Bridge Data Center AI Systems with Edge Computing for Actionable Information Retrieval

A Preprint

Zhengchun Liu
Data Science and Learning division
Argonne National Laboratory
Lemont, IL 60439
zhengchun.liu@anl.gov

Ahsan Ali
Data Science and Learning division
Argonne National Laboratory
Lemont, IL 60439

Peter Kenesei
X-ray Science Division
Argonne National Laboratory
Lemont, IL 60439

Antonino Miceli
X-ray Science Division
Argonne National Laboratory
Lemont, IL 60439

Hemant Sharma
X-ray Science Division
Argonne National Laboratory
Lemont, IL 60439

Nicholas Schwarz
X-ray Science Division
Argonne National Laboratory
Lemont, IL 60439

Dennis Trujillo
X-ray Science Division
Argonne National Laboratory
Lemont, IL 60439

Hyunseung Yoo
Data Science and Learning division
Argonne National Laboratory
Lemont, IL 60439

Ryan Coffee
SLAC National Accelerator Laboratory
Menlo Park, CA 94025

Ryan Herbst
SLAC National Accelerator Laboratory
Menlo Park, CA 94025

Jana Thayer
SLAC National Accelerator Laboratory
Menlo Park, CA 94025

Chun Hong Yoon
SLAC National Accelerator Laboratory
Menlo Park, CA 94025

Ian Foster
Data Science and Learning division
Argonne National Laboratory
Lemont, IL 60439

ABSTRACT

Extremely high data rates at modern synchrotron and X-ray free-electron lasers (XFELs) light source beamlines motivate the use of machine learning methods for data reduction, feature detection, and other purposes. Regardless of the application, the basic concept is the same: data collected in early stages of an experiment, data from past similar experiments, and/or data simulated for the upcoming experiment are used to train machine learning models that, in effect, learn specific characteristics of those data; these models are then used to process subsequent data more efficiently than would general-purpose models that lack knowledge of the specific dataset or data class. Thus, a key challenge is to be able to train models with sufficient rapidity that they can be deployed and used within useful timescales. We describe here how specialized data center AI systems can be used for this purpose.

1 Introduction

The increased coherence and brilliance of next-generation X-ray light sources, such as the APS-U and LCLS-II, will lead to increasingly large and rich data sets produced at high data rates. Such multi-modal data, captured in situ, can provide new insights into rare events such as crack initiation and phase transformations. However, the complexity and velocity of these data means that extracting the desired physical information becomes a considerable computing challenge, particularly when information is needed rapidly, for example to steer experiments. This data extraction challenge is not easily addressed by using conventional analytical methods. Such methods take too long to run on individual processors, and even if efficiently parallelizable (factors like irregular memory accesses often make them latency bound), need many processors to complete analyses in near real-time [Bicer et al., 2017]. Fortunately, machine learning (ML)-based surrogate models can provide an effective alternative to conventional methods in these contexts. A suitably trained ML surrogate can approximate the results of an analytical method with high accuracy, and while it may perform more floating-point operations that the analytical method, can be executed at high speeds on specialized AI accelerators such as GPU, TPU, and NPU [Liu et al., 2020a], which are sufficiently inexpensive to deploy near to data sources.

There is a growing body of work on ML/AI methods for processing of x-ray source data, for example in tomography [Pelt et al., 2018, Liu et al., 2020b, 2019a], serial crystallography [Ke et al., 2018, Souza et al., 2019], and x-ray diffraction [Oviedo et al., 2019, Vecsei et al., 2019]. For example, Pelt et al. [2018] used deep convolutional neural networks to improve tomographic reconstruction from limited measurement (e.g., sparse X-ray projections, short exposure time and limited angles). Others have used DNNs to improve tomography images with budgeted X-ray dosage and to guide data collection and enhance streaming tomography [Liu et al., 2020b, Wu et al., 2020, Liu et al., 2019a]. Using a pre-trained DNN model as a prior, Aslan et al. incorporated learned priors into the generic reconstruction framework for the joint ptycho-tomography problem [Aslan et al., 2020], and have deployed models to AI accelerators (such as Google edgeTPU and NVIDIA Jetson) for low-cost data processing at the edge [Abeykoon et al., 2019]. These methods, when combined with AI accelerators, thus make it feasible, in principle, to analyze data rapidly, at or near the point of data acquisition, rather than streaming them to a remote HPC system.

