AI Models in Malaysia Using Rented Servers,” Tom’s Hardware. Jun. 13, 2025. [Online]. Available: https://www.tomshardware.com/tech-industry/artificial-intelligence/chinese-ai-outfits-smuggling-suitcases-full-of-hard-drives-to-evade-u-s-chip-restrictions-training-ai-models-in-malaysia-using-rented-servers. [Accessed: Oct. 13, 2025].

[39] Bay Area Water Supply and Conservation Agency (BAWSCA), “Hetch Hetchy System,” BAWSCA. [Online]. Available: https://bawsca.org/water/supply/hetchhetchy. [Accessed: Oct. 13, 2025].

[40] Energy Sciences Network (ESnet), “Proposal Process,” ESnet. [Online]. Available: https://www.es.net/network-r-and-d/experimental-network-testbeds/100g-sdn-testbed/proposal-process/. [Accessed: Oct. 13, 2025].

[41] E. Kissel and M. Swany, “Validating Linux Network Emulation,” University of Delaware, Newark, DE, Tech. Rep. 2012/004, 2012. [Online]. Available: https://www.open.sice.indiana.edu/downloads/papers/netem_validate.pdf. [Accessed: Oct. 13, 2025].

[42] The Linux man-pages project, “tc(8) - Linux manual page,” man7.org. [Online]. Available: https://man7.org/linux/man-pages/man8/tc.8.html. [Accessed: Oct. 13, 2025].

[43] The Linux man-pages project, “tc-netem(8) - Linux manual page,” man7.org. [Online]. Available: https://man7.org/linux/man-pages/man8/tc-netem.8.html. [Accessed: Oct. 13, 2025].

[44] Intel, “What is Intel QuickAssist Technology (Intel QAT)?,” Intel. [Online]. Available: https://www.intel.com/content/www/us/en/products/docs/accelerator-engines/what-is-intel-qat.html. [Accessed: Oct. 13, 2025].

[45] Energy Sciences Network (ESnet), “Hardware Selection,” ESnet Fasterdata website. Jan. 9, 2025. [Online]. Available: https://fasterdata.es.net/DTN/hardware/. [Accessed: Oct. 13, 2025].

[46] Intel, “Intel Xeon Gold 5418N Processor (45M Cache, 1.80 GHz) Specifications,” Intel. [Online]. Available: https://www.intel.com/content/www/us/en/products/sku/232392/intel-xeon-gold-5418n-processor-45m-cache-1-80-ghz/specifications.html. [Accessed: Oct. 13, 2025].

[47] Red Hat, “The State of Virtualization [Ebook],” Red Hat. [Online]. Available: https://www.redhat.com/en/resources/state-of-virtualization-ebook. [Accessed: Oct. 13, 2025].

[48] NVIDIA, “Single Root I/O Virtualization (SR-IOV),” MLNX OFED v23.10-0.5.5.0 Documentation. [Online]. Available: https://docs.nvidia.com/networking/display/mlnxofedv23100550/single+root+io+virtualization+%28sr-iov%29. [Accessed: Oct. 13, 2025].

[49] Amazon Web Services, “Overview of the multipart upload feature,” *Amazon S3 User Guide*. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/mpuoverview.html. [Accessed: Oct. 13, 2025].

[50] S. Acharya and A. Sirinterlikci, “Introducing Engineering Design Through an Intelligent Rube Goldberg Implementation,” *The Journal of Technology Studies (JOTS)*, vol. 36, no. 2, Fall 2010. [Online]. Available: https://scholar.lib.vt.edu/ejournals/JOTS/v36/v36n2/acharya.html. [Accessed: Oct. 13, 2025].

[51] NVIDIA, “What’s a DPU, Data Processing Unit?” NVIDIA Blog. [Online]. Available: https://blogs.nvidia.com/blog/whats-a-dpu-data-processing-unit/. [Accessed: Oct. 13, 2025].

[52] Zettar, “Zettar Advances Data Movement in Collaboration with MiTAC Computing and NVIDIA,” HPCwire. Nov. 20, 2024. [Online]. Available: https://www.hpcwire.com/off-the-wire/zettar-advances-data-movement-in-collaboration-with-mitac-computing-and-nvidia/. [Accessed: Oct. 18, 2025].

[53] T. Moriya, Y. Yamada, M. Yamamoto, and T. Senda, “GoToCloud optimization of cloud computing environment for accelerating cryo-EM structure-based drug design,” *Communications Biology*, vol. 7, no. 1, Oct. 2024, doi: 10.1038/s42003-024-07031-6. [Online]. Available: https://www.nature.com/articles/s42003-024-07031-6. [Accessed: Oct. 23, 2025].

[54] High Energy Accelerator Research Organization (KEK), “KEK Official Website,” KEK. [Online]. Available: https://www.kek.jp/en/. [Accessed: Oct. 13, 2025].

[55] Amazon Web Services, “aws-cli: Universal Command Line Interface for Amazon Web Services,” [Computer Software], Version 2, GitHub Repository. [Online]. Available: https://github.com/aws/aws-cli. [Accessed: Oct. 13, 2025].

[56] C. Fang, “Design of Composite Laminates by a Monte Carlo Method,” J. of Composite Materials, vol. 27, no. 7, pp. 721–753, 1993. [Online]. Available: https://ui.adsabs.harvard.edu/abs/1993JCoMa..27..721F/abstract. [Accessed: Oct. 13, 2025].

[57] Supermicro, “BigTwin SYS-222BT-DNR,” Supermicro. [Online]. Available: https://www.supermicro.com/en/products/system/bigtwin/2u/sys-222bt-dnr. [Accessed: Oct. 13, 2025].

[58] C. Fang, R. “Les” A. Cottrell, A. B. Hanushevsky, W. Kroeger, and W. Yang, “High Performance Data Transfer for Distributed Data Intensive Sciences,” SLAC National Accelerator Laboratory, Menlo Park, CA, USA, Tech. Rep. SLAC-TN-16-001, 2016. [Online]. Available: https://www.slac.stanford.edu/pubs/slactns/tn05/slac-tn-16-001.pdf. [Accessed: Oct. 13, 2025].

[59] C. Fang, *Data Movement at Speed & Scale, The 4th IT Pillar,* Samsung Forum, 2019. [Video]. YouTube. Available: https://www.youtube.com/watch?v=XvmWFUeZyUg. [Accessed: Oct. 13, 2025].

[60] C. Fang and E. Kissel, *High-performance Data Movement Services,* Rice OGHPC Technical Talk, 2021. [Video]. YouTube. Available: https://www.youtube.com/watch?v=LIQxIZVAxks. [Accessed: Oct. 13, 2025].