Scaling Study of Flow Simulations on Composable Cyberinfrastructure

Sambit Mishra
PhD candidate, Department of Ocean Engineering, Texas A&M University
sambit98@tamu.edu

Freddie Witherden
Assistant Professor, Department of Ocean Engineering, Texas A&M University
fdw@tamu.edu

Dhruva K. Chakravorty
Director for User Services and Research, High Performance Research Computing, Division of Research, Texas A&M University
chakravorty@tamu.edu

Lisa M. Perez
Director for Advanced Computing Enablement, High Performance Research Computing, Division of Research, Texas A&M University
perez@tamu.edu

Francis Dang
Director for Advanced Computing Systems, High Performance Research Computing, Division of Research, Texas A&M University
francis@tamu.edu

ABSTRACT

Cyberinfrastructure (CI) systems employing composable approaches give researchers the capability to define resources best suited to meet the needs of their computational workflows. Among these approaches, composing disaggregated computing resources over a software-defined network offers the promise of supporting workloads requiring dynamic access to a large pool of accelerators or memory. Much remains to be understood about the architecture-induced constraints of this approach, and how it impacts scientific and engineering applications software. Here, we study the performance of the highly scalable open-source flow solver, PyFR, in a GPU-based composable environment orchestrated using a software-defined PCIe Gen4 fabric. PyFR emphasizes communication between GPUs and helps understand how GPUs on disaggregated resources can be optimally configured for performance. Strong-scaling and weak-scaling performance studies on composed configurations are compared to a traditional CPU-GPU cluster with InfiniBand interconnect. Factors affecting performance, and the need for new benchmark suites for composable devices are discussed.

CCS CONCEPTS

• **Applied computing** → Physical sciences and engineering; • **Computer systems organization** → Architectures; Other architectures; • **Computing methodologies** → Modeling and simulation; Simulation types and techniques; Distributed simulation.

PEARC '23, July 23–27, 2023, Portland, OR, USA

ACM ISBN 978-1-4503-9985-2/23/07.
https://doi.org/10.1145/3569951.3597565

KEYWORDS

Composable disintegrated infrastructure, PyFR, Computational Fluid Dynamics, Scalability, Taylor Green Vortex, Cyberinfrastructure, Resource allocation, Benchmarking

ACM Reference Format:
Sambit Mishra, Freddie Witherden, Dhruva K. Chakravorty, Lisa M. Perez, and Francis Dang. 2023. Scaling Study of Flow Simulations on Composable Cyberinfrastructure. In *Practice and Experience in Advanced Research Computing (PEARC '23), July 23–27, 2023, Portland, OR, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3569951.3597565

1 INTRODUCTION

Traditional distributed architectures allocate a fixed set of computing resources such as CPUs, GPUs and memory, resulting in a rigid configuration of resources for computing workflows. Composable disaggregated infrastructure (CDI) is a powerful approach where a pool of resources, such as compute, memory, accelerators and networking can be allocated to meet the requirements of a research workload. These allocated resources are commonly referred to as "configurations" that can be composed in advance (cold composed) or on-demand (dynamically composed). Such capabilities make composability attractive for research applications that can scale to several GPUs and thousands of cores on advanced cyberinfrastructure resources [1, 2]. Benchmarking the scalability of code on high performance computing (HPC) systems offers opportunities to test the usability of these systems, improve the code performance and optimize workflows [3]. Since CDIs can be composed into computing resources in different ways, i.e. the same number of accelerators can be used in different ways, we take configurations into consideration as we benchmark scalability of application software.

To accurately understand the impact of configurations, it is important to find a workload that is scalable across CPUs and GPUs, moves data to test the interconnects, and offers opportunities to contrast against traditional heterogeneous GPU-enabled HPC topologies connected over InfiniBand. PyFR, an open-source Python-based Computational Fluid Dynamics (CFD) solver employs high-order

Flux Reconstruction technique for solving fluid flow problems on streaming architectures. It efficiently operates on high-performance computing clusters by utilizing distributed memory parallelism through the Message Passing Interface (MPI) and maximizing the overlap of communication and computation. Scalability is achieved by implementing persistent, point-to-point (P2P), non-blocking MPI requests and organizing kernel calls in a manner that enables rank-local computations while exchanging ghost states [4]. PyFR effectively leverages low-latency inter-GPU communication and is seen to be capable of strong-scaling on unstructured grids [5], making it a suitable choice for benchmarking purposes.