In order to use ML models effectively, we also need to be concerned with *retraining*, for example when a sample or experiment setup changes. *Training* involves the use of one of a variety of techniques to learn, from supplied training data (often a collection of input-output pairs), a mapping function from inputs to outputs. The output of this process is a trained model that can then be applied to other inputs (a process often referred to as *inference*) to obtain an estimate of the corresponding outputs. In the case of deep neural networks (DNNs), training uses a

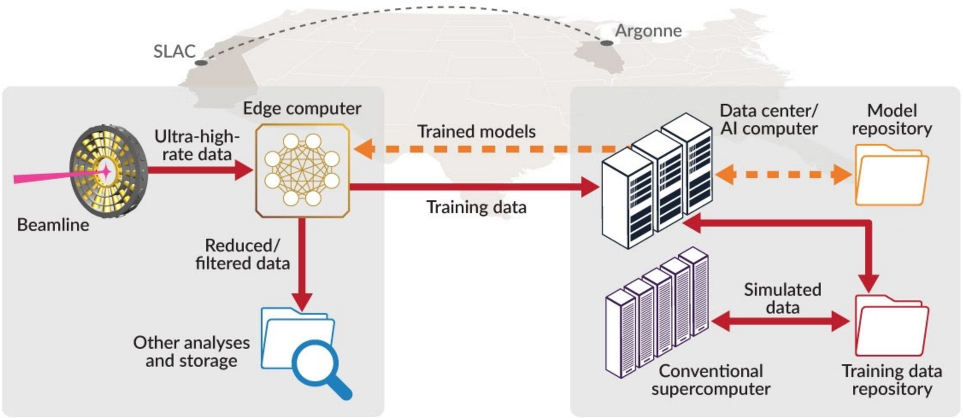

Figure 1: Our actionable information architecture uses an edge computer colocated with the experimental apparatus (here, a SLAC beamline) for rapid reduction/filtering of high-velocity data; the ML model used to perform reduction/filtering is trained on a remote data center AI computer (here, at the Argonne Leadership Computing Facility). Solid crimson arrows are data flows; dashed orange are models.

computationally intensive process called stochastic gradient descent to solve the optimization problem of finding good DNN parameter values.

2 Solution and Gaps

We define a data center AI (DCAI) system as an AI accelerator that that must be deployed in a data center due to its cooling, power supply, ventilation, and fire suppression requirements. DCAI systems can training ML models much more rapidly than computing clusters that are maximally deploy-able within the experiment facility. Once the model is trained, we use another set of AI accelerators specialized for model inference, called `edge-AI` (e.g., FPGA, GPU with Tensor cores, edge TPU [Abeykoon et al., 2019]), to process experiment data near the data acquisition. Since model inference is much less computing intensive than model training and it only needs to be as fast as the data generation rate (i.e., real/wall- time speed), `edge-AI` can be lightweight enough to be deployed within the experiment facility.

The basic concept of utilizing ML for real-time actionable information retrieval is the same irrespective of the application, as illustrated in Figure 1: data generated by a simulation close to the experiment, and/or collected in early stages of an experiment, and/or data from past similar experiments, are used to train models that, in effect, learn specific characteristics of the supplied training data; these models can then be used to process subsequent data more efficiently than general-purpose models that lack knowledge of the specific datasets or classes of data [Liu et al., 2019a, 2020a, Cherukara et al., 2020]. For example, in HEDM, a single scan of 1440–3600 frames can be obtained in 5–10 minutes now and in perhaps 50–100 seconds at APS-U. When measuring a single sample on a layer-by-layer basis, similar data quality is observed repeatedly. Thus, an AI model trained on early layers can be used to process latter layers. In serial crystallography [Nass et al., 2020, Meents et al., 2017], Bragg spots from past experiments can be used for training detectors for use in the current experiment. In single particle imaging [Seibert et al., 2011, Hosseinizadeh et al., 2017], the particles are unique for each experiment; realistic simulations can be used to train a neural network for classifying images of interest. In ptychography, the diffraction patterns collected at early stages of an experiment and their corresponding phase retrieved using the conventional solution of an inverse problem can be used to train a ML model, such as PtychoNN [Cherukara et al., 2020], and the trained model can then be used for phase retrieval for subsequent diffraction patterns much more rapidly.

