Journal of Computational Science xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

The MVAPICH project: Transforming research into high-performance MPI library for HPC community

Dhabaleswar Kumar Panda *, Hari Subramoni, Ching-Hsiang Chu, Mohammadreza Bayatpour

Network-based Computing Laboratory, Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA

ARTICLE INFO

Keywords:
HPC
MPI
MVAPICH
RDMA

ABSTRACT

High-Performance Computing (HPC) research, from hardware and software to the end applications, provides remarkable computing power to help scientists solve complex problems in science, engineering, or even daily business. Over the last decades, Message Passing Interface (MPI) libraries have been powering numerous scientific applications, such as weather forecasting, earthquake simulations, physic, and chemical simulations, to conduct ever-large-scale experiments. However, it is always challenging to practice HPC research. In this article, we use the well-established MVAPICH project to discuss how researchers can transform the HPC research into a high-performance MPI library and translate it to the HPC community.

1. Introduction

The field of High-Performance Computing (HPC) has been seeing steady growth during the last 25 years. For example, based on the TOP500 list [1], the number one Supercomputer in 1995 used to deliver only 170 GFlops/s. In the latest June 2020 list, the number one supercomputer delivers 415.5 PFlops/s. The world is also heading into the ExaFlop era soon [2]. Such progress has been made possible by design and development in hardware and software technologies as well as applications.

Most of the parallel applications, during the last 25 years, continue to use Message Passing Interface (MPI) libraries conforming to the MPI Standard [3]. As the underlying hardware technologies (processor and networking) continue to evolve, it is the responsibility of the MPI library to extract and deliver performance, scalability, and fault-tolerance to the parallel applications. Thus, designing a high-performance, scalable, fault-tolerant, and production quality MPI library is important to the progress of the HPC field.

2. Overview of the MVAPICH project and its evolution

HPC systems in the late nineties were using proprietary networking technologies like Myrinet and Quadrics. A new open-standard networking technology called InfiniBand [4] was introduced in October 2000 for datacenters. However, there was no MPI library available to take advantage of this technology for HPC systems. The MVAPICH Project [5] at the Ohio State University took a giant step in 2001 to design an MPI library to exploit the advanced features of the InfiniBand networking technology. The name 'MVAPICH' was chosen to reflect the fact that the original implementation was an MPI implementation over the InfiniBand VAPI interface based on the MPICH implementation. MVAPICH is pronounced as "em-vah-pich".

The MVAPICH open-source MPI library with support for MPI-1 features was introduced to the HPC community at Supercomputing 2002. The MVAPICH library was updated to conform with the MPI-2 standard in 2004 with the release of the MVAPICH2 software stack. Since then, the MVAPICH-2 library has evolved to offer support to newer MPI standards such as MPI 2.1, 2.2, and 3.1. With the increasing adaptation of the MVAPICH2 library, the MVAPICH library, which only supported the MPI-1 standard, was phased out during 2009–2010 and encountered its End of Life (EOL) during June 2010. Enhanced versions of the MVAPICH2 libraries have been introduced in recent years to target Partitioned Global Address Space (PGAS) and Hybrid MPI + PGAS programming models, GPU architectures, Intel MIC architecture, energy-awareness, and virtualization/cloud environments. The MVAPICH team has also designed a comprehensive Micro-benchmark suite, called OSU Micro-benchmark suite (OMB), that supports benchmarking of MPI primitives for various MPI, CUDA, PGAS features.

The MVAPICH project is in its 19th year currently. The MVAPICH2 libraries are currently being used by more than 3100 organizations in 89

* Corresponding author.
E-mail addresses: panda@cse.ohio-state.edu (D.K. Panda), subramon@cse.ohio-state.edu (H. Subramoni), chu.368@osu.edu (C.-H. Chu), bayatpour.1@osu.edu (M. Bayatpour).

https://doi.org/10.1016/j.jocs.2020.101208
Received 22 July 2020; Received in revised form 10 August 2020; Accepted 25 August 2020
Available online 5 September 2020
1877-7503/

countries. It has enabled many TOP500 systems during the last 15 years. The project is supported by funding from U.S. National Science Foundation, U.S. DOE Office of Science, U.S. Department of Defense, Ohio Board of Regents, Ohio Department of Development, AMD, ARM, Broadcom, Cisco Systems, Cray, Intel, Linux Networx, Mellanox, Microsoft, NVIDIA, Pattern Computer, QLogic, and Sun Microsystems;

