# Contents

**Summary**: The *CAIRO* coalition of experts in the fields of algorithmic theory, artificial intelligence (AI), and high-performance computing (HPC) aims to transform research and education in the broader field of AI through the co-design of a new class of higher-order algorithms in conjunction with a secure, scalable, and accessible advanced cyberinfrastructure (CI) that we anticipate will result in a significant portion of new and existing AI applications achieving **speedup of two to three orders of magnitude**.

The extensive library of higher-order algorithms co-developed by *CAIRO* 's experts will be made available through a truly distributed, widely applicable, and openly available system and infrastructure. It will support a wide variety of target architectures, be easy to deploy and maintain, and provide good portability, productivity, and scalability on hardware ranging from laptops, to leadership HPC clusters and CPUs, to specialized accelerators. Our initial aim will be to provide end users with our algorithms through an API compatible with NumPy, PyTorch, and Tensorflow, thus ensuring a reasonably simple adoption path for the majority of users. The *CAIRO* CI will democratize access to a new optimization system, facilitate its use within new and existing applications, and provide a platform for training, collaboration, and benchmarking. It will ensure that AI researchers have standards-based tooling capable of empowering them to solve today's problems in a fraction of their current times and successfully tackle problems currently beyond the realm of possibility. The *CAIRO* CI deliverables will be incorporated into the multi-institution graduate program, summer AI workshops, and workforce development activities coordinated by the institute.

*CAIRO*'s innovative partnership with the LSU Ethics Institute will produce project-driven, publishable research in the areas of AI ethics, AI ethical risk management, and cutting-edge AI Ethics training for our inter-institutional partners across eight universities.

**Intellectual Merit**: The proposed AI research is both foundational and use-inspired, driven by four science drivers in the areas of natural language processing, hurricane forecasting, sustainable energy, and geohazard simulation. It provides transformative intellectual contributions in areas of AI, distributed systems, and algorithms that will enable a leap in size, complexity, and accuracy of solvable AI problems, ensuring a substantial reduction of the time-to-solution for a broad set of AI applications of national importance.

One of the most salient and innovative features is the close coupling of AI, Algorithms, HPC, and visualization tools. *CAIRO* 's AI engine, powered by scalable higher-order methods, is based on an optimization pipeline that communicates between the layers of the software stack. Visualization and performance tools close the loop, incorporating user feedback. *CAIRO* answers to the need to develop new algorithmic solutions on a wide spectrum of computing systems. This collaboration between theorists, computer scientists, and AI researchers will serve as a high-impact use case for how to improve efficiency of AI on HPC and will inform new approaches to efficient, scalable computing in general. *CAIRO* 's advances in AI and parallelization technology for higher-order optimization algorithms will significantly impact the science of modeling, AI, and HPC. The advancements in algorithmic research will directly benefit domains beyond the field of AI, such as data analysis and modeling, statistics, and the general field of theoretical mathematics.

**Broader Impact**: *CAIRO* has the potential to impact domains far beyond those represented by our four science drivers. Providing scientists and application developers with new insight and services for optimizing AI/ML applications with excellent scalability and parallel efficiency, *CAIRO* will enable new types of applications to be written and maintained. *CAIRO*'s proposed workflow is designed so that code will be able to perform efficiently on current and future architectures. The new, higher-order optimization techniques and their application to four distinct science domains, combined with novel system implementations ensuring best possible scalability across a wide range of computing resources, lays the foundation for programmer productivity and portability of both codes and their performance, greatly reducing maintenance burdens.

*CAIRO* will also have a number of direct societal benefits. Funding this research fosters the growth and development of AI and HPC in Louisiana, home of *CAIRO* (LSU) and an EPSCoR state. Additionally, the project will directly provide undergraduate, graduate, and post-graduate opportunities to the citizens of eight States, beyond just Louisiana, which is vital in fostering existing and creating new industries with AI/ML technology. *CAIRO*, in particular, lays a solid foundation for technology transfer from academia to industry. By creating a software layer that industrial partners can confidently rely on, the project will help fill the gap between academic innovation and commercial application.

*CAIRO* funds will further support societal values through targeted inventions aimed at broadening participation in AI and investing in AI ethics research and training, thus actively building the next generation of talent for a diverse, well-trained workforce.

**Keywords**: Advanced Cyberinfrastructure for Artificial Intelligence, Scalable Higher-Order Optimizations

# 1 Overview and Rationale for Institute Approach

Some of today's most visible and, indeed, remarkable achievements in artificial intelligence (AI) have come from advances in deep learning (DL).The formula for the success of DL has been compute power—artificial neural networks are a decades-old idea, but it was the use of powerful accelerators, mainly GPUs, that truly enabled DL to blossom into its current form [165]. As significant as the impacts of DL have been, there is a realization that current approaches are merely scratching the surface of what might be possible and that researchers could more rapidly conduct exploratory research on ever larger and more complex systems—if only more compute power could be effectively applied. There are three emerging trends that, if properly harnessed, could enable a such a boost in compute power applied to AI, thereby enabling another major advance in AI capabilities.

**Optimization algorithms based on higher-order derivatives** are well-established numerical methods, offering superior convergence characteristics and inherently exposing more opportunities for scalable parallel performance than first-order methods commonly applied today. Despite their potential advantages, these algorithms have not yet found their way into mainstream AI applications, as they require significantly more powerful computational resources and must manage significantly larger amounts of data.

**High-performance computing (HPC)** brings more compute power to bear via parallel programming techniques and large-scale hardware clusters and will be required to satisfy the resource requirements of higher-order methods. That DL is not currently taking advantage of HPC resources is not due to lack of imagination or lack of initiative in the community. Rather, matching the needs of DL systems with the capabilities of HPC platforms presents significant challenges that can only be met by coordinated advances across multiple disciplines.

**Hardware architecture advances** continue apace, with diversification and specialization increasingly being seen as a critical mechanism for increased performance. Cyberinfrastructure (CI) and runtime systems that insulate users from hardware changes, coupled with tools that support performance evaluation and adaptive optimization of AI applications, are increasingly important to achieving high user productivity, code portability, and application performance.

Only a use-inspired, synergistic collaboration of research expertise encompassing AI, numerical algorithms, HPC, computer systems, CI, and more will be able to realize the necessary breakthroughs in size, complexity, and capability to amplify the impact of AI on science, technology, industry, and society. That is, matching DL to HPC requires a focused national-scale research institute. Accordingly, we propose the *"National Coalition for Artificial Intelligence Research on Scalable Optimizations (CAIRO)"*, a nationwide Research Institute that will enable AI to take full advantage of current and future HPC platforms via novel scalable CI specialized for higher-order algorithms.

The **Vision of *CAIRO*** is to ensure a significant and transformative **reduction of the time-to-solution by two to three orders of magnitude** for a broad set of AI applications by *co-designing* a new class of algorithms based on higher-order optimization methods alongside new software infrastructure to support their efficient use across a spectrum of computational resources. This will **enable a fundamental advance in the size, complexity, and accuracy of solvable AI problems**, catalyze a fundamental shift in the training and application of AI models, and create new educational pathways to develop a more diverse AI-savvy STEM workforce. *CAIRO* will realize its vision through **use-inspired research** originating from four challenging science drivers of national importance.

## 1.1 Objectives

Program objectives include (see also Figure 1): **(1)** Development of the next generation of higher-order optimization algorithms, focusing on specific needs of AI; **(2)** Incorporation of the algorithms into an AI-focused CI, offering efficiency and scalability on current and next-generation supercomputers; **(3)** Deployment of the algorithms and CI in the form of easy-to-use web-services and software libraries, making them available to the global scientific community; **(4)** Significant advances in the domain areas of our science drivers; **(5)** Education for the next generation of a highly-skilled STEM workforce, all of whom will require AI expertise; and **(6)** Production of project-driven, publishable research in the area of AI ethics and managing ethical risks in AI through an innovative partnership with the LSU Ethics Institute. Preliminary performance results suggest that by the end of the project we can expect to see speedups of our applications by two to three orders of magnitude, with expected order-of-magnitude increases resulting from each of: the new

optimization algorithms, our scalable software infrastructure realizing those algorithms, and performance portability enabled by the infrastructure that will allow applications to rapidly and fully utilize current and future advanced heterogeneous hardware architectures.

## 1.2 Overview of Approach

**Scalable Optimization Algorithms.** The kernels (both for training and inference) in many DL applications are essentially linear algebra operations that have been highly tuned to take advantage of modern CPU, GPU, and specialized tensor processing hardware. To a certain extent, inference problems compute efficiently, though their size may be limited by available memory. In contrast, training requires substantially more computation, and the standard numerical optimization algorithms for training, e.g., stochastic gradient descent (SGD), appear to have characteristics that inherently limit their scalability beyond a relatively small number of compute nodes. With the *CAIRO* project, we propose to advance the theory and practice of using second-order optimization algorithms for scalable DL training, enabling larger and more complex models to be trained in significantly less time. We will develop and deploy an extensive library of second-order algorithms, evaluate and characterize their convergence behavior and numerical performance on significant problems of interest, and maintain an online dashboard of results and trained networks. Particular approaches will include Newton-Krylov, quasi-Newton/secant, conjugate direction, inexact Newton, and other novel methods to be developed during the course of this work. Since materializing the Hessian that is at the core of second-order methods is not feasible, novel approaches to Hessian-free and limited-memory variants of these algorithms will be investigated. Approaches for improving global convergence behavior, such as trust region, line-search, filter methods, and combination/hybrid methods, will also be developed. Our catalog of algorithms will be available through a set of web services as well as libraries that can be easily integrated with existing applications and mainstream frameworks.

**Scalable Infrastructure.** Although second-order approaches offer the potential for better convergence behavior, they have the reputation of being more expensive computationally. However, the additional computational cost is ameliorated by fewer training epochs and reduced communication—and by offering much more opportunity for scalability—but a suitable infrastructure is required to support it. A truly distributed, widely applicable, and openly available system and infrastructure for processing very large amounts of data in the field of AI has yet to emerge. Such a system must support a wide variety of target architectures, be easy to deploy and maintain, and provide good portability, productivity, and scalability even on the largest HPC resources available. We will develop such a distributed framework and infrastructure that exposes a set of high-performance algorithmic primitives, usable by our science drivers and the broader community.

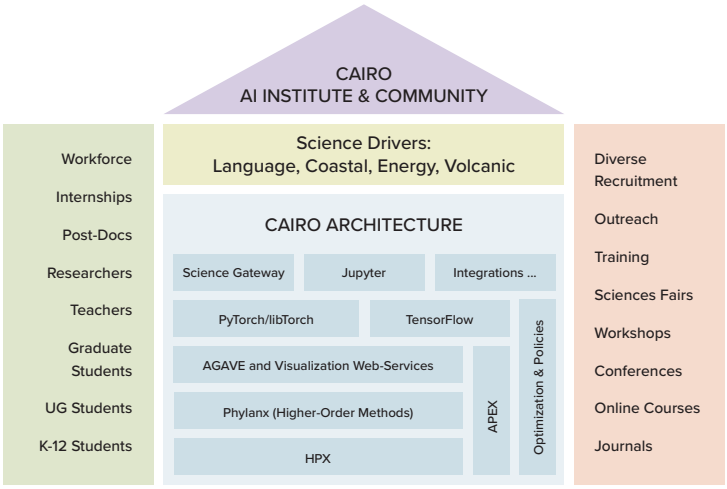



Figure 1: *CAIRO* targets broad outreach activities enabling its integration with other AI institutes, educating the next generation of a highly-skilled STEM workforce with AI expertise.

**Science Drivers (SD).** Our work in *CAIRO* will be **use-inspired** and informed by four science drivers: **SD1** Natural Language Processing (see Section 5.1), **SD2** Hurricane Storm Surge and Flood forecasting (see Section 5.2), **SD3** Sustainable energy resource applications (see Section 5.3), and **SD4** Geohazards and volcanological simulations (see Section 5.4). These science drivers reflect previous strong collaborations among the participating organizations, providing a strong foundation for the proposed research program that relies on close collaboration between theorists, computational scientists, and domain scientists. The selected applications have a strong impact on different areas of national importance and will greatly benefit from the results of the proposed work. The Coalition will effect strong synergistic connections among the researchers, enabling transformative advances in the targeted domains that will have direct impact on other science and application domains.

**State of the Art Teaching and Training.** Our comprehensive education and broadening participation plans (see Sections 6 and 7) position *CAIRO* at the nexus of recruiting, retaining, and training a diverse cohort of undergraduate, graduate, and postdoctoral students, helping to produce the next generation of AI researchers and workers. The distributed structure of *CAIRO* is a major asset for achieving broader social, economic, and educational impacts. Led by LSU, an R1 University in an EPSCoR state, we bring together researchers and research institutions that serve over 220,000 undergraduate and graduate students.

### 1.3 Institute Capabilities

The *CAIRO* Institute connects a **coherent multidisciplinary team of scientists, engineers, and educators** with internationally-recognized expertise in machine learning, high-performance computing, runtime systems, performance analysis and visualization tools, optimization, computational science, and key scientific domains. The members of the Coalition are (see Figure 2): Louisiana State University (LSU, lead institution, ❶), George Mason University (GMU, ❷), Chapman University (CU, ❸), Missouri S&T (MST, ❹), University of Arizona, Tucson (UA, ❺), University of Washington (UW, ❻), University of Oregon (UO, ❼), and University of Texas at Austin (UT, ❽) The Coalition will **create significant new research capabilities** by utilizing their existing leading positions in the field and by building on synergies between their various domains of expertise. *CAIRO* also benefits from having access to existing national and regional HPC resources, e.g., at LSU, CU, and UO. In addition, TACC and LONI have generously offered to allow leveraging their infrastructure (see letters of collaboration). The team has an established track record of delivering high-quality open-source software (see data management plan).

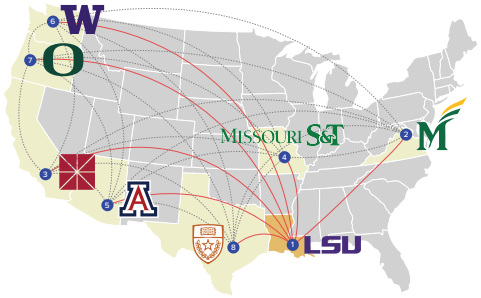



Figure 2: *CAIRO* will be a truly nationwide center of excellence connecting eight universities across the country.

## 2 Background

The detection of AI solutions to what were once considered impossible problems is an almost common occurrence in today's AI landscape. All of us in the computer science and engineering community can rightly feel a measure of pride in these accomplishments—they have been made possible because of the availability of efficiently programmed, immensely powerful hardware. GPU technology in particular has been noted as being responsible for the current "renaissance" in DL [165]. At the same time, current DL approaches are not taking advantage of modern HPC platforms to the same extent that scientific computing is. If DL were to fully use HPC platforms, we would likely see another "punctuational change" in AI, similar to what was experienced when GPUs arrived on the scene. Enabling this punctuational change will require coordinated advances in algorithms and cyberinfrastructure. In this section, we review current approaches, obstacles to scaling them, and opportunities available with higher-order methods.

### 2.1 First-Order Optimization Methods

Training a DNN is an optimization process that seeks to find the set of parameters $\boldsymbol{\theta}^*$ such that

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \mathscr{L}\left(\mathscr{N}(\boldsymbol{\theta}, \boldsymbol{X}), \mathscr{H}(\boldsymbol{X})\right) = \arg\min_{\boldsymbol{\theta}} \frac{1}{|\boldsymbol{X}|} \sum_j \ell\left(n(\boldsymbol{\theta}, \boldsymbol{x}_j), h(\boldsymbol{x}_j)\right)$$

Here, $J : \mathbb{R}^{m_L} \to \mathbb{R}$ is an objective associated with $\mathscr{N}$ that encodes the task for which we wish to train the network. In the case of supervised learning (e.g., which we show here without loss of generality), the objective is defined in terms of a loss function $\mathscr{L}$ that measures the difference between the outputs of the model and some prescribed targets. Also without loss of generality, we can define $\mathscr{L}$ as the average of a single-sample loss function $\ell$ over the training set $\boldsymbol{X}$.

The general form of optimization algorithms for training is shown in Algorithm 1. The primary computational steps are: (line 2) obtain some number of samples from the data set to create a minibatch, (3) apply the model to each sample to compute a corresponding output, (4) compute the loss for the outputs, (5) compute the gradient corresponding to the loss (typically via backpropagation), (6) determine a new

**Algorithm 1** Proto-Optimization loop for DL training.

1: **for** $k = 0$ **to** $\frac{|\boldsymbol{X}|}{|\boldsymbol{B}|} \cdot \text{epochs}$ **do**
2: $\quad \boldsymbol{B}^k \leftarrow \mathcal{S}(\boldsymbol{X})$ ▷ Sample batch from training set $\boldsymbol{X}$ || Data parallel
3: $\quad \boldsymbol{Y}^k \leftarrow \mathcal{N}(\boldsymbol{\theta}^k, \boldsymbol{B}^k)$ ▷ Feed forward through model $\mathcal{N}$ || Model parallel
4: $\quad \boldsymbol{q}^k \leftarrow \mathcal{L}\left(\boldsymbol{Y}^k, \mathscr{H}(\boldsymbol{B}^k)\right)$ ▷ Compute loss against targets $\mathscr{H}(\boldsymbol{B}^k)$ || Collective
5: $\quad \boldsymbol{g}^k \leftarrow -\nabla_{\boldsymbol{\theta}} \boldsymbol{q}^k$ ▷ Compute gradient, using back prop || Model parallel
6: $\quad \boldsymbol{p}^k \leftarrow \boldsymbol{A}^k \boldsymbol{g}^k$ ▷ Compute update direction || Model/Collective
7: $\quad \alpha^k \leftarrow \arg\min J(\boldsymbol{\theta}^k + \alpha^k \boldsymbol{p}^k)$ ▷ Compute step length || Collective
8: $\quad \boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^k + \alpha^k \boldsymbol{p}^k$ ▷ Update $\boldsymbol{\theta}^k$ with update rule || Collective
9: **end for**

search direction based on the gradient, (7) determine a step size with which to scale the search direction, and (8) then use the scaled search direction to update the model parameters.

Gradient descent methods seek to choose $\boldsymbol{p}^k$ to maximize the local decrease in $J$ from one step to the next. Using the first order Taylor expansion of $J$,

$$J(\boldsymbol{\theta}^k + \alpha^k \boldsymbol{p}^k) \approx J(\boldsymbol{\theta}^k) + \alpha^k [\nabla J(\boldsymbol{\theta}^k)]^\top \boldsymbol{p}^k \tag{1}$$

we can observe that the decrease will be locally maximized when $\boldsymbol{p}^k = -\nabla J(\boldsymbol{\theta}^k)$ [46,174]. Gradient descent is realized in Algorithm 1 by choosing $\boldsymbol{A}^k = \boldsymbol{I}$. Stochastic gradient descent (SGD) is gradient descent coupled with a stochastic sampling process for forming $\boldsymbol{B}^k$.

Although SGD is the most popular optimizer for training DNNs, in its elementary form it has a number of notable shortcomings: it is relatively slow to converge, liable to get stuck in saddle points, and computationally inefficient. Moreover, these issues are inter-related and sensitive to the choice of hyper-parameters [127,198]. For example, although numerous extensions to SGD have been proposed to mitigate these issues [58,122,168,268,270], training a DNN with an SGD optimizer is a trial and error tuning process that intertwines learning rate and minibatch size [71, 198, 214]. These issues are only exacerbated when attempting distributed training of DNNs with SGD [210].

## 2.2 Second-Order Optimization Methods

Rather than using the first-order Taylor expansion as in (1), we can choose an update $\boldsymbol{p}^k$ to minimize the second-order expansion:

$$J(\boldsymbol{\theta}^k + \boldsymbol{p}^k) \approx J(\boldsymbol{\theta}^k) + [\nabla J(\boldsymbol{\theta}^k)]^\top \boldsymbol{p}^k + \frac{1}{2}(\boldsymbol{p}^k)^\top \boldsymbol{H}(\boldsymbol{\theta}^k)\boldsymbol{p}^k \quad \Longrightarrow \quad \boldsymbol{p}^k = -\boldsymbol{H}(\boldsymbol{\theta}^k)^{-1}\nabla J(\boldsymbol{\theta}^k).$$

Here, $\boldsymbol{H}(\boldsymbol{\theta}^k)$ is the Hessian (second derivative) of $J$. The resulting algorithm is the well-known Newton method, which can be realized in Algorithm 1 by choosing $\boldsymbol{A}^k = \boldsymbol{H}(\boldsymbol{\theta}^k)^{-1}$ and setting the batch to be the entire training set.

Newton's method has some appealing features. As is well known, since it uses second-order information, it converges quadratically (under appropriate conditions). Moreover, Newton's method offers multiple opportunities for computational efficiency and parallelization, most notably in that it can use a "maxibatch"—a minibatch consisting of the entire training set (thus exposing maximum data parallelism). Parallelization opportunities at each stage of Newton's method are noted in the right-most column of Algorithm 1 and the different modalities of parallelization for DNN training are shown in Figure 3. Recent experience with second-order methods has borne out the expectations of improved convergence and larger batch size in practice [7, 18, 66, 82, 152, 175, 262].

On the other hand, there are some significant difficulties with Newton's method as-is. Most problematically, and the issue that has likely impeded adoption of second-order methods to this point, forming the complete Hessian (which is not sparse [8]) is not feasible simply due to the storage requirements for anything but toy problems. Thus, the linear system solution $\boldsymbol{H}(\boldsymbol{\theta}^k)\boldsymbol{p}^k = \boldsymbol{g}^k$ (line 6 in Algorithm 1) must be approximated in some way. Forming the full Hessian has been a long-standing issue in the optimization community and a variety of approaches have been developed to deal with it, a selection of which are described below and summarized in Table 1.

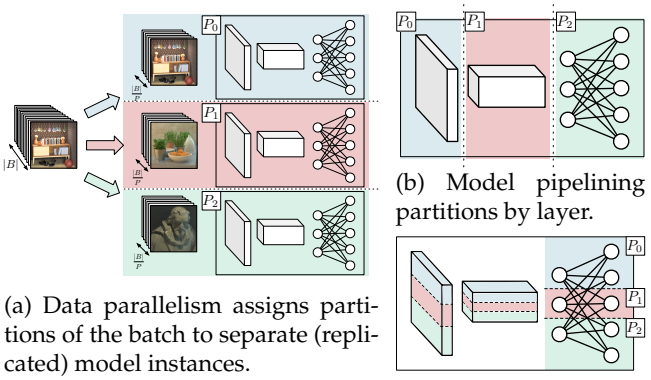
| Method | Direction computation | Approximation |
|---|---|---|
| Newton | Solve $\boldsymbol{H}(\boldsymbol{\theta}^k)\boldsymbol{p}^k = \boldsymbol{g}^k$ for $\boldsymbol{p}^k$ | |
| SGD | $\boldsymbol{p}^k = \boldsymbol{g}^k$ | $\boldsymbol{H} = \boldsymbol{I}$ |
| Newton-Krylov | Solve $\boldsymbol{H}(\boldsymbol{\theta}^k)\boldsymbol{p}^k = \boldsymbol{g}^k$ for $\boldsymbol{p}^k$ | $\boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{p} \approx (\nabla J(\boldsymbol{\theta} + \alpha\boldsymbol{p}) - \nabla J(\boldsymbol{\theta}))/\alpha$ |
| Secant (direct) | Solve $\boldsymbol{F}^k\boldsymbol{p}^k = \boldsymbol{g}^k$ for $\boldsymbol{p}^k$ | $\boldsymbol{F}$ is secant approximation to $\boldsymbol{H}$ |
| Secant (inverse) | $\boldsymbol{p}^k \leftarrow \boldsymbol{G}^k\boldsymbol{g}^k$ | $\boldsymbol{G}$ is secant approximation to $\boldsymbol{H}^{-1}$ |
| AdaHessian | $\boldsymbol{p}^k \leftarrow \text{diag}(\boldsymbol{H})^{-1}\boldsymbol{g}^k$ | $\text{diag}(\boldsymbol{H})$ is the diagonal of $\boldsymbol{H}$ |

Table 1: Selected families of approximate Newton methods.