Model (re)training usually has time constraints that require us to deliver the model within a limited time-frame, particularly for use in-situ experiments. A key challenge is thus to be able to train models quickly enough that they can be deployed and used within useful timescales. The emergence of purpose-built accelerators (e.g., TPU, Cerebras, SambaNova) for high-speed, high-velocity, and/or big data AI models allows for scenarios in which model training is performed on a specialized accelerator while inference is performed at the edge, near the data source, for real-time data analysis. However, usually it is not feasible to deploy these AI systems near an experiment facility because AI systems usually require a significant amount of hardware and software infrastructure, including power subsystems, stable and uninterruptible power supplies, proper ventilation, high-quality cooling systems, fire suppression, reliable backup generators, and connections to external networks. Instead these AI systems, are usually deployed in a data center, for which reason we refer to them as data center AI (DCAI) systems.

In this paper, we proposed an automated method by which DCAI systems are integrated into workflow solutions to facilitate model training on remote DCAI systems and model deployment on edge devices. Thus, one of the important elements to retrieve actionable information in real time is the use of powerful DCAI systems to enable rapid (re)training of ML/AI models that are then deployed on edge devices for production use, as shown in Figure 1. A second major driver for the use of data center computers is for the generation of simulated data for model training. Here we face two distinct needs: high-throughput runs to generate large quantities of simulation data, and low-latency runs to generate results rapidly during an experiment.

3 Workflow Design and Implementation

We used the Globus Flows [Chard et al., 2019a], funcX [Chard et al., 2019b], and Globus file transfer [Foster, 2011] services to realize the workflow to automatically (re)train the ML model with given data or simulations rapidly using DCAI and HPC.

Flows introduces *Action Providers*, *Actions*, and *Flows* to create custom processes solving particular research data management problems. An Action Provider is an HTTP accessible service which acts as a single step in a process and implements the Action Provider Interface. An Action represents a single, discrete invocation of an Action Provider. A Flow represents a single process that orchestrates a series of services into a self contained operation. One can think of a Flow as a declaratively defined ordering of Action Providers with condition handling to define expected success or failure scenarios. A Flow is defined and deployed to the Flows service by any user and can easily and safely be shared among users. Access control to the deployed Flow is provided by Globus Auth [Tuecke et al., 2016], which is also used to authenticate all interactions with Action Provider Actions and Flows. **funcX** provides the function-as-a-service capability to execute functions across a federated ecosystem of funcX endpoints. So we build our computation actions, including simulation, data annotation and model training, using funcX service and wrap it as an action of Globus Flows; The **Globus** transfer service provides a secure, unified interface to the data. It allows us to easily start and manage transfers between endpoints, while automatically tuning parameters to maximize bandwidth usage, managing security configurations, providing automatic fault recovery, and notifying users of completion and problems. We use Globus to transfer data and our ML model between systems within and across organizations. We have wrapped each file transfer requirement as an action of Globus Flows.

Figure 2 shows the overall architecture of the system. Basically, all computing functions, e.g., simulation for training data generation, data curation and model training, are abstracted as a funcX function and wrapped as an action of Globus Flows; and all data dependencies are