3. Research innovation of MVAPICH project for HPC

The research of MVAPICH project is primarily driven by the developments in the HPC community as summarized in Fig. 1. The innovations of MVAPICH project can be roughly classified into four categories: *(1) Enabling new programming models for HPC, (2) Leveraging cutting-edge software/hardware technology, (3) Designing High-performance MPI communication middleware, and (4) Powering novel scientific applications using MPI.* As shown in Fig. 1, the MVAPICH project was launched in 2001 to address the need for a high-performance communication layer over Remote Direct Memory Access (RDMA) networks such as InfiniBand [6]. In this section, we highlight the research milestones achieved by the MVAPICH project and its relevant background. More technical details are available in our publications [7].

3.1. MPI for new programming models

After the initial support of the MPI-1 programming model in MVAPICH1 in 2002, there was a need in the community to understand and evaluate the performance of MPI primitives (irrespective of the underlying MPI library) on various HPC platforms in a stand-alone manner. Thus, a comprehensive OSU MPI-level micro-benchmark suite (OMB) was designed and developed in 2003 and was made available with the MVAPICH1 library. Once the new MPI-2 programming model standard started getting adopted by the HPC community, the MVAPICH team introduced a newer version of the library with the MPI-2 support (*MVAPICH2*) for the HPC community [8]. A new generation of PGAS programming models (OpenSHMEM, UPC, and Co-Array Fortran) were introduced to the HPC community in 2010. The MVAPICH2 team quickly adopted these programming models and introduced a newer version called *MVAPICH2-X* to enable both PGAS and hybrid MPI+PGAS programming models [9]. Similarly, once the MPI-3 standard was introduced by the MPI Forum in 2012, the MVAPICH2 team was able to support all MPI-3 primitives in its software release in 2012. Broadly, the MVAPICH project has been keeping pace with the evolution of the programming models and has been able to deliver high-performance implementations of these programming models on the latest generation HPC platforms.

3.2. MPI with cutting-edge hardware technologies

Since 2010, the HPC community has started adopting cutting-edge networking technologies (InfiniBand, iWARP, RoCE, and Omni-Path), processor technologies (x86, OpenPOWER, and ARM), co-processors (Intel Xeon Phi or Many Integrated Core (MIC) architecture), and accelerator (general-purpose Graphics Processing Unit (GPU)), to power the HPC systems and significantly accelerate various scientific applications. These technologies have provided many new mechanisms, including memory management schemes, for efficient data movement within and across computing nodes. The MVAPICH team has conducted relevant research and developments to exploit these mechanisms and provide high-performance communication schemes in the MVAPICH libraries. For example, a novel concept of GPU-Aware (CUDA-aware) MPI design was proposed in [10]. Moreover, the MVAPICH team worked closely with NVIDIA to exploit the GPUDirect RDMA (GDR) technology, enabling peer-to-peer and RDMA-based transfer for high-performance GPU communication, into the publicly available *MVAPICH2-GDR* library in 2014. Similarly, the *MVAPICH2-MIC* library was introduced in 2014 to exploit the Intel MIC architectures. Recently, RDMA technology has been widely applied to cloud environments such as Amazon web services and Microsoft Azure. Accordingly, the MVAPICH team has conducted research of RDMA-based MPI on the cloud [11] and made such optimized designs available through *MVAPICH2-Azure* and *MVAPICH2-X-AWS* libraries. In all these research and development efforts, the MVAPICH project has played a vital 'bridge' between cutting-edge hardware and MPI-enabled applications.

3.3. Designing MPI library with high-performance, scalability, and fault-tolerance

During the past 19 years, various novel MPI-level designs have been incorporated into the MVAPICH project to achieve performance, scalability, and fault-tolerance. In this section, we briefly outline some of these designs and contributions.

Shared-address-space based Communication: Dense multi-/many-core architectures power modern HPC systems, and this trend is expected to grow for future systems. However, existing approaches relying on POSIX shared-memory have various bottlenecks for high core-density systems [12,13]. To address the performance challenges posed by high-core density architectures, we have developed a "shared-address-space"-based intra-node communication framework thatallows a multi-process model of MPI to have thread-like load/store accesses between different address spaces. Further, this framework provides novel designs for various point-to-point and collective communications [14].

