15th JLESC Workshop

Tuesday, 21 March 2023 - Thursday, 23 March 2023

LaBRI

Book of Abstracts

Contents

Short Talks on Applications / **6**

HPC challenges for ocean and atmosphere simulations

Author: Martin Schreiber[1]

[1] *Université Grenoble Alpes*

The INRIA AIRSEA team in Grenoble is specialized on ocean and atmosphere models.
This short talk will briefly outline our current HPC research topics:

- Domain specific languages: For patch-based Ocean simulations CROCO and NEMO
- Load balancing: Non-equal time-varying workload, AMR, coupling, homogeneous and heterogeneous systems
- Data-intensive data-flows: Part of data assimiliation and uncertainty quantification
- Dynamic resources: Getting rid of the static resource allocation in MPI
- Parallel-in-time methods: Targeting atmosphere and ocean simulations

JLESC topic:

Short Talks on Distributed Resources / **7**

Towards an application-driven dynamic resource approach for HPC

Authors: Dominik Huber[1]; Martin Schreiber[2]

[1] *Technical University of Munich (TUM)*

[2] *Université Grenoble Alpes*

This short talk provides an introduction to the ongoing research at UGA /TUM (EuroHPC Time-X) on an application-driven dynamic resource approach for HPC. Time-X targets the area of parallel-in-time (PinT) integration, where resource dynamic strategies have been shown to improve the performance and efficiency of PinT algorithms.

However, current approaches to enable dynamic resources for HPC applications are often application, programming model or process manager specific or lack integration with the system-wide resource management.

To this end, UGA/TUM (Time-X) collaborates with the PMIx and MPI communities as well as other EuroHPC projects on a standardized, agnostic approach for dynamic resources. This talk discusses some of the basic considerations and challenges of this work.

JLESC topic:

Novel programming models and runtime systems, which allow scientific applications to be updated or reimagined to take full advantage of extreme-scale supercomputers

Project Talks on I/O, Storage and Workflows / **8**

More on I/O scheduling

Author: Lucas Perotin[1]

Co-authors: Anne Benoit [1]; Thomas Herault [2]; Yves Robert ; Frédéric Vivien [1]

[1] *Inria*

[2] *UTK*

This is the report for the project 'Optimization of Fault-Tolerance Strategies for Workflow Applications'

Checkpoint operations are periodic and high-volume I/O operations and, as such, are particularly sensitive to interferences. Indeed, HPC applications execute on dedicated nodes but share the I/O system. As a consequence, interferences surge when several applications perform I/O operations simultaneously: each I/O operation takes much longer than expected because each application is only allotted a fraction of the I/O bandwidth.

This is the motivation for our study about I/O interference.
We design and evaluate several new algorithms for bandwidth sharing,
which we compare with existing work. We do not assume any knowledge
of the applications nor any regularity pattern in I/O operations.

Overall, this project talk is NOT about resilience, even though concurrent checkpoints were the initial motivation.

JLESC topic:

I/O

Short Talks on Workflows, I/O and Frameworks / 9

On Temporal I/O Behavior Characterization: Predicting I/O Phases Using Frequency Techniques

Authors: Ahmad Tarraf[None]; Alexis Bandet[None]; Felix Wolf[None]; Francieli Boito[1]; Guillaume Pallez[1]

[1] *Inria*

In this paper, we propose an approach based on signal processing to characterize HPC applications' temporal I/O behavior. In the context of each application, our goal is to detect/predict the temporal aspects of its access pattern, i.e. the I/O phases (each composed of one or many individual I/O requests) and their periodicity. Such information can very useful for optimization techniques such as I/O scheduling, burst buffers
management, I/O-aware batch scheduling, etc. Our approach uses signal processing techniques, namely Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT) in a signal made of the I/O bandwidth over time (for a small time window). We present our approach and validate it with large-scale experiments, but we also discuss scenarios that were crafted to identify the limitations of such signal processing-based approaches for I/O behavior characterization.

JLESC topic:

Short Talks on Applications / 10

Improving IPPL: a performance portable library for grids and particles

Author: Sriramkrishnan Muralikrishnan[None]

Co-authors: Matthias Frey [1]; Andreas Adelmann [2]

[1] *University of St Andrews, UK*

[2] *Paul Scherrer Institut*

Independent Parallel Particle Layer (IPPL) is an open-source performance portable C++ library for generic computations with grids and particles. It is primarily used for large scale kinetic plasma simulations. In this talk, I will briefly introduce the library, and show some of the recent benchmarks we performed on pre-exascale leadership computing systems with thousands of GPUS and CPU cores. I will then discuss the features we would like to have in IPPL in the areas of I/O, in-situ visualization and performance measurement tools which can open the door for potential collaborations.

JLESC topic:

Short Talks on AI/MD/DL / 11

A Feature-Driven Fixed-Ratio Lossy Compression Framework for Real-World Scientific Datasets

Author: Sheng Di[1]

[1] *Argonne National Laboratory*

Today's scientific applications and advanced instruments are producing extremely large volumes of data everyday, so that error-controlled lossy compression has become a critical technique to the scientific data storage and management. Existing lossy scientific data compressors, however, are designed mainly based on error-control driven mechanism, which cannot be efficiently applied in the fixed-ratio use-case, where a desired compression ratio needs to be reached because of the restricted data processing/management resources such as limited memory/storage capacity and network bandwidth. To address this gap, we propose a low-cost compressor-agnostic feature-driven fixed-ratio lossy compression framework (FXRZ). The key contributions are three-fold. (1) We perform an in-depth analysis of the correlation between diverse data features and compression ratios based on a wide range of application datasets, which is a fundamental work for our framework. (2) We propose a series of optimization strategies that can enable the framework to reach a fairly high accuracy in identifying the expected error configuration with very low computational cost. (3) We comprehensively evaluate our framework using 4 state-of-the-art error-controlled lossy compressors on 10 different snapshots and simulation configuration-based real-world scientific datasets from 4 different applications across different domains. Our experiment shows that FXRZ outperforms the state-of-the-art related work by 108X. The experiments with 4,096 cores on a supercomputer show a performance gain of 1.18-8.71X than the related work in overall parallel data dumping.