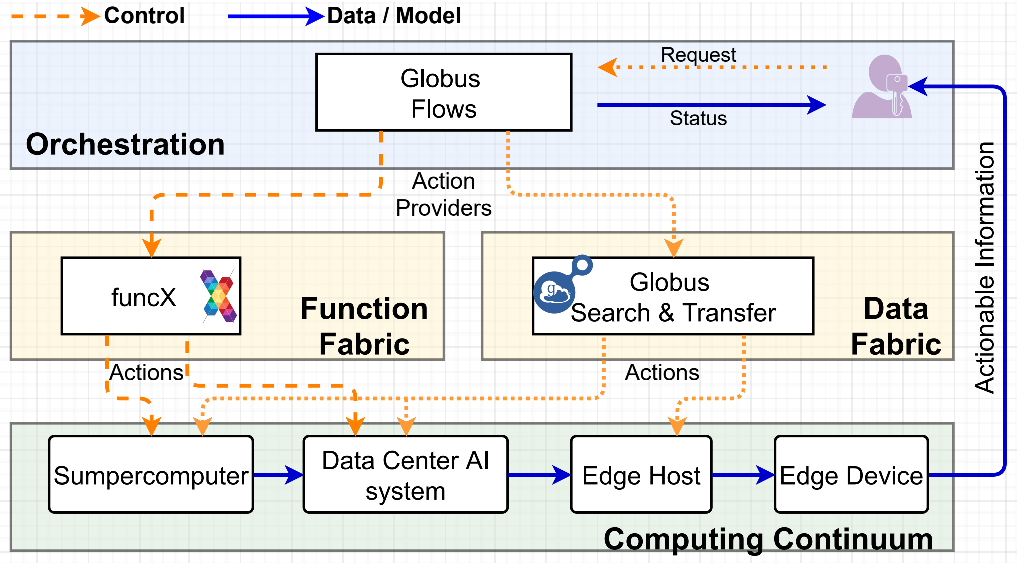

Figure 2: Architecture and building blocks of the workflow. Solid arrows are data flows; dashed arrows are control flows.

solved by wrapping the data transfer as an action of Globus Flows using Globus file transfer service. A developer builds and deploys the workflows to **Flows** and shares with actual users (e.g., experimental scientists). The user only needs to interact with the workflows through a client to initiate the training task and the flow engines will orchestrate resources with action providers to run the workflow and deliver the trained model to the place that defined in the workflow by user.

We open source the workflows shown in this paper at `https://github.com/AISDC/DNNTrainerFlow` and one can use it as a reference to rebuild their own workflow if similar building blocks are used. A recording of the demonstration of the workflow is available at `https://youtu.be/lL6HsIk3xjE`.

4 Experiment and Discussion

We construct the various applications that we consider here in terms of the following six basic operations: **C**ollect a datum; **S**imulate an experiment to generate a datum, d, without an experiment, **A**nalyze a datum using a conventional algorithm (e.g., Bragg peak extraction), generating an analysis (e.g., Bragg peak locations), a; **T**rain (or retrain) a ML model with some number of $\{d, a\}$ pairs, generating a model, m; **D**eploy an ML model an edge-AI device; and **E**stimate an analysis with a previously trained ML model, generating an estimated analysis, $\hat{a}$.

When an operation may be performed at different locations, we denote that by a subscript, such as $\mathbf{A}_{\text{ex}}$ for an analysis performed on a computer at an experiment and $\mathbf{A}_{\text{dc}}$ for an analysis performed in a data center. We represent the movement of data t from location a to location b as $a \xrightarrow{t} b$. Finally, we define $\mathcal{C}(o)$ to be the cost of an operation o. Thus, for example, moving data from experiment to data center for analysis, and then returning result to the experiment (e.g., for steering), will cost $\mathcal{C}(\text{ex}\xrightarrow{d}\text{dc}) + \mathcal{C}(\mathbf{A}_{\text{dc}}) + \mathcal{C}(\text{dc}\xrightarrow{a}\text{ex})$; performing analysis at the experiment will cost $\mathcal{C}(\mathbf{A}_{\text{ex}})$; and estimating an analysis at the experiment will cost $\mathcal{C}(\mathbf{E}_{\text{ex}})$. In comparison, moving a data subset, d', from experiment to data center for analysis and train a model, and then returning a model for **E**stimating the subsequent analysis will cost $\mathcal{C}(\text{ex}\xrightarrow{d'}\text{dc})) + \mathcal{C}(\mathbf{A}_{\text{d}'\text{c}}) + \mathcal{C}(\mathbf{T}_{\text{d}'\text{a}}) + \mathcal{C}(\text{dc}\xrightarrow{m}\text{ex}) + \mathcal{C}(\mathbf{E}_{\text{d - d}'})$

A model for performance analysis will be: Data are generated at SLAC at a rate of R_{slac} B/s. A data subset of size D are transferred to ANL at $R_{\text{slac}\to\text{anl}}$ B/s. Once there, they are analyzed at R_{recon} B/s. The data and reconstruction are then used to train a model of size M, at R_{train} B/s, which is returned to SLAC at $R_{\text{anl}\to\text{slac}}$.