Dynamic and Cooperative Protocols: Many HPC applications have irregular and/or dynamic computation and communication patterns that require different approaches during the run-time to maximize performance. The increasing scale of modern HPC systems, as well as the diversity of emerging architectures, have intensified this problem by making a "one-size-fits-all" policy not feasible. MVAPICH team proposes novel application-aware and dynamic designs that adapt to the applications' computation and communication requirements. These designs deliver improved performance of point-to-point and collective communication operations, fast job startup, and scalable fault-tolerant primitives [15–19].

In-network Computing: Emerging scientific and AI applications

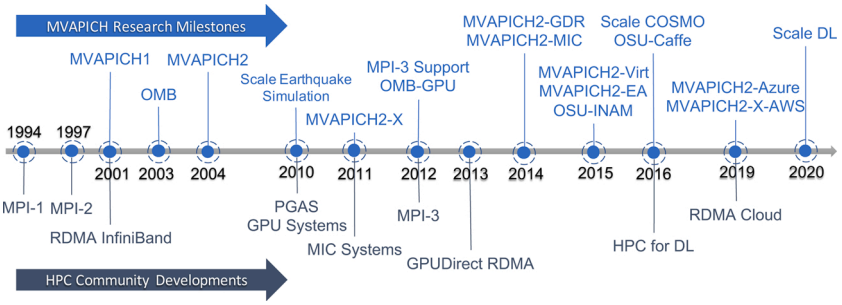


Fig. 1. Research milestones of MVAPICH project along with HPC developments.

that use HPC systems have unprecedented volumes of data that are transferred between various elements of the network. Based on these trends, to increase the scalability and performance of end applications, it is required to bring compute capabilities nearer to the data instead of moving the data to the compute elements. Thus it is crucial to exploit in-network computing features available in modern networking technologies. such as SHARP [20,21], Hardware Multi-cast [22], and Hardware Tag Matching [23,24]. MVAPICH2 libraries have augmented these hardware-based capabilities with novel topology-aware software-based solutions that are dynamically capable of utilizing advanced in-network computing features for progressing MPI operations and scale diverse application communication scenarios.

Fault Tolerance and Resiliency: The MVAPICH2 library provides a set of flexible solutions for fault-tolerance and resiliency, as needed by HPC applications and systems. It provides system-level rollback-recovery capability based on a coordinated Checkpoint-Restart (CR) protocol involving all the application processes. MVAPICH2 also provides an enhanced technique that allows fast checkpoint and restart. It also supports the Fault Tolerance Backplane, which can be used for Checkpoint-Restart and Job Pause-Migration-Restart Frameworks.

3.4. Enabling novel applications and pushing their boundaries using MVAPICH

With continuous research-based solutions being incorporated into the MVAPICH project, we have been empowering and enabling applications in various domains (e.g., from HPC to AI) to efficiently accomplish performance and scalability. For example, in the work of Gordon Bell Prize Finalist for 2010 [25], the MVAPICH2 software successfully helped to scale the earthquake simulation application called AWP-ODC. This production run sustained 220 TFlop/s for 24 h on NCCS Jaguar systems using 223,074 cores. At that time, it was the largest-ever earthquake simulation for earthquake science and engineering. In an international collaboration project with CSCS and Meteo Swiss organizations in Switzerland, the MVAPICH2-GDR library assisted a weather forecasting simulation called COnsortium for Small-scale Modelling (COSMO) to produce high-performance and real-time results in production systems [26]. In [27], we co-designed the well-known Deep Learning (DL) framework, Caffe, with MPI and successfully scaled it to hundreds of GPUs using MVPAPICH2-GDR library while most of the other solutions were only capable of exploiting a few GPUs. This research also opened up the possibility of using MPI library for DL training. Recently, the Horovod framework has adopted MPI-driven solution to scale many emerging DL frameworks such as TensorFlow, PyTorch, and MXNet. In this context, a link-efficient GPU-driven Allreduce scheme [28] proposed in the MVAPICH2-GDR library has been proven to transparently scale distributed DL training up to thousands of GPUs on modern dense-GPU systems.

It is to be noted that Graduate students (Ph.D. and M.S.), Post-doctoral researchers, senior research associated and research scientists have been involved extensively in the publications process of this project. This has resulted into more than 22 Ph.D. Dissertations, 15 M.S. thesis, 10 journal papers and 225 conference/workshop publications.

