

Convergence of Artificial Intelligence and High Performance Computing on NSF-supported Cyberinfrastructure

E. A. Huerta^{*†}, Asad Khan^{*‡}, Edward Davis^{*§}, Colleen Bushell^{*}, William D. Gropp^{*¶}, Daniel S. Katz^{*¶**}, Volodymyr Kindratenko^{*¶||}, Seid Koric^{*††}, William T. C. Kramer^{*¶}, Brendan McGinty^{*}, Kenton McHenry^{*}, Aaron Saxton^{*}

^{*}National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

[†]Department of Astronomy, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

[‡]Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

[§]University of Queensland, St Lucia, QLD 4072 Australia

[¶]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

^{||}Department of Electrical & Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

^{**}School of Information Sciences, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

^{††}Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

Abstract—Significant investments to upgrade or construct large-scale scientific facilities demand commensurate investments in R&D to design algorithms and computing approaches to enable scientific and engineering breakthroughs in the big data era. The remarkable success of Artificial Intelligence (AI) algorithms to turn big-data challenges in industry and technology into transformational digital solutions that drive a multi-billion dollar industry, which play an ever increasing role shaping human social patterns, has promoted AI as the most sought after signal processing tool in big-data research. As AI continues to evolve into a computing tool endowed with statistical and mathematical rigor, and which encodes domain expertise to inform and inspire AI architectures and optimization algorithms, it has become apparent that single-GPU solutions for training, validation, and testing are no longer sufficient. This realization has been driving the confluence of AI and high performance computing (HPC) to reduce time-to-insight and to produce robust, reliable, trustworthy, and computationally efficient AI solutions. In this white paper, we present a summary of recent developments in this field, and discuss avenues to accelerate and streamline the use of HPC platforms to design accelerated AI algorithms.

I. INTRODUCTION

The big data revolution disrupted the digital and computing landscape in the early 2010s [1]. Data torrents produced by corporations such as Google, Amazon, Facebook and YouTube, among others, presented a unique opportunity for innovation. Traditional signal processing tools and computing methodologies were inadequate to turn these big-data challenges into technological breakthroughs. A radical rethinking was urgently needed [2], [3].

Large Scale Visual Recognition Challenges [4] set the scene for the ongoing digital revolution. The quest for novel pattern recognition algorithms [5]–[7] that sift through large, high-quality data sets eventually led to a disruptive combination of deep learning and graphics processing units (GPUs) that enabled a rapid succession of advances in computer vision,

speech recognition, natural language processing, and robotics, to mention just a few [8], [9]. These developments are currently powering the renaissance of AI, which is the engine of a multi-billion dollar industry.

Within just a few years, the emergence of high-quality data sets, e.g., ImageNet [10]; GPU-accelerated computing [11]; open source software platforms to design, train, validate and test AI models; improved AI architectures and novel techniques to enhance the performance of deep neural networks, such as robust optimizers and regularization techniques, led to the rapid development of AI tools that significantly outperform other signal processing tools on many tasks. These developments have been astonishing to witness. Data-driven discovery is now also informing and stirring the design of exascale cyberinfrastructure, in which HPC and data have become a single entity, namely HPCD [2], [12].

II. CONVERGENCE OF AI AND HPC

The convergence of AI and HPC is being pursued in earnest across the HPC ecosystem. Recent accomplishments of this program have been reported in plasma physics [13], cosmology [14], gravitational wave astrophysics [15], multi-messenger astrophysics [16], materials science [17], data management [18], [19] of unstructured datasets, and genetic data [20], among others.

These achievements share a common thread, namely, the algorithms developed to accelerate the training of AI models in HPC platforms have a strong experimental component. To date, there is no rigorous framework to constrain the ideal set of hyper-parameters that ensures rapid convergence and optimal performance of AI models as the number of GPU nodes is increased to accelerate the training stage.