JLESC topic:

Short Talks on Advanced Architectures / 12

SciStream: Architecture and Toolkit for Streaming Data from Instruments to HPC

Author: Rajkumar Kettimuthu[1]

[1] *Argonne National Laboratory*

Modern scientific instruments, such as detectors at synchrotron light sources, generate data at such high rates that online processing is needed for data reduction, feature detection, experiment steering, and other purposes. Leadership computing facilities (e.g., ALCF) are deploying mechanisms that would enable these applications to acquire (a portion of) HPC resources on-demand. These workloads would greatly benefit from memory-to-memory data streaming capabilities from instrument to remote HPC as data transmissions that engage the file system introduce unacceptable latencies. But efficient and secure memory-to-memory data streaming is challenging to realize in practice, due to a lack of direct external network connectivity for scientific instruments; and authentication and security requirements. In response, we have developed SciStream, a middlebox-based architecture and toolkit with appropriate control protocols to enable efficient and secure memory-to-memory data streaming between instruments and HPC. In this talk, we will describe (a) the architecture and protocols that SciStream uses to establish authenticated and transparent connections between instruments and HPC; (b) the design considerations; (c) the implementation approaches for SciStream; and (d) deployment options. We will also present the preliminary results from the experiments that we have conducted evaluate SciStream.

JLESC topic:

HPC clouds

Short Talks on Tasking / 13

DASK enabled external tasks for in transit analytics with DEISA

Authors: Amal Gueroudji[1]; Bruno Raffin[2]; Julien Bigot[1]

[1] *French Alternative Energies and Atomic Energy Commission (CEA Sacaly)*

[2] *Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000 Grenoble, France*

In situ paradigm represents a relevant alternative to classical post hoc workflows as it allows bypassing disk accesses, thus reducing the IO bottleneck. However, as most in situ data analytics tools are built on MPI, and they are complicated to set up and use, especially to parallelize irregular algorithms. In a previous work, we provided a tool that couples MPI simulations with in situ task-based analytics written in Dask Distributed called Deisa[1]. In our old prototype, data and metadata were exchanged synchronously at each timestep (that overloads the scheduler), and a new task graph was submitted to process that step every time (time dependencies need to be managed manually to write algorithms).
In this work, we have addressed these limitations and improved our design by introducing asynchronicity and reducing the traffic to the scheduler. In addition, we avoid metadata fetch and allow submitting time-independent task graphs thanks to three main concept: “deisa virtual arrays” data structure, “contracts” to make selections of only needed data, and “external tasks” in Dask distributed to support getting data from external running programs (a running MPI simulation in our case).
We have implemented these improvements on top of the work presented in [1]. We have added a new Deisa plugin in the PDI Data interface and included our “external tasks” contribution into a forked version Dask Distributed repository. We have tested our approach using a heat equation mini-app with several analytics, such as temporal derivative and incremental PCA, and working on production use cases.

Deisa[1], an in situ analytics tool, [1] A. Gueroudji, J. Bigot and B. Raffin, “DEISA: Dask-Enabled In Situ Analytics,” 2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC), 2021, pp. 11-20, doi: 10.1109/HiPC53243.2021.00015.

JLESC topic:

Short Talks on Tasking / 14

Programming Heterogeneous Architectures using Hierarchical Tasks

Author: Gwenole Lucas[1]

[1] *Inria Bordeaux*

Task-based systems have become popular due to their ability to utilize the computational power of complex heterogeneous systems. A typical programming model used is the Sequential Task Flow (STF) model, which unfortunately only supports static task graphs. This can result in submission overhead and a task graph that is not well-suited for execution on heterogeneous systems. A common approach is to find a balance between the granularity needed for accelerator devices and the granularity required by CPU cores to achieve optimal performance. To address these issues, we have extended the STF model in the STARPU runtime system by introducing the concept of hierarchical tasks. This allows for a more dynamic task graph and, when combined with an automatic data manager, it is possible to adjust granularity at runtime to best match the targeted computing resource. Additionally, submission overhead is reduced by using large-grain hierarchical tasks, as the submission process can now be done in parallel. We have shown that the hierarchical task model is correct and have conducted an early evaluation on shared memory heterogeneous systems using the CHAMELEON dense linear algebra library.

JLESC topic:

Short Talks on Tasking / 15

Memory-Aware Scheduling of Tasks Sharing Data on Multiple GPUs with Dynamic Runtime Systems

Authors: Loris MARCHAL[1]; Maxime GONTHIER[2]; Samuel THIBAULT[3]

[1] *CNRS*

[2] *INRIA Bordeaux*

[3] *Université de Bordeaux*

The use of accelerators such as GPUs has become mainstream to achieve high performance on modern computing systems. GPUs come with their own (limited) memory and are connected to the main memory of the machine through a bus (with limited bandwidth). When a computation is started on a GPU, the corresponding data needs to be transferred to the GPU before the computation starts. Such data movements may become a bottleneck for performance, especially when several GPUs have to share the communication bus.

Task-based runtime schedulers have emerged as a convenient and efficient way to use such heterogeneous platforms. When processing an application, the scheduler has the knowledge of all tasks available for processing on a GPU, as well as their input data dependencies. Hence, it is possible to produce a tasks processing order aiming at reducing the total processing time through three objectives: minimizing data transfers, overlapping transfers and computation and optimizing the eviction of previously-loaded data. We focus on this problem of partitioning and ordering tasks that share some of their input data on multiple GPUs. We present a novel dynamic strategy based on data selection

to efficiently allocate tasks to GPUs and a custom eviction policy, and compare them to existing strategies using either a well-known graph partitioner or standard scheduling techniques in runtime systems.
We present their performance on tasks from tiled 2D, 3D matrix products and Cholesky factorization, as well as a sparse matrix product.