4. Translation process

Fig. 2 depicts the high-level method we follow to perform the various translational research activities are undertaken. To transform our research into the high-performance MPI library and translate it to the HPC community, the MVAPICH project employs research, development, and release cycle as depicted in Fig. 3 [29]. Four primary phases involved in this process are involved as follows. These phases have been getting repeated over the years as computing and networking technologies and programming model standards are evolving.

Laboratory Research (*three to six months*): In the laboratory, research challenges are identified first, and corresponding research is conducted. This work leads to research prototype solutions. These solutions are tested in the laboratory with existing as well as donated hardware equipment from companies. Established cooperation with various institutions, including national laboratories in multiple countries and companies, enables us to remotely perform and evaluate the solutions in controlled environments (i.e., locale) such as the development partition of the supercomputers. Building on top of the existing MVAPICH library, we can quickly conduct required experiments with the newly proposed solutions.

Fig. 2. Translational research process.

Research to Locale and Community (*three to six months*): Next, these research results are presented in various conferences, workshops, and journals. During this time, the research prototypes are strengthened and converted to software prototypes with different options to run on larger-scale systems and obtain performance and scalability numbers. This phase would take several months due to the peer-review process. To accelerate the process, we actively present the preliminary results in tutorials and invited talks at various venues.

MPI Library to Community (*two to four months*): Based on the feedback received from the community during the above two phases, the best designs from these research results are then incorporated into the production codebase. After the research phase is concluded, the development challenges are identified, and corresponding framework development is conducted. The code is integrated into the respective production software bases. The codebase is then tested on dedicated testing facilities (in the locale laboratory and on remote systems), which represent deployment scenarios. After initial bug fixing and testing, we make the MVAPICH2 library publicly available (through open-sourcing and packaging) to the HPC community as Release Candidates (RC). Moreover, we work with system administrators of many supercomputer centers to install and test the new software releases. The community would conduct further testing. Bugs and issues are reported to public mailing lists (mvapich-discuss@cse.ohio-state.edu, and mvapich-help@cse.ohio-state.edu). After rigorous testing, final "General Availability (GA)" releases of the MVAPICH2 libraries are made. The releases are also accompanied by a set of OSU Micro-benchmarks and a set of performance results linked to the project site. Since the code, benchmarks, and performance results are released publicly, it allows easy reproducibility. MVAPICH releases eventually get picked up by community distributions (OpenHPC), community packagers (SPACK), Linux distributions (such as RedHat and SuSE) and software vendors. Bug-fix branches are also maintained for the releases to provide critical fixes to the community.

Community to Laboratory Research: Since 2013, an annual

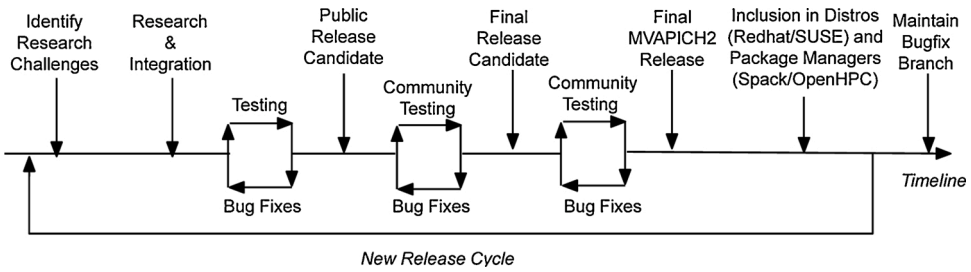


Fig. 3. Established MVAPICH research, development and release process.

MVAPICH User group Meeting [30] has been established to bring together the users and developers of the MVAPICH2 libraries to discuss the latest results, issues, and concerns with the libraries. Moreover, through the highly active discussion list [31] and the annual MUG meeting [30], we continuously interact with the end-users (from academia, supercomputing centers, and industry) to learn and understand the new requirements and challenges from different perspectives. Through the user group meeting and the discussion list, we receive feedback and requests on improving our solutions in MVAPICH2 libraries for real-world applications and on the real-world supercomputers. These requests include, but not limited to, (1) enhancing performance (i.e., low-latency and high-bandwidth data movement), (2) scalability issues on the large-scale supercomputers, (3) new features for cutting-edge hardware and modern applications, (4) deployment on the various systems and (5) bug fixes. Based on the received feedback and new developments in the HPC community, the new research challenges get identified, and the project team members repeat the process mentioned in Fig. 2. Moreover, PIs and senior members discuss with the team to decide the priorities of new research directions based on the requirements from the funding agencies (e.g., NSF and industry), collaborators, end-users, and technology roadmap in the HPC community (e.g., new SW/HW technology, MPI standard).