A commonly used benchmarking test case is the Taylor–Green Vortex (TGV) breakdown simulation, a scalable flow problem that is typically used to test for the stability and accuracy of flow schemes and models [7, 8]. The case is particularly significant for benchmarking due to its ability to highlight turbulence scales and properties in a simulation with simple grid and flow initial conditions [9]. With strong-scaling up to 18,000 NVIDIA K20X GPUs of Titan, PyFR was among the finalists in the ACM Gordon Bell Prize for High-Performance Computing [6], which highlights its exceptional performance. The current study performs benchmarking studies by strong-scaling and weak-scaling the TGV breakdown simulation using PyFR on a CDI cluster and a traditionally designed distributed cluster, and compares various configurations of NVIDIA A100 GPUs on both the clusters.

2 METHOD

This study employs the PyFR 1.15.0 flow solver to simulate the TGV breakdown case. All essential libraries, build instructions used to set up PyFR, and relevant case files are made available [10]. Performance of CFD simulations is heavily influenced by the hardware it is executed on. The current work utilizes NVIDIA A100 GPUs with 40 GB internal memory. These GPUs can efficiently perform double precision computations and have 40-80 GB of internal memory, both of which are often required to perform large scale reliable and accurate CFD simulations. As such, they make good candidates for a scalability study.

The compressible form of Navier-Stokes Equations is solved numerically at each degree of freedom at each time-step. Flow-field variables are evaluated at every time-step by marching in physical time as per the equation given below:

$$\frac{\partial \boldsymbol{v}}{\partial t} = -\nabla \cdot \boldsymbol{f}\left(\boldsymbol{v},\ \nabla \boldsymbol{v}\right)$$

Here $\boldsymbol{v}$ is a vector of the five flow-field variables and $\boldsymbol{f}$ is the flux of field variables. Flow simulation domain is a cube with dimensions $[0,\ 2\pi]^3$ with periodic boundary conditions along each of the three coordinate axes $(x,\ y,\ z)$. Simulation was performed at a Reynolds number of $Re = 1600$ with initial conditions:

$$P = 1 + \frac{U^2}{16}\left(\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right)\right)\left(\cos\left(\frac{2z}{L}\right) + 2\right),$$

$$u = -U\sin\left(\frac{x}{L}\right)\cos\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right),\ v = -U\cos\left(\frac{x}{L}\right)\sin\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right),$$

$$w = 0,\ \rho = \frac{P}{RT},$$

where $\{P,\ u,\ v,\ \rho\}$ are the five field variables. R, T and U are determined such that Mach number is 0.1, and the reference length scale $L = 1$. The domain was discretized into regular hexahedral elements with $(p+1)^3 = 64$ degrees of freedom (DoF) in each element, where p is the order of polynomial used for the case. Performance of the simulation on the hardware is measured as the total number of computations performed per unit wall-time. More specifically, performance is calculated as

$$Performance = \frac{M}{T},$$

where M is the number of computations performed in 10000 time steps and T is the wall-time (in seconds) taken for the simulation to run the same number of time steps. Performance is calculated in Giga Degrees of Freedom per second (GDoF/s).

Strong-scaling tests were performed on a single NVIDIA A100 GPU with the TGV simulation on PyFR. Mesh size was increased from 18^3 to 192^3 DoF while maintaining the simulation's configuration. This increased the workload on the GPU as described in Figure 1. GPU performance is consistent above a mesh size of 48^3 DoF at around 2.66±0.12 GDoF/s. Thus, all multi-GPU scaling tests were performed with a minimum load of at least 48^3 DoF on each GPU.

