Bridging Network and Parallel I/O Research for Improving Data-Intensive Distributed Applications

Debasmita Biswas*, Sarah Neuwirth‡, Arnab K. Paul†, Ali R. Butt*
*Virginia Tech, ‡Goethe-University Frankfurt, †Oak Ridge National Laboratory
{debasmita17, butta}@vt.edu, s.neuwirth@em.uni-frankfurt.de, paula@ornl.gov

Abstract—**The rapidly evolving scene of emerging workloads poses a challenge to the High Performance Computing community in terms of communication and I/O. Significant improvements are required to keep up with the demand of high rate of data transfers, streaming services, and scientific research that deal with extremely large quantities of data, which may impede a system's performance. Networking is a key area that plays a major role in accelerating data transfers within HPC facilities. Though significant research efforts have targeted I/O optimization for storage systems, network optimization to improve the overall storage system performance has been rather overlooked by the research community. In this position paper, we aim to bridge the gap between networks and storage system optimization towards the common goal of accelerating HPC I/O and communication by revealing the various ways in which previously done network optimization research can be applied to improve I/O performance for data-intensive applications.**

Keywords-**Data-intensive Science, HPC, Network Research, High-Performance Data Transfer, Parallel Storage and I/O**

I. Introduction

I/O performance, which typically provides a measure for an application's read and write throughput, is one of the most important factors that determines the merit of a High Performance Computing (HPC) system and its usability. Applications, banking on HPC storage systems to reserve and process a large collection of data, from the fields of scientific research, social media, industries, etc, are expected to deliver fast reads, writes and easy access to data on demand. An important component of a distributed file system is its network that drives data transfers, internode communication and client-to-server communication. For data-intensive computing, network is a pivotal element that can affect the performance as significant overhead may be incurred during interprocess communications. With the increasing rate of data generation today, it is imperative that existing HPC systems adapt themselves to these applications' I/O requirements to stay relevant.

The last decade has witnessed a massive upsurge in the amount of data being generated and consumed by modern applications. It is expected that this trend will continue to grow and the rate of data generation will increase exponentially with more and more data intensive applications being embraced. Some of the major areas that will require super fast data access are data centers, scientific research workflows, data analytics, machine learning, Internet of Things, and content streaming platforms such as Netflix and Hulu. Distributed storage systems have been used in HPC to accelerate the processing of data-intensive applications. However, major upgrades are required to be made to the existing HPC storage designs in order for them to be able to keep up with the demands of fast and easy data transfers that these applications require or they are bound to be bottlenecked by the system I/O performance.

In this paper, we identify several works that address the problem of network optimization for the sake of improving system performance. However, work addressing the direct relationship between network optimization on distributed storage systems for I/O boost seems insufficient in the research community. Performance optimization for distributed storage systems has been traditionally tackled by scaling up the HPC system in use (number of nodes, number of cores, RAM, etc.)

In this paper, we identify key research papers in the field of network optimization and argue how their ideas can be applied towards performance optimization for distributed storage systems in HPC facilities. We also consider works that leverage network components as key parameters for throughput improvement on distributed storage systems. For this exercise, we have gone through recent research papers published between 2015-2021 and labelled them with the corresponding areas of network research they pertain to using the ACM classification tree labels. Then we analyze the papers to evaluate how network research can impact storage research. We hope to provide insights to the HPC community about how closely these two research areas are related and may work together towards the goal of performance optimization for HPC systems running data-intensive workloads.

II. Background and Related Work

Data-intensive science is considered to be the fourth pillar of science and complements the three interrelated paradigms of *empirical*, *theoretical*, and *computational science*. It is seen as a data-driven, exploration-centered style of science, where IT infrastructures and software tools are heavily used to help scientists manage, analyze, and share data. Large-scale scientific HPC workloads are increasingly data-intensive and