*Inexact Newton methods* comprise the general family of methods where $\boldsymbol{H}(\boldsymbol{\theta}^k)\boldsymbol{p}^k = \boldsymbol{g}^k$ is solved approximately, for example with an iterative linear solver [45], or by using an approximation to $\boldsymbol{H}$, or both [25]. *Newton-Krylov methods* combine Newton's method with a Krylov solver for the linear solution. The important *matrix-free*—or, in our case, Hessian-free—variants note that the matrix-vector product at the core of all Krylov solvers can be approximated as the difference of two function (resp. gradient) evaluations [25]. (This technique was rediscovered in the ML community where it is known as the "Pearlmutter trick" [179]).

*Quasi-Newton methods* (or *secant methods*) comprise a family of methods where the Hessian (in the case of direct formulations) or the inverse Hessian (in the case of inverse formulations) is approximated through a series of low-rank updates made to an initial estimate as the Newton iteration proceeds. Different families of quasi-Newton methods are categorized by regularization constraints imposed on the low-rank update. The Broyden family update is minimal in Frobenius norm (but loses symmetry of the Hessian). The SR-1 family is symmetric (but may not be minimal). Powell's symmetric Broyden is minimal and symmetric (but not necessarily positive definite). The celebrated Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno are algorithms from the same family, with updates that are symmetric and positive definite. Each of these four families in turn has four formulations (for a total of 16 methods): direct and dual-inverse, which define updates for the Hessian; and dual and inverse, which define updates for the inverse of the Hessian. The inverse and dual-inverse forms are derived from the direct and dual forms by application of the Sherman-Woodbury-Morrison formula [46]. *Limited-memory* secant methods avoid forming the Hessian explicitly but, instead, store the constituents used for the low-rank updates and apply those when the Hessian is needed—i.e., when computing a matrix-vector product [169].

*Conjugate-Direction methods* are extensions of well-known linear solvers to the nonlinear case. Daniel's method can be viewed as an extension of the conjugate-residual algorithm [199], while the Fletcher-Reeves algorithm can be viewed as an extension of the conjugate-gradient algorithm (the difference between the two pairs is in the choice of norm under which minimization takes place) [174]. L-CG_DESCENT is a limited-memory version of the benchmark nonlinear conjugate gradient method CG_DESCENT [75–77].




(a) Data parallelism assigns partitions of the batch to separate (replicated) model instances.

(b) Model pipelining partitions by layer.

(c) Model parallelism partitions the model.

Figure 3: Primary modes of parallelism for training large DNNs.

The strong quadratic convergence behavior exhibited by Newton methods happens only when iterates are close to a minimum. Hence, a variety of techniques have been developed to provide better *global convergence* behavior, including damping, line-search, trust-region, and homotopy continuation methods [46, 118, 174]. Second-order methods are gaining some recent attention; methods developed specifically for DL include AdaHessian [266], K-FAC [153–155], and Curveball [88]. Most on-going efforts seem to be focused on issues related to convergence with less emphasis on potential for parallelization. Theoretical work on analytic formulations of the Hessian indicate that it has an outer-product structure pointing to alternate representations and efficient storage mechanisms [8].

| Property | ResNet [82] | BERT$_{Large}$ [52] | GPT-2 [186] | GPT-3 [26] |
|---|---|---|---|---|
| $\|\boldsymbol{\theta}\|$ | ∼23-60 M | 340 M | 1.5 B | 175 B |
| Layers | 50-152 | 24 | 48 | 96 |
| TFLOPS | .004-.011 | 6.5E5 | 8.64E8-8.64E9 | 3.14E11 |
| Training set size | 1.28 M images | 3.3 B words | 10.2 B words | 304.7 B words |
| Mini-batch size | 192 images / GPU | 256 K words | 524 K words | 32 K-12 B words |
| Hardware | 8 V100 GPUs | 64 TPU chips | 1024 TPU chips | - |
| Headroom | 835 | 51.5k | 19.5k | 25–9M |

Table 2: Characteristics of selected large neural networks. "Headroom" indicates the ratio between the total training set size and the largest minibatch size, an indication of potentially available data parallelism.

## 2.3 Distributed Computation

Distributed training of Deep Neural Networks (DNN) has become increasingly common in order to keep up with the training requirements of state-of-the-art models [2, 225, 271]. As illustrated in Figures 3, there are three main approaches to partitioning for parallelizing DNN training: partitioning the training set (data parallelism), partitioning across the model (model parallelism or domain parallelism), and partitioning by layer (pipelining). Any of these approaches can be combined (hybrid parallelism). Data parallelism is straightforward to implement and is the most common approach used by existing deep learning frameworks [80]. However, used by itself, it requires that the entire model, as well as the batch partition, fit into the memory of a single worker; additional parallelization is required for very large-scale models.

There are two main challenges in using a distributed SGD. First is the overhead of parallelization itself; communication and synchronization costs are a bottleneck. Some proposed asynchronous variants of distributed SGD seek to decrease the rate of communication, but this later proved not to be scalable [34, 43, 102, 272]. Others attempt to design a more communication-efficient training method [11, 74, 101, 218, 269]. Second, and problematically, attaining greater parallelization requires larger minibatches, but the convergence of SGD and minibatch size are inextricably related. That is, the convergence guarantees provided by different sizes of minibatches have a large variance in the presence of a constant effective batch size [14], and large effective minibatch sizes suffer from a generalization gap. Larger batch sizes per worker allow a larger learning rate and faster training as a result, but the memory constraints of a single worker limit the growth of minibatch size per worker. The instability of SGD training with large effective minibatch sizes imposes ad hoc modifications to the optimizer, namely using a warmup technique and/or a linear scaling rule [139,162]. As of now, the largest reported effective minibatch size (without sparsification) using a distributed SGD is $64K$ [99, 162, 267].

| **Framework** | **Data Par** | **Mod Par** | **Overlap** | **Gran** |
|---|---|---|---|---|
| Mesh-TensorFlow [211] | ✓ | ✓ | | |
| GPipe [91] | compat | ✓ | ✓ | |
| PyTorch DDP [136] | ✓ | | ✓ | |
| Hororvod [207] | ✓ | compat | ✓ | |
| FlexFlow [100] | ✓ | ✓ | ✓ | ✓ |
| Chainer [235] | ✓ | | | |
| BigDL [41] | ✓ | | | |
| MXNET-MPI [150] | ✓ | | | |
| DeepSpeed [189] | ✓ | compat | ✓ | ✓ |

Table 3: Comparison of distributed deep-learning frameworks. **Data Par** indicates support for **data parallelism** on multiple nodes. **Mod Par** indicates support for intra-iteration and/or inter-iteration **model parallelism/pipelining**. **Overlap** indicates ability to compute and communicate concurrently. **Gran** indicates parallelism support down to the **granularity** of an individual operation.

The payoff in using larger minibatch sizes enabled by higher-order methods is potentially enormous. Table 2 shows model characteristics for a few of today's largest models. For each of the models, we show a "headroom" number, which is the ratio between the training set size and the largest minibatch published for that model. This ratio represents how much work could potentially be done in parallel if the entire training set were to be done as one batch. The consequences of not being able to scale due to minibatch size limitations is reflected in the relatively small size of hardware used for these problems (relative to the problem sizes themselves).

Finally, Table 3 shows a comparison of DL frameworks that support distributed memory parallelization [81]. As with the current state of second-order methods, these frameworks tend to aim for modest levels of acceleration rather than HPC scales—and in a very real sense, that is precisely because methods are not available that could scale. Most are also feature incomplete vis-a-vis providing an actual CI.

# 3 Foundational Infrastructure

We describe key extant work that will be used as a foundation for a web-service oriented, end-to-end environment for scalable and highly-optimized distributed array computing. We start by discussing HPX and Phylanx which will underlie the *CAIRO* distributed array computing framework. Next we describe the technology behind the embedded performance evaluation system, APEX, and its associated interactive visualization system, Traveler. Finally, we discuss the JetLag science gateway and the Agave Platform through which we will provide web-based accessibility, automation, and benchmarking to the institute and its user community.

## 3.1 The HPX Runtime System

HPX is the C++ Standard Library for Parallelism and Concurrency [84–86, 108, 111, 112], partially funded by NSF awards (1111888, 1240655, 1339723, and others). It represents an innovative mixture of long-known ideas and concepts such as static and dynamic dataflow, fine-grained *Futures*-based synchronization, and continuation-style programming. It is the combination of these ideas that form the overarching design principles which make HPX unique [111]. HPX addresses problems of scalability, resiliency, energy efficiency, runtime adaptivity, and dynamic resource management that continue to grow in importance as the industry faces increasing demands in supporting highly distributed systems with heterogeneous architectures. To achieve these goals, HPX introduces an asynchronous C++ programming model that departs from today's prevalent parallel programming models with the aim of increasing parallel efficiency. This programming model mitigates common limitations, such as implicit and explicit (global and local) barriers, coarse-grained parallelism, and lack of easily achievable overlap between computation and communication.

HPX exposes a coherent programming model unifying all the different types of parallelism available in today's computer systems. By modeling the API after the interfaces defined by the C++ standards [226–228], programmers are able to write fully asynchronous code using hundreds of millions of HPX-threads (tasks) in a familiar environment. This ease of programming extends to both parallel and distributed applications. HPX is the first open source runtime system to implement the ParalleX execution model [110,232] on a wide range of conventional systems. Further, HPX provides services and APIs allow it to coordinate and manage code execution on GPUs and accelerators in distributed systems. HPX has a worldwide, open, active, and thriving developer and user community.

In *CAIRO*, we plan to use HPX because of its dynamic scheduling and global data addressing capabilities and because of its ISO C++ standards conformance. The shared memory abstractions introduced by HPX have already been adopted in the most recent ISO C++ standard, and HPX's distributed memory abstractions are also standards conforming extensions. Using the *Futurization* concept in HPX, developers can express complex dataflow execution graphs that generate billions of tasks that are scheduled to execute only when their dependencies are satisfied [44]. HPX integrates with the APEX performance measurement and adaptive tuning framework (see Section 3.2).

We expect that many of the core algorithms for *CAIRO* can be implemented elegantly using HPX's higher-level API, which also opens up a natural upgrade path to acceleration. All algorithmic primitives that already exist and those that will be developed by *CAIRO* directly target HPX, which ensures the best possible performance and excellent resource utilization. Our PhySL intermediate representation in Phylanx (see Section 3.4) is directly compiled into a static dataflow execution tree, the nodes in that tree are comprised of primitive operations. The evaluation of this tree produces a dynamic dependency tree of scheduled operations that is executed by HPX with minimal synchronization overhead.

## 3.2 APEX - Autonomic Performance Environment for Exascale

APEX [94, 95] (Autonomic Performance Environment for Exascale) is a performance measurement library for distributed, asynchronous multitasking systems such as HPX. It provides lightweight measurements without perturbing high concurrency through both synchronous and asynchronous interfaces. APEX is integrated into HPX, timing all scheduled tasks and capturing HPX counters. The *policy engine* within APEX provides an API to construct policies that can modify the behavior of the application, execute a desired function in the runtime, or adjust runtime and application parameters. Typically, APEX policies are designed to auto-tune systems in cases where so-called "magic number" parameters (set by expert

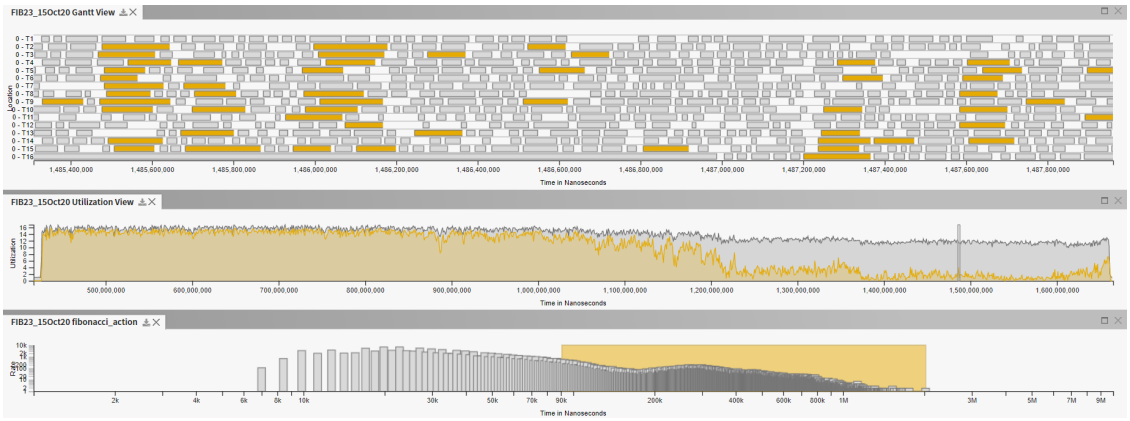



Figure 4: Traveler with three views loaded. The histogram (bottom) shows the distribution of task durations with longer tasks selected (yellow box). This selection is also shown in the individual task timelines (top) and with respect to total utilization (middle). There are fewer short-duration tasks later in the execution.

knowledge and/or best guesses) are used to control algorithms. Examples include default timeouts, queue depths, number of workers, or when to switch algorithms. Examples of APEX policies that have been integrated into HPX include thread throttling during regions of resource contention, concurrency control under soft power caps, network message coalescing, and task inlining. Policies are typically implemented with a guided exploration of a predefined search space using Active Harmony [223], a framework for enabling application adaptation.

To provide policy input, APEX has native support for performance profiling of all tasks scheduled by the runtime. APEX timers and counters contribute to the performance state, used as input to the policy logic. At any point during the execution, the profile maintains the number of times each task was executed and the total time spent executing that type of task. Profile data is also optionally stored to disk in two different formats for postmortem performance analysis. To perform detailed performance analysis involving task dependencies and synchronization overhead, full event traces are captured. APEX is integrated with the OTF2 library [64]—an open, robust format for large scale parallel application event trace data. To capture full task dependencies in HPX, all tasks are uniquely identified by a GUID (globally unique identifier) and the GUID of their parent task. These GUIDs are captured as part of the OTF2 trace output. Figure 4 depicts an execution where OTF2 data is used to capture an HPX benchmark. APEX also captures task dependency graphs that represent the critical task dependencies in an application.

### 3.3 Traveler: A Visualization Framework for Asynchronous Many-Task Models

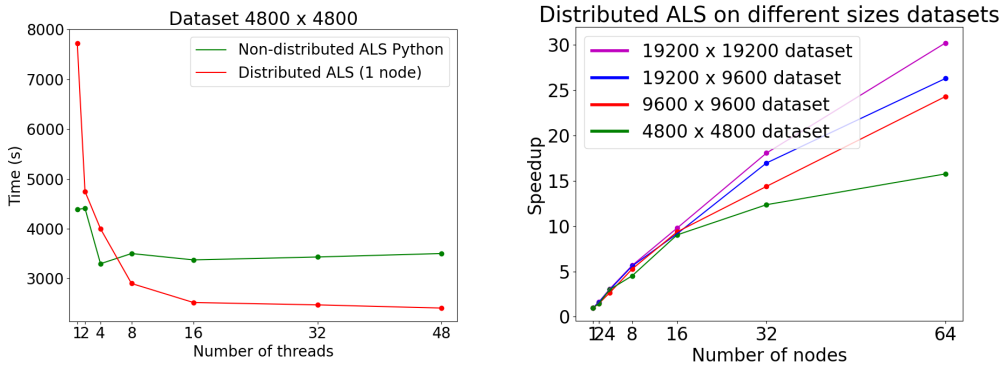
A key component of the *CAIRO* project will be the effective mechanisms to view and analyze the performance of asynchronous paradigms within Phylanx accelerated applications. Traveler is a web-based debugging and performance visualization platform that provides an interactive interface of OTF2 data generated by APEX. Typically, visualization of distributed resources has assumed a homogeneous environment. For example, Gantt charts, which are commonly used to visualize traces, represent the timeline of each thread equally (see Figure 4, top). Many performance visualization platforms do not include explicit support for tasking runtimes [216, 276]. Representations of resources themselves tend to focus on single, specific architectures [197, 212], networks [15, 126, 132], or otherwise represent compute units equally [164, 242].

Traveler houses a variety of *views* of execution and performance. The framework allows users to add and manipulate views as well as link data across views to provide higher dimensional insights. The implemented views include aggregated task graph and expression tree diagrams [253], Gantt charts, line charts and histograms of counter data, and source code. Figure 4 shows an example configuration. Additional views will be added to incorporate data flow and access patterns. In doing so, *CAIRO* will empower users with tooling to better tune and understand their application codes, data, and hardware.

### 3.4 The Phylanx Distributed Array Toolkit

The Phylanx library, funded in part by concluded NSF grant 1737785, supports many NumPy objects and array operations [109, 234]. These operations are highly parallelized and can also run asynchronously to

improve machine throughput. These have been made possible by leveraging HPX's parallel threads, *Futurization* facilities, and dataflow abstractions. Phylanx also includes distributed implementations of a number of these operations and makes them available in Python programs through decorators. Application developers can transparently benefit from all these facilities by adding the Phylanx decorator to any ordinary Python function. Phylanx transforms the body of the function into a PhySL (Phylanx Specialization Layer) expression tree which also acts as an intermediate representation suitable for high-level analyses. Once the PhySL expression tree is compiled down to an HPX expression tree, either explicitly by the user or implicitly upon the first function call, HPX automatically schedules and runs the tasks.




(a) ALS Scaling, one node

(b) ALS Scaling distributed

Figure 5: Distributed Alternating Least Squares (ALS) performance on Queen Bee (Intel Cascade Lake Xeon 64-bit processor, $2 \times 24$-core 2.4GHz processors per node) [21]. Phylanx ALS implementation exploits parallelism and is 1.5x faster than the base NumPy implementation on one node (Figure 5a). Beyond that, Phylanx exhibits improving speedups as the number of nodes and dataset sizes increase (Figure 5b).

In addition to HPX, Phylanx also benefits from other open-source, industry-quality C++ libraries, namely, the Blaze [16] math library for linear algebra operations, and pybind11 [185] for interoperability between Python programs and the underlying C++ implementation of Phylanx. We have published performance results from a Phylanx version of the binary logistic regression analysis (LRA), alternating least squares (ALS), k-means, decision trees, random forests, latent Dirichlet allocation (topic modeling), and training of deep neural networks (DNN) [21, 234]. In Figure 5, we show an exemplar comparison of the achieved performance of the Phylanx ALS implementation and a base NumPy CPU implementation, finding that Phylanx is faster on a single node and exhibits good scalability and improving speedups as the number of nodes increases [21]. In *CAIRO* we expect that Phylanx will serve as the implementation platform for developing the algorithms, ensuring their scalability and efficiency over a wide range of hardware architectures.

## 3.5 The Agave Platform

The Agave Platform (Agave) is an open, Science-as-a-Service (ScaaS) platform for reproducible science [56]. Agave uses standards-based technologies and community-promoted best practices to enable users to run code, manage data, collaborate meaningfully, and integrate anywhere. Since its first launch in 2011 as a proof of concept for what is now the CyVerse project, Agave has grown to power dozens of production science gateways while extending and enhancing the functionality of hundreds more applications. During that time, a rich technology ecosystem has developed around the platform to provide client SDK in multiple languages, a command line interface (CLI), reference web applications, and integrations with many of today's most popular web frameworks and cloud services. Agave features a highly flexible, cloud-native architecture as shown in Figure 6. This has allowed it to power large, multi-institution projects including CyVerse(iPlant) [70], Araport [78], DesignSafe [190], and SD2E [205], while supporting cross-disciplinary use cases for projects such as the Science Gateway Community Institute (SGCI) [251]. In total, Agave has been a core technology on projects representing over $150M of funding from NSF and produced two spin-off projects, Abaco [220]

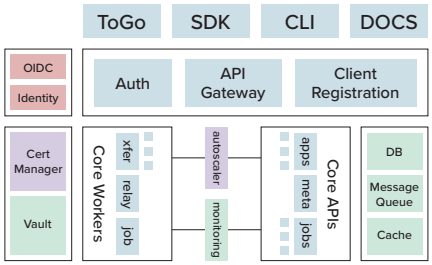



Figure 6: The Agave Platform architecture.

and Tapis [38], each with their own independent funding from NSF. The core functionality behind JetLag's remote execution and data movement capabilities are powered by Agave's core science APIs. Moving forward, *CAIRO* can further leverage Agave's suite of services to track and publish results, provide automated CI services to build, test, benchmark, and publish user codes, run application portably across hosts, incorporate Phylanx codes into existing gateways and workflows, and expose pipelines and applications as reproducible services for community use. This will provide *CAIRO* a proven, extensible research CI platform from which to engage the AI community.

### 3.6 JetLag – An Interactive Frontend to Remote Execution

Manual use of Phylanx and its technology ecosystem currently requires a measure of setup that could potentially be a barrier for some users. The JetLag [20, 21] is an interactive tool focused on lowering the barrier to entry for both new and existing users by facilitating the containerization, deployment, and profiling of Phylanx programs using a Jupyter notebook [19]. Through JetLag, users have an environment preconfigured with Phylanx and its dependencies, active integration with third-party tools and services, and templates for using Phylanx with existing Python analysis scripts. JetLag also provides a built-in orchestration tool to portably package and run user's code on remote systems using a Phylanx Docker (cloud) or Singulary (HPC) container, track the job's lifecycle, collect the performance data, and stage it back to the client for analysis in Traveler. To achieve this functionality, JetLag leverages the Agave Platform [56], its TACC-focused spin-off, Tapis [38], the Jupyter notebook environment, and Singularity.

## 4 Description of the Research Plan of the Institute

This section describes the *CAIRO* Coalition's Work Breakdown **(WB)** planning for fulfilling research objectives outlined in 1 (see also Table 10). *CAIRO* is committed to excellence and will collaborate with third-party experts to ensure the quality and reliability of the *CAIRO* platform as well as developed algorithms and applications described in this section and the next. We will consult Dr. John Leidel from TCL to rigorously evaluate the work; collaborate with Dr. Corey Trahan who will provide test cases for different ML algorithms applicable to coastal hazards; and collaborate with machine learning group at Radiance Technologies for optimization of NLP tasks and various Neural Network Schema's to enhance the combined efforts at meeting the needs of DARPA, AFRL, ARMY, and other customers. Please see attached letters of collaboration.

### 4.1 WB1: Deep Learning via Second-order Optimization

*CAIRO* will develop a comprehensive catalog of advanced second-order methods and realize them as high-quality software artifacts. Our work will put these methods in the hands of ML practitioners, significantly improving their productivity as well as providing a rich toolbox for ML researchers worldwide to investigate and further advance second-order approaches.

**Methods.** Many techniques for unconstrained optimization appear to be potentially applicable to DL, but few have been investigated in much depth. We will develop robust implementations of basic Newton and Gauss-Newton iterations, parameterized in such a way that families of approximate Newton and Quasi-Newton methods can be efficiently composed. Particular methods that we will initially develop include Newton-Krylov, Quasi-Newton families include Broyden, SR-1, Powell's symmetrized Broyden, and DFP/BFGS, conjugate-direction methods (Fletcher-Reeves, Daniel), nonlinear conjugate gradient methods (L-CG_DESCENT, Hager-Zhang), and approximate Newton methods (AdaHessian). Conjugate gradient and conjugate residual solvers will be developed for linear system solution, which will also be parameterized by "matrix" and "vector" types so that limited-memory secant and Hessian-free techniques can be applied. We will also support convex combinations of methods. As more experience is gained with the initial suite of methods, we will investigate new approaches to Hessian initialization in the secant methods, Hessian update formulae, and preconditioners. More recently-developed second-order approaches (such as K-FAC [153, 154]) will also be considered.