2.1 Scaling across multiple GPUs

Scaling simulations were performed on the FASTER [11] and Grace [12] clusters as described in Table 1. FASTER is a 184 node Intel Ice Lake (Xeon 8352Y) composable cluster on which NVIDIA A100 GPUs are composed onto single nodes using the Liqid PCIe Gen4 software-defined Fabric. Grace is a 940-node heterogeneous cluster with Intel Cascade Lake (Xeon 6248R) processors and a variety of NVIDIA GPUs. On Grace, 2 NVIDIA A100 GPUs were available on each node. Both Grace and FASTER use the NVIDIA (Mellanox) HDR100 interconnects to connect servers and the Lustre parallel file system. System administrators manually composed the GPUs on to the FASTER nodes as per the needs of the authors. The Liqid Command Center was used for composing the GPUs to the nodes. A GPU enabled version of the HPL LINPACK test from NVIDIA's HPC benchmark container from NGC [13] was run as a sanity check for the composed GPUs. P2P communication between NVIDIA GPUs was leveraged with the CUDA-aware MPI extension available in MPI 4.1.4, thereby bypassing the need for intermediate host buffers. Unless explicitly mentioned, CUDA-aware MPI was enabled for all simulations. Cubic meshes of required sizes for each of the scaling tests were generated [14] and partitioning of the mesh was outsourced to METIS software [15] to assign computation load to multiple GPUs. Preliminary tests showed that the standard error in mean wall-time per physical time-unit was well below 1% for all simulations and negligibly small differences were found with consecutive runs of the same simulation.

2.2 Strong-scaling

Strong-scaling tests were performed on FASTER-1N and Grace-8N configurations. Three meshes were created as given in Table 1. The mesh for each simulation was divided into partitions that equaled the number of GPUs used.

Strong-scaling test results for each of the three meshes are given in Figure 2. In absence of overheads in inter-GPU communication, strong-scaling of simulations would give a proportional relationship

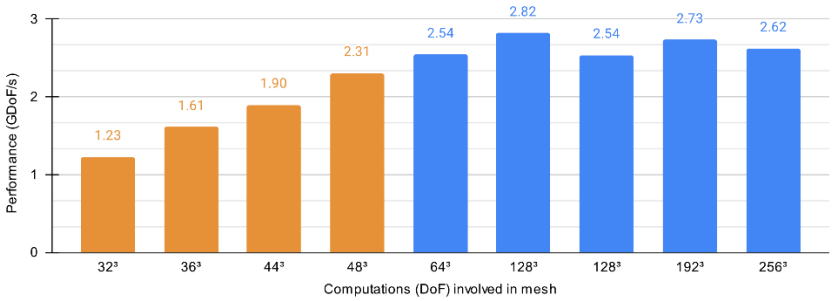

Figure 1: Strong-scaling performance tests on a single GPU. The yellow-marked bars and blue-marked bars correspond to sub-optimal and optimal performance on the A100 GPU with varying computational loads.

Table 1: A comparison between three configurations used for benchmarking, with the naming convention: [Cluster]-[no. of nodes]N. All configurations consisted of 16 NVIDIA A100 GPUs with 40 GB internal memory.

Configuration	Nodes	GPUs connected to node	Node-GPU connection	Inter-node connection
FASTER-1N	1	16	Liqid PCIe Gen4 fabric	None (only one node)
FASTER-4N	4	4	Liqid PCIe Gen4 fabric	InfiniBand
Grace-8N	8	2	Direct connection via PCIe slots	InfiniBand

Table 2: Six sets of meshes created and partitioned to perform all scaling simulations.

Scaling test	Mesh set	Number of meshes in set	Partitions performed on each mesh	Computation load
Strong-scaling	M1	1	9	128^3 DoF
	M2	1	9	256^3 DoF
	M3	1	8*	512^3 DoF
Weak-scaling	G1	9	1	~ 128^3 DoF/GPU
	G2	9	1	~ 256^3 DoF/GPU
	G3	9	1	~ 384^3 DoF/GPU

*The 512^3 DoF mesh fit across multiple GPUs but not within a single GPU due to its 40 GB memory limitation.