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

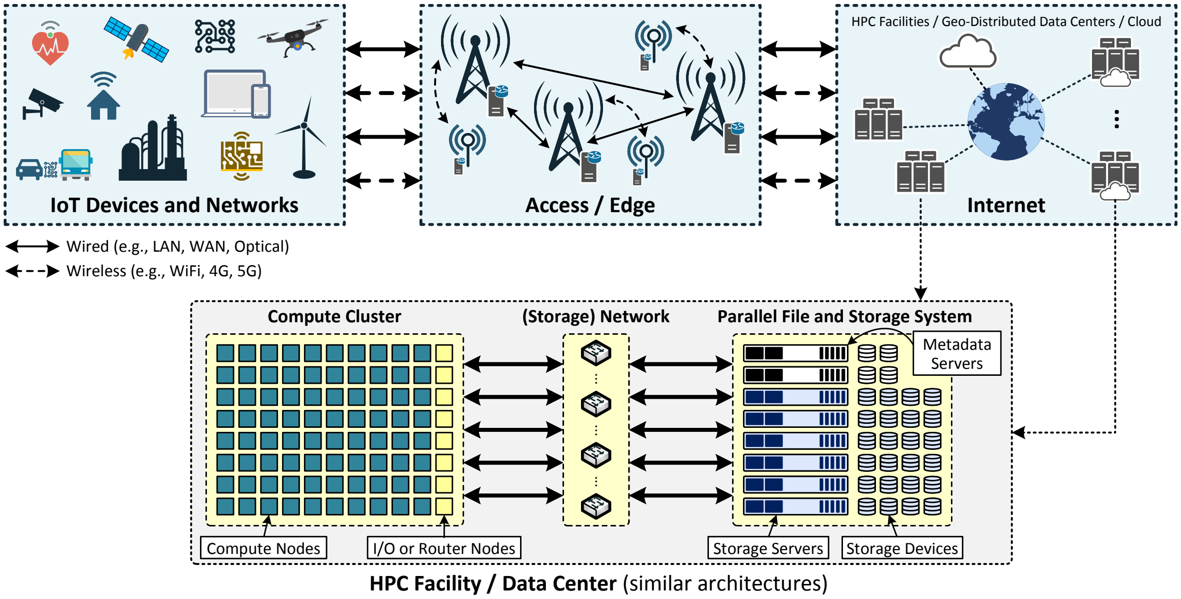
HPC Facilities / Geo-Distributed Data Centers / Cloud
IoT Devices and Networks
Access / Edge
Internet
Wired (e.g., LAN, WAN, Optical)
Wireless (e.g., WiFi, 4G, 5G)
Compute Cluster
(Storage) Network
Parallel File and Storage System
Metadata Servers
Compute Nodes
I/O or Router Nodes
Storage Servers
Storage Devices
HPC Facility / Data Center (similar architectures)

Fig. 1: An architectural overview of different network components. The diverse range spans IoT devices and networks, geo-distributed data centers and HPC facilities, access points equipped with edge equipment, and the larger internet.

put excessive pressure on the interconnection network, which is the backbone for both communication and parallel I/O.

An end-to-end data transfer involves at least three major components: a high-performance storage system, a high-performance network, and the software to tie it all together. Figure 1 provides an overview of different network components and communication networks, and their interconnection. The *Internet of Things* (IoT) describes physical objects that are embedded with sensors, processing ability, software, and other technologies, and that connect and exchange data with other devices and systems over the Internet or other communications networks. The diverse range of IoT devices and networks connect to the *access points*, possibly equipped with edge computing resources and routers, which can communicate amongst themselves and to the larger *Internet*. Geo-distributed data centers, HPC facilities, and cloud resources provide another source of high-performance compute capabilities for data-intensive science. Due to the increasing diversity of devices, networks, and services, it is only natural conclusion to combine the research efforts of the network and parallel I/O communities since they are two sides of the same coin.

As a recent I/O behavior analysis of a year's worth of I/O activity [1] has shown, HPC is no longer solely limited to traditional simulation and modeling workloads, which are typically write-intensive and bursty. Emerging HPC workloads now also include big data analytics [2], [3], machine learning, deep learning [4], high-throughput applications, and data-intensive workflows [5], [6]. These workloads exhibit largely different kinds of I/O patterns than the traditional simulation-based workloads resulting in highly random small file accesses, or non-sequential, metadata-intensive, and small-transaction reads and writes that ultimately are translated into network requests and therefore stress the underlying network.

However, only a few research papers have addressed the gap between network and parallel file system research. Previous work by Tsiftes et al. [7] has introduced the Coffee file system, which provides a programming interface for building efficient and portable storage abstractions for flash-based sensor networks. Their work shows that network layer components such as routing tables and packet queues can be implemented on top of Coffee, leading to increased performance and reduced memory requirements for routing and transport protocols. Ezell et al. [8] present an approach to optimize the I/O router placement and introduce fine-grained routing to ensure that Lustre clients are paired with the closest I/O router for communication with the parallel file system to minimize end-to-end hop counts and network congestion. Mills et al. [9] try to identify the optimal data transfer parameters for performing parallel genomics data transfers by optimizing the interface between parallel file systems and advanced research networks. And finally, Neuwirth [10] looks into accelerating network communication and I/O, but considers both research areas separately. To the best of the authors' knowledge, this is one of the first papers that attempts to align the network and parallel I/O and storage system research, which makes this a timely and valuable contribution to both research communities.