All strategies have been implemented on top of the StarPU runtime, and we show that our dynamic strategy achieves better performance when scheduling tasks on multiple GPUs with limited memory.

JLESC topic:

Short Talks on Workflows, I/O and Frameworks / 18

Bringing Elasticity to HPC Storage and Data Services

Author: Matthieu Dorier[1]

[1] *Argonne National Laboratory*

Elasticity, or the ability to adapt a system to a dynamically changing workload, has been a core feature of Cloud Computing storage since its inception more than two decades ago. In the meantime HPC applications have mostly continued to rely on static parallel file systems to store their data. This picture is now changing as more and more applications adopt custom data services tailored to their needs, including in transit analysis systems, staging areas for coupling, and transient file system aggregating on-compute-node storage capacity. As a result, it is increasingly important to incorporate the concept of elasticity within HPC so that these new data services can dynamically adapt in response to time-varying application requirements.

In this talk, we will present current efforts from the Mochi team to tackle the challenges of data service elasticity. Mochi is an R&d100-awarded collection of building blocks for developing composable HPC data services. It enjoys a growing community of users and contributors, with applications that increasingly need Mochi to support elasticity. We will highlight the different levels at which elasticity can be implemented, from low-level thread-scheduling decisions, to scaling across nodes with data migration. We will show the technical challenges, as well as opportunities from the AI domain to enable self-adapting data services.

JLESC topic:

Project Talks on I/O, Storage and Workflows / 19

Open Platforms and Reproducibility

Authors: Alexandru Costan[1]; Daniel Rosendo[1]; Gabriel Antoniu[1]; Katarzyna Keahey[2]

[1] *INRIA*

[2] *ANL*

Open experimental platforms for Computer Science systems research, like the Chameleon and Grid'5000/FIT testbeds, are a critical tool not only for the support of computer science experimentation but also a key enabler of reproducibility. One of the perennial challenges that scientific instruments of this type grapple with are how they should evolve to support the emergent needs of research. Another is the definition and alignment of abstractions through which these resources should be provided such

that research may be portable across different platforms. Finally, once such abstractions are found, the challenge is to create tools and services that will make packaging experiments for repeatable execution feasible.

This talk will present an update of the collaboration between the Chameleon and Grid'5000 testbeds on all these questions. We will describe the manner in which respectively CHI@Edge (for Chameleon) and FIT (for Grid'5000) address the challenge of supporting edge to cloud experimentation as well as report on the CHI-in-a-Box packaging of Chameleon that has been used to support IndySCC experimentation and is being used to integrate unique resources (such as Fugaku nodes or ARM Thunder nodes), often ephemerally for reproducibility purposes. We will then describe reproducibility tools and workflows and how they leverage abstractions the respective testbeds expose to run edge to cloud experiments across platforms. Finally, I will talk about the recently funded REPETO project that supports international collaboration on fostering practical reproducibility in computer science research and describe how it can be leveraged by JLESC attendees.

JLESC topic:

Project Talks on AI/ML/DL / 20

Leveraging Rehearsal Buffers to Enable Efficient Data-Parallel Continual Learning

Authors: Alexandru Costan[1]; Bogdan Nicolae[2]; Gabriel Antoniu[1]; Thomas Bouvier[None]

[1] *Inria*

[2] *ANL*

Deep Learning (DL) emerged as a way to extract valuable information from ever-growing volumes of data. However, when trained on sequential tasks *ie. without full access to the dataset at the beginning of the training*, typical Deep Neural Networks (DNNs) suffer from catastrophic forgetting, a phenomenon causing them to reinforce new patterns at the expense of previously acquired knowledge. This limitation prevents updating models incrementally, which is problematic in many real-life scenarios where the aforementioned datasets are replaced by data streams generated over time by distributed devices. It is unfeasible to train models from scratch every time new samples are being ingested either, as this would lead to prohibitive time and/or resource constraints.

In this talk, we will present techniques based on rehearsal to achieve Continual Learning at scale. Rehearsal-based approaches leverage representative samples previously encountered during training to augment future minibatches with. The key novelty we address is how to adopt rehearsal in the context of data-parallel training, which is one of the main techniques to achieve training scalability on HPC systems. The goal is to design and implement a distributed rehearsal buffer that handles the selection of representative samples and the augmentation of minibatches asynchronously in the background. We will discuss trade-offs introduced by such a continual learning setting in terms of training time, accuracy and memory usage.

JLESC topic:

Continual Learning at Scale

Short Talks on Distributed Resources / 21

Study of the folding of distributed experiments containing a distributed file system

Authors: Quentin Guilloteau[None]; Olivier Richard[1]; Eric Rutten[2]

[1] *UGA*

[2] *INRIA*

The development and evaluation of grid or cluster middlewares, such as batch schedulers, require to deploy numerous machines to reach an environment close to the full scale of the production system.

To avoid these huge deployments, one can consider folding the system on itself by deploying several "virtual" resources onto one physical resource.

In this study, we investigate the variations in performance for a distributed IO benchmark while folding a cluster of machines which contains a distributed file system.