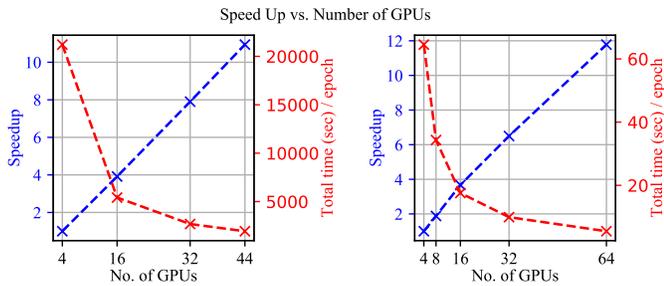


Fig. 1. Left panel: the training of an AI model (which characterizes the signal manifold of spinning binary black hole mergers) is reduced, achieving state-of-the-art performance, from one week to 17 hours by distributing the training workload up to up to 44 V100 GPUs using `Horovod` on the HAL cluster. Right panel: as the left panel, but now the training is reduced from 5 hours to 90 seconds using the entire HAL cluster. This AI model is used to classify and label galaxy images observed by two electromagnetic surveys.

In the context of NSF-supported infrastructure, we present two sample cases of AI and HPC convergence using the Hardware-Accelerated Learning (HAL) cluster [21] at NCSA.

The HAL cluster has 64 NVIDIA V100 GPUs distributed evenly across 16 nodes, and connected by NVLink 2.0 [21] inside the nodes and EDR InfiniBand across the nodes. Using this system, Figure 1 presents two science drivers: (1) an AI model to characterize the signal manifold of binary black mergers that is trained with time-series signals that describe gravitational wave signals [22]; (2) an AI model that classifies galaxy images collected by the Sloan Digital Sky Survey (SDSS), and automatically labels images collected by the Dark Energy Survey (DES) [14]. Using a single V100 GPU, these two models take on average one week and five hours to train, respectively. However, when the training of these models is distributed across HAL, one may fully train them achieving state-of-the-art performance within 17 hours and 90 seconds, respectively.

These examples clearly underscore the importance of coupling AI with HPC, i.e., accelerating the training stage enables: (1) the exploration of domain-inspired architectures and optimization schemes that are critical for the design of rigorous, trustworthy and interpretable AI solutions; (ii) the use of larger training data sets to boost the accuracy and reliability of AI models.

III. SOFTWARE AND HARDWARE CHALLENGES

While open source software platforms have played a key role in the swift evolution of AI, they present a number of challenges when used in HPC platforms. This is because open source software platforms such as `TensorFlow` [23] and `PyTorch` [24] are updated at a much faster pace than libraries deployed cluster-wide on HPC platforms. Furthermore, producing AI models usually requires a unique set of package dependencies. Therefore, the traditional use of modules has limited effectiveness since software dependencies change between projects and sometimes evolve even during a single project. Common solutions to give users more fine-grained

control over software environments include containerization (e.g., `Singularity` [25] or `Kubernetes` [26]), and virtual environments (e.g., `Anaconda` [27], which is extensively used by deep learning practitioners). We provide below a number of recommendations to streamline the use of HPC resources for AI research:

- 1) Provide up-to-date documentation and tutorials to set up containers and virtual environments, and adequate help desk support to enable smooth, fast-paced project life-cycles.
- 2) Maintain a versatile, up-to-date base container image, and base virtual environment that users can easily clone and modify for their specific needs.
- 3) Distributed training software stacks such as `TensorFlow` depend on distributed training software stacks (e.g., `Horovod` [28]), which in turn depend on system architecture and specific versions of MPI installed by the root administrator. It is important to have clear up-to-date documentation on system architecture and MPI versions installed, and clear instructions on how to install/update distributed training software packages like `Horovod` into the user’s container/virtual environment.