III. Survey Techniques

In this position paper, we follow different guidelines [11], [12] for undertaking a brief methodical survey that focuses on the identification of network optimization research that can also be leveraged to enhance the storage system designs in HPC. We focus on articles published between 2015 and 2021. The goal is to reveal the applicability of these research works for performance optimization of data-intensive science applications and their emerging workloads.

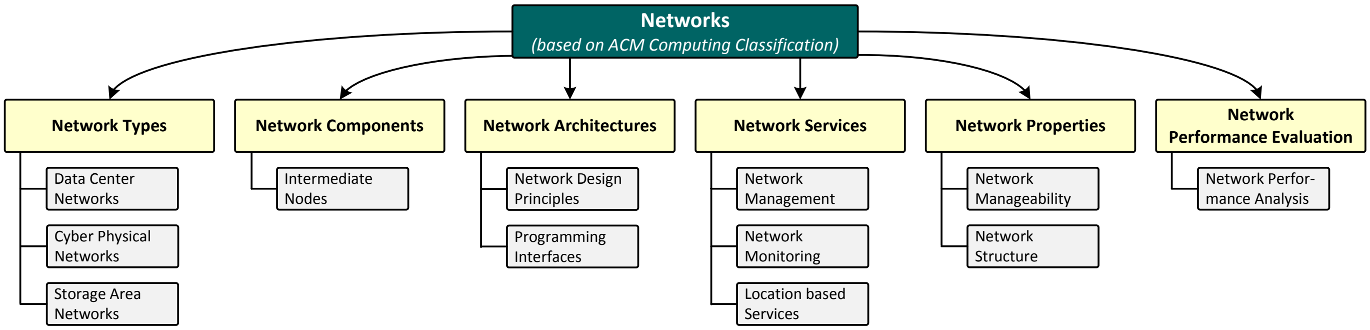

Fig. 2: Network research classification derived from the ACM Computing Classification System.

We use the *ACM Computing Classification System* (CCS) to identify relevant network research subcategories and identify search keywords. First, we narrow down the network research areas that are most appropriate for our goal and traverse through the child branches under each parent branch that fall under the *Networks* research label of interest. Next, we search the ACM library and used a brute force method on Google Scholar to identify pertinent research work not listed by ACM. The final set of labels used to categorize the selected papers is depicted in Figure 2. Note that a single publication can be categorized under multiple labels.

We apply a variety of combinations of search keywords, including: *Network Optimization*, *HPC Storage Systems*, *Data centers*, *Storage Area Network*, *IoT network*, *Edge*, *I/O optimization* and *Data-intensive applications/workloads*. Finally, we identify 17 research articles, each with at least one potential design of significance for network integrated HPC storage system I/O throughput increment in emerging workloads that pertain to data intensive applications.

IV. Network and I/O Research

As introduced in Section III, a subset of the ACM network classification is used to group publications on network optimization. Following Figure 2, we describe the optimization techniques for each network classification and estimate how they can be applied to parallel I/O optimization research dealing with data-intensive workloads.

A. Network Types

1) Data Center Networks: Data centers are critical components for hosting large-scale storage devices with extremely large amounts of data, and find application in big data science, data-intensive applications like Facebook, and Google, and over-the-top (OTT) platforms such as Netflix and HBO Max. Data centers are not always limited to a specific location and may be geographically distributed. As such, high storage I/O and data transmission rates are very critical to their smooth functionality and usability. *CliqueMap* [13] is a hybrid RMA/RPC (remote memory access/remote procedure call) caching system used at the Google Data Center for three years at the time of publication, which highlights the I/O benefits data centers can receive from careful distribution of work between RPC and RMA across data plane, control plane, and management operations, while optimizing the CPU overhead. This observation will be beneficial to data streaming applications where the number of read operations supersede write operations. *Chameleon* [14] increases the network utilization of data centers by reallocating resources at the edges of multiple geo-distributed data centers. Such network optimization can also be used to improve data transmission rates across various distributed storage systems that deal with multiple scientific datasets to aid collaborative efforts among scientists located at different parts of the world.