This translation process involves multiple parties from industry, universities, national laboratories, and supercomputer centers as indicated in Fig. 2. It typically does not happen in a conventional NSF project, mostly terminating at publications and reports rather than being adopted in real-world systems and applications. It is worth noting that the designs and enhancements mentioned in Section 3 were possible due to the translational process integrated into the project. Thus, the project distinguishes itself from being a simple applied research project. We highlight a few of these cases below.

1. **A collaboration with national laboratories and supercomputer centers**: Fast and scalable job startup designs and solutions [17] were made possible only when researchers from Lawrence Livermore National Laboratory (LLNL) and other institutions (Texas Advanced Supercomputing Center (TACC) and San Diego Supercomputing Center (SDSC)) deployed the MVAPICH2 software stack on the supercomputer centers and reported the job start up performance issues at an extremely large scale.
2. **An international collaboration with MeteoSwiss and CSCS** (from Switzerland) enabled the MVAPICH2-GDR software to be thoroughly tested on the production systems for real-time weather forecasting. Based on the feedback received from the engineers and end-users, the MVAPICH2 team then designed a high-performance MPI datatype processing solution [26] and intelligently applied it to the advanced GPU-Aware MPI designs [10] for the real-world weather forecasting applications.
3. **Long-term cooperation with industry**, such as Mellanox, NVIDIA, and AWS has produced high-performance solutions in the MVAPICH2 software libraries. This cooperation allowed the project team to explore cutting-edge software and hardware technologies, features, and protocols, such as GPUDirect technology, GDRCopy, InfiniBand Direct Connect (DC) Protocol, Scalable Reliable Datagram (SRD) protocol supported by the AWS Elastic Fabric Adapter (EFA). It also brings innovative solutions to the MVAPICH2 libraries through the translation process as discussed in this section.

5. Impact and lessons learned

As of August 2020, the MVAPICH2 software libraries are being used by more than 3100 organizations in 89 countries. The list of registered organizations (in a voluntary manner) is available from the 'users' tab of the project's website [5]. Furthermore, more than 810,000 downloads have taken place from the project site. This software is also being distributed by many vendors as part of their software distributions. MVAPICH2 software is also powering many top supercomputers, including the 4th ranked Sunway TaihuLight, 8th ranked Frontera, 12th ranked ABCI, and the 18th ranked Nurion. (based on June 2020 Top500 Rankings). The MVAPICH2 software libraries enable hundreds of thousands of MPI users worldwide to make giant leaps and breakthroughs in their respective domains on a daily basis. The availability of the MVAPICH2 libraries has enabled the InfiniBand industry (introduced in 2001) to grow into a multi-billion dollar industry during the last decade. Currently, 29.8% of TOP500 supercomputers use InfiniBand.

The MVAPICH project has remained sustainable during the last 19 years. It is supported by funding from U.S. National Science Foundation, U.S. DOE Office of Science, U.S. Department of Defense, Ohio Board of Regents, Ohio Department of Development, Switzerland Supercomputing Center (CSCS), arm, Cisco Systems, Cray, Intel, Linux Networx, Mellanox, Microsoft, NVIDIA, Pattern Computer, QLogic, and Sun Microsystems.

Different funding has been used in different stages of the translation process. For example, funding from the NSF core programs have been used for Laboratory Research and Designing Research Prototype (Stages 1 and 2 of the translation process). Funding from NSF CSSI programs, DOE, DoD have been used for designing production quality software and their deployment on production systems (stages 3 and 4 of the translation process). Funding from industry and other collaborating organizations have been used to identify problems from stage 4 of the translation process and repeat it through the complete cycle to have a robust and production quality software solution.

During the last 19 years, many of the designs proposed in the MVAPICH2 library have been adopted by other MPI libraries (open-source and commercial). This situation has led to competition among these libraries for the end-users. However, the MVAPICH project has maintained steady momentum by bringing innovations in its underlying designs through in-depth research. It is worth noting that the students and staff involved in the translation process are greatly benefiting from it. The students' theses are tackling real-world problems using practical solutions. Moreover, the students and staff get exposed to the HPC community directly (industry and academia) and have the requisite skills. The translation process also prepares them to be highly competitive in the job market because they gain experience through countless hands-on exercises as well as production quality software design and development process.