In addition to these considerations, the AI model architecture, data set, and training optimizer prevent a seamless use of distributed training. Stochastic gradient decent (SGD) and its variants are the workhorse optimizer for AI training. The common way to parallelize training is to use “mini-batches” with SGD. In principle, a larger mini-batch may naively utilize more GPUs (or CPUs). Training time to solution will often scale linearly with small batch size. Figure 1 shows good generalization at 64 GPUs, which amounts to a global batch size of 128 samples. However, it is known that as data sets and number of features grow, naively scaling number of GPUs, and subsequently batch size, will often take more epochs to achieve an acceptable validation error. The state-of-the art in AI training at scale was reported in [29], who trained `ResNet-50` using a batch size of 64k samples, run across 2048 Tesla P40s. While achieving this level of scaling required a lot of experimental work, this benchmark, and others [30], indicate that scaling AI models to larger data and feature sets is indeed possible. However, it requires a considerable amount of human effort to tune the model and training pipeline. A mixture of fast human model development cycle mixed with automated hyperparameter tuning is a candidate solution to tackle this problem.

IV. CLOUD COMPUTING AND HPC

Cloud computing and containerization became popular for developing customer facing web apps. It allowed a DevOps team to keep strict control of the customer facing software, while new features and bug fixes were designed, developed, and tested in an environment that “looked the same” as a live one. Depending on the business cycle, companies could dynamically scale their infrastructure with virtually no overhead

of purchasing hardware, and then relinquish it when it was no longer needed.

HPC would do well to adopt a DevOps cycle like the ones seen in startup culture. However HPC has some unique challenges that make this difficult. 1) Data storage separated from compute in the form of a shared file system and an instance on maintaining a traditional tree like file system. Cloud computing delivers a unit of compute and storage in tandem as a single instance and isolates distinct resources. A developer using cloud resources treats a compute instance as only the host for their code and must explicitly choose how to move large volumes of data on and off. This is usually done by allocating a specialized cloud instance of a data store (e.g., SQL databases). Improved cloud solutions provide `Kubernetes` (and other cluster manager) recipes to allocate a skeleton of these resources, but it is still up to the developers to choose exactly how data are moved between the resources and to code the specific functions of their app. 2) HPC is a shared resource. That is, many users with different projects see the same file system and compute resource. Each developer must wait their turn to see their code run. In cloud computing, a resource belongs and is billed to the developer on demand. When the resource is released, all of its state-full properties get reset. 3) HPC is very concerned with the compute resources interconnect. To have high bandwidth and low latency between cloud compute instances, one pays a premium.

In the case of distributed training, one needs to ascertain whether the cloud or HPC platforms provide an adequate solution. On-demand, high throughput or cloudbursting of single-node applications are ideally suited for the cloud. For instance, in the case of genetic data analysis, the `KnowEng` platform [20] is implemented as a web application where the compute cluster is managed by `Kubernetes`, and provides an example of a workflow that can be expanded to include methods for intuitively managing library compatibility and cloud bursting. This cloud-based solution includes: (1) the ability to access disparate data; (2) set parameters for complex AI experiments effortlessly; (3) deploy computation in a cloud environment; (4) engage with sophisticated visualization tools to evaluate data and study results; and (5) save results and access parameter settings of prior runs.

However, large distributed training workloads, that run for many hours or days will continue to excel on a high-end HPC environment. For instance, the typical utilization of the HAL cluster at NCSA, which tends to be well above 70%, would require a monthly investment of around \$100k in comparable cloud compute resources; this is far higher than the amortized cost of the HAL cluster and its support.

V. INDUSTRY APPLICATIONS

The confluence of AI and HPC is a booming enterprise in the private sector. NCSA is spearheading its application to support industry partners from the agriculture, healthcare, energy, and financial, sectors to stay competitive on the global market by analyzing bigger and more complex data to uncover hidden patterns, reveal market and cash flow trends, and

identify customer preferences. The confluence of modeling and simulation and AI is another area of growing interest among manufacturing and life science partners, promising to significantly accelerate many extremely difficult and computationally expensive methods and workflows in model-based design and analysis [31]–[33].