**Towards Global Convergence.** To improve the global convergence behavior of second-order methods, we will define and implement filter, trust region, line-search, and step length selection techniques, composable with the second-order optimization algorithms. We will also investigate techniques for providing better initial solution estimates, including "warm up" with SGD and continuation methods.

**Theory.** We will seek to formally characterize the convergence behavior of the different methods (Newton-Krylov as well as quasi-Newton families), beginning with particular model problems. Of particular interest is understanding the characteristics (e.g., the spectra) of the different approaches to approximating the Hessian. We will also seek to characterize the impact of batch size on optimizer behavior.

**Hessian.** We will focus particular attention on developing effective strategies for choosing, approximating, and constructing the Hessian matrix. We also propose to scrutinize dynamic sampling strategies for constructing stochastic gradients that not only reduce the variance of those gradients but also provide a natural way to extract second-order information of the objective function. Since we solve stochastic composite quadratic subproblems inexactly (refer to subsection 4.1), careful coordination is needed on the dynamic sampling size, the inexact solution of the composite quadratic subproblem, and the variance reduction strategies to ensure overall efficiency.

**Evaluation.** We will conduct extensive experimental evaluations of our catalog of approaches, for well studied problems, for benchmarks such as MLPerf [158], for our science drivers, and for other problems of interest that emerge during the course of the institute. To the extent practicable, we will integrate measurements of performance, scalability, and convergence behavior into our continuous-integration process, with the results available as an on-line dashboard. The results of these evaluations will be used to optimize the performance of our methods.

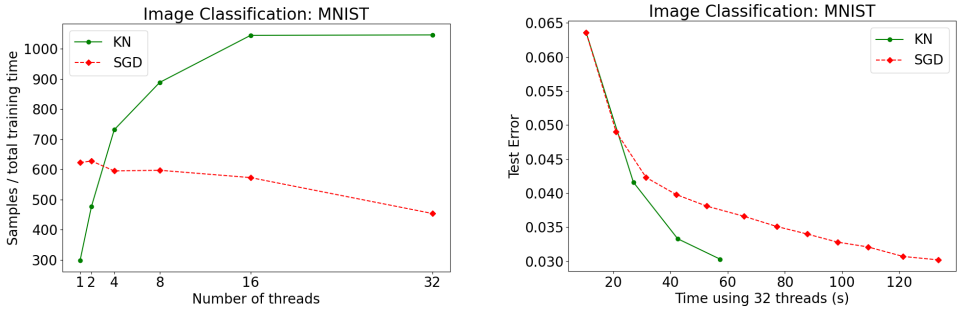
**Software.** Our implementations will be continually updated and made available in a publicly available repository. This will include compatible libraries for mature ML systems such as PyTorch, Tensorflow, and MXNet. We will also make selected trained networks (including training history) available to enable additional studies.

**Inexact proximal stochastic second-order methods for nonconvex composite optimization.** SI Zhang's group proposed a framework of Inexact Proximal Stochastic Second-order (IPSS) methods for solving nonconvex composite optimization, where the objective function consists of the loss function $J(\boldsymbol{\theta})$ and a possibly nonsmooth convex regularization function [246]. This IPSS framework incorporates variance reduction techniques and allows solving stochastic composite quadratic subproblems inexactly to an adaptive accuracy derived from theoretical analysis. The IPSS guarantees global convergence with desired computational complexity, even when the subproblems are solved inexactly. In particular, given $\varepsilon > 0$, it is shown in [246] that the number of stochastic gradient evaluations required by IPSS to achieve an $\varepsilon$-accuracy solution, i.e., $\mathbb{E}[\|\mathbf{g}(\boldsymbol{\theta}^{\mathbf{k}})\|^{\mathbf{2}} \leq \varepsilon]$, can be bounded by $\mathcal{O}\left(n + n^{2/3}/\varepsilon\right)$, which is the best-known complexity bound when $J(\boldsymbol{\theta})$ is nonconvex [191]. Here, $\mathbb{E}[\cdot]$ denotes the expectation, $\mathbf{g}(\boldsymbol{\theta}^{\mathbf{k}})$ is the proximal gradient at iteration $\boldsymbol{\theta}^k$ and $n$ is the number of elements of data $\boldsymbol{X}$. Moreover, when $J(\boldsymbol{\theta})$ is only $\nu$-weakly smooth with $\nu \in (0, 1)$, for obtaining an $\varepsilon$-accuracy solution, IPSS can achieve the stochastic gradient complexity and iteration complexity as $\mathcal{O}\left(n + n^{\frac{1+\nu}{2+\nu}}/\varepsilon^{1/\nu}\right)$ and $\mathcal{O}\left(1/\varepsilon^{\frac{1}{\nu}}\right)$, respectively, which are again the best-known bounds [246]. The Hessian matrix in the composite quadratic subproblem of IPSS can be used to capture second-order information of $J(\boldsymbol{\theta})$. However, efficiently choosing this Hessian matrix and solving the subproblems remains under investigation. We will also identify the computational cost of each component of IPSS and rigorously establish the overall computational complexity for solving this target possibly nonconvex optimization.

**Preliminary Results: Hessian-free Krylov-Newton Method.** We show the scalability and convergence of a Krylov-Newton method (described in § 2.2), implemented as a PyTorch optimizer and compare its performance with SGD on an image classification task, using multi-layer perceptron (MLP) network with one hidden layer. Here we use the MNIST dataset, and the hidden layer size is set to 150. For SGD, the learning rate is 0.333, the momentum is 0.9, and the mini-batch size is 64. Figure 7 demonstrates the scalability of KN method. Specifically, we compare the throughput of KN and SGD on different numbers of threads. It can be seen that when using 4 or more threads, KN obtains higher throughput than SGD. When using 32 threads, KN achieves 2.3x better throughput than SGD.

## 4.2 WB2: Runtime System and Program Optimization

**Motivation.** Here we consider optimization of *CAIRO*'s runtime system, computation graphs and data management schemes. As an example, for DNN compilers, optimizing the computation graph to minimize communication for distributed execution is a challenging problem. Proposed optimization techniques are hardware-independent and can be applied to various backend targets [133]. The frontend optimizations involve data/code passes traversing the nodes of the computation graph. We apply various algorithmic




(a) SGD and KN Method Scaling (b) SGD and KN Method Convergence

Figure 7: Scalability of a Krylov-Newton (KN) method implemented as a PyTorch optimizer, on an Intel Cascade Lake Xeon 64-bit processor, $2 \times 24$-core 2.4GHz processors per node. We compare the throughput of KN with SGD on different numbers of threads (Figure 7a). When using 32 threads, KN obtain $2.3\times$ higher throughput than SGD (Figure 7b).

techniques to optimize the passes. Four types of optimization problems are considered with respect to *CAIRO* and other similar AI-focused HPC systems and layout frameworks for building practical solutions based on sound and strong theoretical foundations.

**Outcome and Evaluation.** By creating an optimization toolkit built upon strong theoretical foundations we hope to make *CAIRO* adaptable to 1) a wide range of AI and scientific computing solvers, 2) various hardware architectures and 3) changing user needs. The groups led by co-PI Banerjee and co-PI Li will emphasize theory and general purpose algorithmic solutions. The Ph.D. students and a postdoc will work with co-PI Banerjee and co-PI Li to adapt these solutions to *CAIRO*. Section 4.5 discusses adopting and implementing some of these techniques as part of optimization policies of *CAIRO*'s runtime system.

**Distributed Data Partitioning and Tensor Tiling.** Linear algebra operations involving large matrices and tensors (which can be either sparse or dense) are important computational drivers in *CAIRO*. To efficiently distribute these types of operations, we need to solve several combinatorial optimization problems. Various techniques have been proposed [42, 47, 90] for partitioning distributed data to reduce communication. Underlying theoretical problems can be modeled as partitioning graphs (or hypergraphs) [47, 87] which are known to be inapproximable within a constant factor [3]. Co-PI Banerjee, with the Phylanx team, is developing a data partitioning framework for dense matrices. As part of this work, we show that the problem is approximation hard even for simple cases. We also provide a greedy algorithm based on hypergraph coloring that simultaneously looks at the algorithm selection and partitioning. We plan to extend our work along two directions: We will 1) apply smoothed analysis [217] to give better theoretical prediction of runtime and approximation performance of our algorithms. Smoothed analysis is particularly suited in this regard since unknown program parameters before execution can be modeled as distributions over a collection of integer programs; and 2) extend data tiling schemes to sparse matrices / tensors as well as their low rank approximation to reduce communication (see next section). We also plan to consider graphical data which have gained popularity due the advent of graph neural networks.

**Sketching Data Layout and Memory Allocation.** We propose using the random sampling method to sketch and approximate inputs to significantly reduce execution time. We expect that in Phylanx we will be able to process the data with a much smaller approximation size and provable performance guarantees. The approach of sketching matrices has been popularized by Frieze, Kannan, and Vempala [69]. In *CAIRO*, we will study various combinations of matrix decomposition and preprocess, such combinations offline, to find the best sketch. We will also incorporate random sampling into machine learning algorithms (such as perceptron, $k$-mean). Methods used to evaluate these sampling techniques include smoothed analysis [27, 28, 163] and instant-optimality [31, 79]. Methods will be evaluated offline and the promising candidates will be implemented in Phylanx. As a side effect we believe that training with simplified data or NN models may reduce generalization errors and improve scalability [263].

**Minimizing Completion Time for Tasks with Dependencies.** We propose to study how to execute tasks optimally in parallel under hard resource constraints in Phylanx. Our proposed solution is based on the previous work by co-PI Li: minimizing the number of task execution paths as well as the path costs to complete a given set of tasks with dependencies in a graph [131, 135]. We have proposed near-optimal

offline solutions. However, the algorithmic challenge in the proposed work is to consider the online version of this problem. One approach that we are taking is to learn task dependencies in an online manner and incorporate the offline algorithm's idea to generate a reasonable performance guard. We also consider multi-criteria optimization techniques [4,233] to solve data partitioning along with several other objectives such as makespan minimization, algorithm selection, and resource type allocation (e.g., CPU vs GPU vs TPU). In [9], co-PI Banerjee developed an evolutionary algorithm based on a local interaction model which can be extended in the multi-objective case. For this part, co-PI Banerjee will closely collaborate with researchers from the Intelligent Systems Center at Missouri S&T.

**Boosting Algorithms for Load Distribution.** We propose to develop online learning approaches to generate load distribution for Phylanx in order to improve its weighted throughput. To optimize weighted throughput for tasks with various CPU and memory requirements, we learn tasks' resource requirements in a runtime manner. The challenge is that the tasks arrive and are executed over time, and the learning procedure should adapt the workload in a dynamic way. This problem is a generalized version of the 2D online stochastic bin packing problem [145, 209]. In co-PI Li's previous work [245], we provided a $1.25$-competitive randomized algorithm for a more strict version of this problem against an oblivious adversary. We also applied boosting algorithms to maximize weighted throughput [274]. In this project, we extend our previous research to study learning and dispatching CPU-extensive and memory-extensive tasks for Phylanx. Dr. Songqing Chen at GMU shares training data with co-PI Li for learning algorithms.

## 4.3 WB3: Phylanx and HPX

Phylanx will be the base implementation framework for *CAIRO*'s scalable, distributed, higher-order optimization algorithms. We will improve the distributed capabilities of Phylanx by adding more primitives, prioritized by our science drivers. This will include adding primitives to support new algorithms as well as alternative implementations of current algorithms. Supporting multiple implementations, optimized for particular characteristics of the computation and the consumed data, will provide better and more flexible opportunities for scheduling and optimization at runtime. In the same manner, we will add first-class support for hardware accelerators.

We will also provide facilities to improve and automatically apply both architecture-oblivious and architecture-specific optimizations on the PhySL tree (see Section 4.7). Every expression Phylanx evaluates has its own required capabilities, data transfer requirements, and algorithmic complexity. Based on the research results from Section 4.2, we will develop optimization methods that enable the system to use that information to **(a)** select better tiling (blocking) strategies that minimize the amount of data transfer between nodes, **(b)** apply additional structural optimizations to the PhySL representation targeting the reduction of algorithmic operations, and **(c)** design and implement optimizations specific to the target architecture (accelerator devices, etc.) that the code will be running on.

**GPU Performance Support.** The goals of the accelerator code work are to provide kernels that support second-order training and provide the performance modeling needed so that the autotuner can find a configuration (kernels set, data partitioning, etc.) that realizes the chosen objective (such as minimize time to solution or minimize the cost of computation with available resources). Project members have experience tuning GPU code, as part of a model-driven autotuner for stencil codes [22,89,204] under NSF-1265449 and the development of accelerated graph analytics code [116,117] under NRL N00173-16-2-C901.

Here, kernel refers to a parameterized piece of code that can perform some operation, along with a performance model that provides execution time estimates and other data. Kernels' performance models are used by an autotuner and scheduler to choose a desirable execution configuration and to make scheduling decisions. The initial work will extend Phylanx so that it can target accelerators, initially GPUs, and will provide basic support for second-order training workloads. The kernels and kernel models will be extended to support kernel composition, an increasingly important capability for DNN compilers [134]. That is, the models will provide a time estimate for a group of dependent kernels in which dependent kernels specialize their data layouts, computation, and other factors based on the others' needs. This is especially important for accelerators which have less forgiving memory hierarchies. The use of composable kernels for autotuning will result in better configurations.

Distributing a workload across multiple nodes will complicate scheduling due to the irregular timing of data exchange. For that, autotuning will select kernels' grain (input) size. A larger grain size favors efficiency (lower execution overhead) at the expense of lower utilization, and is the typical way workloads

are executed. This work will extend prior work on dynamic grain sizing in HPX [243]. This work will include modeling kernel execution, taking into account resources used by other data-driven kernels. HPX performance counters and APEX timers will provide data to validate the models at runtime.

## 4.4 WB4: *CAIRO* and Computing Hardware Evolution

To support ML workloads, CPUs and GPUs have been evolving, and new technologies are being developed, such as custom tensor-processing chips and devices performing analog computation. Recent generations of GPUs have added features to accommodate DNN workloads [170, 171], and this trend is expected to continue. Features include mixed-precision multiply/add units and tensor cores that consist of arithmetic units interconnected in such a way as to reduce energy-consuming register accesses. Due to the similarity between the computation-dominating operations in first- and second-order methods these features will be of equal value to both. In fact, where second-order methods use existing library functions (such as those provided by cuBLAS and cuDNN), *CAIRO* will benefit as those libraries are updated. Some low-level, second-order code may need to be written as part of the project. The project will incorporate new features and re-tune such code as hardware evolves and becomes available (see Budget and Budget Justification).

The research community for some years has been looking at reducing DNN inference cost by imposing weight sparsity [83, 178]. These ideas reached a commercial product in modest form in the NVidia A100 [171]. Sparse weights, though primarily intended to realize an inference benefit, must be supported by training code to be useful. The project will monitor sparsity features and support them as needed.

Google's TPUs, used for production workloads, feature one or two large systolic arrays [105, 106]. They are available through cloud services where they can be programmed at a fairly low level. The project will consider targeting these devices. There is a great deal of research on other specialized devices for DNN calculation. Proposed designs differ in how multiply/accumulate units are interconnected and on where data is buffered [177, 264]. The Cerebras CS-1, an ambitious commercial realization of these ideas, consists of a wafer-filling 2D mesh of simple cores [195]. Network models that would otherwise span multiple chips can execute on the CS-1, and so avoid communication and memory access latency, potentially outperforming less costly devices. The project will monitor the CS-1. Some have looked to more exotic technologies, such as using analog electronics to perform multiply accumulate operations [57,92]. Though analog technologies show promise, they are too far from practical implementation to impact *CAIRO* over the life of the project.

## 4.5 WB5: Performance Monitoring, Measurement, and Runtime Control

In *CAIRO*, we will need distributed monitoring at an application-level scope of telemetry that goes beyond traditional system administration monitoring [1, 156, 176, 215]. We will leverage our experience in past performance monitoring efforts [93, 121, 255, 258, 273] and integrate performance monitoring from HPX and APEX into *CAIRO*'s service infrastructure. APEX already monitors hardware and operating system data related to networking, filesystems, processors, and memory, and will integrate with tools such as TensorBoard [257] to ensure algorithmic progress during training. For metrics that aren't well represented by TensorBoard, we will integrate with Grafana [72] or other relevant Agave services for effective system monitoring using a broader application of the work done by [13] on IO monitoring.

**Performance Debugging and Runtime Control.** In the current APEX implementation, performance data for policy execution and postmortem analysis is retained locally in memory until the end of execution or periodically flushed to disk if necessary. As a consequence, performance data is only available for local policy decisions and analysis. In the *CAIRO* project, APEX will be extended to optionally aggregate performance measurements (using asynchronous HPX futures and communication collectives) for global policy execution. *CAIRO* policies that require a global state include general load balancing, evaluation of tiling and data partitions, and overall algorithmic progress. Detailed performance debugging at runtime is also not currently available. Event trace data is written to disk on a per-process basis and only unified at the end of execution. Because of the sheer volume of data that is collected, highly efficient binary file formats such as OTF2 [64] are typically used exclusively for postmortem processing. We will explore the use state-of-the-art services and interfaces such as Jaeger Tracing [229], OpenTracing [230] or OpenZipkin [231], although another streaming binary format may be needed. Task dependency graphs could also be useful at runtime, and APEX would need a scalable, dynamic graph representation that would aggregate data across all processes in the application. Typical APEX task dependency graphs only represent dependency classes,

not discrete task instances, and are, therefore, only on the order of hundreds of nodes. We will evaluate solutions such as TurboFlux [120] to aggregate streaming data.

**Policies.** Several dynamic runtime control decisions have been discussed in subsections 4.2 and 4.3. Where appropriate, these runtime control decisions will be implemented as APEX policies, enabling the *CAIRO* infrastructure to optimize for a given metric. While hyperparameter tuning is critical for efficient execution of first order algorithms, tuning within *CAIRO* can be implemented as APEX policies, potentially eliminating the need for exhaustive tuning sessions involving hundreds of tuning instances. Finally, we will implement new Active Harmony plugins that implement new search strategies, such as neural networks or reinforcement learning (in collaboration with University of Maryland, see letter of support).

## 4.6 WB6: Visualizing Computation State: Performance, Debugging, Interpretability

Data visualization can aid in understanding and hypothesis generation, especially when little is known about the data or the questions that should be asked of it. In addition to supporting our science drivers (see Section 5.2), we aim to address such scenarios at the intersection of AI and parallelism in this project. We plan to leverage data collected through the coupling of the computational runtime and performance measurement capabilities to elucidate core structures within our approach, allow exploration of correspondences between models and performance, and expose views to explain model behavior and outcome.

Due to the scale of the data, there will be design trade-offs in what can be shown. Interactivity, auxiliary views, and semi-automated anomaly detection will be necessary. Our approach, following best practice methodologies [159, 196, 206], will be to identify what properties need to be preserved and communicated and then to iteratively develop visual encodings and responsive, scalable methods for rendering them. We will leverage the web services developed in Sections 4.5 and 4.7 to concurrently process the data for visual presentation. We will evaluate our solutions both computationally and qualitatively with human participant studies. The resulting solutions will not only aid our team's efforts, but those of others working with large-scale matrices, optimization, or distributed computing. We foresee the following major topics:

**Scalable Representations of Multi-dimensional Arrays.** Multi-dimensional arrays are a core data type in *CAIRO*. A mental model of these arrays or, for some approaches, how their array-less counterparts relate to the array, can aid research, e.g., understanding structures in the data for more efficient algorithms. Understanding the distribution of tiles (Section 4.2) and their performance may help improve our tiling cost functions. We will design new interactive multi-dimensional array visualizations to serve these tasks and integrate them with our existing performance visualization framework (Section 3.3). We expect the design will require focus+context [39] and row/column re-ordering [12, 181] techniques and build on prior work in large matrix visualization [62, 63, 130].

**Combining Performance and Model Data.** By collecting performance and model behavior data together, we have an opportunity to debug issues that manifest in how the *values* in a program change over time. This approach is used by Anteater [67], which traces variable values to reveal issues such as gradient explosion due to poorly chosen training rates. Bugs like these do not halt computation but lead to divergent results. We will integrate with existing platforms like TensorBoard [257] that show model progress and develop additional dashboards showing performance and application behavior *in situ* where existing tools are insufficient. Such an interface has the potential to aid in steering, e.g., setting notifications based on behavior or stopping iterations early.

**Visual Components for Interpretability.** Another benefit of the integration of measurement capabilities with our runtime is the ability to collect data used for *interpretability*—insight into the internals of models and their state, ultimately to help reason about why they produce the results they do. Several visualizations of the model state exist [33, 107, 142, 193, 257], but are often geared towards the specifics of the particular model used. Leveraging our approach as a general framework, we propose to abstract common elements of existing interpretability visualizations and make them available in a composable front-end that integrates with data collection capabilities. We expect this approach will provide flexibility to support a wide set of models while also being familiar and easy to use because of its grounding in the interpretability space.

## 4.7 WB7: Research and Integration Platform

In order to make *CAIRO* 's technologies accessible to a broad audience of users, we propose to deploy a dedicated instance of the Agave Platform to be hosted primarily at LSU, with scale-out capacity available at

Chapman, and the University of Oregon. We will leverage the Agave technology ecosystem to 1) simplify and broaden access to *CAIRO* 's second order optimization technology and highly efficient trained domain models available by exposing them as secure, scalable cloud services, 2) provide tools and libraries to access *CAIRO* in a wide range of languages and technologies, 3) provide hosted visualization and analysis services that can be used and shared with others in the community, and 4) provide a SaaS interface for the community to create, collaborate, and integrate their applications as they see fit.

**Hosted Platform Services.** The Agave Platform features a collection of REST APIs providing job submission, data management, application registration, metadata management, high-level access controls, and collaboration, among its capabilities. We will use Agave's API Gateway feature to augment these APIs with microservice wrappers of the high performance ML models developed by *CAIRO* 's science drivers. These services will be registered as first class REST APIs in the *CAIRO* Agave Platform deployment and available for use by the project's science drivers as well as third-party applications. Agave's existing job submission service will be extended to support Common Workflow Language (CWL) job submission syntax, thus allowing greater interoperability with other research cyberinfrastructure providers, and a bidirectional path between campus, commercial, and NSF-funded HPC, HTC, and Cloud resource providers. A natural byproduct of this work will be explicit support for the kind of integration into edge and fog models of computing necessary to bridge between campus CI and the national cyberinfrastructure ecosystem that will begin to take shape over the life of this project. To ensure that users have the lowest possible barrier to entry, *CAIRO* 's Agave instance will utilize Custos [188] to provide InCommon [249] identity federation. This will streamline the onboarding process by allowing users to login with their campus identities. As new runtime and storage infrastructure technologies emerge over the life of the institute, Agave's web-service abstraction will continue to allow CAIRO users to leverage new technology with little or no impact on their existing workflow. An example of one such technology comes from Gregory Kurtzer's company Control Command, Inc. HPCng project (see letters of collaboration).

**Platform Tooling and Libraries.** By registering the *CAIRO* services as first class APIs in Agave, we ensure they will be accessible as extensions to Agave's existing tooling and interfaces. This includes a command line interface (CLI), software development kit (SDK) in multiple languages, and REST API. All tooling will be made available as open source software and distributed through the standard distribution channels of the target language and environment.