In this section, we will demonstrate one workflow designed for rapid model [Liu et al., 2020a] (re)training of High-Energy X-Ray Diffraction Microscopy (HEDM) data with the above discussed steps. We will compare the performance with using an high-end GPU that can be deployed locally within the experiment facility.

4.1 Experiment Setup

In this experiment, we mimic a case in which the SLAC National Accelerator Laboratory needs to train a model for an experiment. We compare and contrast two scenarios: one in which model training is performed remotely, using a Cerebras CS-1 DCAI system at the Argonne Leadership Computing Facility (ALCF), and one in which training is performed locally, on an NVIDIA V100 GPU co-located with the experiment. We use BraggNN [Liu et al., 2020a], a machine learning-based method designed to localize Bragg peak much more rapidly than conventional pseudo-Voigt peak fitting thus can be run on edge-AI near data acquisition, as a use case to demonstrate the concept of using remote DCAI for rapid model (re)training. The training dataset consists of two files, of size 24 GB and 1.3 MB respectively; it produces a model file of 3 MB.

In the remote case, the DCAI system is 3000 km distant, with a network round trip time over the Energy Sciences Network (ESNet, a high-speed computer network serving United States Department of Energy scientists and their collaborators worldwide) of about 48 ms. As the training dataset is produced at SLAC, we must transfer it from SLAC to ALCF, train the model with the dataset, and transfer the trained model back to SLAC. In the local case, there is no wide-area data transfer overhead.

Table 1: Time breakdown of the model retraining workflow when using either a remote Cerebras CS-1 DCAI system or a local NVIDIA V100 GPU.

Time Mode	Data Transfer	Model Training	Model Transfer	End-to-End
Remote	76s	126s	5s	207s
Local	N/A	4051s	N/A	4051s

4.2 Results and discussion

Table 1 presents the time breakdown for each of the operations/steps in the workflow and compares it with training the same model with the same dataset using a single local NVIDIA V100 GPU. End-to-end time denotes the time that the experimental scientist needs to wait, from when they initiate the (re)training process till they receive the trained model. As one can see, although the data and model transfer (i.e., cost when use remote resources) occupied 1/3 of the end-to-end time, using a remote DCAI system is still 20 times faster than using local resources. Although a local cluster can incorporate many GPUs to provide similar model training times compared to remote DCAI systems, we do not consider the use of such clusters here because the required power, space, and cooling requirements tend to be in limited supply near experimental instruments.

We note that the effective transfer rate of the training dataset is only about 320 MB/s because the dataset consists of one large file and a small file that cannot benefit from concurrent transfers [Liu et al., 2019b, 2017, Kettimuthu et al., 2018]. Since we have the flexibility to divide the training dataset into multiple files, we can perform a joint-optimization between file read/write and file transfer to find the optimal file organization for the dataset.

5 Related work

Wilamowski et al. [2021] used Globus Flows to automate real-time SSX data analysis, but did not use ML models or AI accelerators as here. Rocki et al. [2020] discussed the possibility of using DCAI systems for PDE codes in scientific applications, and demonstrated the benefit over using conventional CPU or GPU based solutions. Emani et al. [2021] explored the suitability of SambaNova, another DCAI system, for diverse AI for Science workloads and observed significant performance gains over traditional hardware. Acciarri et al. [2020] advanced state-of-the-art accuracy for an important neutrino physics image segmentation problem leveraging the large memory of DCAI systems which are not possible to fit on the highest-end GPU because of the large tensor size (convolutions neural networks with images beyond 50k x 50k resolution).