2) Cyber Physical Networks: The field of Cyber-Physical systems is interdisciplinary in nature and is a combination of computational, network and physical processes. It provides the foundation for IoT technology and for this paper, we will limit the scope of Cyber-Physical systems to the context of the same. Borah et al. [15] use a game theoretic approach for context-based routing for oppNets, to identify the best next-hop to forward data packets efficiently. It is based on the context information, encounter index, and distance of the corresponding node from the destination. This is designed for opportunistic IoT and will also greatly help in reducing latency related issues in I/O bound real-time data analysis problems.

3) Storage Area Network: Storage Area Networks (SAN) form the backbone of any large-scale HPC or data center I/O infrastructure by helping connect storage devices to storage servers. *BlueDBM* [16] incentivises the usage of distributed flash storage as a low cost and energy efficient alternative to DRAM to boost storage I/O for complex big data applications requiring random non-sequential memory accesses. To account for overheads caused due to network latency, BlueDBM is designed with its own in-store processors, integrated network routers and flash controllers. The design may also find application in local data centers.

BAShuffler [17] is an application level bandwidth-aware shuffle scheduler that tries to maximize the overall network utilization on a cluster. BAShuffler increases network utilization even in a heterogeneous network infrastructure with nodes displaying varying uplink and downlink bandwidth capacities. It exploits the available bandwidth from the available nodes in the cluster and increases the network utilization by incrementing the shuffle throughput, thereby increasing the overall cluster performance. Thus, BAShuffler can find applications in any HPC storage setup to leverage the full potential of the network resources and optimize I/O for increasingly data intensive workloads, such as, big data analytics.

SANs also benefit from the implementation of *Software Defined Networking* (SDN). *Ceph* [18] is a distributed object storage system that increases fault tolerance by creating two replicas of each object across its Object Storage Daemons (OSD) driven by its CRUSH algorithm [19]. Wu et al. [20] propose the use of SDN to detect network status and load of an OSD node, and use those values to select storage nodes for incoming objects that by default relies only on the OSD node storage capacity. This approach increases the read performance of a cluster significantly which is useful within distributed storage facilities that serve read intensive workloads like content streaming platforms (for example, Netflix) demanding very high throughput. It also opens up the possibility of integrating SDNs into distributed storage architectures and exploring their benefits towards parallel I/O.

B. Network Components – Intermediate Nodes

Network optimization on intermediate nodes is very beneficial in optimizing I/O on storage systems based on hierarchical structure (i.e., Lustre [21]). *Argo* [22] – a user space distributed shared memory system built atop MPI, proposes a novel cache coherence protocol that mitigates the latency produced due to communications between distant nodes. It maximizes localized decision making process facilitating faster synchronization between nodes. Argo does not use message handlers and delegates associated operations to RMA by requesting nodes. The work is preliminary and the full potential of Argo is yet untapped. It will be interesting to see the applicability of such a system on large-scale HPC centers. *NICE* (network-integrated cluster-efficient) [23] proposes a novel way of reducing network latency during request routing in a storage system through the implementation of a ring of virtual storage nodes over the traditional ring model of physical storage nodes in a Network Oblivious (NOOB) Storage system architecture. This approach leverages SDN for node virtualization. NICE uses the virtual storage nodes' IP while hashing an access request from the client and then maps the selected virtual node's IP address to the physical node's IP address with the help of the storage system's metadata service. The network packets are routed directly to the IP address of the physical storage nodes, foregoing the additional network hops between physical storage servers and thus reducing the overall network latency. Focusing on key-value storage system, this architecture is deployed on the prototype storage system NICEKV, which shows significant get/put performance boost and a reduced network latency. As the industry shows a growing trend to incorporate SDN for network management, the results promise a much needed performance boost in most deployments of HPC storage clusters that include data centers and edge computing.

C. Network Architecture

1) Network Design Principles: Fog computing in IoT tries to reduce the network latency during communication between IoT end devices and data center storage by scheduling IoT tasks to nearby edge devices within one hop distance. He et al. [24] address the lack of research into how multiple fog topologies influence the over cost in terms of average number of hops by proposing, implementing and evaluating two Integer Linear Programming (ILP) models on star and ring fog topologies on real time and mobile IoT tasks. The approach is tested on an *iCloudFog* (integrated cloud and fog framework) framework which shows that the star topology outperforms fully connected mesh topology, and ring topology costs can theoretically increase with increasing system complexity when compared to fully connected mesh topology. For emerging data-intensive workloads, fog computing will play a part between HPC system storage and IoT devices where these ILP models could be used to determine the optimal fog topology.