**Hosted Visualization Services.** The Traveler application described in Sections 3.3 and 4.6 currently focuses on visualization and interaction of one or more jobs for an individual user. We will use Agave's Serverless infrastructure to provide a secure, scalable, multi-user deployment of Traveler that integrates with Agave's existing data and job execution capabilities within the institute. This will ensure that anyone seeking to analyze the performance of their Phylanx-enabled applications has a persistent place to publish and reference their output, ensuring the same degree of privacy for their visualizations as for their data. As support for OpenTelemetry output in Phylanx matures, we will evolve the visualization service to also support ingestion of this data and rendering in familiar, open source, OpenTelemetry-compliant web interfaces such as Zipkin, Jaeger, and Kibana. This activity will come after year 2 of the project, so we will continue to monitor the evolution of the state of the art in OpenTelemetry and make a concrete technology choice at that time.

**Security.** Security of the Platform is of the highest concern. Agave has undergone a successful security review by the Center for Trustworthy CI. In addition to drawing on leading expertise from the *CAIRO* Coalition, the Platform will also benefit from other experts' knowledge, such as Dr. Irfan Ahmed from VCU. We are aware that some data sets may contain sensitive or proprietary information and we will take appropriate measures to ensure that data at rest and in transit is secured against both accidental and intentional exposure or tampering. This will involve using standard data encryption methods and sanitization techniques, already implemented by default in Agave, to securely wipe data that is no longer in use, both in the runtime and on non-volatile storage.

**Graphical User Interfaces.** *CAIRO* will have two user interfaces through which users can visually interact with the technology. The primary being a Science Gateway serving as *CAIRO* 's hosted SaaS application, and the other being a Jupyter Notebook. Both will be made available as open source projects that users can self-host and run for their own needs.

**SaaS.** The *CAIROWeb* science gateway will be the primary user-facing web interface for the institute. Building on top of the server-side version of Agave's reference science gateway, Agave ToGo, *CAIROWeb* will allow users to login using their campus identities, create, manage, and discover apps, define SciOps pipelines,

view historical performance data, perform A/B testing, create custom automation, share and manage data, and aggregate their own heterogeneous collection of resources to construct a bespoke digital lab. We will integrate the gateway with technologies from the Coastal Emergency Risks Assessment (CERA) tool (see Figure 8), which is one of the most well-known and recognized web visualization tools to disseminate storm surge forecasts to stakeholders during tropical storm events. In the context of *CAIRO* we intend to extend this technology to **(a)** serve as a validation tool for the science drivers (see Section 5), **(b)** help training future generation of talent to work with AI output (see Section 6), and **(c)** demonstrate national importance, i.e. how to bring the results to *real* end-users.

By building from a mature, existing technology in Agave ToGo, the development time of *CAIROWeb* can be significantly shortened. This will allow us to follow an Agile, interactive development cycle, incorporating user feedback from the earliest phases of the project and producing a product more aligned with the needs of our user community.

**Jupyter Notebooks.** We will build upon the existing work in the JetLag project to provide a preconfigured Jupyter environment distributed as a Docker image and Helm chart through which users can interactively build, optimize, and run their applications locally and on a Kubernetes cluster. The image will contain tutorials introducing the user to *CAIRO*, teaching them how to instrument an application with the *CAIRO* framework, build and run their application on multiple resources using the *CAIRO* CLI and SDK, construct multistep ML pipelines utilizing their application, and finally scale up their pipeline to run on *CAIRO* 's campus HPC infrastructure using discretionary allocations provided by the participating institutions.

# 5 Science Drivers (SD)

Many of today's AI applications are directly hampered by the amount of data and the size of models that can be reasonably trained. *CAIRO* has selected science drivers from four different domains that will serve as representative applications and which allow us to **conduct use-inspired research** that both informs foundational AI advances and drives innovations in related sectors of science and engineering. The selected applications have a strong impact on different aspects of national and societal importance. Specifically, the outcomes of AI/CI research in the areas of **faster learning rates for ML models, physics-informed surrogate models, robust data-driven closure models, and approximate adjoints for ML models** will be integrated into the science driver domain applications. It is expected that each of the application characteristics will be exploited in the performance improvements of domain-specific optimizers.

Postdoctoral researchers in each science domain will be designated as *CAIRO* liaisons to ensure effective communication and knowledge transfer and exchange between *CAIRO*-affiliated research units.

## 5.1 SD1: Natural Language Processing (NLP)

The ultimate goal of NLP in AI is to enable computers to understand human languages so computers can assist in different language-related tasks, such as important applications for text classification, information extraction, question answering, and machine translation. NLP research has recently witnessed a major breakthrough where extremely large language models with deep stacks of Transformer neural networks (e.g., BERT, GPT-3) [26, 52, 141, 144, 186, 239] are pre-trained on enormous amounts of text data with self-supervised learning objectives. This has helped to set a new standard for successful systems for almost all NLP tasks [140,240,275]. Despite such impressive progress, the community is facing at least two limitations that prevent existing pre-trained language models to have broader impacts. Both of them cannot be solved with most currently used systems due to their constraints in terms of reasonably supported data and model sizes, making *CAIRO* crucial for the development of pre-trained language models.

**Domain-specific pre-trained language models**: Existing pre-trained language models are mostly trained on general domain texts, e.g., Wikipedia, [52, 144, 265], without considering domain-specific texts and knowledge bases to capture valuable knowledge to support various decision making problems, e.g., the MIMIC-III knowledge base for the medical domain [103] and the National Vulnerability Database for the cybersecurity domain [17]. This has led to unsatisfactory performance of such pre-trained models for NLP tasks of domains that involve different word distributions and writing styles (e.g., in biomedical and cybersecurity domains [128, 151]).

**Multilingual pre-trained language models**: These models are trained on texts of multiple languages (e.g., multilingual BERT, XLMR) [40, 52, 104], thus enabling zero-shot cross-lingual transfer learning for NLP

models [260]. Unfortunately, the distribution of commonly used text data (e.g., Wikipedia) tends to be highly skewed toward higher-resource languages, causing low-resource languages to be underrepresented in existing pre-trained language models [261]. Additionally, there is a mismatch between the advances in mono- and multilingual language models as many recent monolingual language models for English (e.g., ELECTRA [37], GPT-3 [26]) have not been explored and evaluated for multilingual settings.

These limitations call for fundamental development of novel technologies to effectively exploit domain-specific texts and knowledge bases, and efficiently consume texts in low-resource languages, that in all require significant computational resources and time to support extensive training and evaluation of novel methods. For instance, the large XLNet model [265] consists of 340 million parameters trained on 330 billion words in 2.5 days with 512 TPU chips (for a single training time). Owing to its ability to significantly accelerate the training of deep learning architectures and handle large-scale data sets, the efficient framework for second order optimization in *CAIRO* will serve as a game changer to make large-scale language model development for different domains and languages feasible and accessible to the public. Nguyen's group at UO has collected domain-specific texts and knowledge bases for medical and cybersecurity domains [147,148,151] and created various NLP systems based pre-trained language models [125,183,184,241] to serve as the evaluation framework for this research.

## 5.2 SD2: Hurricane Storm Surge and Flood forecasting

The well-being of all Americans depends on the environmental integrity and sustainable productivity of the ocean, our coasts, and coastal watersheds. More than half of the population of the United States lives in coastal watershed counties or parishes, and generate 58% ($8.3 trillion) of the Nation's GDP, even though they comprise only 25% of the Nation's land area [237]. Coastal communities and associated infrastructure are especially susceptible to wind and flooding due to tropical storms, hurricanes, and heavy rainfall events, which are increasing in frequency and intensity.

Because of its accuracy and flexibility, the Advanced Circulation model (ADCIRC [30,54,55]) has been used extensively by national agencies (e.g. FEMA, USACE, U.S. Coast Guard, DoT) in predicting and analyzing hurricane storm surge and is—in conjunction with the CERA visualization framework (see Figure 8)—an important tool for state emergency management agencies along the U.S. coast to forecast storms as they approach landfall and to make decisions about evacuations, deployment of first-responders, transportation, etc. The ADCIRC model, however, is only capable of capturing flooding from storm surge. In light of recent tropical cyclone events on the U.S. coast, such as Hurricanes Harvey and Florence, it has become evident that modeling approaches that combine flooding induced by rainfall runoff and storm surge should be considered to inform real-time decision-making and resilience of coastal regions to future events. No comprehensive modeling of compound storm surge and rainfall events exists beyond academic research or region-specific models and have not been systematically coupled over a wide spectrum of flow regimes. Recent explosive advances in the use of ML algorithms for watershed modeling are a promising avenue for coupling coastal models with ML models.

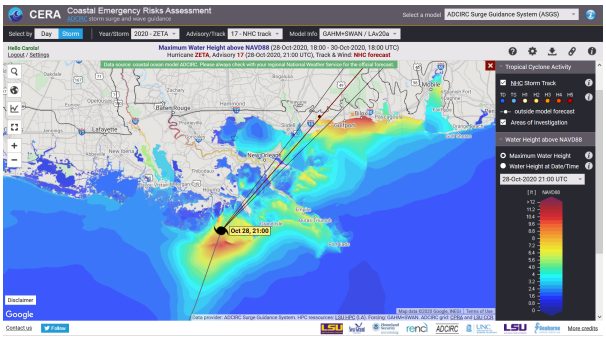



Figure 8: Predicted storm surge from ADCIRC for hurricane ZETA on Oct 28, 2020, disseminated by the CERA web visualization tool.

In the context of *CAIRO*, we will investigate three topics using AI for coastal modeling: **ML for storm surge applications**: previous work [96, 194] done a decade ago, still promises new avenues for computing fast and accurate surrogate models that may be used, for example, in parameter estimation [73, 157], in determining flood risks at particular locations, and in quantifying uncertainties in hurricane track and intensity (e.g. in a forecast scenario). This is a novel application of ML with medium-risk, high-reward potential. **ML for rainfall runoff modeling**: in prior work we have shown that compared to conventional watershed models, a deep learning sequence model can simultaneously achieve significantly quicker calibration, faster prediction, and improved accuracy [115, 124, 137, 138]. Coupling ML models for time series data with coastal models such as ADCIRC can provide ADCIRC with critical information needed to sim-

ulate compound floods. We anticipate that an efficient second order optimization algorithm can notably speed up the training process of the ML models for watershed modeling. **ML algorithms for compound flooding**: Dawson's group at UT Austin has developed an open-source Python framework, Water Coupler, for two-way dynamic coupling of ADCIRC, hydrologic models (GSSHA, HEC-RAS), and ML models to simulate compound flooding events [35,36].

## 5.3 SD3: Sustainable Energy Resource Applications

High performance computing applications such as computational fluid dynamics (CFD) and geothermal reservoir simulations can provide large parametric ensemble data sets for the optimization of specific objective functions such as the designs for the lowest bed shear stress or the exploration of target subsurface formations with the highest extraction rates for enthalpy. It is expected that exploiting the curvature information in the second-order methods [221,247] will improve the computational performance of ML surrogate training by at least an order of magnitude. Two main engineering applications are proposed here that can benefit from the development of the scalable AI optimizers from *CAIRO* research.

**Surrogate models using machine learning for fracture network stimulation during geothermal reservoir exploration and extraction processes**: Parametric study of engineering applications associated with the sustainable energy resources exploration and extraction are addressed through simulations of coupled Multiphysics processes in the deep subsurface environments [123].

However, there are large uncertainties and insufficient information to characterize the subsurface environment accurately, and, therefore, a large ensemble of reservoir simulations must be carried out to understand the potential of heat extraction [222]. Typically, enhanced geothermal systems require fracture network stimulation to provide a large heat exchange surface area for increased heat transfer to the fluids between the injector and producer wells. Peridynamics [213], as well as phase field approaches, will be adopted for modeling the coupled physics phenomena of creating and evolving a fracture network during the geothermal heat extraction process. Simulation results will provide image data sets and time series about multiple fields to build physics informed neural networks (PINNs) via deep learning [97,180]. Using second-order optimizers for PINNS would involve increased calculations and storage during each epoch and requires a computationally scalable and efficient ML surrogate development approach [187].

**Bayesian optimization for the design of experiments for Offshore Coastal and Petroleum engineering structures**: Coastal engineering structures, offshore petroleum drilling, and production platforms are subjected to time-varying loads due to waves [172], tides, and frequent hurricanes. The Gulf of Mexico oil and gas production, as well as coastal sustainability, is of significant national and regional importance in terms of energy independence, sustainability of dependent industries, and the environment. Specifically, the development of reduced-order models for the processes related to the sediment bed scour around the coastal structures using a machine learning approach on high-resolution CFD datasets is proposed [129] using open-source toolkits such as OpenFOAM [98,250]. Bayesian Optimization using Gaussian processes with second-order derivatives information [259] will be used for searching the design space. The CERA and Traveler visualization frameworks will be adopted in this SD application.

## 5.4 SD4: Geological Simulations

Explosive volcanic eruptions can inject voluminous amounts of ash into the atmosphere, potentially crippling global infrastructure, as witnessed by the recent relatively small eruption of Eyjafjallajökull in remote Iceland in 2010. This incident caused losses of over $\$4.7$ billion due to far reaching dispersal of volcanic ash over European air space [61,219]. Closer to a volcano, volcanic particles and gas can collapse into pyroclastic density currents (PDC) that can rapidly inundate nearby populations (e.g. Fuego, Guatemala 2018). Both volcanic plumes and PDC are multiphase, often turbulent and compressible, multiscale flows [59]. While micro-physical interactions between fluids and solids can have significant impacts on the propagation of these flows [65,224], fully resolving these flows is not tractable even in the smallest volcanic eruption. Further, while high resolution research simulations [167,173,238] have been useful in elucidating the dynamics of eruptive events, these approaches are too computationally costly and slow to be performed for hazard analyses during an ongoing volcanic crisis.

In the context of *CAIRO* we plan to apply the developed AI techniques to **improve understanding of dynamics and in predicting ash dispersal** that is currently hampered by a limited description of the proximal dynamics of volcanic eruptions and by our limited understanding of the volcanic processes occurring

near the vent on spatial scales of constituent particles and gases [248]. A complimentary effort will involve **the prediction of the emergent flow behavior of PDC** based on integrating discrete element method approaches as training sets for coarse-graining [23] granular flows and incorporation as a rheological surrogate in continuum approaches.

Efforts are underway in co-PI Dufek's group to develop a set of libraries of high resolution simulations that can be used as starting conditions for dispersal models (in collaboration USGS, see letter of support). These simulations will provide training sets to develop surrogate ML models that can be readily applied to conditions at a particular volcano. An ML approach offers a solution that is much faster than a suite of simulations during a crisis, is flexible to better incorporate information from on-going observations (satellite, seismic, infrasound), and provides a platform for more robust quantification of uncertainty that can be communicated to dispersal simulations. This approach has been applied successfully in related applications incorporating remote sensing data into Earth systems applications [60, 192]. Much as in the plume modeling case, applying ML to PDC related simulations enables a quantification of uncertainty, will be more rapid than concurrent DEM simulations, and will better capture the physical processes at play.

# 6 Education and Workforce Development

*CAIRO* is a *distributed* AI Institute, bringing together 8 institutions, led by LSU. The PIs have extensive experience in mentoring dozens of graduate students and early career researchers. In our broadening participation plan (see Section 7), we have outlined ways to foster a diverse pipeline of talent and build awareness of AI opportunities, including informal summer K-12 programs targeting underrepresented groups in STEM and a focus on providing structured undergraduate research mentorship at each participating institution.

*CAIRO* will recruit, retain, and train a diverse cohort of undergraduate, graduate and postdoctoral students, helping to produce the next generation of AI researchers and workers. It offers innovative rotational training opportunities and research-driven pedagogies through a distributed model of delivery. New graduate courses, developed at LSU, will be accessible to center affiliates at any of the *CAIRO* sites. In addition, *CAIRO* emphasizes the ethical use of AI and will foster ethical AI research and train researchers to identify important risks associated with AI. Please see our Ethics Plan for a detailed roadmap on ethics-centered education and programming for all institute affiliates.

**Graduate and Post-Doctoral Education: Leveraging the Consortium.** In addition to these initiatives, *CAIRO* offers an innovative model for graduate and post-graduate education, that leverages and integrates the diverse research expertise of our affiliates:

- *CAIRO* will offer its inter-institutional cohort of graduate students bimonthly research seminars (organized by the postdoctoral fellows) focusing on an area of inquiry, science driver, or applied technique.
- Graduate students will be able to take advantage of summer research rotations, working in the various *CAIRO* affiliated institutions and labs under the direction of one of our PIs.
- An annual intra-institute conference will allow graduate students and postdoctoral students to share their research results and get feedback from leading researchers.
- In year 2-5, *CAIRO* will roll out three inter-disciplinary, broadly accessible graduate course designs. The first course, entitled "Applied Distributed AI" will teach interested parties how to use existing AI tools and workflows. Topics will include training to use distributed AI and ML frameworks, including accessing distributed systems, scripting under the frameworks, executing in parallel and obtaining results.
- *CAIRO* will facilitate industry and lab internships for undergraduates, graduate students, and postdocs. Please see the letter of collaboration from Dr. Victor Shen offering internship position, for eligible participating institutes, at MITRE. Many of our PIs enjoy robust connections with industry leaders. As an example of industry engagement, co-PI Tyagi was a founding member and co-director of "Enabling Process Innovation through Computation (EPIC)" consortium that brought in industry-funded research projects and hosted high-profile seminar speakers during 2013-2018.
- co-PI Dawson and SI C. Kaiser have long term experience in reaching out to regional and national first-time responders with workshops and tutorials on analyzing hurricane forecasting data. *CAIRO* will leverage these efforts to introduce data visualization to postdocs, graduate students, and the wider community. Please see letter of collaboration from Dr. Robert Twilley from Sea Grant which has a strong research, education and outreach program to solve problems along the coast.

# 7 Broadening Participation Plan

The distributed structure of *CAIRO* is a major asset for achieving broader social, economic, and educational impacts. Led by LSU, an R1 University in an EPSCoR state, *CAIRO* brings together researchers and research institutions that overall serve over 220,000 undergraduates and graduate students. The *CAIRO* consortium is comprised of 8 public and land-grant universities, including one designated Hispanic Serving Institution (University of Arizona). *CAIRO* 's senior personnel are already contributing extensively to inclusion and diversity efforts in CS and STEM, with a proven track record of outreach efforts, and recruiting/mentoring women and minority students, e.g.:

- PI Kaiser and SI Diehl have participated in Google Summer of Code program for the past seven years [53], as well as the LSU REU program, and have mentored over 20 students through these venues. This work has resulted in several publications and significant student involvement resulting in over 60 pull requests to HPX and other open source projects, and mentoring.
- Both co-PI Isaacs and co-PI Huck have mentored through PI Kaiser's Google Summer of Code program. SI Brandt has overseen the LSU Beowulf Bootcamp, which teaches high school and junior high students from underrepresented groups about HPC through talks and hands-on activities since 2007.
- co-PI Isaacs (Arizona) is the faculty advisor to the UArizona Women in Information and Computer Science (WICS) student organization and hosts students through ASEMS, a STEM diversity program focusing on undergraduate research at Arizona.
- co-PI Banarjee, member of the CS undergraduate committee at MST, works closely with a host of outreach and women's programs including SWE, WIE, SDOWP, and ACM-W to achieve undergraduate Diversity, Equity, and Inclusion (DEI) goals.
- co-PI Dooley has participated as an undergraduate mentor and guest lecturer to HBCU through the Science Gateways Summer Institute since 2014.

*CAIRO* will leverage this collective experience and the networks of our PIs to existing diversity and inclusion efforts to execute on the following three foci for broadening participation in AI: **(a)** fostering inclusive undergraduate research; **(b)** recruiting and mentoring a diverse cohort of STEM practitioners from undergraduate through post-graduate; **(c)** Summer educational enrichment initiatives for underrepresented junior high and high school students. Extensive research supports the "pipeline model" for achieving reaching diversity and inclusion goals. In this model, practitioners recruit and retain under-represented groups (including women and BIPOC) at the various stages of the educational lifecycle. *CAIRO* pursues a BP approach that will make impactful interventions for K-12, undergraduate, graduate, and postgraduate levels, with a special focus on driving inclusive undergraduate research in AI and productive mentoring arrangements for graduate and postgraduates.

**Undergraduate Education & Research**: Research demonstrates that undergraduate research experiences for undergraduates increase their knowledge and confidence in STEM fields, and leads to students pursuing STEM graduate degrees and careers at higher rates [146, 208]. Increasing participation in undergraduate research for underrepresented groups is a priority at many of our member institutions, and programs like McNair Scholars.

**Goals:** PIs at member institutions will commit to specific goals for supporting inclusive undergraduate research in their labs. *CAIRO* affiliates will mentor a minimum of 15 undergraduate researchers distributed across its 8 campuses. Graduate and postgraduate students affiliated with *CAIRO* will lead monthly research webinars for our undergraduate researchers and serve as mentors.

**Strategy and Evaluation:** Working with the offices of undergraduate research at our respective institutions to ensure diversity, we have identified partners including LSU Aspire and Discover programs, McNair diversity scholar programs and Distributed Research Experiences for Undergraduates (REU). As the lead organization, LSU can also draw upon the Halliburton Research Scholar program, which funds experiential learning opportunities to underrepresented undergraduate. EDAB will help senior personnel identify undergraduate research partnerships and evaluate impacts over the life-cycle of the grant.

**Broadening Participation through Recruitment & Retention**: In the US, two-year (community) colleges account for 41% of undergraduate enrollment, including the majority of those students underrepresented in STEM fields [182]. Because nearly half of all students earning bachelor's degrees in STEM begin progress towards their degree at community colleges, broadening participation in AI requires a commitment to recruiting for AI fields at community colleges [119]. Minority-Serving Institutions (Including HBCUs and

Hispanic-Serving Institutions-HSI) recruit, retain, and graduate minority undergraduate students in STEM at impressive rates. "HBCUs represent seven of the top eight institutions that graduate the highest number of Black undergraduate students who go on to earn S&E doctorates" [256].
**Goals:** Increase diversity of AI researchers and workforce by recruiting undergraduate and graduate students from community colleges and minority-serving institutions into STEM and AI fields.
**Strategy and Evaluation:** *CAIRO* PIs will commit to establishing or extending outreach and recruiting efforts to CS and IT programs at 2-year colleges. co-PI Dooley will coordinate these efforts. *CAIRO* affiliates will deliver accessible presentations or webinars at local two-year colleges on AI prospects, problems, and social impacts. *CAIRO* PIs will recruit graduate students and postdoctoral students from HBCUs. LSU will leverage existing strong relationships with Southern University, Xavier, and Dillard State University (HBCUs). Co-PI Isaacs' lab is also located at the University of Arizona, a designated HSI. Impact will be assessed according to the number of outreach efforts initiated and students recruited.

**K-12 Informal Education & Outreach**: The National Research Council (NRC) reports that informal learning environments can be effective tools for increasing awareness about STEM and increasing the perception of its value among communities under-represented in STEM, when designed to be intellectually and emotionally engaging, culturally responsive, and connected to learning experiences (NRC 2009, 2015). LSU CCT, the lead *CAIRO* institution, has successfully run a summer CS/IT engagement program aimed at under-represented junior and senior high school student, Beowulf Bootcamp, for 13 years.
**Goals:** *CAIRO* will develop and run (at all member sites) an AI Summer Program (based on the successful model of LSU Beowulf Bootcamp). The goal of this summer program is to host up to 20 students at each project site one week each summer, beginning in year 2 of the grant. Topics will include demonstrations of algorithms, ML and optimization, and explorations of the ethical and social aspects of AI.
**Strategy and Evaluation:** Building on LSU's Beowulf experience of successful engagement, the LSU team, led by SI Diehl and the LSU Ethics Institute, will develop a framework and curriculum for *CAIRO* AISP that can then be replicated at the member institutions. Impact will be assessed in terms of number of students served, inclusive representation, and success of skills transfer.