6 Conclusion and Future work

We have presented an automated workflow for rapid deep neural networks training using remote resources, which automates the model (re)training and achieves a turnaround time between initialization and model delivery to edge host of less than 4 minutes including all data movement overhead. As a comparison, it takes 67 minutes when training the same model locally using one high-end GPU that is deploy-able within the data acquisition machine. This workflow proves the feasibility of using powerful yet remote data center AI systems to enable rapid (re)training of deep neural networks for production use on edge devices. We automated the rapid DNN training, using remote data center AI (DCAI) systems (e.g., Cerebras CS-1). The workflow simplifies the use of remotely hosted DCAI systems for domain scientists.

Acknowledgements

This material was based upon work supported by the U.S. Department of Energy, Office of Science, under contract DE-AC02-06CH11357 and Office of Basic Energy Sciences under Award Number FWP-35896. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. The authors would like to thank Ryan Chard (Argonne National Laboratory) for providing funcX tutorial and technical support.

References

Tekin Bicer, Doğa Gürsoy, Vincent De Andrade, Rajkumar Kettimuthu, William Scullin, Francesco De Carlo, and Ian T Foster. Trace: A high-throughput tomographic reconstruction engine for large-scale datasets. *Advanced Structural and Chemical Imaging*, 3(1):1–10, 2017.

Zhengchun Liu, Hemant Sharma, Jun-Sang Park, Peter Kenesei, Jonathan Almer, Rajkumar Kettimuthu, and Ian Foster. BraggNN: Fast x-ray Bragg peak analysis using deep learning. *arXiv preprint arXiv:2008.08198*, 2020a.

Daniel M. Pelt, Kees Joost Batenburg, and James A. Sethian. Improving tomographic reconstruction from limited data using mixed-scale dense convolutional neural networks. *Jour-*

nal of Imaging, 4(11), 2018. ISSN 2313-433X. doi:10.3390/jimaging4110128. URL `http://www.mdpi.com/2313-433X/4/11/128`.

Zhengchun Liu, Tekin Bicer, Rajkumar Kettimuthu, Doga Gursoy, Francesco De Carlo, and Ian Foster. TomoGAN: Low-dose synchrotron x-ray tomography with generative adversarial networks. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, 37 (3):422–434, 2020b.

Zhengchun Liu, Tekin Bicer, Rajkumar Kettimuthu, and Ian Foster. Deep learning accelerated light source experiments. In *IEEE/ACM Third Workshop on Deep Learning on Supercomputers*, pages 20–28. IEEE, 2019a.

T-W Ke, Aaron S Brewster, Stella X Yu, Daniela Ushizima, Chao Yang, and Nicholas K Sauter. A convolutional neural network-based screening tool for x-ray serial crystallography. *Journal of Synchrotron Radiation*, 25(3):655–670, 2018.

Artur Souza, Leonardo B Oliveira, Sabine Hollatz, Matt Feldman, Kunle Olukotun, James M Holton, Aina E Cohen, and Luigi Nardi. DeepFreak: Learning crystallography diffraction patterns with automated machine learning. *arXiv preprint arXiv:1904.11834*, 2019.

Felipe Oviedo, Zekun Ren, Shijing Sun, Charles Settens, Zhe Liu, Noor Titan Putri Hartono, Savitha Ramasamy, Brian L DeCost, Siyu IP Tian, Giuseppe Romano, et al. Fast and interpretable classification of small x-ray diffraction datasets using data augmentation and deep neural networks. *npj Computational Materials*, 5(1):1–9, 2019.

Pascal Marc Vecsei, Kenny Choo, Johan Chang, and Titus Neupert. Neural network based classification of crystal symmetries from x-ray diffraction patterns. *Physical Review B*, 99(24): 245120, 2019.

Ziling Wu, Tekin Bicer, Zhengchun Liu, Vincent De Andrade, Yunhui Zhu, and Ian T. Foster. Deep learning-based low-dose tomography reconstruction with hybrid-dose measurements. In *2020 IEEE/ACM Workshop on Machine Learning in High Performance Computing Environments (MLHPC) and Workshop on Artificial Intelligence and Machine Learning for Scientific Applications (AI4S)*, pages 88–95, 2020. doi:10.1109/MLHPCAI4S51975.2020.00017.

Selin Aslan, Zhengchun Liu, Viktor Nikitin, Tekin Bicer, Sven Leyffer, and Doga Gursoy. Distributed optimization with tunable learned priors for robust ptycho-tomography. *arXiv preprint arXiv:2009.09498*, 2020.