2) Programming Interfaces: A consistent bandwidth is extremely important for a quality user experience in the case of data-intensive applications like video gaming and content streaming platforms. As such, a dynamic self-driving network interface that can capture the application behavior and regulate the network bandwidth by intelligently reallocating available resources during application run, as proposed by Madanapalli et al. [25], can be a great breakthrough. Such self-driving network interfaces will also find application for data-intensive scientific workloads streaming between storage systems that see a high network concurrency, by setting application-specific priority and assisting and de-assisting the application based on a certain threshold. *PacketMill* [26] introduces a method for software packet metadata management and code optimizations pushing for maximum utilization of underlying commodity hardware. It increases throughput and reduces latency for non-trivial packet processing at 100GBps using one core. IoT applications that require a continues to/from flow of data from cloud through edge routers will benefit from such packet metadata management and network hardware optimizations.

D. Network Services

1) Network Management: Mill et al. [9] suggest an equation that leverages the *Bandwidth-Delay-Product* to predict the optimal TCP socket buffer size and the number of TCP streams for data transmission. The bandwidth-delay-product is computed as the product of the amount of data in flight and RTT (roundtrip delay time). The equation can be represented as: BDP $\leq$ buffer $\times$ streams. However, the equation does not take into account network losses. To mitigate this limitation, the authors suggest that a higher number of streams can be used versus increasing the socket buffer size as a larger number of streams is more favourable for quicker recovery in case of loss. This equation will be useful as SDNs, which can dynamically adapt network parameters, find more and more applications in HPC facilities.

2) Network Monitoring: Cloud computing companies that provide services to multiple businesses rely on shared computing resources that result in high network traffic concurrency. It can be observed that heavy hitter flows from certain tenants using the shared services may lead to contention in intermediate nodes that serve as load balancers, and cloud gateways. As such, it may significantly degrade the expected performance of the other tenants. Song et al. [27] note that widely deployed x86 boxes as intermediate nodes that use flow based hashing to load balance the packets received among available cores, may suffer from CPU overload from heavy hitter applications in cases where traffic from heavy hitter applications are hashed to the same CPU cores. The authors propose a cloudscale per-flow backpressure system designed in Alibaba Cloud that involves an FPGA(Field-programmable gate array)-based heavy hitter flow detection when CPU utilization has crossed a threshold and then conduct backpressuring the heavy hitter flows to the traffic source – this is done by the intermediate node through sending a notification packet to the traffic source and conducting heavy hitter rate, therefore limiting by adding corresponding meter table entries on the virtual switch (VS) of the packet sender. It is evaluated that an FPGA accelerated VS can positively influence the packet processing rate by 5 times, halve network latency and double traffic throughput compared to a software virtual switch. Researchers may leverage this to improve shared HPC storage performance by reducing CPU contention and maintain expected I/O throughput for all users. This is relevant for a shared facility where certain applications may generate unheralded high bursts of data which may negatively impact other applications.

3) Location Based Services: High speed data transfers are critical for data-intensive scientific applications. Mills et al. [9] focus on the optimization of the network configuration to produce higher bandwidth for genomics data transfers between two geographically distributed HPC storage clusters connected over a CloudLab testbed using Internet2 and SDN. They investigate the usage of GripFTP with multiple parallel file systems, namely BeeGFS, OrangeFS, GlusterFS and Ceph in association with Infiniband and TCP. For streaming a single file of 834GB, it is evaluated that the best data transmission performance was found in BeeGFS with 4-8 nodes connected by InfiniBand over GridFTP using at least 5 parallel TCP streams with a 16 MiB TCP socket buffer size. Storage clusters using BeeGFS as the PFS can use this configuration for comparable file sizes within other scientific fields that require large data transmission from remote facilities to accelerate the availability of data to geo-distributed scientists.

E. Network Properties

1) Network Manageability: Intermittent network changes during maintenance in a production data center deployment can affect the overall I/O as the network utilization is reduced, which in turn may affect user QoS and incur high cost in dynamic network traffic conditions. Though most providers fall back on MRC based planning to maximize the residual network, it is a costly undertaking especially under dynamic network conditions. *Janus* [28] operates at less cost and optimally plans for maximizing network throughput under reduced network capacity taking advantage of the high degree of symmetry in data center networks. Operating principles of Janus include finding blocks of equivalent switches, finding equivalent sub plans, scaling cost estimation and accounting for failures. Evaluation on large-scale Clos and Facebook traffic show that Janus generates plans in real time and incurs 33 to 70 percent of the cost compared to SoA and can adapt to varying network change policies like different cost functions and deadlines. We believe Janus' potential can be applied to different kind of applications and storage centers that host heterogeneous workloads such as IoT data and streaming data during network downtime.