# 8 Collaboration and Knowledge Transfer

The synergies possible in *CAIRO* due to the extensive range of expertise and multifaceted research agenda will be facilitated by deliberate cross-institute coordination. We will build on the experiences collected during long-standing, very successful collaborations between different organizations and individuals involved with *CAIRO*. For the structure and responsibilities of the management team and its operation see Section 9.

**Intra Institute Coordination.** Continuous interaction, data-sharing, and cross-training activities among the eight institutions, as well as with the wider community, will be conducted using team collaboration tools such as Slack (or similar). Software development activities will be focused around github and its CI/CD toolchains. Project meetings will be held using teleconference tools such as Zoom or Microsoft Teams. To the extent possible we will seek to integrate the various technology platforms used for collaboration so that information in all its forms can flow freely among *CAIRO* personnel. The entire group (Project Execution Team, collaborators, students, and postdocs), as well as relevant invitees, will meet at least once a year (to include a kickoff meeting at project initiation). Locations for these meetings (including whether they are in-person or virtual) will be determined as the project progresses. These meetings will include progress reports as well as discussions and demonstrations of new techniques and approaches. Meeting proceedings and the availability of *CAIRO* capabilities will be shared on the project web site, individual PI web pages, and more broadly via a social media presence that will be established for *CAIRO*. Smaller, focused, in-person meetings will also be scheduled in conjunction with major conferences such as SC or NeurIPS.

**The Proposed Research in Relation to Other Research Groups.** The scale of the proposed research plus its relationship with other current and planned research projects dictate effective coordination and collaboration. For the development of new algorithms and techniques, we are collaborating with other groups beyond the *CAIRO* project. Broad collaboration among our groups minimizes redundancy in code development while increasing the efficiency of dissemination and analysis of results. *CAIRO* will actively invite researchers and leads from other established AI Institutes to exchange the lessons learned and issues to avoid. Developed algorithms and applications will be of more utility than the scope of the project and it will be essential to provide detailed information to the research community, beyond releasing the code

under open source licenses. All source code, data, and additional materials produced by the project will be made available publicly, through the project website, Github repositories, or other means as outlined in the Data Management Plan.

**Outreach.** Multiple mechanisms for inclusion in the group's activities will be established and these will be open to all interested parties, fueling *CAIRO*'s growth into a nation-wide center of excellence and a vital information dissemination hub. To encourage broader engagement, calls for participation for *CAIRO* activities will be promoted through social media, appropriate professional society SIGs, member institutions, and institute partners. We will offer tutorials on major pieces of *CAIRO* technology and establish one or more regular workshop series on scalable AI and related areas.

# 9 Key Personnel, Management and Integration Plan

The management structure for *CAIRO* (Figure 9) is designed to effectively implement and assess the project goals, promote project-wide participation in leadership, ensure effective communication among collaborating institutions, and establish *CAIRO* as a vital presence in the national and international AI research community. PI Kaiser will serve as the overall Project Director (PD) for the duration of the project. The project director will be assisted by a dedicated project manager (PM) who will reside at LSU and oversee the day-to-day activities of the project. Co-PI Lumsdaine will serve as the Chief Scientist for the project and will oversee and coordinate all major technical activities within the Institute.

A Project Execution Team (PET) for the project will be comprised of the following members, who will be responsible for defined tasks as follows (see also Figure 9):

- The Project PIs from the universities will oversee the activities related to their respective work items and will perform the communication and coordination with the science driver teams.
- co-PI Tyagi who will oversee the proposed External Engagement, and SI Diehl will oversee all Workforce Development activities.
- The Evaluation and Assessment (E&A) will be coordinated by co-PI Isaacs who will assess all activities of the the PET and be responsible for providing assessment data to the External Review Board.
- SI Goldgaber will serve as the Ethics and Diversity Advisor who will interact regularly with the PET to assess progress towards achieving project broadening participation and diversity goals and ensuring all center affiliates complete the ethics requirements.

The Project Execution Team (PET) will oversee the activities of the project and provide direction and guidance to project participants in each of the science driver and computational teams. Each science driver team will have a dedicated postdoc responsible for the coordination with other groups. The PET members are all experienced scientists with the requisite management skills to lead large multidisciplinary projects. They will play vital roles in ensuring effective communications between and within institutions. The co-leads will assume leadership responsibilities to ensure a smooth succession of leadership, should it become necessary. The PD will meet with the PM and the PET on a monthly basis.

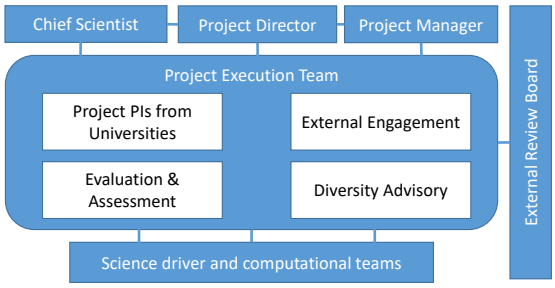



Figure 9: The *CAIRO* Management Structure

An External Advisory Board (EAB) comprised of diverse internationally-recognized experts in *CAIRO*'s focus areas related to research, diversity, workforce development, external engagement, and assessment will conduct regular reviews of program activities. A subset of the EAB will be comprised of individuals from industry to insure *CAIRO* has that vital perspective. EAB reviews will cover all aspects of the project, including research conducted by the institute, broader impacts, evaluation, and assessment data provided by the E&A lead, and will include an annual site visit. The EAB will provide objective guidance, feedback, and recommendations to the PD and PET to ensuring program goals and objectives are being met.

**Qualifications of Key Personnel.** PI Kaiser has extensive experience in leading a large research group and leading successful collaborative efforts involving multi-institutional, inter-disciplinary groups of scientists. Although a specific individual has yet to be identified as a program manager, the Center of Computation and Technology at LSU has a long history of hiring exceptional PMs for large projects and empowering them to lead projects to success. Co-PI Lumsdaine has similarly participated in a variety of large research

| Task | WB1 | WB2 | WB3 | WB4 | WB5.1 | WB5.2 | WB5.3 | WB6.1 | WB6.2 | WB6.3 | WB7.1 | WB7.2 | WB7.3 | WB7.4 | WB7.5 | SD1 | SD2 | SD3 | SD4 | WB9 | WB10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LSU | ⊙ | ⊙ | ● | ● | | | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ● | ⊙ | ● | ● |
| GMU | ⊙ | ● | | | | | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | | | | ⊙ | | | | | ● | ● |
| CU | | | | | ⊙ | ⊙ | | | | | ● | ● | ● | ● | ● | | | | | ● | ● |
| MS&T | ⊙ | ● | | | | | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | | ⊙ | | ⊙ | | | | | ● | ● |
| UA | | ⊙ | | | | ⊙ | | ● | ● | ● | ⊙ | | ⊙ | | ⊙ | | | ⊙ | | ● | ● |
| UW | ● | ⊙ | | | | | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | | | | ⊙ | | ⊙ | ⊙ | ⊙ | ● | ● |
| UO | | ⊙ | ⊙ | ⊙ | ● | ● | ● | | ⊙ | | ⊙ | | ⊙ | ⊙ | ⊙ | ● | ⊙ | ⊙ | ● | ● | ● |
| UT | | | | | | | | | | | ⊙ | | | | ⊙ | ⊙ | ● | ⊙ | ⊙ | ● | ● |

Figure 10: Project work breakdown (WB) structure for research activities. Lead institutions are indicated with a •, participating institutions are indicated with a ⊙. Labels: *WB1 Deep Learning via Second-order Optimization (§4.1); WB2 Runtime System and Program Optimization (§4.2); WB3 Phylanx and HPX (§4.3); WB4 CAIRO and Computing Hardware Evolution (§4.4); WB5.1 Performance Monitoring (§4.5); WB5.2 Aggregation and Streaming (§4.5); WB5.3 CAIRO Policies (§4.5); WB6.1 Matrix Visualization (§4.6); WB6.2 Model and Performance Visualization (§4.6); WB6.3 Interpretability Visualization (§4.6); WB7.1 Platform API development (§4.7); WB7.2 Tooling and libraries (§4.7); WB7.3 Hosted visualization services (§4.7); WB7.4 SaaS Science Gateway (§4.7); WB7.5 Jupyter environment (§4.7); WB8.1 SD1: Natural Language Processing (NLP) (§5.1); WB8.2 SD2: Hurricane Storm Surge and Flood forecasting (§5.2); WB8.3 SD3: Sustainable Energy Resource Applications (§5.3); WB8.4 SD4: Geological Simulations (§5.4); WB9 Workforce Development (§6); and WB10 Broadening Participation (§7).*

projects, both as an investigator and in leadership roles. The PIs from *CAIRO*'s participating institutions are all leaders in their fields and, significantly, already have a history of working with each other.

In summary, by leveraging current facilities, we will build nationwide interdisciplinary research collaborations involving domain scientists, computer experts, applied mathematicians, and theorists. We will build a sustainable interdisciplinary and inter-institutional AI science graduate program.

## 10 Broader Impacts

The ultimate goal of *CAIRO* is to enable a "punctuational change" in AI. If successful, *CAIRO* will have far-ranging impacts across science, technology, industry, and society—really, on any researcher, developer, or consumer of AI. With the expected orders-of-magnitude performance improvements, scientists, AI researchers, and practitioners will be able to develop immensely more sophisticated models and conduct entirely new classes of explorations. These models will in turn be able to solve entirely new classes of problems, opening up new scientific vistas and providing new end-user experiences to consumers.

The *CAIRO* PIs established track record of developing and delivering high-quality open-source software will ensure that the artifacts resulting from this work will be readily accessible and able to be rapidly incorporated into end-user applications. By creating a software layer that industrial partners can rely on, the project will help fill the gap between academic innovation and commercial application.

*CAIRO* will have numerous direct societal benefits, including those resulting from our Broadening Participation plan described in Section 7. Project funds will support societal values through targeted inventions aimed at broadening participation in AI and investing in AI ethics research and training. *CAIRO*'s partnership with the LSU Ethics Institute, which includes funding for a postdoctoral researcher, will produce project-driven, publishable research in the domain of AI ethics and managing ethical risks in AI, from discrimination and bias in algorithms to privacy and security challenges. This postdoctoral fellow and the Director of the LSU Ethics Institute (SI Goldgaber) will serve as the Ethics and Diversity Adviser on the PET.cutting-edge AI Ethics training the inter-institutional partners across 8 universities (see "Ethics Plan").

The lead institution (LSU) resides in an EPSCoR state. Funding this research will foster the growth and development of AI and HPC in Louisiana. The project will directly provide undergraduate, graduate, and post graduate opportunities to institutions in eight states, which is vital to fostering existing industries and creating new industries with AI/ML technology. *CAIRO*, in particular, lays a solid foundation for technology-transfer from academia to industry.

# 11 Results from Prior NSF Support

**Hartmut Kaiser, LSU**, *Phylanx: Python-based Array Processing in HPX* (NSF grant 1737785, $373,200, 8/15/17 – 31/7/19) that creates an infrastructure for distributed array processing and machine learning. *Intellectual Merit:* The project focuses on the interrelationship of parallelism and overhead, ultimately determining the practical range of attainable scalability and includes immediate impacts in advancing the specific science domain. It *broadly impacts* many problems related to machine learning applications and has supported several graduate students. *Research Products:* Publications [20,21,80] and open source tools [109].

**Clint Dawson, UT Austin** SI2-SSI: Collaborative Research: STORM: A Scalable Toolkit for an Open Community Supporting Near Realtime High Resolution Coastal Modeling, ACI-1339801, $540,0000, 09/01/2014-08/31/2018. *Intellectual merit.* The project goal is to evolve ADCIRC, a free-surface-flow coastal circulation model, into a dynamic framework that readily admits new/recently-developed solution algorithms. *Broader Impact*: The STORM project is expected to have a great impact on the coastal ocean modeling, computational mathematics, and computer science communities. ADCIRC is widely-used with a large and expanding user base. *Products*: Publications [5,24,29,160,161,200–203,254] and 3 Ph.D. theses.

**Rion Dooley, Chapman University**, OAC-1906052, *The Agave Platform: An Open Science-as-a-Service Cloud Platform For Reproducible Science* (10/1/2018-7/31/2021; $1,240,246). *Intellectual Merit:* The goal of this project is to close the capability gap between academic and commercial infrastructure by hardening the Agave Platform, an open, Science-as-a-Service cloud platform for reproducible science. *Research Products:* 18 publications, one Master's Thesis, over a dozen open source products, two independently funded spin-off technologies. A key technology recommended by the Science Gateway Community Institute.

**Kevin A. Huck, University of Oregon**, *Phylanx: Python-based Array Processing in HPX* (NSF grant 1737785, $93,300, 8/15/17 – 31/7/19). Built on HPX, the Phylanx project focuses on creating an infrastructure for distributed array processing and machine learning. *Intellectual Merit:* This project's goal is to build a distributed Machine Learning platform that provides users a high-level Python interface with HPC performance. *Broader Impacts:* Our results have led to a better understanding of the performance ramifications of solving general array processing and specific machine learning algorithms using a predefined set of parallel operations and employing algorithms which optimize execution and data layout from a user-provided expression graph. *Research Products:* Publications [234,244] and open source software [95].

**Katherine E. Isaacs, University of Arizona**, IIS-1656958, *CRII: III: Scalable and Interactive Dependency Visualization to Accelerate Parallel Program Analysis* (7/1/2017 to 6/30/2021; $174,518). *Intellectual Merit:* This project's goal is identifying design factors that increase the efficacy of visual tools for large-scale parallel program dependency analysis. *Broader Impacts:* Visualizations developed during this project have aided researchers understand emergent behavior in program control flow and parallel runtime performance. The project has supported three students and led to the development of a new human-computer interaction course. *Research Products:* Publications [50,51,253] and open-source tools [10,48,49,252].

**Fei Li, George Mason University**, CCF-1216993, *Algorithmic Approaches to Energy-Efficient Computing* (08/01/2012 to 07/31/2016; $128,325.00) *Intellectual Merit:* This project studies algorithmic methods for improving energy efficiency of data processing and storage in large-scale networked computing systems. It addressed both specific energy optimization problems and produced new algorithmic techniques, as well as deepened understanding of the adequacy of standard performance enhancement tools (e.g., caching and load balancing) for improving energy efficiency. *Research Products:* More than 20 publications in top journals and conferences; supported two women Ph.D. students.

**Andrew Lumsdaine, University of Washington**, ACI 1716828 SI2-SSE: GraphPack: Unified Graph Processing with Parallel Boost Graph Library, GraphBLAS, and High-Level Generic Algorithm Interfaces (10/01/2016 - 09/30/2019, $499,386). *Intellectual Merit.* This work will provide insight into how aspects such as graph structure, parallelization, runtime, and hardware have on the performance and scalability of graph computation and will provide a robust open-source software platform for the larger research community. *Broader Impacts.* GraphPack will improve the use of graph algorithms in diverse areas including knowledge discovery, genomics, proteomics, electronic design automation, power grid management, etc. with immediate appeal to students, forming a natural path from intuitively familiar things to their computational underpinnings. *Research Products:* Publications [6,32,68,113,114,143,149].

# References


[1] A. Agelastos, B. Allan, J. Brandt, P. Cassella, J. Enos, J. Fullop, A. Gentile, S. Monk, N. Naksinehaboon, J. Ogden, M. Rajan, M. Showerman, J. Stevenson, N. Taerat, and T. Tucker. The lightweight distributed metric service: A scalable infrastructure for continuous monitoring of large scale computing systems and applications. In *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (2014), pp. 154–165. http://dx.doi.org/10.1109/SC.2014.18.

[2] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems* (2017), pp. 1709–1720.

[3] K. Andreev and H. Racke. Balanced graph partitioning. *Theory of Computing Systems 39*, 6 (2006), 929–939.

[4] E. Angel, E. Bampis, and L. Gourvés. Approximating the pareto curve with local search for the bicriteria tsp (1, 2) problem. *Theoretical Computer Science 310*, 1-3 (2004), 135–146.

[5] H. Arabshahi. *Space–Time Hybridized Discontinuous Galerkin Methods for Shallow Water Equations.* PhD thesis, The University of Texas at Austin, 8 2016.

[6] A. Azad, M. M. Aznaveh, S. Beamer, M. Blanco, J. Chen, L. D'Alessandro, R. Dathathri, T. Davis, K. Deweese, J. Firoz, et al. Evaluation of graph analytics frameworks using the gap benchmark suite. In *2020 IEEE International Symposium on Workload Characterization* (2020).

[7] J. Ba, R. Grosse, and J. Martens. Distributed second-order optimization using kronecker-factored approximations. Open Review submission, 2016.

[8] C. Bakker, M. J. Henry, and N. O. Hodas. The Outer Product Structure of Neural Network Derivatives. *arXiv:1810.03798 [cs, stat]* (Oct. 2018). arXiv: 1810.03798, http://arxiv.org/abs/1810.03798.

[9] I. Banerjee and P. Das. Group technology based adaptive cell formation using predator–prey genetic algorithm. *Applied Soft Computing 12*, 1 (2012), 559–572.

[10] J. Bartels. Roundtrip. https://github.com/hdc-arizona/roundtrip, 2019.

[11] D. Basu, D. Data, C. Karakus, and S. Diggavi. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. In *Advances in Neural Information Processing Systems* (2019), pp. 14695–14706.

[12] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, and J.-D. Fekete. Matrix reordering methods for table and network visualization. *Comput. Graph. Forum 35*, 3 (June 2016), 693–716.

[13] E. Betke and J. Kunkel. Real-time i/o-monitoring of hpc applications with siox, elasticsearch, grafana and fuse. In *International Conference on High Performance Computing* (2017), Springer, pp. 174–186.

[14] O. Bhardwaj and G. Cong. Inefficiency of stochastic gradient descent with larger mini-batches (and more learners). Open Review submission, 2016.

[15] A. Bhatele, N. Jain, Y. Livnat, V. Pascucci, and P.-T. Bremer. Analyzing network health and congestion in dragonfly-based supercomputers. In *Proceedings of the IEEE International Parallel & Distributed Processing Symposium* (may, 2016), IPDPS.

[16] Blaze high-performance C++ math library for dense and sparse arithmetic, 2020. https://bitbucket.org/blaze-lib/blaze/src/master/.

[17] H. Booth, D. Rike, and G. Witte. The national vulnerability database (nvd): Overview. *NIST Technical Report* (2013).

[18] A. Botev, H. Ritter, and D. Barber. Practical gauss-newton optimisation for deep learning. *arXiv preprint arXiv:1706.03662* (2017).

[19] S. R. Brandt. Docker repository for the integrated phylanx jupyter notebook solution. https://hub.docker.com/r/stevenrbrandt/trav, 2020. Last Accessed October 2020.

[20] S. R. Brandt, A. R. Bigelow, S. A. Sakin, K. Williams, K. E. Isaacs, K. Huck, R. Tohid, B. Wagle, S. Shirzad, and H. Kaiser. Jetlag: An interactive, asynchronous array computing environment. In *Practice and Experience in Advanced Research Computing* (July 2020).

[21] S. R. Brandt, B. Hasheminezhad, N. Wu, S. A. Sakin, A. R. Bigelow, K. E. Isaacs, K. Huck, and H. Kaiser. Distributed asynchronous array computing with the jetlag environment. In *Proceedings of the 9th Workshop on Python for High-Performance and Scientific Computing* (Nov. 2020).

[22] S. R. Brandt, D. M. Koppelman, and Y. Hu. Chemora kernel mapping optimization, August 2015. http://www.ece.lsu.edu/koppel/pubs/chemora.pdf.

[23] E. C. Breard, J. Dufek, L. Fullard, and A. Carrara. The basal friction coefficient of granular flows with and without excess pore pressure: implications for pyroclastic density currents, water-rich debris flows, rock and submarine avalanches. *Journal of Geophysical Research: Solid Earth* (2020), e2020JB020203.

[24] M. Bremer, K. Kazhyken, H. Kaiser, C. Michoski, and C. Dawson. Performance comparison of hpx versus traditional parallelization strategies for the discontinuous Galerkin method. *Journal of Scientific Computing 80* (2019), 878–902.

[25] P. N. Brown and Y. Saad. Hybrid Krylov Methods for Nonlinear Systems of Equations. *SIAM Journal on Scientific and Statistical Computing 11*, 3 (May 1990), 450–481. http://epubs.siam.org/doi/10.1137/0911026.

[26] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krüger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *ArXiv abs/2005.14165* (2020).

[27] T. Brunsch, N. Goyal, L. Rademacher, and H. Roglin. Lower bounds for the average and smoothed number of pareto-optima. *Theory of Computing 10*, 10 (2014), 237–256.

[28] T. Brunsch and H. Roglin. Improved smoothed analysis of multiobjective optimization. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC)* (2012), pp. 407–426.

[29] S. R. Brus, D. Wirasaet, J. J. Westerink, and C. Dawson. Performance and scalability improvements for discontinuous Galerkin solutions to c onservation laws on unstructured grids. *Journal of Scientific Computing 70*, 1 (Jan 2017), 210–242. https://doi.org/10.1007/s10915-016-0249-y.

[30] S. Bunya, J. C. Dietrich, J. J. Westerink, B. A. Ebersole, J. M. Smith, J. H. Atkinson, R. Jensen, D. T. Resio, R. A. Luettich, C. Dawson, V. J. Cardone, A. T. Cox, M. D. Powell, H. J. Westerink, and H. J. Roberts. A High-Resolution Coupled Riverine Flow, Tide, Wind, Wind Wave, and Storm Surge Model for Southern Louisiana and Mississippi. Part I: Model Development and Validation. *Monthly Weather Review 138*, 2 (02 2010), 345–377. https://doi.org/10.1175/2009MWR2906.1.

[31] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[32] J. Chamberlin, M. Zalewski, S. McMillan, and A. Lumsdaine. Pygb: Graphblas DSL in python with dynamic compilation into efficient C++. In *2018 IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPS Workshops 2018, Vancouver, BC, Canada, May 21-25, 2018* (2018), IEEE Computer Society, pp. 310–319. https://doi.org/10.1109/IPDPSW.2018.00059.

[33] A. Chatzimparmpas, R. M. Martins, I. Jusufi, and A. Kerren. A survey of surveys on the use of visualization for interpreting machine learning models. *Information Visualization* (2020), 1473871620904671.

[34] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz. Revisiting distributed synchronous sgd. *arXiv preprint arXiv:1604.00981* (2016).

[35] G. K. Choudhary. *Coupled atmospheric, hydrodynamic, and hydrologic models for simulation of complex phenomena*. Doctoral dissertation, The University of Texas at Austin, 2019. http://dx.doi.org/10.26153/tsw/7729.

[36] G. K. Choudhary and C. Dawson. pyADCIRC: A Python interface for accessing functions and variables of ADCIRC in Python. Virtual ADCIRC Users Group Meeting, 30-31 Mar 2020.

[37] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR* (2020).

[38] S. Cleveland, A. Jamthe, S. Padhy, J. Powell, J. Stubbs, M. Daniels, S. Pierce, and G. Jacobs. Tapischords integration: Time-series data support in science gateway infrastructure. In *Science Gateways 2019* (10 2019). http://dx.doi.org/10.17605/OSF.IO/NR3X6.

[39] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv. 41*, 1 (Jan. 2009). https://doi.org/10.1145/1456650.1456652.

[40] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116* (2019).

[41] J. J. Dai, Y. Wang, X. Qiu, D. Ding, Y. Zhang, Y. Wang, X. Jia, C. L. Zhang, Y. Wan, Z. Li, et al. Bigdl: A distributed deep learning framework for big data. In *Proceedings of the ACM Symposium on Cloud Computing* (2019), pp. 50–60.

[42] T. A. Davis, S. Rajamanickam, and W. M. Sid-Lakhdar. A survey of direct methods for sparse linear systems. *Acta Numerica 25* (2016), 383–566.