Vibhatha Abeykoon, Zhengchun Liu, Rajkumar Kettimuthu, Geoffrey Fox, and Ian Foster. Scientific image restoration anywhere. In *IEEE/ACM 1st Annual Workshop on Large-scale Experiment-in-the-Loop Computing*, pages 8–13. IEEE, 2019.

Mathew J Cherukara, Tao Zhou, Youssef Nashed, Pablo Enfedaque, Alex Hexemer, Ross J Harder, and Martin V Holt. AI-enabled high-resolution scanning coherent diffraction imaging. *Applied Physics Letters*, 117(4):044103, 2020.

Karol Nass, Lars Redecke, M. Perbandt, O. Yefanov, M. Klinge, R. Koopmann, F. Stellato, A. Gabdulkhakov, R. Schönherr, D. Rehders, J. M. Lahey-Rudolph, A. Aquila, A. Barty, S. Basu, R. B. Doak, R. Duden, M. Frank, R. Fromme, S. Kassemeyer, G. Katona, R. Kirian, H. Liu, I. Majoul, J. M. Martin-Garcia, M. Messerschmidt, R. L. Shoeman, U. Weierstall, S. Westenhoff, T. A. White, G. J. Williams, C. H. Yoon, N. Zatsepin, P. Fromme, M. Duszenko, H. N. Chapman, and C. Betzel. In cellulo crystallization of Trypanosoma brucei IMP dehydrogenase enables the identification of genuine co-factors. *Nature Communications*, 11(1):1–13, dec 2020. ISSN 20411723. doi:10.1038/s41467-020-14484-w.

A. Meents, M. O. Wiedorn, V. Srajer, R. Henning, I. Sarrou, J. Bergtholdt, M. Barthelmess, P. Y.A. Reinke, D. Dierksmeyer, A. Tolstikova, S. Schaible, M. Messerschmidt, C. M. Ogata, D. J. Kissick, M. H. Taft, D. J. Manstein, J. Lieske, D. Oberthuer,

R. F. Fischetti, and H. N. Chapman. Pink-beam serial crystallography. *Nature Communications*, 8(1):1281, dec 2017. ISSN 20411723. doi:10.1038/s41467-017-01417-3. URL http://www.ncbi.nlm.nih.gov/pubmed/29097720http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5668288.

M. Marvin Seibert, Tomas Ekeberg, Filipe R.N.C. Maia, Martin Svenda, Jakob Andreasson, Olof Jönsson, Duško Odić, Bianca Iwan, Andrea Rocker, Daniel Westphal, Max Hantke, Daniel P. Deponte, Anton Barty, Joachim Schulz, Lars Gumprecht, Nicola Coppola, Andrew Aquila, Mengning Liang, Thomas A. White, Andrew Martin, Carl Caleman, Stephan Stern, Chantal Abergel, Virginie Seltzer, Jean Michel Claverie, Christoph Bostedt, John D. Bozek, Sébastien Boutet, A. Alan Miahnahri, Marc Messerschmidt, Jacek Krzywinski, Garth Williams, Keith O. Hodgson, Michael J. Bogan, Christina Y. Hampton, Raymond G. Sierra, Dmitri Starodub, Inger Andersson, Săa Bajt, Miriam Barthelmess, John C.H. Spence, Petra Fromme, Uwe Weierstall, Richard Kirian, Mark Hunter, R. Bruce Doak, Stefano Marchesini, Stefan P. Hau-Riege, Matthias Frank, Robert L. Shoeman, Lukas Lomb, Sascha W. Epp, Robert Hartmann, Daniel Rolles, Artem Rudenko, Carlo Schmidt, Lutz Foucar, Nils Kimmel, Peter Holl, Benedikt Rudek, Benjamin Erk, André Hömke, Christian Reich, Daniel Pietschner, Georg Weidenspointner, Lothar Strüder, Günter Hauser, Hubert Gorke, Joachim Ullrich, Ilme Schlichting, Sven Herrmann, Gerhard Schaller, Florian Schopper, Heike Soltau, Kai Uwe Kühnel, Robert Andritschke, Claus Dieter Schröter, Faton Krasniqi, Mario Bott, Sebastian Schorb, Daniela Rupp, Marcus Adolph, Tais Gorkhover, Helmut Hirsemann, Guillaume Potdevin, Heinz Graafsma, Björn Nilsson, Henry N. Chapman, and Janos Hajdu. Single mimivirus particles intercepted and imaged with an x-ray laser. *Nature*, 470(7332):78–82, feb 2011. ISSN 14764687. doi:10.1038/nature09748.