2) Network Structure: Revisiting BlueDBM [16], we can learn from its low latency high bandwidth transport layer network infrastructure to support its storage controller network. In the packet switching mesh network of BlueDBM, each storage device in the cluster is connected with serial links that form a separate network among themselves, each of them having multiple network ports that can route packets across the network without a separate router/switch. For inter storage device network traffic, this structure removes the overhead of going to the host software to access individual storage devices, hence boosting I/O. The multi-port structure of BlueDBM also creates an environment that can support multiple network topologies by only changing the physical cable links between devices while the routing is dynamically updated by the software component. Further, each endpoint is given the choice of maintaining end-flow-control or doing away with it to reduce latency. It is a trade off that can pay off in terms of higher I/O performance. Similarly, integrated network with storage devices can be implemented in large-scale computing facilities that require faster data access like in data analytics.

F. Network Performance Evaluation – Analysis

Network Traffic Classification (NTC) will play an important role in the growing landscape of diverse network traffic for network planning, network behavior analysis, and network management. With the evolution of applications that arouse high internet traffic, NTC will provide insight into network patterns and can be instrumental in understanding, classifying and dynamically adjusting to the volume, variety and velocity of variable network traffic generated, especially in a shared resource setup.

Shahraki et al. [29] address the challenges imposed by the three popular methods (port-based, pay-load based and statistical modelling based using one or ML models) used for NTC. The paper [29] suggests port-based methods are not the optimal solution as the industry adapts to new protocols and pay-load based methods are not suitable when it comes to encrypted traffic and avoiding heavy overheads. Statistical models in NTC are restricted to basic ML models so far. They propose a deep learning based method to classify the network traffic in communication systems and networks using a combination of multiple convolutional neural networks (CNN) to build an ensemble of classifiers. The outputs from these models are combined to generate a final prediction with an average accuracy of 98 percent on the Cambridge Internet

Traffic dataset. We estimate that this model can be applied to data-intensive applications generating erratic network traffic patterns like in IoT, shared high performance computing facilities for scientific research and other evolving workloads that rely on SDNs to boost storage system performance by efficiently analyzing the network utilization and dynamically adjusting the networking parameters for maximum I/O.

V. Key Insights and Research Challenges

Through this exercise, we get an insight on how closely related network and storage systems are in the HPC architecture for the purpose of I/O optimization. Network optimizations and the application of appropriate network configurations can affect many layers of distributed storage functionality that leads to overall performance boost of the system. Further, with the continuously upgrading workload characteristics and demand of fast, on-demand and real time data accesses from various sectors, I/O throughput becomes one of the key components in judging the feasibility of any system for data intensive applications. As CPU performance scaling has slowed down and almost reached its cap, the HPC community can pour more attention towards research on this topic to accelerate system performance. Software defined networks are very interesting and a viable option for implementation in storage system architectures, data centers, and edge routers due to their potential of dynamically adjusting to changing network characteristics and workloads.

The findings of Madanapalli et al. [25] suggest another choice for configuring the network based on the relationship between BDP, number of TCP streams and TCP socket buffer size to optimize throughput for large data transmission in geo-distributed data centers. Given the expected rise in the number data intensive IoT applications in the near future, we anticipate network load balancing and utilization paradigms will be major drivers towards avoiding network congestion, reducing latency and maintaining the expected quality of service for end users. The HPC community may explore how network load balancers and resource managers can be integrated into large-scale storage system designs that reduce latency due to a large number of internode communications during an application run, and boost the I/O performance for data-intensive applications. However, many of these undertakings may require major system design changes, which are complex and time consuming.

Another challenge will be to determine the best design approach for non-homogeneous workloads hosted on a single HPC storage cluster. Further research is necessary to find the optimal approach that takes into account the networking challenges posed by new and emerging workloads and how they affect the storage system throughput.

VI. Conclusion

This position paper presents a snapshot of the recent network research landscape targeting data-intensive science applications from a network perspective and identifies possible synergy effects between network and parallel file and storage system research. We hope that the identified key insights and research challenges will benefit the software and hardware environment needed to serve data-intensive application architectures from both the network and parallel I/O perspective.