[43] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, et al. Large scale distributed deep networks. *Advances in neural information processing systems 25* (2012), 1223–1231.

[44] C. Dekate, M. Anderson, M. Brodowicz, H. Kaiser, B. Adelstein-Lelbach, and T. L. Sterling. Improving the scalability of parallel N-body applications with an event driven constraint based execution model. *The International Journal of High Performance Computing Applications abs/1109.5190* (2012). http://arxiv.org/abs/1109.5190.

[45] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact newton methods. *SIAM Journal on Numerical Analysis 19*, 2 (1982), 400–408. http://dx.doi.org/10.1137/0719025.

[46] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics, Jan. 1996. http://epubs.siam.org/doi/book/10.1137/1.9781611971200.

[47] K. D. Devine, E. G. Boman, R. T. Heaphy, R. H. Bisseling, and U. V. Catalyurek. Parallel hypergraph partitioning for scientific computing. In *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium* (2006), IEEE, pp. 10–pp.

[48] S. Devkota. CFGExplorer. http://github.com/hdc-arizona/cfgexplorer, 2018.

[49] S. Devkota and A. R. Ahmed. Stress-Plus-X. https://github.com/devkotasabin/SPX-graph-layout, 2019.

[50] S. Devkota, A. R. Ahmed, F. De Luca, K. Isaacs, and S. Kobourov. Stress-Plus-X (SPX) Graph Layout. In *Proceedings of the 27th Symposium on Graph Drawing and Network Visualization* (Sept. 2019).

[51] S. Devkota and K. E. Isaacs. CFGExplorer: designing a visual control flow analytics system around basic program analysis operations. *Computer Graphics Forum (Proceedings of EuroVis 2018) 37*, 3 (2018). http://dx.doi.org/10.1111/cgf.13433.

[52] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT* (2019).

[53] P. Diehl. Google Summer of Code Statistics, 2017. https://stellar-group.org/2017/11/google-summer-of-code-statistic/.

[54] J. C. Dietrich, S. Bunya, J. J. Westerink, B. A. Ebersole, J. M. Smith, J. H. Atkinson, R. Jensen, D. T. Resio, R. A. Luettich, C. Dawson, V. J. Cardone, A. T. Cox, M. D. Powell, H. J. Westerink, and H. J. Roberts. A High-Resolution Coupled Riverine Flow, Tide, Wind, Wind Wave, and Storm Surge Model for Southern Louisiana and Mississippi. Part II: Synoptic Description and Analysis of Hurricanes Katrina and Rita. *Monthly Weather Review 138*, 2 (02 2010), 378–404. https://doi.org/10.1175/2009MWR2907.1.

[55] J. C. Dietrich, J. J. Westerink, A. B. Kennedy, J. M. Smith, R. E. Jensen, M. Zijlema, L. H. Holthuijsen, C. Dawson, J. Luettich, R. A., M. D. Powell, V. J. Cardone, A. T. Cox, G. W. Stone, H. Pourtaheri, M. E. Hope, S. Tanaka, L. G. Westerink, H. J. Westerink, and Z. Cobell. Hurricane Gustav (2008) Waves and Storm Surge: Hindcast, Synoptic Analysis, and Validation in Southern Louisiana. *Monthly Weather Review 139*, 8 (08 2011), 2488–2522. https://doi.org/10.1175/2011MWR3611.1.

[56] R. Dooley, S. R. Brandt, and J. Fonner. The agave platform: An open, science-as-a-service platform for digital science. In *Proceedings of the Practice and Experience on Advanced Research Computing* (New York, NY, USA, 2018), PEARC '18, Association for Computing Machinery. https://doi.org/10.1145/3219104.3219129.

[57] Y. Du, L. Du, X. Gu, J. Du, X. S. Wang, B. Hu, M. Jiang, X. Chen, S. S. Iyer, and M. F. Chang. An analog neural network computing engine using cmos-compatible charge-trap-transistor (ctt). *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 38*, 10 (2019), 1811–1819. http://dx.doi.org/10.1109/TCAD.2018.2859237.

[58] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research 12*, 7 (2011).

[59] J. Dufek. The fluid mechanics of pyroclastic density currents. *Annual Review of Fluid Mechanics 48* (2016), 459–485.

[60] K. M. Duffy, T. Vandal, S. Li, S. Ganguly, R. Nemani, and A. R. Ganguly. Deepemsat: Deep emulation for satellite data mining. *Frontiers in Big Data 2* (2019), 42.

[61] O. Economics. The economic impacts of air travel restrictions due to volcanic ash, report for airbus, 2010.

[62] N. Elmqvist, T. Do, H. Goodell, N. Henry, and J. Fekete. Zame: Interactive large-scale graph visualization. In *2008 IEEE Pacific Visualization Symposium* (2008), pp. 215–222. http://dx.doi.org/10.1109/PACIFICVIS.2008.4475479.

[63] N. Elmqvist, N. Henry, Y. Riche, and J.-D. Fekete. Melange: Space folding for multi-focus interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), CHI '08, Association for Computing Machinery, p. 1333–1342. https://doi.org/10.1145/1357054.1357263.

[64] D. Eschweiler, M. Wagner, M. Geimer, A. Knüpfer, W. E. Nagel, and F. Wolf. Open trace format 2: The next generation of scalable trace formats and support libraries. In *Advances in Parallel Computing*, vol. 22. IOS Press, Amsterdam, NL, 2012, pp. 481–490.

[65] J. Estep and J. Dufek. Discrete element simulations of bed force anomalies due to force chains in dense granular flows. *Journal of volcanology and geothermal research 254* (2013), 108–117.

[66] K. Fan, Z. Wang, J. Beck, J. Kwok, and K. A. Heller. Fast second order stochastic backpropagation for variational inference. In *Advances in Neural Information Processing Systems* (2015), pp. 1387–1395.

[67] R. Faust, K. Isaacs, W. Z. Bernstein, M. Sharp, and C. Scheidegger. Anteater: Interactive visualization for program understanding. *CoRR abs/1907.02872* (2019). http://arxiv.org/abs/1907.02872.

[68] J. S. Firoz, M. Zalewski, and A. Lumsdaine. A synchronization-avoiding distance-1 grundy coloring algorithm for power-law graphs. In *28th International Conference on Parallel Architectures and Compilation Techniques, PACT 2019, Seattle, WA, USA, September 23-26, 2019* (2019), IEEE, pp. 421–432. https://doi.org/10.1109/PACT.2019.00040.

[69] A. Frieze, R. Kannan, , and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM) 51*, 6 (2004), 1025–1041.

[70] S. Goff, M. Vaughn, S. McKay, E. Lyons, A. Stapleton, D. Gessler, N. Matasci, L. Wang, M. Hanlon, A. Lenards, A. Muir, N. Merchant, S. Lowry, S. Mock, M. Helmke, A. Kubach, M. Narro, N. Hopkins, D. Micklos, U. Hilgert, M. Gonzales, C. Jordan, E. Skidmore, R. Dooley, J. Cazes, R. McLay, Z. Lu, S. Pasternak, L. Koesterke, W. Piel, R. Grene, C. Noutsos, K. Gendler, X. Feng, C. Tang, M. Lent, S.-j. Kim, K. Kvilekval, B. Manjunath, V. Tannen, A. Stamatakis, M. Sanderson, S. Welch, K. Cranston, P. Soltis, D. Soltis, B. O'Meara, C. Ane, T. Brutnell, D. Kleibenstein, J. White, J. Leebens-Mack, M. Donoghue, E. Spalding, T. Vision, C. Myers, D. Lowenthal, B. Enquist, B. Boyle, A. Akoglu, G. Andrews, S. Ram, D. Ware, L. Stein, and D. Stanzione. The iPlant Collaborative: Cyberinfrastructure for Plant Biology. *Frontiers in Plant Science 2* (2011), 34. https://www.frontiersin.org/article/10.3389/fpls.2011.00034.

[71] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677* (2017).

[72] Grafana Labs. Grafana: The open observability platform. https://grafana.com, 2020. Accessed: 2020-11-22.

[73] L. Graham, T. Butler, S. Walsh, C. Dawson, and J. nnes J. Westerink. A measure-theoretic algorithm for estimating bottom friction in a coasta l inlet: Case study of Bay St. Louis during Hurricane Gustav (2008 ). *Monthly Weather Review 145* (2017), 929–954.

[74] F. Haddadpour, M. M. Kamani, M. Mahdavi, and V. Cadambe. Trading redundancy for communication: Speeding up distributed sgd for non-convex optimization. In *International Conference on Machine Learning* (2019), pp. 2545–2554.

[75] W. W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAMOPT 16* (2005), 170–192.

[76] W. W. Hager and H. Zhang. Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent. *TOMS 32* (2006), 113–137.

[77] W. W. Hager and H. Zhang. The limited memory conjugate gradient method. *SIAMOPT 23* (2013), 2150–2168.

[78] M. R. Hanlon, M. Vaughn, S. Mock, R. Dooley, W. Moreira, J. Stubbs, C. Town, J. Miller, V. Krishnakumar, E. Ferlanti, and E. Pence. Araport: an application platform for data discovery: Araport: an application platform for data discovery. *Concurrency and Computation: Practice and Experience 27*, 16 (Nov. 2015), 4412–4422. http://doi.wiley.com/10.1002/cpe.3542.

[79] J. Hartline and T. Roughgarden. Optimal mechanism design and money burning. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)* (2008), pp. 75–84.

[80] B. Hasheminezhad, S. Shirzad, N. Wu, P. Diehl, H. Schulz, and H. Kaiser. Towards a scalable and distributed infrastructure for deep learning applications. In *Proceedings of the Deep Learning on Supercomputers Workshop* (Nov. 2020).

[81] B. Hasheminezhad, S. Shirzad, N. Wu, P. Diehl, H. Schulz, and H. Kaiser. Towards a scalable and distributed infrastructure for deep learning applications. *arXiv preprint arXiv:2010.03012* (2020).

[82] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.

[83] K. Hegde, J. Yu, R. Agrawal, M. Yan, M. Pellauer, and C. W. Fletcher. UCNN: Exploiting computational reuse in deep neural networks via weight repetition. In *Proceedings of the 45th Annual International Symposium on Computer Architecture* (Piscataway, NJ, USA, 2018), ISCA '18, IEEE Press, pp. 674–687. https://doi.org/10.1109/ISCA.2018.00062.

[84] T. Heller, H. Kaiser, P. Diehl, D. Fey, and M. A. Schweitzer. Closing the Performance Gap with Modern C++. In *High Performance Computing* (2016), M. Taufer, B. Mohr, and J. M. Kunkel, Eds., vol. 9945 of *Lecture Notes in Computer Science*, Springer International Publishing, pp. 18–31.

[85] T. Heller, H. Kaiser, and K. Iglberger. Application of the ParalleX Execution Model to Stencil-based Problems. In *Proceedings of the International Supercomputing Conference ISC'12, Hamburg, Germany* (2012). http://stellar.cct.lsu.edu/pubs/isc2012.pdf.

[86] T. Heller, H. Kaiser, A. Schäfer, and D. Fey. Using HPX and LibGeoDecomp for Scaling HPC Applications on Heterogeneous Supercomputers. In *Proceedings of the Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems* (New York, NY, USA, 2013), ScalA '13, ACM, pp. 1:1–1:8. http://doi.acm.org/10.1145/2530268.2530269.

[87] B. Hendrickson and T. G. Kolda. Graph partitioning models for parallel computing. *Parallel computing 26*, 12 (2000), 1519–1534.

[88] J. F. Henriques, S. Ehrhardt, S. Albanie, and A. Vedaldi. Small steps and giant leaps: Minimal newton solvers for deep learning, 2018.

[89] Y. Hu, D. M. Koppelman, and S. R. Brandt. Thoroughly exploring GPU buffering options for stencil code by using an efficiency measure and a performance model. *IEEE Transactions on Multi-Scale Computing Systems 4*, 3 (July 2018), 477–490. http://dx.doi.org/10.1109/TMSCS.2017.2705139.

[90] C.-C. Huang, Q. Chen, Z. Wang, R. Power, J. Ortiz, J. Li, and Z. Xiao. Spartan: A distributed array framework with smart tiling. In *USENIX Annual Technical Conference* (2015), pp. 1–15.

[91] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in neural information processing systems* (2019), pp. 103–112.

[92] Y. Huang, Z. Yang, J. Zhu, and T. T. Ye. Analog circuit implementation of neurons with multiply-accumulate and ReLU functions. In *Proceedings of the 2020 on Great Lakes Symposium on VLSI* (New York, NY, USA, 2020), GLSVLSI '20, Association for Computing Machinery, p. 493–498. https://doi.org/10.1145/3386263.3406941.

[93] K. A. Huck, A. D. Malony, S. Shende, and A. Morris. Taug: Runtime global performance data access using mpi. In *European parallel virtual machine/message passing interface users' group meeting* (2006), Springer, pp. 313–321.

[94] K. A. Huck, A. Porterfield, N. Chaimov, H. Kaiser, A. D. Malony, T. Sterling, and R. Fowler. An autonomic performance environment for exascale. *Supercomputing frontiers and innovations 2*, 3 (2015), 49–66.

[95] Huck, Kevin. Apex: An autonomic performance environment for exascale. `https://github.com/khuck/xpress-apex`, 2020. Accessed: 2020-11-22.

[96] J. Irish, D. Resio, and M. Cialone. A surge response function approach to coastal hazard assessment. Part 2: Quantification of spatial attributes and response functions. *Natural Hazards 51* (2009), 183–205.

[97] A. Jagtap and G. Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *arXiv: Computational Physics* (2019).

[98] H. K. Jang, C. E. Ozdemir, J.-H. Liang, and M. Tyagi. Oscillatory flow around a vertical wall-mounted cylinder: Flow pattern details. *Physics of Fluids* (2020 Under Review).

[99] X. Jia, S. Song, W. He, Y. Wang, H. Rong, F. Zhou, L. Xie, Z. Guo, Y. Yang, L. Yu, et al. Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes. *arXiv preprint arXiv:1807.11205* (2018).

[100] Z. Jia, M. Zaharia, and A. Aiken. Beyond data and model parallelism for deep neural networks. *arXiv preprint arXiv:1807.05358* (2018).

[101] P. Jiang and G. Agrawal. A linear speedup analysis of distributed deep learning with sparse and quantized communication. In *Advances in Neural Information Processing Systems* (2018), pp. 2525–2536.

[102] P. H. Jin, Q. Yuan, F. Iandola, and K. Keutzer. How to scale distributed deep learning? *arXiv preprint arXiv:1611.04581* (2016).

[103] A. E. Johnson, T. J. Pollard, L. Shen, L. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific data* (2016).

[104] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651* (2016).

[105] N. P. Jouppi, D. H. Yoon, G. Kurian, S. Li, N. Patil, J. Laudon, C. Young, and D. Patterson. A domain-specific supercomputer for training deep neural networks. *Commun. ACM 63*, 7 (June 2020), 67–78. `https://doi.org/10.1145/3360307`.

[106] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P.-l. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, and D. H. Yoon. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture* (New York, NY, USA, 2017), ISCA '17, Association for Computing Machinery, p. 1–12. `https://doi.org/10.1145/3079856.3080246`.

[107] M. Kahng, N. Thorat, D. H. Chau, F. B. Viégas, and M. Wattenberg. Gan lab: Understanding complex deep generative models using interactive visual experimentation. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 310–320. `http://dx.doi.org/10.1109/TVCG.2018.2864500`.

[108] H. Kaiser, B. Adelstein-Lelbach, T. Heller, and A. B. et.al. HPX V1.5.1: The C++ Standards Library for Parallelism and Concurrency, 2020. http://dx.doi.org/10.5281/zenodo.598202, `http://dx.doi.org/10.5281/zenodo.598202`.

[109] H. Kaiser, P. Amini, S. Brandt, and B. Hasheminezhad. Phylanx: An Asynchronous Distributed C++ Array Processing Toolkit, 2019. https://github.com/STEllAR-GROUP/phylanx, `https://github.com/STEllAR-GROUP/phylanx`.

[110] H. Kaiser, M. Brodowicz, and T. Sterling. ParalleX: An Advanced Parallel Execution Model for Scaling-Impaired Applications. In *Parallel Processing Workshops* (Los Alamitos, CA, USA, 2009), IEEE Computer Society, pp. 394–401. http://doi.ieeecomputersociety.org/10.1109/ICPPW.2009.14.

[111] H. Kaiser, T. Heller, B. Adelstein-Lelbach, A. Serio, and D. Fey. HPX: A Task Based Programming Model in a Global Address Space. In *Proceedings of the 8th International Conference on Partitioned Global Address Space Programming Models* (New York, NY, USA, 2014), PGAS '14, ACM, pp. 6:1–6:11. http://doi.acm.org/10.1145/2676870.2676883.

[112] H. Kaiser, T. Heller, D. Bourgeois, and D. Fey. Higher-level parallelization for local and distributed asynchronous task-based programming. In *Proceedings of the First International Workshop on Extreme Scale Programming Models and Middleware* (New York, NY, USA, 2015), ESPM '15, ACM, pp. 29–37. http://doi.acm.org/10.1145/2832241.2832244.

[113] T. A. Kanewala, M. Zalewski, and A. Lumsdaine. Parallel asynchronous distributed-memory maximal independent set algorithm with work ordering. In *24th IEEE International Conference on High Performance Computing, HiPC 2017, Jaipur, India, December 18-21, 2017* (2017), IEEE Computer Society, pp. 52–61. https://doi.org/10.1109/HiPC.2017.00016.

[114] T. A. Kanewala, M. Zalewski, and A. Lumsdaine. Distributed, shared-memory parallel triangle counting. In *Proceedings of the Platform for Advanced Scientific Computing Conference, PASC 2018, Basel, Switzerland, July 02-04, 2018* (2018), ACM, pp. 5:1–5:12. https://doi.org/10.1145/3218176.3218229.

[115] I.-F. Kao, Y. Zhou, L.-C. Chang, and F.-J. Chang. Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting. *Journal of Hydrology* (2020), 124631.

[116] R. Karimi, D. M. Koppelman, and C. J. Michael. GPU road network graph contraction and SSSP query. In *Proceedings of the ACM International Conference on Supercomputing* (New York, NY, USA, 2019), ICS '19, ACM, pp. 250–260. http://doi.acm.org/10.1145/3330345.3330368.

[117] R. Karimi, D. M. Koppelman, and C. J. Michael. Fast GPU graph contraction by combining efficient shallow searches and post-culling. In *Proceedings of the 2020 IEEE High Performance Extreme Computing Conference* (2020), IEEE.

[118] H. B. Keller. Global Homotopies and Newton Methods. In *Recent Advances in Numerical Analysis*. Elsevier, 1978, pp. 73–94. https://linkinghub.elsevier.com/retrieve/pii/B9780122083600500097.

[119] B. Khan, C. Robbins, and A. Okrent. Science and Engineering Indicators 2020: The State of U.S. Science and Engineering. Biennial report NSB-2020-1, National Science Board, Alexandria, VA, Jan. 2020. https://ncses.nsf.gov/pubs/nsb20201/.

[120] K. Kim, I. Seo, W.-S. Han, J.-H. Lee, S. Hong, H. Chafi, H. Shin, and G. Jeong. Turboflux: A fast continuous subgraph matching system for streaming graph data. In *Proceedings of the 2018 International Conference on Management of Data* (New York, NY, USA, 2018), SIGMOD '18, Association for Computing Machinery, p. 411–426. https://doi.org/10.1145/3183713.3196917.

[121] M. Kim, J. Kress, J. Choi, N. Podhorszki, S. Klasky, M. Wolf, K. Mehta, K. Huck, B. Geveci, S. Phillip, et al. In situ analysis and visualization of fusion simulations: Lessons learned. In *International Conference on High Performance Computing* (2018), Springer, pp. 230–242.

[122] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[123] O. Kolditz, U. Görke, H. Shao, and W. Wang. *Thermo-Hydro-Mechanical-Chemical Processes in Porous Media: Benchmarks and Examples*. Springer, 01 2012.

[124] F. Kratzert, D. Klotz, G. Shalev, G. Klambauer, S. Hochreiter, and G. Nearing. Benchmarking a catchment-aware long short-term memory network (LSTM) for large-scale hydrological modeling. *arXiv preprint arXiv:1907.08456* (2019).

[125] V. D. Lai, T. N. Nguyen, and T. H. Nguyen. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2020).

[126] A. G. Landge, J. A. Levine, A. Bhatele, K. E. Isaacs, T. Gamblin, M. Schulz, S. H. Langer, P.-T. Bremer, and V. Pascucci. Visualizing network traffic to understand the performance of massively parallel simulations. *IEEE Transactions on Visualization and Computing Graphics, Proceedings of InfoVis 18*, 12 (2012), 2467–2476. http://dx.doi.org/10.1109/TVCG.2012.286.

[127] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng. On optimization methods for deep learning. In *ICML* (2011).

[128] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* (2020).

[129] S. Lee and D. You. Data-driven prediction of unsteady flow over a circular cylinder using deep learning. *Journal of Fluid Mechanics 879* (2019), 217–254.

[130] F. Lekschas, M. Behrisch, B. Bach, P. Kerpedjiev, N. Gehlenborg, and H. Pfister. Pattern-driven navigation in 2d multiscale visualizations with scalable insets. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 611–621. http://dx.doi.org/10.1109/TVCG.2019.2934555.

[131] F. Li and M. Thottan. End-to-end service quality measurement using source-routed probes. In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)* (2006), pp. 1–12.

[132] J. K. Li, M. Mubarak, R. B. Ross, C. D. Carothers, and K.-L. Ma. Visual analytics techniques for exploring the design space of large-scale high-radix networks. In *2017 IEEE International Conference on Cluster Computing (CLUSTER)* (sep, 2017), CLUSTER, pp. 193–203. http://dx.doi.org/10.1109/CLUSTER.2017.26.

[133] M. Li, Y. Liu, X. Liu, Q. Sun, X. You, H. Yang, and Z. Luan. The deep learning compiler: A comprehensive survey. *arXiv*, 2002.03794v4 (2020), 1–34.

[134] M. Li, Y. Liu, X. Liu, Q. Sun, X. You, H. Yang, Z. Luan, L. Gan, G. Yang, and D. Qian. The deep learning compiler: A comprehensive survey. *IEEE Transactions on Parallel and Distributed Systems 32*, 3 (2021), 708–727. http://dx.doi.org/10.1109/TPDS.2020.3030548.

[135] N. Li, F. Li, and J. Offutt. Better algorithms to minimize the cost of test paths. In *Proceedings of 2012 IEEE 5th International Conference on Software Testing, Verification and Validation (ICST)* (2012), pp. 280–289.

[136] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, et al. Pytorch distributed: Experiences on accelerating data parallel training. *arXiv preprint arXiv:2006.15704* (2020).

[137] W. Li, A. Kiaghadi, and C. Dawson. Exploring the best sequence LSTM modeling architecture for flood prediction. *Neural Computing and Applications* (2020), 1–10.

[138] W. Li, A. Kiaghadi, and C. N. Dawson. High temporal resolution rainfall runoff modelling using long-short-term-memory (LSTM) networks. *arXiv preprint arXiv:2002.02568* (2020).

[139] T. Lin, S. U. Stich, K. K. Patel, and M. Jaggi. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217* (2018).

[140] Y. Lin, H. Ji, F. Huang, and L. Wu. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020).

[141] Q. Liu, M. J. Kusner, and P. Blunsom. A survey on contextual embeddings. *ArXiv abs/2003.07278* (2020).

[142] S. Liu, X. Wang, M. Liu, and J. Zhu. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics 1*, 1 (2017), 48 – 56. http://www.sciencedirect.com/science/article/pii/S2468502X17300086.