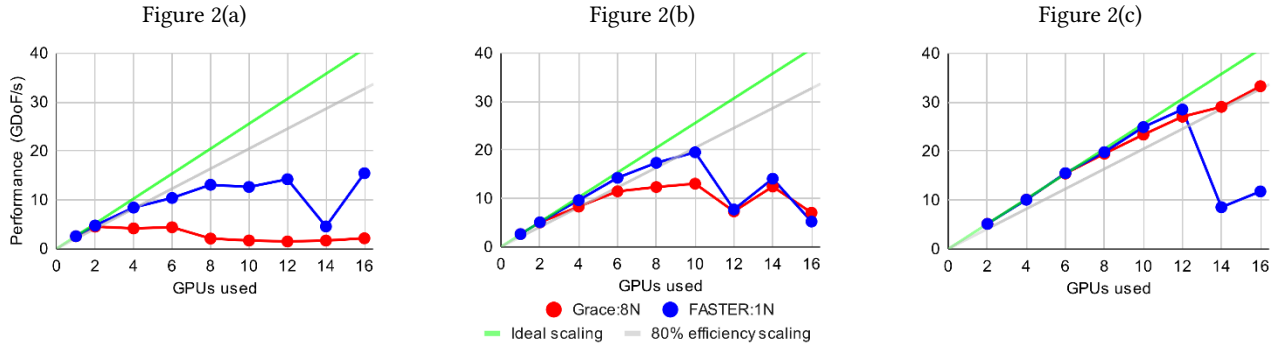

Figure 2: Strong-scaling tests performed on FASTER-1N and Grace-8N on up to 16 NVIDIA A100 GPUs. Figure 2(a): M1 mesh set, Figure 2(b): M2 mesh set, Figure 2(c): M3 mesh set.

between performance and number of GPUs used. When strong-scaling simulations an 80% strong-scale performance efficiency in simulations is usually considered an acceptable trade-off [16]. Single-GPU performance was extrapolated to determine the ideal strong-scaling and 80% efficiency strong-scaling, which were plotted in Figure 2 for reference.

Figure 3(a)

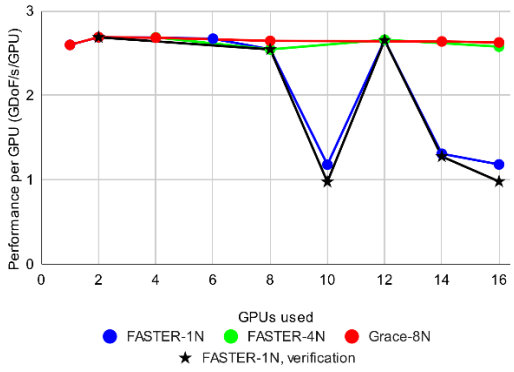

Figure 3(b)

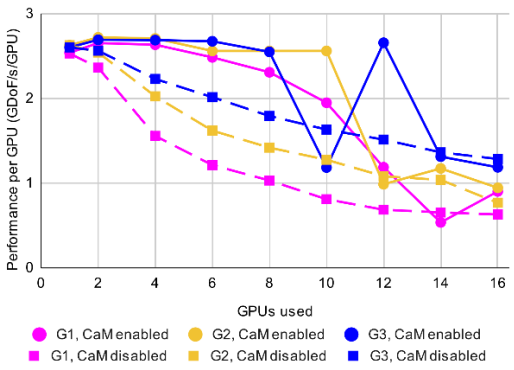

Figure 3: Weak-scaling tests performed on multiple configurations on up to 16 GPUs. CaM: CUDA-aware MPI. Figure 3(a): Tests on G3 mesh set on FASTER-1N, FASTER-4N and Grace-8N with CUDA-aware MPI enabled. Performance of FASTER-1N was verified by switching to a different node. Figure 3(b): G1, G2 and G3 mesh sets on FASTER-1N with CUDA-aware MPI enabled and disabled.

2.3 Weak-scaling

Weak-scaling tests were conducted on three different sets of meshes G1, G2 and G3 described in Table 2. Meshes were created and partitioned such that computation load per GPU remains uniform in each mesh set. Simulations were performed on two composed configurations on FASTER: FASTER-1N and FASTER-4N. A study was performed on Grace (Grace-8N) to establish benchmark. To further verify the performance of FASTER-1N configuration, weak-scaling simulations for G3 mesh set were performed on a different node. Additionally, the impact of enabling CUDA-aware MPI was also investigated by running with FASTER-1N configuration with CUDA-aware MPI enabled and disabled.

3 RESULTS AND DISCUSSION

Strong-scaling and weak-scaling tests were conducted and compared on multiple configurations on FASTER and Grace. Strong-scaling tests run on 1-2 GPUs on FASTER-1N and Grace-8N achieved 98.4% efficiency, with the relative performance difference between the clusters below 0.5%. Tests using 4-12 GPUs on FASTER-1N were seen to consistently perform better than Grace-8N, with the largest relative performance difference of about 1029% observed for M1 mesh strong-scaling simulations at 12 GPUs. Also, with increasing mesh size, the relative performance improvement of FASTER-1N over Grace-8N decreased. Most simulations performed on 12-16 GPUs found the performance of both FASTER-1N and Grace-8N configurations starting to fluctuate with increasing GPU count. Simulations performed on FASTER-1N with more than 12 GPUs and particularly in the case of those performed on the M3 mesh saw a sharp drop in performance by up to ~70% relative to the Grace-8N configuration.