Cross-pollination in AI research between academia and industry will continue to inform these activities, making an optimal use of HPC and cloud resources, to design and deploy solutions that transform AI innovation into tangible societal as well as business benefits.

VI. CONCLUSION

The convergence of AI and HPC is strongly poised to fully exploit the potential of AI in science, engineering and industry. Realizing this goal demands a concerted effort between AI practitioners, HPC and domain experts. It is essential to design and deploy commodity software across HPC platforms to facilitate a seamless use of state-of-the-art open source software platforms for AI research. It is urgent to go beyond experimental approaches that lack generality to optimally use oversubscribed NSF resources. An initial step in this direction includes making open source existing solutions that scale well while exhibiting good generalization in mid-scale clusters.

ACKNOWLEDGMENT

EAH, AK, DSK, and VK gratefully acknowledge National Science Foundation (NSF) awards OAC-1931561. EAH and VK also acknowledge NSF award OAC-1934757. This work utilized XSEDE resources through the NSF award TG-PHY160053, and the NSF’s Major Research Instrumentation program, award OAC-1725729, as well as the University of Illinois at Urbana-Champaign.

REFERENCES

- [1] M. Asch, T. Moore, R. Badia, M. Beck, P. Beckman, T. Bidot, F. Bodin, F. Cappello, A. Choudhary, B. de Supinski, E. Deelman, J. Dongarra, A. Dubey, G. Fox, H. Fu, S. Girona, W. Gropp, M. Heroux, Y. Ishikawa, K. Keahey, D. Keyes, W. Kramer, J.-F. Lavignon, Y. Lu, S. Matsuoka, B. Mohr, D. Reed, S. Requena, J. Saltz, T. Schulthess, R. Stevens, M. Swamy, A. Szalay, W. Tang, G. Varoquaux, J.-P. Vilotte, R. Wisniewski, Z. Xu, and I. Zacharov, “Big data and extreme-scale computing: Pathways to convergence-toward a shaping strategy for a future software and data ecosystem for scientific inquiry,” *The International Journal of High Performance Computing Applications*, vol. 32, no. 4, pp. 435–479, 2018.
- [2] National Academies of Sciences, Engineering, and Medicine, *Opportunities from the Integration of Simulation Science and Data Science: Proceedings of a Workshop*. Washington, DC: The National Academies Press, 2018.
- [3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
- [5] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.
- [6] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, May 2015.