This work is joint with Eric Rutten (INRIA) and Olivier Richard (UGA)

JLESC topic:

experiments

Short Talks on Tasking / **22**

A Communication Module for FMSolvr

Authors: Ivo Kabadshow[1]; Theresa Werner[None]

[1] *Juelich Supercomputing Centre*

In the process of modularizing the molecular dynamics simulation library FMSolvr of JSC, we do not only want to improve our numerical methods but also take a look at communication and whether we can achieve an improvement on that front. We conducted a systematic literature review to see what has already been done in this field and picked out two promising communication schemes for further analysis: Shift communication as by Plimpton (1993) and a processor team-based approach by Driscoll, Georganas and Koanantakool (2013). Now, we are focused on finding a formal way of modelling these approaches in the hopes that it will help us with finding a formula for the trade-off point between one and the other method. The calculation of this trade-off point will be part of the communication module, which based on the result decides which communication method is best applied for the given input values of the simulation.

JLESC topic:

communication in HPC

Poster Session / **23**

A Communication Module for FMSolvr

Authors: Ivo Kabadshow[1]; Theresa Werner[None]

[1] *Juelich Supercomputing Centre*

In the process of modularizing the molecular dynamics simulation library FMSolvr of JSC, we do not only want to improve our numerical methods but also take a look at communication and whether we can achieve an improvement on that front. We conducted a systematic literature review to see what has already been done in this field and picked out two promising communication schemes for

further analysis: Shift communication as by Plimpton (1993) and a processor team-based approach by Driscoll, Georganas and Koanantakool (2013). Now, we are focused on finding a formal way of modelling these approaches in the hopes that it will help us with finding a formula for the trade-off point between one and the other method. The calculation of this trade-off point will be part of the communication module, which based on the result decides which communication method is best applied for the given input values of the simulation.

JLESC topic:

communication in HPC

Project Talks on I/O, Storage and Workflows / 24

LibPressio: A Unifiying Data Compression Interface for Users and Developers

Author: Robert Underwood[1]

[1] *Argonne National Laboratory*

Scientists have lots of data that they need to store, transport, and use. Lossy compression could be the solution, but there are 32+ compressors, each with its own interface and the interfaces of the most recent compressors often evolve. Moreover, compressors are missing key features: provenance and configuration parameter optimization. LibPressio addresses all these issues by providing a unifying interface with advanced engines for provenance and configuration optimization. This talk will highlight recent new capabilities for GPU-enabled other features for performance portable compression.

JLESC topic:

lossy compression

Short Talks on AI/MD/DL / 25

JuLES: AI Super-Resolution Models for Large-Scale Simulations at Scale

Author: Mathis Bode[1]

[1] *Forschungszentrum Jülich GmbH*

Super-resolution tools have been originally invented for image super-resolution but are also increasingly used for improving scientific simulations or data-storage. Examples range from cosmology to urban prediction. One particular network framework, physics-informed enhanced super-resolution generative adversarial networks (PIESRGANs), has been shown to be a powerful tool for subfilter modeling. It is the basis for JuLES (JUelich Large-Eddy Simulation) which has been recently developed to generate AI super-resolution models at scale and accelerate large-scale simulations significantly. This talk highlights important modeling aspects employing PIESRGAN with applications to HPC simulations. The examples range from simple homogeneous isotropic turbulence to finite-rate-chemistry premixed flame kernels. A priori and a posteriori results are presented.

JLESC topic:

Poster Session / 26

On MPI+(task-based OpenMP) performances

Author: Romain PEREIRA rpereira[None]

The architecture of supercomputers is evolving to expose massive parallelism by considerably increasing the number of compute units per node. HPC users must adapt their applications to remain efficient on current and ultimately on future hardware. Open Multi-Processing (OpenMP) and the Message Passing Interface (MPI) are two HPC programming standards widely used and both aim at performant and portable codes but work on different parallelism levels: the shared and the distributed memory levels.

OpenMP proposes a task-based programming model which composability shall enable seemless hybridization with other asynchronous programming models such as MPI. Composing the two standards thus appears as a well-suited solution for performant and portable codes.

This poster presents the level of performance to expect from an optimized MPI+(task-based OpenMP) software stack. Our methodology consists in modeling hybrid applications to a unified task graph scheduling problem on which some metrics are defined to analyze application performances. Proxy-applications are ported, analyzed, and improved through both user-code and runtime optimizations wherever it is the most suitable to preserve real-world code representativeness. Performance results show 1.9 speedup weak-scaled to 16,000 cores from our task-based over the for-loop parallel version on LULESH. We show the highest performances on fine dependant tasks of 100 us. in average with about 80% communication overlap.

JLESC topic:

Project Talks on AI/ML/DL / 27

Machine Learning for Predicting Flow Fields

Authors: Mario Ruettgers[1]; Ando Kazutu[2]

Co-authors: Wolfgang Schröder [3]; Makoto Tsubokura [4]; Andreas Lintermann [5]

[1] *Institute of Aerodynamics and Chair of Fluid Mechanics (AIA, RWTH Aachen University), Jülich Supercomputing Centre (JSC, FZ Jülich), Jülich Aachen Research Alliance - Center for Simulation and Data Science (JARA-CSD)*

[2] *RIKEN Center for Computational Science*

[3] *Institute of Aerodynamics and Chair of Fluid Mechanics (AIA, RWTH Aachen University), Jülich Aachen Research Alliance - Center for Simulation and Data Science (JARA-CSD)*

[4] *RIKEN Center for Computational Science, Graudate School of Systems Informatics (Kobe University)*

[5] *Jülich Supercomputing Centre (JSC, FZ Jülich), Jülich Aachen Research Alliance - Center for Simulation and Data Science (JARA-CSD)*

This JLESC collaboration focuses on the prediction of flow fields using machine learning (ML) techniques. The basis for the project are jointly developed convolutional neural networks (CNNs) with an autoencoder-decoder type architecture, inspired by the work in [1]. These CNNs are used to investigate dimension-reduction techniques for a three-dimensional flow field [2]. That is, the CNNs are trained to identify the different modes of the flow, and the results are compared to conventional techniques for mode decomposition. The basic loss function considers the mean-squared error between the predicted flow field, expressed by the sum of all modes, and the flow field used as input to the CNNs. Additionally, the influence of physical loss functions that consider the dominating frequency and energy of a mode on predictions is investigated. Furthermore, time-evolution of the

reduced-order space is evaluated using a reduced-order model (ROM) based on long short-term memory (LSTM) networks and gated recurrent units (GRUs). The neural networks are implemented with a performance-effective distributed parallel scheme on Fugaku.