Even though the project has been successful in attracting funding to sustain the project for the last 19 years, we would like to indicate that designing a production-quality software stack in an academic environment is quite challenging. It will be more beneficial if more industries (including cluster integrators), large-scale supercomputer centers, and National Laboratories extend funding for the design, development, and testing of the software stack. Furthermore, it will be beneficial for more industries to work in a collaborative manner with the project team to optimize the software stack for their hardware and platforms. Such collaboration can help the supercomputing centers, research laboratories, and industries to train students for their respective projects in their institutions. For example, several students from the MVAPICH team have joined NVIDIA, Mellanox, Amazon, Microsoft, and AMD in recent years after graduation, and they are the experts in that area from the first day of the work. Finally, with our expertise, industries can take advantage of our proposed designs and get feedback for their product in the early stages. *CUDA-Aware MPI* is an excellent example that the MVAPICH team worked closely with NVIDIA to introduce this novel concept [10]. Based on our research efforts and feedback, NVIDIA developed the GPUDirect technology [32] that has been powering various MPI libraries and applications on GPU-enabled systems for the HPC community [33].

6. Conclusion

In this paper, we discuss how high-impact computer science research can be translated into production-quality software for the community. The MVAPICH project, an academic project sustained during the last 19 years, has been successfully transformed into a production-quality high-performance MPI library for the HPC community. The project involves the standard research and development process, release cycle, participation in open source communities and deployment on production HPC clusters. As a result, the MVAPICH project has proven to benefit many applications in HPC and AI domains. Through a steady and continuous effort, the MVAPICH project is looking forward to sustaining the momentum for powering a diverse set of HPC and AI applications on the next-generation *ExaScale* and *ZettaScale* systems.

Authors' contributions

Conception and design of study: D.K. Panda, H. Subramoni, C.-H. Chu, M. Bayatpour; acquisition of data: D.K. Panda, H. Subramoni, analysis and/or interpretation of data: C.-H. Chu, M. Bayatpour

Drafting the manuscript: C.-H. Chu, M. Bayatpour, H. Subramoni, D. K. Panda revising the manuscript critically for important intellectual content: C.-H. Chu, M. Bayatpour, H. Subramoni, D.K. Panda

Acknowledgment

This research is supported in part by NSF grants #1931537, #1450440, #1664137, #1818253, and XRAC grant #NCR-130002.