Weak-scaling tests performed on three configurations further exposed the performance drop in large mesh simulations. The tests performed on G3 set of meshes on FASTER-1N were performed on two different nodes to find a trend of abrupt decrease in performance for the simulations weak-scaled to 10, 14 and 16 GPUs. No such drop in performance was observed for FASTER-4N and Grace-8N configurations, suggesting that the source of the performance in FASTER-1N may be due to its high GPU-to-node ratio. The weak-scaling test results obtained on FASTER-1N revealed that, although most tests favored enabling CUDA-aware MPI, those that employed more than 8 GPUs on a single node did not consistently exhibit this advantage over different numbers of GPUs. This finding was unexpected, as enabling P2P communication between GPUs was anticipated to yield better performance compared to transferring memory between GPUs using an intermediate buffer in the node memory. A GPU benchmarking test code was run [17], and the test results revealed bidirectional bandwidth between different GPUs was 30.5±7.3 GB/s with P2P communication enabled and 14.2±0.6 GB/s with P2P communication disabled [8]. Given the superior bandwidth between GPUs with P2P communication enabled, implementing CUDA-aware MPI in PyFR was expected to leverage this P2P communication between GPUs, ultimately leading to a more efficient simulation process. Thus, there was an unexpected lack in performance benefits when CUDA-aware MPI was enabled in some simulations using more than 12 GPUs composed on to one node.

Finally, the benchmarking tests were compared with other benchmarking tests performed on FASTER. The Hovorod Tensorflow Resnet50 model showed good linear scaling behaviour on the FASTER composable up to 20 GPUs connected to a node [19]. In tests performed with a molecular dynamics software suite, LAAMPS, showed an unsteady performance for larger than 8 NVIDIA A100 GPUs connected to a node for one of the three strong-scaling tests [18]. In comparison with these other benchmarking tests, our tests with PyFR showed variance in performance that were not picked up by traditional benchmarking suites like HPL, suggesting that while traditional infrastructure can effectively support the functionalities required for efficient execution of the flow solver, composable infrastructure sometimes has issues doing so for larger than 8 GPUs connected to a node. We hypothesize that

PyFR's persistent, point-to-point (P2P), non-blocking MPI communication between multiple GPUs may not be fully actualized within a composable infrastructure context. As part of our future work, we aim to explore the specific aspects of PyFR contributing to this performance drop in composable infrastructure. Upon understanding these factors, we can develop tailored benchmarking suites to address the efficacy of this feature in the cluster.

4 CONCLUSION

In this study, we performed strong-scaling and weak-scaling tests on composable disaggregated infrastructure with TGV breakdown simulations on PyFR, benchmarking the performance of nodes composed with 16 NVIDIA A100 GPUs. Our findings suggest that while performance of PyFR on composable infrastructure aligned with that of traditional infrastructure when running with fewer than 8 GPUs, spurious performance drops occurred when scaling simulations to large number of GPUs, particularly when PyFR was configured with CUDA-aware MPI enabled. Furthermore, standard benchmarking tests failed to uncover these issues, highlighting the need for new benchmarks tailored for composable infrastructure. Our future work will focus on understanding the specific PyFR features contributing to this performance drop, with the goal of developing specialized benchmarking suites for this novel infrastructure type.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (NSF) award number 1925764 SWEETER - SouthWest Expertise in Expanding, Training, Education and Research, and NSF MRI award NSF award number 2019129 FASTER - Fostering Accelerated Scientific Transformations, Education, and Research, staff and students at Texas A&M High Performance Research.

REFERENCES

[1] National Research Platform. Retrieved April 18, 2023, from https://nationalresearchplatform.org/.

[2] ACES (Accelerating Computing for Emerging Sciences). Retrieved April 18, 2023, from https://hprc.tamu.edu/aces/.