[1] T. Murata, K. Fukami, and K. Fukagata, "Nonlinear mode decomposition with convolutional neural networks for fluid dynamics", Journal of Fluid Mechanics, vol. 882, 13, 2020, doi:10.1017/jfm.2019.822.
[2] K. Ando, K. Onishi, R. Bale, M. Tsubokura, A. Kuroda, and K. Minami, "Nonlinear mode decomposition and reduced-order modeling for three-dimensional cylinder flow by distributed learning on Fugaku", Proceedings of International Conference on High Performance Computing (ISC2021), Springer, Cham, pp.122–137, 2021, doi:10.1007/978-3-030-90539-2_8.

JLESC topic:

Short Talks on Applications / **28**

Aevol: An experimental evolution simulator (and its mini-Apps)

Author: Jonathan Rouzaud-Cornabas[1]

[1] *Inria / LIRIS*

The Inria Beagle project-team at LIRIS has been developing evolutionary models (Experimental Evolution In Silico) for more than 15 years, and in particular the Aevol software, which makes it possible to identify predictive molecular markers in evolution (emergence of variants, resistance to antibiotics, environmental changes). These markers can be environmental characteristics (living conditions, environmental variations, ...), populational (population size, migrations, ...) or molecular (epistasic interactions, structural variants, ...).

In order to be able to scale these models while increasing their complexity, we apply a co-design approach between modeling approaches and the development of numerical and computational tools and methods related to high-performance computing.

First, we will present the application, its usage and its computational and memory access patterns. Second, we will shortly present previous works on it (in collaboration with Inria Avalon) and the related mini-Apps. Last, as part of the transition to exascale, we will present our roadmap to a rewrite Aevol to take into account a programming model facilitating code variability, performance portability (CPU, Vectorization, GPU) and in-situ data analysis.

JLESC topic:

Short Talks on Numerical Methods / **29**

ExODE : Scaling the solving of Ordinary Differential Equation for Computational Biology

Authors: Arsène Marzorati[1]; Jonathan Rouzaud-Cornabas[2]; Samuel Bernard[1]; Thierry Gauthier[3]

[1] *Inria / ICJ*

[2] *Inria / LIRIS*

[3] *Inria / LIP*

In biology, the vast majority of systems can be modeled as ordinary differential equations (ODEs). Modeling more finely biological objects leads to increase the number of equations. Simulating ever

larger systems also leads to increasing the number of equations. Therefore, we observe a large increase in the size of the ODE systems to be solved. A major lock is the limitation of ODE numerical resolution software (ODE solver) to a few thousand equations due to prohibitive calculation time. The AEx ExODE tackles this lock via 1) the introduction of new numerical methods that will take advantage of the mixed precision that mixes several floating number precisions within numerical methods, 2) the adaptation of these new methods for next generation highly hierarchical and heterogeneous computers composed of a large number of CPUs and GPUs. For the past year, a new approach to Deep Learning has been proposed to replace the Recurrent Neural Network (RNN) with ODE systems. The numerical and parallel methods of ExODE will be evaluated and adapted in this framework in order to improve the performance and accuracy of these new approaches.

After presenting the Inria Exploratory Action ExODE, we will present our early results and collaboration opportunities.

JLESC topic:

Short Talks on AI/MD/DL / 30

Malicious Peers Detection in Federated Learning

Author: Cedric PRIGENT[1]

[1] *INRIA*

Federated Learning (FL) is proposed as a solution to collaboratively learn a shared model in massively distributed environments without sharing private data of the participating parties.
While taking advantage of edge resources to compute model updates from a massive number of clients, it may lead to security risks.
Selected clients for a training round get access to the global model in order to update it with their local data.
However, such access to the global model is an entry for potential poisoning attacks.
Malicious clients that are sampled in a given round can manipulate the weights of the model before sending them back to the server.
In this work, an approach for discarding malicious updates in Federated Learning is proposed.
This approach is leveraging auto-encoders to generate synthetic data that are used to evaluate client updates.

JLESC topic:

Short Talks on Advanced Architectures / 31

Controlling the Energy Efficiency of HPC Nodes - A Reinforcement Learning Based Approach

Authors: Akhilesh Raj[1]; Swann Perarnau[2]

[1] *Student Researcher at Argonne National Lab*

[2] *Argonne National Laboratory*

Exascale systems draw a significant amount of power. As each application deployed map to the various heterogeneous computing elements of these platforms, managing how power is distributed across components becomes a priority. The ECP Argo project is developing an infrastructure for node-local control loops that can observe application behavior and adjust resources dynamically, power included, for better performance. We have recently developed a control loop using reinforcement learning, with a proximal

policy optimization algorithm, trained on an existing mathematical model of application progress response to power capping. This dependency on the mathematical model is a hindrance: progress/instantaneous performance is stochastic (noisy) under a dynamic workload and therefore a good approximation model demands more data, and lengthy characterization studies. Therefore, we are exploring methods for bypassing this mathematical model, like actor-critic methods, and are looking for collaborations with know-how on other options, for example: real-time training, existing fully characterized applications, alternative control loop designs.