A. Hosseinizadeh, G. Mashayekhi, J. Copperman, P. Schwander, A. Dashti, R. Sepehr, R. Fung, M. Schmidt, C.H. Yoon, B.G. Hogue, G.J. Williams, A. Aquila, and A. Ourmazd. Conformational landscape of a virus by single-particle x-ray scattering. *Nature Methods*, 14(9), 2017. ISSN 15487105. doi:10.1038/nmeth.4395.

Ryan Chard, Kyle Chard, and Ian Foster. Globus Automate: A distributed research automation platform. *figshare*, 2019a.

Ryan Chard, Tyler J Skluzacek, Zhuozhao Li, Yadu Babuji, Anna Woodard, Ben Blaiszik, Steven Tuecke, Ian Foster, and Kyle Chard. Serverless supercomputing: High performance function as a service for science. *arXiv preprint arXiv:1908.04907*, 2019b.

Ian Foster. Globus Online: Accelerating and democratizing science through cloud-based services. *IEEE Internet Computing*, 15(3):70–73, 2011.

Steven Tuecke, Rachana Ananthakrishnan, Kyle Chard, Mattias Lidman, Brendan McCollam, Stephen Rosen, and Ian Foster. Globus auth: A research identity and access management platform. In *2016 IEEE 12th International Conference on e-Science (e-Science)*, pages 203–212. IEEE, 2016.

Yuanlai Liu, Zhengchun Liu, Rajkumar Kettimuthu, Nageswara Rao, Zizhong Chen, and Ian Foster. Data transfer between scientific facilities–bottleneck analysis, insights and optimizations. In *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 122–131. IEEE, 2019b.

Zhengchun Liu, Prasanna Balaprakash, Rajkumar Kettimuthu, and Ian Foster. Explaining wide area data transfer performance. In *Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing*, pages 167–178, 2017.

Rajkumar Kettimuthu, Zhengchun Liu, David Wheeler, Ian Foster, Katrin Heitmann, and Franck Cappello. Transferring a petabyte in a day. *Future Generation Computer Systems*, 88:191–198, 2018.

Mateusz Wilamowski, Darren A Sherrell, George Minasov, Youngchang Kim, Ludmilla Shuvalova, Alex Lavens, Ryan Chard, Natalia Maltseva, Robert Jedrzejczak, Monica Rosas-Lemus, Nickolaus Saint, Ian T. Foster, Karolina Michalska, Karla J. F. Satchell, and Andrzej Joachimiak. 2'-o methylation of RNA cap in SARS-CoV-2 captured by serial crystallography. *Proceedings of the National Academy of Sciences*, 118(21), 2021.

Kamil Rocki, Dirk Van Essendelft, Ilya Sharapov, Robert Schreiber, Michael Morrison, Vladimir Kibardin, Andrey Portnoy, Jean Francois Dietiker, Madhava Syamlal, and Michael James. Fast stencil-code computation on a wafer-scale processor. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press, 2020. ISBN 9781728199986.

Murali Emani, Venkatram Vishwanath, Corey Adams, Michael E Papka, Rick Stevens, Laura Florescu, Sumti Jairath, William Liu, Tejas Nama, and Arvind Sujeeth. Accelerating scientific applications with sambanova reconfigurable dataflow architecture. *Computing in Science & Engineering*, 23(2):114–119, 2021.

R Acciarri, C Adams, C Backhouse, W Badgett, L Bagby, V Basque, Q Bazetto, A Bhanderi, A Bhat, D Brailsford, et al. Cosmic background removal with deep neural networks in sbnd. *arXiv preprint arXiv:2012.01301*, 2020.

Government License