[143] X. Liu, J. S. Firoz, M. Zalewski, M. Halappanavar, K. J. Barker, A. Lumsdaine, and A. H. Gebremedhin. Distributed direction-optimizing label propagation for community detection. In *2019 IEEE High Performance Extreme Computing Conference, HPEC 2019, Waltham, MA, USA, September 24-26, 2019* (2019), IEEE, pp. 1–6. https://doi.org/10.1109/HPEC.2019.8916215.

[144] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv abs/1907.11692* (2019).

[145] A. Lodi, S. Martello, and D. Vigo. Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics 123*, 1-3 (2002), 379–396.

[146] D. Lopatto. Undergraduate Research Experiences Support Science Career Decisions and Active Learning. *CBE—Life Sciences Education 6*, 4 (Dec. 2007), 297–306. https://www.lifescied.org/doi/10.1187/cbe.07-06-0039.

[147] Q. Lu, N. de Silva, D. Dou, T. H. Nguyen, P. Sen, B. Reinwald, and Y. Li. Exploiting node content for multiview graph convolutional network and adversarial regularization. In *Proceedings of the International Conference on Computational Linguistics (COLING)* (2020).

[148] Q. Lu, N. de Silva, S. Kafle, J. Cao, D. Dou, T. H. Nguyen, P. Sen, B. Hailpern, B. Reinwald, and Y. Li. Learning electronic health records through hyperbolic embedding of medical ontologies. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, BCB 2019* (2019).

[149] A. Lumsdaine, L. D'Alessandro, K. Dewesee, J. Firoz, and S. Mcmillian. Triangle counting with cyclic distributions. In *2020 IEEE High Performance Extreme Computing Conference (HPEC)* (09 2020).

[150] A. R. Mamidala, G. Kollias, C. Ward, and F. Artico. Mxnet-mpi: Embedding mpi parallelism in parameter server task model for scaling deep learning. *arXiv preprint arXiv:1801.03855* (2018).

[151] H. Man Duc Trong, D. Trong Le, A. Pouran Ben Veyseh, T. Nguyen, and T. H. Nguyen. Introducing a new dataset for event detection in cybersecurity texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2020).

[152] J. Martens. Deep learning via hessian-free optimization. In *ICML* (2010), vol. 27, pp. 735–742.

[153] J. Martens, J. Ba, and M. Johnson. Kronecker-factored curvature approximations for recurrent neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings* (2018), OpenReview.net. https://openreview.net/forum?id=HyMTkQZAb.

[154] J. Martens and R. B. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015* (2015), F. R. Bach and D. M. Blei, Eds., vol. 37 of *JMLR Workshop and Conference Proceedings*, JMLR.org, pp. 2408–2417. http://proceedings.mlr.press/v37/martens15.html.

[155] J. Martens and I. Sutskever. Training deep and recurrent networks with hessian-free optimization. In *Neural Networks: Tricks of the Trade - Second Edition*, G. Montavon, G. B. Orr, and K. Müller, Eds., vol. 7700 of *Lecture Notes in Computer Science*. Springer, 2012, pp. 479–535. https://doi.org/10.1007/978-3-642-35289-8_27.

[156] M. L. Massie, B. N. Chun, and D. E. Culler. The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing 30*, 7 (2004), 817 – 840. http://www.sciencedirect.com/science/article/pii/S0167819104000535.

[157] S. Mattis, K. Steffen, T. Butler, C. Dawson, and D. Estep. Learning quantities of interest from dynamic systems for observation-consistent inversion. *Journal of Computational Physics* (2020).

[158] P. Mattson, C. Cheng, C. Coleman, G. Diamos, P. Micikevicius, D. Patterson, H. Tang, G.-Y. Wei, P. Bailis, V. Bittorf, D. Brooks, D. Chen, D. Dutta, U. Gupta, K. Hazelwood, A. Hock, X. Huang, A. Ike, B. Jia, D. Kang, D. Kanter, N. Kumar, J. Liao, G. Ma, D. Narayanan, T. Oguntebi, G. Pekhimenko, L. Pentecost, V. J. Reddi, T. Robie, T. S. John, T. Tabaru, C.-J. Wu, L. Xu, M. Yamazaki, C. Young, and M. Zaharia. Mlperf training benchmark, 2020.

[159] M. Meyer and J. Dykes. Criteria for rigor in visualization design study. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 87–97. http://dx.doi.org/10.1109/TVCG.2019.2934539.

[160] C. Michoski, A. Alexanderian, C. Paillet, E. Kubatko, and C. . Dawson. Stability of nonlinear convection-diffusion-reaction systems in disconti nuous Galerkin methods. *Journal of Scientific Computing 70* (2017), 516–550.

[161] C. Michoski, C. Dawson, E. Kubatko, D. Wirasaet, S. Brus, and J. W. esterink. A comparison of artificial viscosity, limiters, and filter, for high order discont inuous Galerkin solution in nonlinear settings. *Journal of Scientific Computing* (2015). http://dx.doi.org/10.1007/s10915.015.0027.2.

[162] H. Mikami, H. Suganuma, Y. Tanaka, Y. Kageyama, et al. Massively distributed sgd: Imagenet/resnet-50 training in a flash. *arXiv preprint arXiv:1811.05233* (2018).

[163] A. Moitra and R. O'Donnell. Pareto optimal solutions for smoothed analysts. *SIAM Journal on Computing 41*, 5 (2012), 1266–1284.

[164] C. Muelder, F. Gygi, and K.-L. Ma. Visual analysis of inter-process communication for large-scale parallel computing. *IEEE Transactions on Visualization and Computer Graphics, Proceedings of InfoVis 15*, 6 (2009), 1129–1136. http://dx.doi.org/10.1109/TVCG.2009.196.

[165] National Defense Magazine. How to build a well-rounded ai workforce? https://www.nationaldefensemagazine.org/articles/2020/10/21/how-to-build-a-well-rounded-ai-workforce, Last Accessed November 2020.

[166] L. Nazareth. A Relationship between the BFGS and Conjugate Gradient Algorithms and Its Implications for New Algorithms. *SIAM Journal on Numerical Analysis 16*, 5 (Oct. 1979), 794–800. http://epubs.siam.org/doi/10.1137/0716059.

[167] A. Neri, A. Di Muro, and M. Rosi. Mass partition during collapsing and transitional columns by using numerical simulations. *Journal of volcanology and geothermal research 115*, 1-2 (2002), 1–18.

[168] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In *Doklady an ussr* (1983), vol. 269, pp. 543–547.

[169] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation 35*, 151 (Sept. 1980), 773–773. http://www.ams.org/jourcgi/jour-getitem?pii=S0025-5718-1980-0572855-7.

[170] NVIDIA Corporation. NVIDIA tesla v100 GPU architecture. Tech. rep., NVIDIA Corporation, August 2017.

[171] NVIDIA Corporation. NVIDIA A100 tensor core GPU architecture. Tech. rep., NVIDIA Corporation, 2020.

[172] E. D. Obasaju, P. W. Bearman, and J. M. R. Graham. A study of forces, circulation and vortex patterns around a circular cylinder in oscillating flow. *Journal of Fluid Mechanics 196* (nov 1988), 467–494.

[173] D. E. Ogden, G. A. Glatzmaier, and K. H. Wohletz. Effects of vent overpressure on buoyant eruption columns: implications for plume stability. *Earth and Planetary Science Letters 268*, 3-4 (2008), 283–292.

[174] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Elsevier, 1970. https://linkinghub.elsevier.com/retrieve/pii/C20130112639.

[175] K. Osawa, Y. Tsuji, Y. Ueno, A. Naruse, R. Yokota, and S. Matsuoka. Large-scale distributed second-order optimization using kronecker-factored approximate curvature for deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 12359–12367.

[176] J. T. Palmer, S. M. Gallo, T. R. Furlani, M. D. Jones, R. L. DeLeon, J. P. White, N. Simakov, A. K. Patra, J. Sperhac, T. Yearke, et al. Open xdmod: A tool for the comprehensive management of high-performance computing resources. *Computing in Science & Engineering 17*, 4 (2015), 52–62.

[177] A. Parashar, P. Raina, Y. S. Shao, Y. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S. W. Keckler, and J. Emer. Timeloop: A systematic approach to DNN accelerator evaluation. In *2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)* (2019), pp. 304–315. https://ieeexplore.ieee.org/abstract/document/8695666.

[178] A. Parashar, M. Rhu, A. Mukkara, A. Puglielli, R. Venkatesan, B. Khailany, J. Emer, S. W. Keckler, and W. J. Dally. SCNN: An accelerator for compressed-sparse convolutional neural networks. In *Proceedings of the 44th Annual International Symposium on Computer Architecture* (New York, NY, USA, 2017), ISCA '17, ACM, pp. 27–40. http://doi.acm.org/10.1145/3079856.3080254.

[179] B. A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Comput. 6*, 1 (Jan. 1994), 147–160. https://doi.org/10.1162/neco.1994.6.1.147.

[180] W. Peng, W. Zhou, J. Zhang, and W. Yao. Accelerating physics-informed neural network training with prior dictionaries. *arXiv: Computer Science* (2020).

[181] C. Perin, P. Dragicevic, and J. Fekete. Revisiting bertin matrices: New interactions for crafting tabular visualizations. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2082–2091. http://dx.doi.org/10.1109/TVCG.2014.2346279.

[182] K. Phillippe. American Association of Community Colleges Fast Facts Sheet 2020, Jan. 2020. https://www.aacc.nche.edu/research-trends/fast-facts/.

[183] A. Pouran Ben Veyseh, F. Dernoncourt, D. Dou, and T. H. Nguyen. Exploiting the syntax-model consistency for neural relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020).

[184] A. Pouran Ben Veyseh, N. Nouri, F. Dernoncourt, Q. H. Tran, D. Dou, and T. H. Nguyen. Improving aspect-based sentiment analysis with gated graph convolutional networks and syntax-based regulation. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (2020).

[185] Seamless operability between C++11 and Python, 2020. https://github.com/pybind/pybind11.

[186] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, , and I. Sutskever. Language models are unsupervised multitask learners. In *OpenAI Technical Report* (2019).

[187] M. Raissi, Z. Wang, M. Triantafyllou, and G. Karniadakis. Deep learning of vortex-induced vibrations. *Journal of Fluid Mechanics 861* (2019), 119–137.

[188] I. Ranawaka, S. Marru, J. Graham, A. Bisht, J. Basney, T. Fleury, J. Gaynor, D. Wannipurage, M. Christie, A. Mahmoud, E. Afgan, and M. Pierce. Custos: Security middleware for science gateways. In *Practice and Experience in Advanced Research Computing* (New York, NY, USA, 2020), PEARC '20, Association for Computing Machinery, p. 278–284. https://doi.org/10.1145/3311790.3396635.

[189] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020), p. 3505–3506.

[190] E. M. Rathje, C. Dawson, J. E. Padgett, J.-P. Pinelli, D. Stanzione, A. Adair, P. Arduino, S. J. Brandenberg, T. Cockerill, C. Dey, M. Esteva, F. L. Haan, M. Hanlon, A. Kareem, L. Lowes, S. Mock, and G. Mosqueda. DesignSafe: New Cyberinfrastructure for Natural Hazards Engineering. *Natural Hazards Review 18*, 3 (Aug. 2017), 06017001. http://ascelibrary.org/doi/10.1061/%28ASCE%29NH.1527-6996.0000246.

[191] S. J. Reddi, S. Sra, B. Póczos, and A. J. Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In *Proceedings of the 30th International Conference on Neural Information Processing* (2016), NIPS, Ed., pp. 1153–1161.

[192] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, et al. Deep learning and process understanding for data-driven earth system science. *Nature 566*, 7743 (2019), 195–204.

[193] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 61–70. http://dx.doi.org/10.1109/TVCG.2016.2598828.

[194] D. Resio, J. Irish, and M. Cialone. A surge response function approach to coastal hazard assessment. Part 1: Basic concepts. *Natural Hazards 51* (2009), 163–182.

[195] K. Rocki, D. Van Essendelft, I. Sharapov, R. Schreiber, M. Morrison, V. Kibardin, A. Portnoy, J. F. Dietiker, M. Syamlal, and M. James. Fast stencil-code computation on a wafer-scale processor. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (2020), SC '20, IEEE Press.

[196] J. Rogers, A. H. Patton, L. Harmon, A. Lex, and M. Meyer. Insights from experiments with rigor in an evobio design study. *IEEE Transactions on Visualization and Computer Graphics* (2021). http://dx.doi.org/10.1109/TVCG.2020.3030405.

[197] P. Rosen. A visual approach to investigating shared and global memory behavior of cuda kernels. *Computer Graphics Forum 32*, 3pt2 (2013), 161–170. https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12103.

[198] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747* (2016).

[199] Y. Saad. *Iterative Methods for Sparse Linear Systems*, 2nd ed. Society for Industrial and Applied Mathematics, USA, 2003.

[200] A. Samii. *A hybridized discontinuous Galerkin method for nonlinear dispersive w ater waves*. PhD thesis, The University of Texas at Austin, 2017.

[201] A. Samii and C. Dawson. An explicit hybridized discontinuous Galerkin method for Serre-Gre en-Naghdi wave model. *Computer Methods in Applied Mechanics and Engineering 330* (2018), 447–470.

[202] A. Samii, C. Michoski, and C. Dawson. A parallel and adaptive hybridized discontinuous Galerkin method for ani sotropic nonhomogeneous diffusion. *Computer Methods in Applied Mechanics and Engineering 304* (6 2016), 118–139. http://dx.doi.org/10.1016/j.cma.2016.02.009.

[203] A. Samii, N. Panda, C. Michoski, and C. Dawson. A hybridized discontinuous Galerkin method for the nonlinear Korteweg–d e Vries equation. *Journal of Scientific Computing 68*, 1 (2016), 191–212. http://dx.doi.org/10.1007/s10915-015-0133-1.

[204] E. Schnetter, M. Blazewicz, S. R. Brandt, D. M. Koppelman, and F. Löffler. Chemora: A PDE-solving framework for modern high-performance computing architectures. *Computing in Science Engineering 17*, 2 (Mar 2015), 53–64. http://dx.doi.org/10.1109/MCSE.2015.2.

[205] Synergistic Discovery and Design Environment (SD2E). https://sd2e.org/.

[206] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the statcks. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2431–2440.

[207] A. Sergeev and M. Del Balso. Horovod: fast and easy distributed deep learning in tensorflow. *arXiv preprint arXiv:1802.05799* (2018).

[208] E. Seymour, A.-B. Hunter, S. L. Laursen, and T. DeAntoni. Establishing the benefits of research experiences for undergraduates in the sciences: First findings from a three-year study. *Science Education 88*, 4 (July 2004), 493–534. http://doi.wiley.com/10.1002/sce.10131.

[209] J. Sgall. Online bin packing: Old algorithms and new results. In *Proceedings of the Conference on Computability in Europe (CiE)* (2014), pp. 362–372.

[210] C. J. Shallue, J. Lee, J. Antognini, J. Sohl-Dickstein, R. Frostig, and G. E. Dahl. Measuring the effects of data parallelism on neural network training. *arXiv preprint arXiv:1811.03600* (2018).

[211] N. Shazeer, Y. Cheng, N. Parmar, D. Tran, A. Vaswani, P. Koanantakool, P. Hawkins, H. Lee, M. Hong, C. Young, et al. Mesh-tensorflow: Deep learning for supercomputers. In *Advances in Neural Information Processing Systems* (2018), pp. 10414–10423.

[212] C. Sigovan, C. Muelder, K. Ma, J. Cope, K. Iskra, and R. Ross. A visual network analysis method for large-scale parallel i/o systems. In *2013 IEEE 27th International Symposium on Parallel and Distributed Processing* (Los Alamitos, CA, USA, 2013), pp. 308–319. http://dx.doi.org/10.1109/IPDPS.2013.96.

[213] S. A. Silling. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids 48* (2000), 175—-209.

[214] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489* (2017).

[215] M. J. Sottile and R. G. Minnich. Supermon: A high-speed cluster monitoring system. In *Proceedings. IEEE International Conference on Cluster Computing* (2002), IEEE, pp. 39–46.

[216] W. Spear, A. D. Malony, C. W. Lee, S. Biersdorff, and S. Shende. An approach to creating performance visualizations in a parallel profile analysis tool. In *Proceedings of the 2011 International Conference on Parallel Processing - Volume 2* (Berlin, Heidelberg, 2012), Lecture Notes in Computer Science, Springer-Verlag, pp. 156–165. http://dx.doi.org/10.1007/978-3-642-29740-3_19.

[217] D. A. Spielman and S.-H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM) 51*, 3 (2004), 385–463.

[218] S. U. Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767* (2018).

[219] A. Stohl, A. Prata, S. Eckhardt, L. Clarisse, A. Durant, S. Henne, N. Kristiansen, A. Minikin, U. Schumann, P. Seibert, et al. Determination of time-and height-resolved volcanic ash emissions for quantitative ash dispersion modeling: the 2010 eyjafjallajökull eruption. *Atmospheric Chemistry & Physics Discussions 11*, 2 (2011).

[220] J. Stubbs, R. Dooley, and M. Vaughn. Containers-as-a-service via the Actor Model. In *Proceedings of The 11th Gateway Computing Environments Conference* (San Diego, CA, Jan. 2017). https://figshare.com/articles/Containers-as-a-service_via_the_Actor_Model/4490747.

[221] L. Sun, H. Gao, S. Pan, and J.-X. Wang. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering 361* (2020), 112732. http://www.sciencedirect.com/science/article/pii/S004578251930622X.

[222] E. Suwartadi, S. Krogstad, and B. Foss. Adjoint-based surrogate optimization of oil reservoir water flooding. *Optimization and Engineering 16* (2015), 441–481.

[223] C. Tapus, I.-H. Chung, and J. Hollingworth. Active harmony: Towards automated performance tuning. In *SC '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing* (2002).

[224] J. Telling and J. Dufek. An experimental evaluation of ash aggregation in explosive volcanic eruptions. *Journal of volcanology and geothermal research 209* (2012), 1–8.

[225] Y. Teng, W. Gao, F. Chalus, A. E. Choromanska, D. Goldfarb, and A. Weller. Leader stochastic gradient descent for distributed training of deep learning models. In *Advances in Neural Information Processing Systems* (2019), pp. 9824–9834.

[226] The C++ Standards Committee. ISO International Standard ISO/IEC 14882:2011, Programming Language C++. Tech. rep., Geneva, Switzerland: International Organization for Standardization (ISO)., 2011. http://www.open-std.org/jtc1/sc22/wg21.

[227] The C++ Standards Committee. ISO International Standard ISO/IEC 14882:2014, Programming Language C++. Tech. rep., Geneva, Switzerland: International Organization for Standardization (ISO)., 2014. http://www.open-std.org/jtc1/sc22/wg21.

[228] The C++ Standards Committee. ISO International Standard ISO/IEC 14882:2017, Programming Language C++. Tech. rep., Geneva, Switzerland: International Organization for Standardization (ISO)., 2017. http://www.open-std.org/jtc1/sc22/wg21.

[229] The Jaeger Authors, The Linux Foundation. Jaeger: open source, end-to-end distributed tracing. `https://www.jaegertracing.io`, 2020. Accessed: 2020-11-20.

[230] The OpenTelemetry Authors. Opentelemetry. `https://opentelemetry.io`, 2020. Accessed: 2020-11-20.

[231] The OpenZipkin Authors. OpenZipkin: A distributed tracing system. `https://zipkin.io`, 2020. Accessed: 2020-11-20.

[232] Thomas Sterling. ParalleX Execution Model V3.1. unpublished specification., 2013.

[233] V. T'kindt and J.-C. Billaut. *Multicriteria scheduling: theory, models and algorithms*. Springer Science & Business Media, 2006.

[234] R. Tohid, B. Wagle, S. Shirzad, P. Diehl, A. Serio, A. Kheirkhahan, P. Amini, K. Williams, K. Isaacs, K. Huck, S. Brandt, and H. Kaiser. Asynchronous execution of python code on task-based runtime systems. In *2018 IEEE/ACM 4th International Workshop on Extreme Scale Programming Models and Middleware (ESPM2)* (2018), pp. 37–45.

[235] S. Tokui, R. Okuta, T. Akiba, Y. Niitani, T. Ogawa, S. Saito, S. Suzuki, K. Uenishi, B. Vogel, and H. Yamazaki Vincent. Chainer: A deep learning framework for accelerating the research cycle. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019), pp. 2002–2011.

[236] Y. Ueno, K. Osawa, Y. Tsuji, A. Naruse, and R. Yokota. Rich information is affordable: A systematic performance analysis of second-order optimization using k-fac. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020), pp. 2145–2153.

[237] U.S. Global Change Research Program and Subcommittee on Global Change Research. Our Changing Planet: The U.S. Global Change Research Program for Fiscal Year 2020. Tech. rep., U.S. Global Change Research Program, 2020. `https://www.globalchange.gov/browse/reports/our-changing-planet-FY-2020`.

[238] G. A. Valentine and K. H. Wohletz. Numerical models of plinian eruption columns and pyroclastic flows. *Journal of Geophysical Research: Solid Earth 94*, B2 (1989), 1867–1887.

[239] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems* (2017).

[240] A. P. B. Veyseh, F. Dernoncourt, M. Thai, D. Dou, and T. H. Nguyen. Multi-view consistency for relation extraction via mutual information and structure prediction. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)* (2020).

[241] A. P. B. Veyseh, M. T. Thai, T. H. Nguyen, and D. Dou. Rumor detection in social networks via deep contextual modeling. In *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2019), F. Spezzano, W. Chen, and X. Xiao, Eds.

[242] T. Vierjahn, M.-A. Hermanns, B. Mohr, M. S. Müller, T. W. Kuhlen, and B. Hentschel. Using directed variance to identify meaningful views in call-path performance profiles. In *Proceedings of the Third International Workshop on Visual Performance Analysis* (2016), VPA, pp. 9–16.

[243] B. Wagle, M. A. H. Monil, K. Huck, A. D. Malony, A. Serio, and H. Kaiser. Runtime adaptive task inlining on asynchronous multitasking runtime systems. In *Proceedings of the 48th International Conference on Parallel Processing* (New York, NY, USA, 2019), ICPP 2019, Association for Computing Machinery. https://doi.org/10.1145/3337821.3337915.

[244] B. Wagle, M. A. H. Monil, K. Huck, A. D. Malony, A. Serio, and H. Kaiser. Runtime adaptive task inlining on asynchronous multitasking runtime systems. In *Proceedings of the 48th International Conference on Parallel Processing* (2019), pp. 1–10.

[245] H. Wang, J. X. Zhang, and F. Li. Worst-case performance guarantees of scheduling algorithms maximizing weighted throughput in energy-harvesting networks. *Sustainable Computing: Informatics and Systems 4*, 3 (2014), 172–182.

[246] X. Wang and H. Zhang. Inexact stochastic proximal quasi-newton method for nonconvex optimization. *OMS 35* (2020), 808–835.

[247] J. Watt, R. Borhani, and A. K. Katsaggelos. *Machine Learning Refined: Foundations, Algorithms, and Applications*. Cambridge University Press, 2020.

[248] P. Webley and L. Mastin. Improved prediction and tracking of volcanic ash clouds. *Journal of Volcanology and Geothermal Research 186*, 1-2 (2009), 1–9.

[249] V. Welch, A. Walsh, W. Barnett, and C. A. Stewart. A roadmap for using nsf cyberinfrastructure with incommon. In *Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery* (New York, NY, USA, 2011), TG '11, Association for Computing Machinery. https://doi.org/10.1145/2016741.2016771.

[250] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics 12*, 6 (1998), 620–631.

[251] N. Wilkins-Diehr, M. Zentner, M. Pierce, M. Dahan, K. Lawrence, L. Hayden, and N. Mullinix. The science gateways community institute at two years. In *Proceedings of the Practice and Experience on Advanced Research Computing* (New York, NY, USA, 2018), PEARC '18, Association for Computing Machinery. https://doi.org/10.1145/3219104.3219142.