JLESC topic:

Short Talks on Numerical Methods / 32

Beyond 2D block-cyclic: extended patterns for distributed linear algebra

Authors: Olivier Beaumont[1]; Lionel Eyraud-Dubois[1]; Julien Langou[2]; Mathieu Vérité[1]

[1] *Inria*

[2] *UC Denver*

The 2D block-cyclic pattern is a well-known solution to distribute the data of a matrix among homogenous nodes. Its ease of implementation and good performance makes it widely used.

With the increased popularity and efficiency of task-based distributed runtime systems, it becomes feasible to consider more exotic patterns. We have recently proposed improvements in two different contexts:
1. For symmetric operations, there exist patterns that take advantage of the symmetry of the matrix to reduce the communication volume.
2. When the number of nodes P cannot be expressed as $P = p \times q$ with close values of p and q, we can find patterns that use all the nodes while keeping optimal load balancing and low communication volume.

For each context, we showed that using an exotic pattern with an efficient runtime system yields increased performance. We believe these ideas can be explored further, for example: to improve the scheduling of communication operations, to extend to other operations, to consider non-homogeneous nodes, ...

JLESC topic:

Short Talks on Distributed Resources / 33

Optimize heterogenous storage resources use on HPC systems with simulations

Author: Julien Monniot[1]

[1] *INRIA*

Large-scale infrastructures are increasingly required to store and retrieve massive amounts of data in order to execute scientific applications at scale. The severe need for I/O performance is now often handled by new intermediate tiers of storage resources, deployed throughout HPC systems (node-local storage, burst-buffers, ...) and backed by more and more specialized hardware (NVRAM, NVMe, ...). Unfortunately, these costly resources are vastly heterogeneous and require advanced techniques to be correctly allocated and sized, otherwise risking to be underutilized. In an effort to help mitigate such issues, we recently presented StorAlloc, a simulator used as a testbed for assessing

storage-aware job scheduling algorithms and evaluating various storage infrastructures. Achieving the main goal behind StorAlloc – allocating HPC storage in a similar way as compute resources – now requires to extend on this initial work. To do so, we turn to state of the art simulation frameworks such as WRENCH and Simgrid to further develop the ideas presented in StorAlloc.

JLESC topic:

Project Talks on AI/ML/DL / 34

Architecture and Hyperparameter Search for Super-Resolution Networks Operating on Medical Images

Authors: Xin Liu[1]; Mario Ruettgers[2]; Romain Egele[3]; Marcel Aach[4]; Prasanna Balaprakash[5]; Andreas Lintermann[6]

[1] *Juelich Supercomputing Centre, Germany*

[2] *Juelich Supercomputing Centre, Germany; Institute of Aerodynamics, RWTH Aachen University, Germany*

[3] *Argonne National Laboratory, USA; Universit´e Paris-Saclay, France*

[4] *Juelich Supercomputing Centre, Germany; University of Iceland, Iceland*

[5] *Argonne National Laboratory, USA*

[6] *Jülich Supercomputing Centre (JSC, FZ Jülich), Jülich Aachen Research Alliance - Center for Simulation and Data Science (JARA-CSD)*

Super-resolution networks (SRNs) are employed for enhancing the resolution of Computer Tomography (CT) images. In previous works of the JSC group, respiratory flow simulations were integrated into a data processing pipeline to facilitate diagnosis and treatment planning in rhinology [1]. However, obtaining accurate simulation results is often hindered by low CT image resolutions in clinical applications. SRNs have the potential to increase the CT image resolution, from which computational meshes are generated and used for simulations. The baseline SRN for the project has a U-net architecture with residual learning blocks and is trained with fine CT images as ground truth and down-sampled coarse CT images as input. The performance of the SRN is validated by comparing Computational Fluid Dynamics (CFD) simulations results based on its predictions, fine, coarse, and interpolated CT data of three test patients. The pressure loss between the inflow regions (nostrils) and the outlet (pharynx) of the simulations based on the SRN's predictions deviate by only 1.6%, 0.9%, and -0.3% from the case with fine CT data, compared to deviations of -8.5%, -8.7%, and 10.8% for coarse CT data, and -20.5%, -85.0%, and -0.5% for interpolated CT data.

The collaboration between Juelich Supercomputing Centre (JSC) and Argonne National Laboratory (ANL) focuses on SRN optimization and uncertainty quantification using DeepHyper [3] and AutoDEUQ [2], which are frameworks developed at ANL. Finding the optimal architectures and hyperparameters is limited by computational resources as the search space is often too large to explore exhaustively. DeepHyper tackles the challenge by employing an asynchronous Bayesian optimization approach at HPC scale. The SRN of the previously mentioned baseline case will be further optimized with DeepHyper, and the performance, scalability, and accuracy of DeepHyper will be analyzed and juxtaposed to similar tools, such as Ray Tune [4]. Best-practice for using DeepHyper will be collected and shared among other users and it eventually will be deployed as a standard module on JSC's HPC systems. The findings of the current project will further help to increase the number of CT recordings that are usable for flow simulations, and therefore help to improve CFD-based diagnoses and treatments of pathologies in the human respiratory system.

References

[1] Rüttgers, M., Waldmann, M., Schröder, W., & Lintermann, A. A machine-learning-based method for automatizing lattice-Boltzmann simulations of respiratory flows. Applied Intelligence (2022), 1-21.

[2] Egele, R., Maulik, R., Raghavan, K., Balaprakash, P., & Lusch, B. AutoDEUQ: Automated Deep Ensemble with Uncertainty Quantification. In 2022 International Conference on Pattern Recognition (ICPR).