- [7] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. [Online]. Available: <https://doi.org/10.1162/neco.1989.1.4.541>
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.
- [9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.
- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.
- [11] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *NIPS*, 2012.
- [12] National Academies of Sciences, Engineering, and Medicine, *Future Directions for NSF Advanced Computing Infrastructure to Support U.S. Science and Engineering in 2017-2020*. Washington, DC: The National Academies Press, 2016.
- [13] A. Svyatkovskiy, J. Kates-Harbeck, and W. Tang, "Training distributed deep recurrent neural networks with mixed precision on gpu clusters," in *Proceedings of the Machine Learning on HPC Environments*, ser. MLHPC'17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: <https://doi.org/10.1145/3146347.3146358>
- [14] A. Khan, E. A. Huerta, S. Wang, R. Gruendl, E. Jennings, and H. Zheng, "Deep learning at scale for the construction of galaxy catalogs in the Dark Energy Survey," *Physics Letters B*, vol. 795, pp. 248–258, Aug 2019.
- [15] H. Shen, E. A. Huerta, and Z. Zhao, "Deep Learning at Scale for Gravitational Wave Parameter Estimation of Binary Black Hole Mergers," *arXiv e-prints*, p. arXiv:1903.01998, Mar 2019.
- [16] E. A. Huerta *et al.*, "Enabling real-time multi-messenger astrophysics discoveries with deep learning," *Nature Rev. Phys.*, vol. 1, pp. 600–608, 2019.
- [17] L. Ward, B. Blaiszik, I. Foster, R. S. Assary, B. Narayanan, and L. Curtiss, "Machine learning prediction of accurate atomization energies of organic molecules from low-fidelity quantum chemical calculations," *MRS Communications*, vol. 9, no. 3, p. 891–899, 2019.
- [18] L. Marini, R. Gutierrez-Polo, R. Kooper, S. Satheesan, M. Burrnette, T. Nichoson, O. M. Zhao, Y., J. Lee, and K. McHenry, "Clowder: Open source data management for long tail data," *PEARC*, 2018.
- [19] S. Padhy, J. Alameda, E. Black, D. M. K. P. Diesendruck, L., R. Kooper, J. Lee, R. Liu, R. Marciano, L. Marini, D. Mattson, B. Minsker, C. Navarro, M. Slavenas, W. Sullivan, J. Votava, and K. McHenry, "Brown dog: Leveraging everything towards autocuration," *IEEE Big-Data*, 2015.
- [20] C. Blatti, A. Emad, M. J. Berry, L. Gatzke, M. Epstein, D. Lanier, P. Rizal, J. Ge, X. Liao, O. Sobh, M. Lambert, C. S. Post, J. Xiao, P. Groves, A. T. Epstein, X. Chen, S. Srinivasan, E. Lehnert, K. R. Kalari, L. Wang, R. M. Weinshilboum, J. S. Song, C. V. Jongeneel, J. Han, U. Ravaioli, N. Sobh, C. B. Bushell, and S. Sinha, "Knowledge-guided analysis of 'omics' data using the knoweng cloud platform," *bioRxiv*, 2019. [Online]. Available: <https://www.biorxiv.org/content/early/2019/05/19/642124>
- [21] NCSA, "HAL Cluster," <https://wiki.ncsa.illinois.edu/display/ISL20/HAL+cluster>.
- [22] A. Das, A. Khan, and E. A. Huerta, "The signal manifold of spinning binary black hole mergers. A Deep Learning Perspective," *In Preparation*.
- [23] M. Abadi, A. Agarwal *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *ArXiv e-prints*, Mar. 2016.
- [24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.
- [25] G. M. Kurtzer, "Singularity 2.1.2 - Linux application and environment containers for science," Aug. 2016. [Online]. Available: <https://doi.org/10.5281/zenodo.60736>
- [26] Kubernetes, <https://kubernetes.io/>.
- [27] Anaconda, <https://www.anaconda.com/>.
- [28] A. Sergeev and M. Del Balso, "Horovod: fast and easy distributed deep learning in TensorFlow," *ArXiv e-prints*, Feb. 2018.
- [29] X. Jia, S. Song, W. He, Y. Wang, H. Rong, F. Zhou, L. Xie, Z. Guo, Y. Yang, L. Yu, T. Chen, G. Hu, S. Shi, and X. Chu, "Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes," 07 2018.
- [30] Y. You, Z. Zhang, C.-J. Hsieh, J. Demmel, and K. Keutzer, "Imagenet training in minutes," 2018. [Online]. Available: <https://doi.org/10.1145/3225058.3225069>
- [31] D. W. Abueidda, S. Koric, and N. A. Sobh, "Machine learning accelerated topology optimization of nonlinear structures," *arXiv e-prints*, p. arXiv:2002.01896, Jan 2020.
- [32] S. Luo, J. Cui, M. Vellakal, J. Liu, E. Jiang, S. Koric, and V. Kirdratenko, "Review and Examination of Input Feature Preparation Methods and Machine Learning Models for Turbulence Modeling," *arXiv e-prints*, p. arXiv:2001.05485, Jan 2020.
- [33] S. G. Rosofsky and E. A. Huerta, "Artificial neural network subgrid models of 2-D compressible magnetohydrodynamic turbulence," *arXiv e-prints*, p. arXiv:1912.11073, Dec 2019.