[252] K. Williams. Atria. https://github.com/kawilliams/expression-trees, 2019.

[253] K. Williams, A. Bigelow, and K. E. Isaacs. Visualizing a moving target: A design study on task parallel programs in the presence of evolving data and concerns. *To appear in IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis '19)* (Jan. 2020). http://dx.doi.org/10.1109/TVCG.2019.2934285.

[254] D. Wirasaet, S. Brus, C. Michoski, E. Kubatko, and J. . Westerink. Artificial boundary layers in discontinuous Galerkin solutions to shallow water equations in channels. *J. Comp. Physics 299* (2015), 597–612.

[255] M. Wolf, J. Choi, G. Eisenhauer, S. Ethier, K. Huck, S. Klasky, J. Logan, A. Malony, C. Wood, J. Dominski, et al. Scalable performance awareness for in situ scientific applications. In *2019 15th International Conference on eScience (eScience)* (2019), IEEE, pp. 266–276.

[256] W. Wondwossen. The science behind HBCU success, Sept. 2020. https://beta.nsf.gov/science-matters/science-behind-hbcu-success.

[257] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mané, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg. Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (Jan 2018), 1–12. http://dx.doi.org/10.1109/TVCG.2017.2744878.

[258] C. Wood, S. Sane, D. Ellsworth, A. Gimenez, K. Huck, T. Gamblin, and A. Malony. A scalable observation system for introspection and in situ analytics. In *2016 5th workshop on extreme-scale programming tools (ESPT)* (2016), IEEE, pp. 42–49.

[259] A. Wu, M. C. Aoi, and J. W. Pillow. Exploiting gradients and hessians in bayesian optimization and bayesian quadrature. *arXiv: Machine Learning* (2017).

[260] S. Wu and M. Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019).

[261] S. Wu and M. Dredze. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP* (2020).

[262] P. Xu, F. Roosta, and M. W. Mahoney. Second-order optimization for non-convex machine learning: An empirical study. In *Proceedings of the 2020 SIAM International Conference on Data Mining* (2020), SIAM, pp. 199–207.

[263] A. Yaguchi, T. Suzuki, S. Nitta, Y. Sakata, and A. Tanizawa. Scalable deep neural networks via low-rank matrix factorization. *arXiv preprint arXiv:1910.13141* (2019).

[264] X. Yang, M. Gao, Q. Liu, J. Setter, J. Pu, A. Nayak, S. Bell, K. Cao, H. Ha, P. Raina, C. Kozyrakis, and M. Horowitz. Interstellar: Using halide's scheduling language to analyze DNN accelerators. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems* (New York, NY, USA, 2020), ASPLOS '20, Association for Computing Machinery, p. 369–383. https://doi.org/10.1145/3373376.3378514.

[265] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems* (2019).

[266] Z. Yao, A. Gholami, S. Shen, M. Mustafa, K. Keutzer, and M. W. Mahoney. Adahessian: An adaptive second order optimizer for machine learning, 2020.

[267] C. Ying, S. Kumar, D. Chen, T. Wang, and Y. Cheng. Image classification at supercomputer scale. *arXiv preprint arXiv:1811.06992* (2018).

[268] Y. You, Z. Zhang, C.-J. Hsieh, J. Demmel, and K. Keutzer. Imagenet training in minutes. In *Proceedings of the 47th International Conference on Parallel Processing* (2018), pp. 1–10.

[269] H. Yu, R. Jin, and S. Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. *arXiv preprint arXiv:1905.03817* (2019).

[270] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).

[271] S. Zhang, A. E. Choromanska, and Y. LeCun. Deep learning with elastic averaging sgd. In *Advances in neural information processing systems* (2015), pp. 685–693.

[272] W. Zhang, S. Gupta, X. Lian, and J. Liu. Staleness-aware async-sgd for distributed deep learning. *arXiv preprint arXiv:1511.05950* (2015).

[273] X. Zhang, H. Abbasi, K. Huck, and A. D. Malony. Wowmon: A machine learning-based profiler for self-adaptive instrumentation of scientific workflows. *Procedia Computer Science 80* (2016), 1507–1518.

[274] Z. Zhang, F. Li, and S. Chen. Online learning approaches in maximizing weighted throughput. In *Proceedings of the International Performance Computing and Communications Conference (IPCCC)* (2010), pp. 206–213.

[275] Z. Zhang, J. Yang, and H. Zhao. Retrospective reader for machine reading comprehension. *ArXiv abs/2001.09694* (2020).

[276] I. Zhukov, C. Feld, M. Geimer, M. Knobloch, B. Mohr, and P. Saviankou. Scalasca v2: Back to the future. In *Tools for High Performance Computing 2014*. Springer, 2015, pp. 1–24.

# Ethics Plan

The *CAIRO* Coalition will expect its personnel to uphold the integrity of the Center's mission and to work conscientiously in maintaining a respectful, responsible, and professional environment throughout the Center's lifetime. To ensure all staff carry out their research and related work in an ethical way, the Center PIs have established a three-part plan to aid all personnel in conducting themselves responsibly. In implementing its plan, the Center will rely on several existing resources at Center campuses and from federal agencies to guarantee its staff carry themselves with integrity. Responsible conduct of research encompasses a wide variety of issues, including responsible data management and sharing, human subject protection, and negotiating authorship on scholarly works. The National Science Foundation (NSF) has required Responsible Conduct of Research (RCR) training information be made available. Moreover, the Center will make use of NSF's Online Ethics Center for Science and Engineering, its online collaborative resource, particularly regarding best practices.

## Overview

The commitment of *CAIRO* to ethical and responsible research (ER2) and ethical innovation is evidenced by its formal partnership with the LSU Ethics Institute and its investment in a postdoctoral fellowship in the critical domain of AI Ethics and emerging technologies.

This postdoctoral fellow will play a lead role in: designing the 4 modules for ER2-AI (below); coordinating the AI Ethics webinar series for the institute affiliates (below); designing the educational resources for the *CAIRO* AISP enrichment program for junior and high school students (see broader impacts/BPAI). Also, the postdoctoral fellow will pursue an independent research project in the domain of AI ethics and social risk. Working directly with *CAIRO* and Ethics Institute affiliates, this research aims to identify a framework for ethical AI research design, to include operational definitions of fairness, bias, and privacy. This sort of normative research is crucial to the future of not only AI and ML but human society as well. Academic, industry, and policy leaders lament the absence of "big think" in AI research, and *CAIRO* is investing in normative as well as technical innovation.

## Certifications

Certification in Ethics and Responsible Conduct of Research in AI (ER2-AI) is required for all *CAIRO* affiliates. This certification is managed by the Ethics and Diversity Advisory Board (EDAB).

**The ER2-AI Ethics Requirement may be satisfied by:**

1. completion of four ethics training modules (offered annually, both synchronously and asynchronously), OR;
2. an approved course on AI/Applied Ethics at affiliates' home institution (list provided to participants upon request) + Module 4
   (a) **Module 1.** Values in Science, Research Ethics & Integrity, Intellectual Property: Roles and Responsibilities for Researchers
   (b) **Module 2.** 'Ethically Aligned Design' and Responsible Innovation; Transparent, Interpretable, Trustworthy AI
   (c) **Module 3.** Mitigating bias and discrimination in AI; Ethics of Big Data; Privacy and Security Challenges
   (d) **Module 4.** Strategies for Inclusion and Diversity in AI Research

## Additional Ethics Resources for *CAIRO* Affiliates

**Webinar Series**: biannual synchronous, zoom-based talks (synchronous) on issues in the Ethics and Social Impact of AI.

**Consultative support** in the following areas: mitigating ethical risks, ethical research design and ethics auditing (organized through EDAB).

## Postdoctoral Researcher Mentoring Plan

As part of this project, the involved institutions will support, mentor, and train postdoctoral researchers. Postdocs will be based at their respective universities and will interact with other collaborators, senior personnel, postdocs, and students in the *CAIRO* project as well at their home institutions. The PIs on this project have previous experience mentoring postdoctoral researchers.

**Assist in career development, planning, and research vision**: Starting from the beginning of the postdoctoral term, the PIs will help the postdocs develop well-defined, creative, and ambitious but realistic career and research goals, and work towards those goals. To help the postdocs formulate these goals, the PIs will meet with them to discuss interests and long-term goals, and break down those goals into feasible projects. Recommended background reading will be suggested and discussed. Postdocs will be encouraged to submit their own proposals and be supported in their submission. Based on the long-term goals and specific aims of the proposal, the group will work together to shape short-term goals. Postdocs will be advised in seeking out opportunities, preparing application materials, and preparing for interviews.

**Broaden postdocs' visualization and computer science**: Postdocs will be exposed to broad training opportunities to encounter new research areas that supplement and complement their own, thereby enhancing their ability to make the kind of important connections that lead to new discoveries. The postdoc will be encouraged to attend relevant interdisciplinary seminars and participate in journal clubs. In addition, the postdoc will have the opportunity to engage in interdisciplinary projects with collaborators in high performance computing, statistics, or environmental science. Guidance on how to effectively collaborate with researchers from diverse backgrounds and disciplinary areas is inherent in this interdisciplinary project and will be emphasized in interactions with PI and their collaborators.

**Gain training in professional research ethics**: Each institution has online courses that employees are required to complete regarding ethical conduct in the workplace.

**Participate in professional development activities**: Postdocs will be advised to participate in exceptional training opportunities specific to postdoctoral researchers available at their home institutions (and virtually with collaborative institutions when possible). Postdocs will also present their research findings in lab meeting and conferences, and they will participate in training workshops.

**Obtain mentoring and outreach experience**: Postdocs will contribute to the supervision of graduate and undergraduate students. They will shadow all student meetings related to the CAIRO project, and mentoring and management strategies will be discussed in the postdocs' recurring meetings with the PIs. They will also have the opportunity to work with other students and through the outreach activities.

**Provide personal support**: Finally, the PIs will provide a supportive environment where postdocs are comfortable discussing work-life balance issues, career options, as well as any gender, ethnic, or cultural concerns.

# Data Management Plan

This section explains the specific actions that will be taken in this project to conform to NSF policy on the dissemination and sharing of research results. In accordance with NSF proposal guidelines, all data and other supplementary materials for future research and educational outreach will be stored and made accessible to other researchers, and the following text will detail the archiving and dissemination methods.

**Types of Data** This project will manage various types of data as follows:

- Software developed, including tools, compilers, and frameworks will be optimized and used by the project team. Since many of them are open-source projects under open-source licenses (e.g., BSD), PIs will use them and archive different versions of releases on GitHub (https://github.com) for public access (see Section 'Deliverables' for more details below).
- Results and experimental data from each research project that uses this project will be disseminated through publications and products. In general, the investigators expect that intellectual property issues on contributions from the PI, co-PIs, senior personnel, and collaborators will follow their license terms. All contributions will be reviewed and approved, providing a means to detect violations.

Furthermore, the project expects to support multidisciplinary research projects that would produce unprecedented amounts of data. In general, each collaborator's research project will follow its original data management plan.

In addition to the public information, the project will maintain social and relational information among researchers and research groups; and technical information on each instrument and equipment. Since such information is related to security and privacy, it will be stored on the system management server secured by a security system. Only personnel including the PI, co-PIs, senior personnel, and project administrators will have an access to such data.

## Source code repositories

The PI and Co-PIs plan to manage and archive source codes, data and results at Git repositories maintained on Github (https://github.com). Access to any data at the Git repository can be allowed or restricted on a per folder basis to everyone, anyone, or a group. The data from the project will be stored throughput the life of the project. As all data and source codes are updated, Git will handle versioning automatically.

## Project/User Discussions

The project team will maintain a Slack team and a wiki website to keep all discussions from users and developers regarding the project, such as questions and experiments. The wiki website will also contain training materials, such as examples of scripts for experimentation and tutorials on how to use the hardware and software (e.g., systems research, different domains' applications research).

## Storage Systems for Preservation and Archiving

The long-term data storage and archival needs of the *CAIRO* project researchers will be served by CCT in conjunction with LSU's Office of Information Technology Services (ITS). PIs expect that this system will satisfy the primary big data storage needs of researchers associated with *CAIRO*. In addition to those large storage systems, CCT maintains SVN and Git repositories and wiki services for research projects, and those services are expected to be supported long after the life of the project. These servers are routinely backed up, protecting against system failure and vandalism. CCT is regularly updating its data management plan in concert with the needs of LSU's research community and NSF guidelines.

## Access, Sharing, and Intellectual Property Issues

With few exceptions data will be publicly available. This includes anonymous read-only access to the code depository. Any user will be able to obtain development versions as well as release versions of the code. The majority of software elements we have produced in the past are licensed under the 3-clause BSD or similar licenses, and we plan to use the same license for this project. This license is one of the most permissive licenses. It is widely used and analyzed by legal experts, making it friendly to both commercial and academic users. All contributions will be reviewed and approved, providing a means to detect violations.

## Privacy

A privacy statement will be provided to users of this project wiki and repository. That statement will explain that posts are public, which should be obvious to users. Code contributors will be made aware of the license.

## Deliverables

Our intention with delivering our software is not simply to make it available for download, but to create a real and vital user community. To that end, our development process will be hosted on github.com, which provides and integrates a number of important tools and features for creating and supporting the kind of community we envision.

**Code and Documentation Repository** Of course, Github functions as a repository for source code as well as for high-quality documentation, and this provides the simplest engagement mechanism for end-users: anyone is able to download and use the code, per the associated license (described below).

**Continuous Integration** Continuous Integration / Continuous Delivery (CI/CD) pipelines comprise a set of software development practices and tools for shortening development cycles and removing barriers between developers and end users. The Github CI/CD pipelines are fully customizable but typically consist of three phases: a build phase to check for successful compilation, a test phase to check that unit tests run successfully, and a deployment phase which releases a successful production pipeline to users of the software. These builds are run inside of containers hosted by github and can be configured with a variety of compilers and library support. That is, the build phase may consist of builds under multiple compilers, operating systems, etc. Github can choose to create CI/CD pipelines whenever developers change the codebase, at prescheduled times, or when invoked by a developer.

**Community-Based Development** An open-source code repository is a two-way street, enabling users to not only download code but to contribute to the code base as well. There are different levels of involvement that a user may have, from making suggestions on the issues list to becoming a full-fledged member of the development team. To facilitate communication among developers through all stages of the development process, Github provides an issues tracker, a wiki, Gists, and notifications and feeds. Our development process will include code contributions via pull requests and code review will be conducted via Github.

**User Engagement.** Outside of the development environment itself, we will conduct events such as workshops, BoFs, demonstrations, poster sessions, and participate in standardization bodies. Furthermore, we plan to organize at least one workshop (additional NSF funds) when the software reaches a level of maturity. We will reach out to other communities by organizing BoFs, demos, and presenting posters at related events.

## Metrics

We will gather metrics from the hosting environment used for development, which are provided directly by Github's tools. These will include code quality metrics, such as results from code-quality tools (e.g., clang-tidy), build tests, unit tests, integration tests, and bug reports. In addition, we will gather metrics for developer activity, such as number of commits, number of external developers, number of mailing list messages, and amount of contribution to collaborative environments (e.g., wiki). As software for which mathematical accuracy and computational performance are critical issues, we will also implement and monitor a number of micro and macro kernels for verification and validation as well as performance. Finally, we will gather metrics regarding the effectiveness of our process, e.g., outstanding time between bug reporting and bug fixing. All metrics will be published on our Github site.

## Facilities

LSU Center for Computation & Technology (CCT) hosts a small research cluster named **Rostam**. This cluster is mainly used by researchers for the development and test of their codes before moving to bigger clusters. CCT will make this cluster and its storage available to all members of the project. The resources include (`https://wiki.rostam.cct.lsu.edu/en/cluster/hardware`):

| **Resource** | **Description** | | | | |
|---|---|---|---|---|---|
| **Compute** | Number | Cores | Memory | Accelerator | OS |
| | 16 | 40 (Skylake) | 96 GB | None | CentOS 8 |
| | 16 | 16 (Sandy Bridge) | 48 GB | None | CentOS 8 |
| | 1 | 40 (Skylake) | 386 GB | 4 x V100 | CentOS 8 |
| | 1 | 20 (Haswell) | 256 GB | 2 x V100 | CentOS 8 |
| | 1 | 20 (Haswell) | 128 GB | 4 x K80 | CentOS 8 |
| | 1 | 20 (Ivy Bridge) | 128 GB | 2 x R9 Fury | CentOS 8 |
| | 1 | 20 (Ivy Bridge) | 128 GB | K40 + R9 Fury | CentOS 8 |
| | 1 | 16 (Sandy Bridge) | 64 GB | None | CentOS 8 |
| | 4 | 4 (Cortex-A72) | 4 GB | None | Ubuntu 20.04 |
| | 8 | 4 (Cortex-A53) | 1 GB | None | Ubuntu 20.04 |
| **Storage** | 120 TB ZFS | | | | |
| **Ethernet** | 25 Gb | | | | |
| **Infiniband** | 56 Gb | | | | |
| **Authentication** | LDAP + Two-Factor Authentication | | | | |
| **Others** | Jenkins Build System, SLURM work Scheduler, Wiki | | | | |

In order to support the goal of the *CAIRO* coalition to adapt the produced CI infrastructure to the most modern accelerator hardware we plan to extend the **Rostam** cluster by purchasing additional nodes in years one and three as outlined in the budget and budget justification:

- **First year:**
  - Two servers with four Nvidia A100 GPUs, 60 CPU cores, 512GB memory and infiniband connectivity. These servers give us the capability of testing the scalability of the project, both on a single locality and also on a minimal distributed environment. These types of machines (2 cpu, 4 gpu) are more inline with current HPC environments such as SUMMIT and give us the opportunity to test our programs before moving on to any external HPC system.
  - An storage solution with 100TB capacity.
    In AI problems the raw input data and intermediate and final results could sum up terabytes of data. We would need a fast and reliable storage solution in close proximity to be able to store these data locally.
- **Third year:**
  - Two servers with four next generation Nvidia GPUs (or similar), two next generation CPUs.
    We will repeat our experiments and test our progress with the next generation of GPUs and CPUs, we plan for two similar machines with next generation hardware.

Also Louisiana State University (LSU) and LONI (Louisiana Optical Network Initiative) will make their high-performance computing and storage resources, hosted by "HPC@LSU" available for all members of this project. These computing and storage resources, operated by LSU's Information Technology Services (ITS) and the LSU Center for Computation & Technology (CCT), include (see: `http://www.hpc.lsu.edu/resources/hpc/index.php`) the following.

| System | Description | SUs | Peak |
|---|---|---|---|
| **SuperMIC** (NSF-funded; An XSEDE resource) | Include 382 nodes, each with two 10-core 2.8GHz Intel Ivy Bridge-EP processors. 380 compute nodes have 64 GB of SuperMIC memory and 500 GB of local HDD storage, FDR InfiniBand. 360 compute nodes have 2 Intel Xeon Phi 7120P coprocessors. 20 compute nodes have 1 Intel Xeon Phi 7120P coprocessor and 1 NVIDIA Tesla K20X. 840 TB Lustre high-performance disk storage subsystem. | 66,576,000 | 1050 TF |
| **SuperMike-II** | Includes 443 node (7,424 total cores) Red Hat Enterprise Linux (RHEL v6) cluster (Intel 2.6 GHz 64-bit Sandy Bridge processors). 50 nodes have dual NVIDIA M2090 GPUs. 32 GB RAM each for 404 nodes, 64 GB RAM each for 52 GPU nodes, 256 GB RAM each on 8 nodes. QDR InfiniBand interconnect. 1 node with 40 cores and 1 TB memory. 400 TB Lustre high-performance disk storage subsystem. | 65,034,240 | 228 TF |
| **QB3** (LONI resource) | QB-3 is an 856 TFLOPS peak performance cluster: 202 nodes, each with two 24-core 2.4GHz Intel Xeon Platinum 8260 64-bit processors (9696 total cores). Each node contains 192GB RAM and has a HDR100 InfiniBand interconnect. Eight compute nodes contain 2 V100 NVIDIA accelerators per node. Two big memory nodes have 1.5TB RAM each. The cluster has a 1.7 PB Lustre high-performance storage subsystem. | 84,936,960 | 856 TF |
| **QB2** (LONI resource) | A 1.5 PFLOPS peak performance, 504 nodes, each with two 10-core 2.8GHz Intel Ivy Bridge-EP 64-bit processors (10,080 total cores). Each node contains 64GB RAM. 480 compute nodes contain 2 NVIDIA Tesla K20X accelerators per node, FDR InfiniBand interconnect. 16 compute nodes have 2 Intel Xeon Phi 7120P coprocessors per node. 4 big memory nodes have 1.5TB RAM each. 4 visualization nodes contain 2 NVIDIA Tesla K40 accelerators. 2.8 PB Lustre high-performance storage subsystem. | 88,300,800 | 1530 TF |
| **TOTAL:** | | **304,848,000** | |

| Storage | Description | Total |
|---|---|---|
| LSU Storage | 1240 TB of high-performance storage running Lustre;<br>200 TB of long-term storage running Lustre. | 1,440 TB |
| LONI Storage | 4300 TB of high-performance storage running Lustre (1.5PB on QB3) | 4,300 TB |
| **TOTAL:** | | **5,740 TB** |

**Networking:** The LSU Center for Computation & Technology has spearheaded an initiative (LONI: Louisiana Optical Network Initiative) in Louisiana to improve statewide networking and computing resources via the installation of a 100 Gigabit optical network connecting all major Louisiana research sites. The LONI network provides a 100 Gb/s connection to Internet2. The LSU SuperMIC and SuperMike-II, and LONI's QB3 and QB2 are connected by 40 Gb/s circuits to both LONI and Internet2; other clusters like Eric are connected by 10 Gb/s circuits. CCT itself is currently connected by 4x10 Gb/s connections to the LSU network core, and by two 40 Gb/s to LONI and Internet2 giving an aggregate bandwidth of over 100Gb/s.

**Visualization and Display:** The Center for Computation & Technology has built an advanced visualization and digital arts auditorium housed in the Digital Media Center (DMC). Central to this facility is the DMC Theatre, a 202 seat, 5700 square foot auditorium designed for visualization and teleconferencing. A Cisco C90 VTC teleconferencing unit that includes multiple inputs and outputs for up to 1080p HD video is used for simultaneous broadcasting of audio, video and presentation materials to multiple sites using H.323 teleconferencing standards. The main function of this unit is for distance learning.

**Other Facilities and Resources:** LSU's Center for Computation & Technology is housed in a 94,000 square foot Louisiana Digital Media Center that is also home to 400 video game development workers who work for EA Sports' North American Testing Center. This three-story structure on LSU's main campus provide a contemporary, permanent home for the CCT's research and computing facilities. In addition, the new building contains a 22-rack data center with 40Gb/s connectivity to the building core, nine multimedia conference rooms, four multimedia classrooms and four research laboratories.

CCT has a strong internal computing infrastructure including fault-tolerant virtualized hosts that provide mail, web, code management (SVN), DB, Wiki, and LDAP servers. The Web and wiki servers are used for the Center portal, and the SVN server supports Center code development. The internal computing resources include a large storage space of 100 TB that can be used by all the users at CCT and affiliates. CCT has wired and wireless networks, with approximately 500 Gigabit Ethernet Drops and 50+ Wireless Access points (a/b/g/n bands). There are approximately one hundred 10Gb/s ports available in the research spaces (both fiber and copper) and the number can be expanded with need.

The Center for Computation and Technology at LSU will provide administrative support to the PIs for managing this project, and for dissemination and event planning through the center's Event Coordination programs. The Visitor Program at the Center for Computation and Technology will be available for hosting visits of contributing members and collaborators to LSU.