[3] Balaprakash, P., Salim, M., Uram, T. D., Vishwanath, V., & Wild, S. M. DeepHyper: Asynchronous hyperparameter search for deep neural networks. In 2018 IEEE 25th international conference on high performance computing (HiPC), pp. 42-51. IEEE, 2018.
[4] Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., & Stoica, I. Tune: A research platform for distributed model selection and training. In 2018 International Conference on Machine Learning (ICML) AutoML workshop

JLESC topic:

Short Talks on Advanced Architectures / 35

KheOps: A Collaborative Environment for the Cost-effective Reproducibility of Edge-to-Cloud Experiments

Authors: Daniel Rosendo[1]; Katarzyna Keahey[2]; Alexandru Costan[1]; Gabriel Antoniu[1]

[1] *INRIA*

[2] *ANL*

Distributed digital infrastructures for computation and analytics are now evolving towards an interconnected ecosystem allowing complex scientific workflows to be executed from IoT Edge devices to the HPC Cloud (aka the Computing Continuum). Understanding and optimizing the performance of workflows in such a complex Edge-to-Cloud Continuum is challenging. This breaks down to reconciling many, typically contradicting application requirements and constraints with low-level infrastructure design choices. One important challenge is to accurately reproduce the relevant behaviors of a given application workflow and representative settings of the physical infrastructure underlying this complex continuum.

Based on the limitations of the main state-of-the-art approaches like Google Colab, Kaggle, and Code Ocean, we propose KheOps, a collaborative environment for the cost-effective reproducibility and replicability of Edge-to-Cloud experiments. KheOps is composed of three core elements to enable reproducible Computing Continuum research: (1) Trovi portal: for sharing experiment artifacts; (2) Jupyter environment: for packaging code, data, environment, and results; and (3) Multi-platform experiment methodology: for abstracting all the complexities to deploy workflows on large-scale scientific testbeds with heterogeneous resources, such as Grid5000, Chameleon, FIT IoT lab, and CHI@Edge.

We illustrate KheOps with a real-life Edge-to-Cloud experiment workflow. Evaluations explore the point of view of the authors of an article (They want to make their experiments reproducible), as well as readers of an article (They want to replicate the article experiments). Results show that KheOps has proven useful for guiding authors and readers to reproduce and replicate Edge-to-Cloud experiments on large-scale scientific platforms.

JLESC topic:

Short Talks on Applications / 36

Extending the COMET component model to support hierarchical composite data: Aevol case study

Authors: Jerry Lacmou Zeutouo[1]; Christian PEREZ[2]; Thierry Gauthier[3]; Jonathan Rouzaud-Cornabas[4]

[1] *Research Engineer at INRIA Lyon*

[2] *Avalon Team Leader, INRIA Lyon*

[3] *Inria / LIP*

[4] *Inria / LIRIS*

Numerical simulation is a key technology for many application domains. It is nowadays considered the third pillar of sciences (with experiment and theory) and is critical to gain a competitive position. Thanks to the democratization of high-performance computers (HPC), complex physics, molecular biology, and more generally complex systems can now be routinely simulated. Aevol (http://aevol.fr) is an example of such a simulator. It consists in simulating millions of generations of an evolving population of micro-organisms. Each generation is made up of a set of populations. For each individual, the model simulates how it evolves through stochastic selection and mutations, which consist in randomly modifying its structures. A simulation is characterized by a Petri dish of bacteria (modeled as a two-dimensional array), where each bacterium has cyclic DNA (modeled as an array of characters) with a few thousand to millions of base pairs.

One problem with Aevol, as with many HPC applications, is that it mixes functional parts of interest to bioinformaticians with HPC concerns. As a result, it is very difficult for bioinformaticians to adapt the code to new use cases without the help of HPC experts. COMET, developed by the AVALON Inria team, is a component-based programming model for HPC applications that aims to address this problem. From a parallel dataflow description of the application, it generates the OpenMP code that performs the tasks at runtime, reducing the need for user expertise in HPC. However, the current version of the COMET model is not expressive enough to fully support Aevol. This talk will present current efforts to define and support hierarchical composite data types that will enable the composition of parallel Aevol code that manipulates Petri dishes and DNA. We will present Aevol and its requirements, the envisioned evolutions for COMET with respect to hierarchical composite data types, as well as preliminary results.

JLESC topic:

Short Talks on Tasking / **37**

A task-based data-flow model for distributed and heterogeneous applications

Author: KEVIN SALA[1]

[1] *BARCELONA SUPERCOMPUTING CENTER (BSC)*

Applications traditionally leverage MPI to run efficiently on HPC systems and scale up to thousands of processors. Since one decade ago, developers have also been adapting their applications to heterogeneous systems by offloading the most time-consuming computation kernels to the available GPUs. To achieve optimal performance in such applications, developers must use the non-blocking and asynchronous services provided by the MPI and GPU-offload APIs; otherwise, the application threads would waste CPU host resources waiting on the synchronous completion. But managing the asynchronicity of communications and GPU-offloading from the application is challenging, tedious, and repetitive among applications. Furthermore, overlapping computation with communication or GPU operations is even harder.

For this purpose, we present a data-flow model that allows distributed and heterogeneous applications to easily benefit from asynchronous communications and GPU-offloading operations, so they avoid dealing with low-level details, such as progress and completion checks. The idea consists of taskifying the application with standard OpenMP tasks: the computations, the asynchronous communications, and the asynchronous GPU-related operations. The tasks declare data dependencies on the data buffers they read/write to define their execution order constraints. This way, computations naturally overlap with communications and GPU operations. We provide two libraries named Task-Aware MPI and Task-Aware CUDA (or other GPU vendors) that define task-aware asynchronous services and transparently handle all those details mentioned earlier.

Our data-flow model has already shown significant benefits at both performance and programmability levels on multiple benchmarks. This short talk aims at finding collaboration opportunities for porting real-world applications or mini-applications to this model.