References

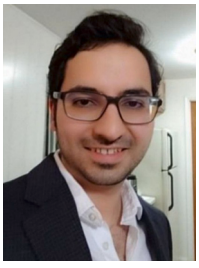
[1] H. Meuer, E. Strohmaier, J. Dongarra, H. Simon, TOP 500 Supercomputer Sites, 2020. http://www.top500.org.
[2] J. Dongarra, S. Gottlieb, W.T.C. Kramer, Race to exascale, Comput. Sci. Eng. 21 (1) (2019) 4–5.
[3] Message Passing Interface Forum, MPI: A Message-Passing Interface Standard, 1994.
[4] InfiniBand Trade Association, http://www.infinibandta.com (2017).
[5] Network-Based Computing Laboratory, MVAPICH: MPI Over InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE, 2001 (accessed 03.09.20), http://mvapich.cse.ohio-state.edu/.
[6] J. Liu, J. Wu, D.K. Panda, High performance RDMA-based MPI implementation over InfiniBand, Int. J. Parallel Program. 32 (3) (2004) 167–198.
[7] Network-Based Computing Laboratory, NOWLABL::Publications, 2001 (accessed 03.09.20), http://nowlab.cse.ohio-state.edu/publications/.
[8] W. Huang, G. Santhanaraman, H. Jin, Q. Gao, D.K. Panda, Design of high performance MVAPICH2: MPI2 over InfiniBand, Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06), vol. 1 (2006) 43–48.
[9] J. Jose, M. Luo, S. Sur, D.K. Panda, Unifying UPC and MPI runtimes: experience with MVAPICH, Proceedings of the Fourth Conference on Partitioned Global Address Space Programming Model (2010) 1–10.
[10] H. Wang, S. Potluri, D. Bureddy, C. Rosales, D.K. Panda, GPU-Aware MPI on RDMA-Enabled clusters: design, implementation and evaluation, IEEE Trans. Parallel Distrib. Syst. 25 (10) (2014) 2595–2605.
[11] S. Chakraborty, S. Xu, H. Subramoni, D. Panda, Designing scalable and high-performance MPI libraries on Amazon elastic fabric adapter, 2019 IEEE Symposium on High-Performance Interconnects (HOTI) (2019) 40–44.
[12] J.M. Hashmi, S. Chakraborty, M. Bayatpour, H. Subramoni, D.K. Panda, Designing efficient shared address space reduction collectives for multi-/many-cores, 2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS) (2018) 1020–1029.
[13] S. Chakraborty, H. Subramoni, D.K. Panda, Contention-aware Kernel-assisted MPI collectives for multi-/many-core systems, 2017 IEEE International Conference on Cluster Computing (CLUSTER) (2017) 13–24.
[14] J.M. Hashmi, S. Chakraborty, M. Bayatpour, H. Subramoni, D.K. Panda, FALCON: efficient designs for zero-copy MPI datatype processing on emerging architectures, 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS) (2019) 355–364.
[15] H. Subramoni, S. Chakraborty, D.K. Panda, Designing dynamic and adaptive MPI point-to-point communication protocols for efficient overlap of computation and communication, High Performance Computing (2017) 334–354.
[16] H. Subramoni, K. Hamidouche, A. Venkatesh, S. Chakraborty, D.K. Panda, Designing MPI library with dynamic connected transport (DCT)of InfiniBand: early experiences. Supercomputing, Springer International Publishing, Cham, 2014, pp. 278–295.
[17] S. Chakraborty, H. Subramoni, A. Moody, A. Venkatesh, J. Perkins, D.K. Panda, Non-blocking PMI extensions for fast MPI startup, 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (2015) 131–140.
[18] S. Chakraborty, M. Bayatpour, J. Hashmi, H. Subramoni, D.K. Panda, Cooperative rendezvous protocols for improved performance and overlap, in: SC18: International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2018, pp. 361–373.
[19] M. Bayatpour, J. Maqbool Hashmi, S. Chakraborty, H. Subramoni, P. Kousha, D. K. Panda, SALaR: scalable and adaptive designs for large message reduction collectives, 2018 IEEE International Conference on Cluster Computing (CLUSTER) (2018) 12–23.
[20] R.L. Graham, et al., Scalable hierarchical aggregation protocol (SHArP): a hardware architecture for efficient data reduction. Proceedings of the First Workshop on Optimization of Communication in HPC, COM-HPC'16, IEEE Press, Piscataway, NJ, USA, 2016, pp. 1–10, https://doi.org/10.1109/COM-HPC.2016.6.
[21] M. Bayatpour, S. Chakraborty, H. Subramoni, X. Lu, D.K. Panda, Scalable reduction collectives with data partitioning-based multi-leader design, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, ACM, 2017, p. 64.
[22] J. Liu, A.R. Mamidala, D.K. Panda, Fast and scalable MPI-level broadcast using InfiniBand's hardware multicast support, 18th International Parallel and Distributed Processing Symposium, 2004. Proceedings (2004) 10.
[23] M. Bayatpour, S.M. Ghazimirsaeed, S. Xu, H. Subramoni, D.K. Panda, Design and characterization of InfiniBand hardware tag matching in MPI, 20th Annual IEEE/ACM International Symposium in Cluster, Cloud, and Grid Computing (2020).
[24] M. Bayatpour, J.H. Maqbool, S. Chakraborty, K.K. Suresh, S.M. Ghazimirsaeed, B. Ramesh, H. Subramoni, D.K. Panda, Communication-aware hardware-assisted MPI overlap engine, ISC High Performance 2020 (2020).
[25] Y. Cui, K.B. Olsen, T.H. Jordan, K. Lee, J. Zhou, P. Small, D. Roten, G. Ely, D. K. Panda, A. Chourasia, et al., Scalable earthquake simulation on petascale supercomputers, in: SC'10: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2010, pp. 1–20.
[26] C.-H. Chu, K. Hamidouche, A. Venkatesh, D.S. Banerjee, H. Subramoni, D.K. Panda, Exploiting maximal overlap for non-contiguous data movement processing on modern GPU-enabled systems, in: 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2016, pp. 983–992.
[27] A.A. Awan, K. Hamidouche, J.M. Hashmi, D.K. Panda, S-Caffe: co-designing MPI runtimes and caffe for scalable deep learning on modern GPU clusters, Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPoPP'17 (2017) 193–205.
[28] C.-H. Chu, P. Kousha, A.A. Awan, K.S. Khorassani, H. Subramoni, D.K.D.K. Panda, NV-Group: link-efficient reduction for distributed deep learning on modern dense GPU systems, Proceedings of the 34th ACM International Conference on Supercomputing, ICS'20 (2020).
[29] D.K. Panda, K. Tomko, K. Schulz, A. Majumdar, The MVAPICH Project: Evolution and Sustainability of an Open Source Production Quality MPI Library for HPC, 2013. https://figshare.com/articles/The_MVAPICH_Project_Evolution_and_Sustainability_of_an_Open_Source_Production_Quality_MPI_Library_for_HPC/791563/5.
[30] Network-Based Computing Laboratory, Annual MVAPICH User Group (MUG) Meeting, 2013 (accessed 03.09.20), http://mug.mvapich.cse.ohio-state.edu/.