Acknowledgment

This work is sponsored in part by the NSF under the grants: CSR-2106634, CCF-1919113, OAC-2004751, and CSR-1838271. This research used resources of the Oak Ridge Leadership Computing Facility, located in the National Center for Computational Sciences at the Oak Ridge National Laboratory, which is supported by the Office of Science of the DOE under Contract DE-AC05-00OR22725.

References

[1] T. Patel, S. Byna, G. K. Lockwood, and D. Tiwari, “Revisiting I/O Behavior in Large-Scale Storage Systems: The Expected and the Unexpected,” SC ’19, ACM, 2019.

[2] P. Xenopoulos, J. Daniel, M. Matheson, and S. Sukumar, “Big Data Analytics on HPC Architectures: Performance and Cost,” in *2016 IEEE International Conference on Big Data (Big Data)*, pp. 2286–2295, 2016.

[3] P. Xuan, W. B. Ligon, P. K. Srimani, R. Ge, and F. Luo, “Accelerating big data analytics on HPC clusters using two-level storage,” *Parallel Computing*, vol. 61, pp. 18–34, 2017. Special Issue on 2015 Workshop on Data Intensive Scalable Computing Systems (DISCS-2015).

[4] F. Chowdhury, Y. Zhu, T. Heer, S. Paredes, A. Moody, R. Goldstone, K. Mohror, and W. Yu, “I/O Characterization and Performance Evaluation of BeeGFS for Deep Learning,” in *Proceedings of the 48th International Conference on Parallel Processing (ICPP 2019)*, 2019.

[5] R. Ferreira da Silva, R. Filgueira, I. Pietri, M. Jiang, R. Sakellariou, and E. Deelman, “A characterization of workflow management systems for extreme-scale applications,” *Future Generation Computer Systems*, vol. 75, pp. 228–238, 2017.

[6] R. F. da Silva, H. Casanova, K. Chard, T. Coleman, D. Laney, D. Ahn, S. Jha, D. Howell, S. Soiland-Reyes, I. Altintas, D. Thain, R. Filgueira, Y. N. Babuji, R. M. Badia, B. Balis, S. Caíno-Lores, S. Callaghan, F. Coppens, M. R. Crusoe, K. De, F. D. Natale, T. M. A. Do, B. Enders, T. Fahringer, A. Fouilloux, G. Fursin, A. Gaignard, A. Ganose, D. Garijo, S. Gesing, C. A. Goble, A. Hasan, S. Huber, D. S. Katz, U. Leser, D. Lowe, B. Ludäscher, K. Maheshwari, M. Malawski, R. Mayani, K. Mehta, A. Merzky, T. S. Munson, J. Ozik, L. Pottier, S. Ristov, M. Roozmeh, R. Souza, F. Suter, B. Tovar, M. Turilli, K. Vahi, A. Vidal-Torreira, W. R. Whitcup, M. Wilde, A. Williams, M. Wolf, and J. M. Wozniak, “Workflows Community Summit: Advancing the State-of-the-art of Scientific Workflows Management Systems Research and Development,” *CoRR*, vol. abs/2106.05177, 2021.

[7] N. Tsiftes, A. Dunkels, Z. He, and T. Voigt, “Enabling large-scale storage in sensor networks with the Coffee file system,” in *2009 International Conference on Information Processing in Sensor Networks*, pp. 349–360, 2009.

[8] M. Ezell, D. Dillow, S. Oral, F. Wang, D. Tiwari, D. E. Maxwell, D. Leverman, and J. Hill, “I/O Router Placement and Fine-Grained Routing on Titan to Support Spider II,” in *Cray User Group Conference (CUG 2014)*, 2014.

[9] N. Mills, F. A. Feltus, and W. B. Ligon III, “Maximizing the performance of scientific data transfer by optimizing the interface between parallel file systems and advanced research networks,” *Future Generation Computer Systems*, vol. 79, pp. 190–198, 2018.

[10] S. Neuwirth, *Accelerating Network Communication and I/O in Scientific High Performance Computing Environments*. PhD thesis, Heidelberg University, Germany, December 2018.

[11] Y. Charband and N. J. Navimipour, “Online knowledge sharing mechanisms: a systematic review of the state of the art literature and recommendations for future research,” *Information Systems Frontiers*, vol. 18, no. 6, pp. 1131–1151, 2016.

[12] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering – a systematic literature review,” *Information and Software Technology*, vol. 51, no. 1, pp. 7–15, 2009. Special Section - Most Cited Articles in 2002 and Regular Research Papers.