[3] Abhinand Nasari, Hieu Le, Richard Lawrence, Zhenhua He, Xin Yang, Mario Krell, Alex Tsyplikhin, Mahidhar Tatineni, Tim Cockerill, Lisa Perez, Dhruva Chakravorty, and Honggao Liu. Benchmarking the performance of accelerators on national cyberinfrastructure resources for artificial intelligence / machine learning workloads. In Practice and Experience in Advanced Research Computing. ACM, July 2022. https://doi.org/10.1145/3491418.3530772.

[4] F.D. Witherden, A.M. Farrington, and P.E. Vincent. 2014. PyFR: An open-source framework for solving advection–diffusion type problems on streaming architectures using the flux reconstruction approach. 185 (11):3028–3040, November 2014. https://doi.org/10.1016/j.cpc.2014.07.011.

[5] B.C. Vermeire, F.D. Witherden, and P.E. Vincent. 2017. On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools. Journal of Computational Physics, 334:497–521, April 2017. https://doi.org/10.1016/j.jcp.2016.12.049.

[6] P. Vincent, F. Witherden, B. Vermeire, J. S. Park and A. Iyer, "Towards Green Aviation with Python at Petascale," SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, Salt Lake City, UT, USA, 2016, pp. 1-11, doi: 10.1109/SC.2016.1.

[7] Geoffrey Ingram Taylor and Albert Edward Green. 1937. Mechanism of the production of small eddies from large ones. Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences, 158(895):499–521, February 1937. https://doi.org/10.1098/rspa.1937.0036.

[8] Abouelmagd Abdelsamie, Ghislain Lartigue, Christos E. Frouzakis, and Dominique Th évenin. The taylor–green vortex as a benchmark for high-fidelity combustion simulations using low-mach solvers. Computers & Fluids, 223:104935, June 2021. https://doi.org/10.1016/j.compfluid.2021.104935.

[9] Z.J. Wang, Krzysztof Fidkowski, R émi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, H.T. Huynh, Norbert Kroll, Georg May, Per-Olof Persson, Bram van Leer, and Miguel Visbal. High-order CFD methods: current status and perspective. International Journal for Numerical Methods in Fluids, 72(8):811–845, January 2013. https://doi.org/10.1002/fld.3767.

[10] Benchmarking FASTER. Retrieved April 19, 2023, from https://github.com/sambitmishra98/benchmarking.

[11] NVIDIA HPC-Benchmarks 23.3. Retrieved April 21, 2023, from https://catalog.ngc.nvidia.com/orgs/nvidia/containers/hpc-benchmarks.

[12] Grace: A Dell x86 HPC Cluster. Retrieved April 18, 2023, from https://hprc.tamu.edu/wiki/Grace:Intro.

[13] FASTER: A Dell x86 HPC Cluster. Retrieved April 18, 2023, from https://hprc.tamu.edu/wiki/FASTER:Intro.

[14] Basic GMSH code Retrieved April 18, 2023, from https://github.com/WillTrojak/basic_gmsh/blob/main/cube_hex_mesh.py.

[15] Karypis, George; Kumar, Vipin. 1997. METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. Retrieved from the University of Minnesota Digital Conservancy, (1997). https://hdl.handle.net/11299/215346.

[16] Kolev T, Fischer P, Min M, *et al.* Efficient exascale discretizations: High-order finite element methods. The International Journal of High Performance Computing Applications. 2021;35(6):527-552. https://doi.org/10.1177/10943420211020803.

[17] NVIDIA GPU P2P Benchmark. Retrieved April 18, 2023, from https://gist.github.com/joshlk/bbb1aca6e70b11d251886baee6423dcb.

[18] Richard Lawrence, Dhruva K. Chakravorty, Francis Dang, Lisa M. Perez, Wesley Brashear, Zhenhua He, Honggao Liu. Developing Synthetic Applications Benchmarks on Composable Cyberinfrastructure: A Study of Scaling Molecular Dynamics Applications on GPUs. July 23-27, 2023, PEARC Conference Series, Portland, OR, USA. https://doi.org/10.1145/3569951.3597556.

[19] Zhenhua He, Aditi Saluja, Richard Lawrence, Dhruva K. Chakravorty, Francis Dang, Lisa M. Perez, and Honggao Liu. 2023. Performance of Distributed Deep Learning Workloads on a Composable Cyberinfrastructure. July 23-27, 2023, PEARC Conference Series, Portland, OR, USA, 12 pages. https://doi.org/10.1145/3569951.3593601.