[31] Network-Based Computing Laboratory, mvapich-Discuss – Discussion About MVAPICH and MVAPICH2 Software, 2001 (accessed 03.09.20), https://mailman.cse.ohio-state.edu/mailman/listinfo/mvapich-discuss.
[32] NVIDIA, GPUDirect, 2010 (accessed 03.09.20), https://developer.nvidia.com/gpudirect.
[33] J. Kraus, An Introduction to CUDA-Aware MPI, 2013 (accessed 03.09.20), https://developer.nvidia.com/blog/introduction-cuda-aware-mpi/.

Dhabaleswar K. (DK) Panda is a Professor and University Distinguished Scholar of Computer Science and Engineering at the Ohio State University. He has published over 500 papers in the area of high-end computing and networking. The MVAPICH2 (High Performance MPI and PGAS over InfiniBand, Omni-Path, iWARP and RoCE) libraries, designed and developed by his research group (http://mvapich.cse.ohio-state.edu), are currently being used by more than 3100 organizations worldwide (in 89 countries). As of August'20, more than 810,000 downloads of this software have taken place from the project's site. This software is empowering several InfiniBand clusters (including the 4th, 8th, 12th, 18th, and 19th ranked ones) in the TOP500 list. The RDMA packages for Apache Spark, Apache Hadoop and Memcached together with OSU HiBD benchmarks from his group (http://hibd.cse.ohio-state.edu) are also publicly available. These libraries are currently being used by more than 330 organizations in 36 countries. As of August'20, more than 37,100 downloads of these libraries have taken place. MPI-driven approaches to achieve high-performance and scalable versions of Deep Learning frameworks (TensorFlow, PyTorch and MXNet) are available from https://hidl.cse.ohio-state.edu. Prof. Panda is an IEEE Fellow. More details about Prof. Panda are available at http://www.cse.ohio-state.edu/~panda.

Hari Subramoni received the Ph.D. degree in Computer Science from The Ohio State University, Columbus, OH, in 2013. He is a research scientist in the Department of Computer Science and Engineering at the Ohio State University, USA, since September 2015. His current research interests include high performance interconnects and protocols, parallel computer architecture, network-based computing, exascale computing, network topology aware computing, QoS, power-aware LAN-WAN communication, fault tolerance, virtualization, big data, and cloud computing. He has published over 50 papers in international journals and conferences related to these research areas. Recently, Dr. Subramoni is doing research and working on the design and development of MVAPICH2, MVAPICH2-GDR, and MVAPICH2-X software packages. He is a member of IEEE. More details about Dr. Subramoni are available from: http://www.cse.ohio-state.edu/~subramon.

Ching-Hsiang Chu received the Ph.D. degree in Computer Science and Engineering from The Ohio State University, Columbus, Ohio, USA. He received the BS and MS degrees in computer science and information engineering from the National Changhua University of Education, Taiwan, and the National Central University, Taiwan, respectively. His research interests include High-Performance Computing, GPU communication, and wireless networks. He has authored or co-authored over 30 papers in conferences and journals related to these research areas. More details about Dr. Chu are available from: https://kingchc.gitlab.io.



Mohammadreza Bayatpour is a Ph.D. candidate in Computer Science and Engineering from The Ohio State University, Columbus, Ohio, USA. He received the BS degrees in computer engineering from the Sharif University of Technology in Tehran, Iran, and joined the Network-Based Computing Lab at The Ohio State University in 2015. His research interests include High-Performance Computing and parallel computer architecture. He has authored or co-authored over 15 papers in conferences and journals related to these research areas. More details are available from http://web.cse.ohio-state.edu/~bayatpour.1/.