[13] A. Singhvi, A. Akella, M. Anderson, R. Cauble, H. Deshmukh, D. Gibson, M. M. K. Martin, A. Strominger, T. F. Wenisch, and A. Vahdat, “Cliquemap: Productionizing an rma-based distributed caching system,” in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, SIGCOMM ’21, (New York, NY, USA), p. 93–105, Association for Computing Machinery, 2021.

[14] A. Van Bemten, N. erić, A. Varasteh, S. Schmid, C. Mas-Machuca, A. Blenk, and W. Kellerer, “Chameleon: Predictable latency and high utilization with queue-aware and adaptive source routing,” in *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT ’20, (New York, NY, USA), p. 451–465, Association for Computing Machinery, 2020.

[15] S. Borah, S. Dhurandher, I. Woungang, and V. Kumar, “A game theoretic context-based routing protocol for opportunistic networks in an iot scenario,” *Computer Networks*, vol. 129, 07 2017.

[16] S.-W. Jun, M. Liu, S. Lee, J. Hicks, J. Ankcorn, M. King, S. Xu, and Arvind, “Bluedbm: Distributed flash storage for big data analytics,” *ACM Trans. Comput. Syst.*, vol. 34, June 2016.

[17] F. Liang and F. C. Lau, “Bashuffler: Maximizing network bandwidth utilization in the shuffle of yarn,” in *Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing*, HPDC ’16, (New York, NY, USA), p. 281–284, Association for Computing Machinery, 2016.

[18] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, “Ceph: A scalable, high-performance distributed file system,” in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI ’06, (USA), p. 307–320, USENIX Association, 2006.

[19] S. Weil, S. Brandt, E. Miller, and C. Maltzahn, “Crush: Controlled, scalable, decentralized placement of replicated data,” 11 2006.

[20] D. Wu, Y. Wang, H. Feng, and Y. Huan, “Optimization design and realization of ceph storage system based on software defined network,” in *2017 13th International Conference on Computational Intelligence and Security (CIS)*, pp. 277–281, 2017.

[21] P. J.Braam, “The lustre storage architecture (tech. rep.),” tech. rep., Available: http://wiki.lustre.org/., 2004.

[22] S. Kaxiras, D. Klaftenegger, M. Norgren, A. Ros, and K. Sagonas, “Turning centralized coherence and distributed critical-section execution on their head: A new approach for scalable distributed shared memory,” HPDC ’15, (New York, NY, USA), p. 3–14, Association for Computing Machinery, 2015.

[23] I. Kettaneh, A. Alquraan, H. Takruri, S. Yang, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau, and S. Al-Kiswany, “The network-integrated storage system,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 2, pp. 486–500, 2020.

[24] Z. He and L. Peng, “Evaluation of fog topologies in fog planning for iot task scheduling,” in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, SAC ’20, (New York, NY, USA), p. 2177–2180, Association for Computing Machinery, 2020.

[25] S. C. Madanapalli, H. H. Gharakheili, and V. Sivaraman, “Assisting delay and bandwidth sensitive applications in a self-driving network,” in *Proceedings of the 2019 Workshop on Network Meets AI amp; ML*, NetAI’19, (New York, NY, USA), p. 64–69, Association for Computing Machinery, 2019.

[26] A. Farshin, T. Barbette, A. Roozbeh, G. Q. Maguire Jr., and D. Kostić, “Packetmill: Toward per-core 100-gbps networking,” in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS 2021, (New York, NY, USA), p. 1–17, Association for Computing Machinery, 2021.

[27] E. Song, N. Yu, T. Pan, L. Xu, Y. Qiao, J. Lu, Y. Lv, X. Zhang, M. Xie, J. Guo, J. He, J. Mao, C. Jia, and S. Zhu, “A cloud-scale per-flow backpressure system via fpga-based heavy hitter detection,” in *Proceedings of the SIGCOMM ’21 Poster and Demo Sessions*, SIGCOMM ’21, (New York, NY, USA), p. 27–29, Association for Computing Machinery, 2021.

[28] O. Alipourfard, J. Gao, J. Koenig, C. Harshaw, A. Vahdat, and M. Yu, “Risk based planning of network changes in evolving data centers,” pp. 414–429, 10 2019.

[29] A. Shahraki, M. Abbasi, A. Taherkordi, and M. Kaosar, “Internet traffic classification using an ensemble of deep convolutional neural networks,” in *Proceedings of the 4th FlexNets Workshop on Flexible Networks Artificial Intelligence Supported Network Flexibility and Agility*, FlexNets ’21, (New York, NY, USA), p. 38–43, Association for Computing Machinery, 2021.