JLESC topic:

Project Talks on further topics / **38**

Designing Flash-X, a Multiphysics Application for Exascale and Beyond

Author: Anshu Dubey[1]

Co-authors: Jared O'Neal [1]; Mohamed Wahib

[1] *Argonne National Laboratory*

Computing at large scales has become extremely challenging due to increasing heterogeneity in both hardware and software. A positive feedback loop of more scientific insight leading to more complex solvers which in turn need more computational resources has been a continuous driver for development of more powerful platforms. The field of computer architecture is poised for more radical changes in how future platforms are likely to be designed, especially because scientific workflows themselves are growing more complex and diverse. We have enhanced Flash-X, a multiphysics community software, to be able to cope with heterogeneity in and diversity across platform architectures. In this presentation we will distill our experience for achieving performance portability, including its design features, with an emphasis on tools that were developed in collaboration with Riken.

JLESC topic:

Numerical Methods and Algorithms

Project Talks on further topics / **39**

Project Talk on Compression for Instruments

Authors: Kento Sato[1]; Robert Underwood[2]

[1] *Riken*

[2] *Argonne National Laboratory*

This talk will highlight recent updates in the collaboration for streaming data compression for instruments between Argonne National Laboratory and Riken R-CCS. Since the last JLESC, we've shared our compression approaches between organizations, and attempted to use each other's compression approaches. We share our findings, lessons learned, and other progress.

JLESC topic:

Poster Session / **40**

libyt: a Tool for Parallel In Situ Analysis with yt

Authors: Shin-Rong Tsai[1]; Hsi-Yu Schive[2]; Matthew Turk[1]

[1] *University of Illinois at Urbana-Champaign*

[2] *National Taiwan University*

Aims

libyt provides researchers a way to analyze and visualize data using yt (a Python package for analyzing and visualizing volumetric data) or any other Python packages during simulations runtime. User can either use a Python script or enter Python statements to analyze the ongoing data in simulations and get feedbacks instantly. This improves disk usage efficiency and makes analyzing large-scale simulation feasible.

Methods

Connecting Python and Simulation

- Using Python C API and NumPy C API, libyt provides an interface for exchanging data between simulations and Python instance.

In Situ Analysis Under MPI

- Each MPI process contains one piece of simulation code and one Python instance. When launching *N* MPI processes, there will be a total of *N* Python instances working together to conduct in situ analysis. Since yt supports MPI parallelism feature, libyt use it directly.
- In in situ analysis, data are distributed in different processes. We use one-sided communication in MPI, also known as Remote Memory Access (RMA) for data exchange process during in situ analysis.

Applications

Analyzing Fuzzy Dark Matter Vortices Simulation using GAMER + libyt

We use GAMER to simulate the evolution of vortices form from density voids in a Fuzzy Dark Matter halo. In order to investigate the dynamics of these vortices, a very high spatial and temporal resolution in simulation is required.

Each simulation snapshot takes 116 GB, and a total of 321 simulation snapshots are required to capture their evolution clearly. Roughly 37 TB disk space is needed if we were to do this post-processingly. libyt provides a promising approach by using yt function covering_grid to extract our region of interest, which now consumes only 8 GB. It is 15 times smaller than that in post-processing per simulation step.

Analyzing Core-Collapse Supernova Simulation using GAMER + libyt

We use GAMER to simulate core-collapse supernova explosions. libyt facilitates closely monitoring the simulation progress during runtime. We use yt function SlicePlot during in situ analysis to plot the gas entropy distribution. Since entropy is not envolved in simulation's iterative process, these data will only be generated through simulation provided function only when they are needed in in situ analysis. libyt handles data transition between simulation and Python.

Discussion and Conclusion

- libyt is an open source software.
- libyt provides a promising method to analyze and visualize data in parallel during simulation runtime with minimal memory overhead and slightly faster than post-processing under the same environment.
- libyt focuses on using yt as its core analytic method, even though we can call arbitrary Python modules using libyt.

JLESC topic:

C++ library, Embedded Python

Project Talks on I/O, Storage and Workflows / **41**

Update on CI-HPC Project: Github2Gitlab-Integration, SSH-Gitlab-Runner

Author: Jakob Fritz[1]

Co-authors: Ivo Kabadshow [2]; Robert Speck

[1] *FJZ, JSC*

[2] *Juelich Supercomputing Centre*

This project-talk shall give an update on the current status of the CI-HPC project within JLESC. In the last JLESC-meeting some issues and aspects of CI-HPC were raised, that have been taken care of. Two shall be presented here.

First, an approach to combine best of both worlds from GitHub and GitLab: The large community and visibility of GitHub with the rich feature set that is available in the CI of Gitlab. This is especially relevant if there are self-hosted Gitlab-Instances with access to computing infrastructure, that is not reachable from the outside otherwise.

Another aspect was how to simplify the setup of Gitlab-Runners. As the architecture of HPC-Systems changes, the testing of the code that runs on those machines should also change. Therefore, it is important to execute developed code on machines with the same characteristics (e.g. architecture). But not all architectures offer the possibility to run docker-containers. For those cases SSH-Executors can be used in Gitlab-Runners to execute automated tests on remote machines. The talk will introduce a way to also setup this kind of runner easily. This makes it possible to run CI-Jobs on much more machines and on more architectures.

JLESC topic:

CI-HPC

Short Talks on Workflows, I/O and Frameworks / **42**

Optimizing iterative applications using a data-flow programming model

Author: DAVID ALVAREZ ROBERT[1]

[1] *Barcelona Supercomputing Center*

Many HPC applications display iterative patterns, where a series of computations and communications are repeated a specific number of times. This pattern happens, for example, in multi-step simulations, iterative mathematical methods and machine learning training. When these applications are coded using data-flow programming models, much time is spent creating tasks and processing dependencies which are then repeated in regular patterns. This may cause scalability problems or excessive overheads when using very small tasks.

To tackle this issue, we present a new construct for the OmpSs-2 programming model, the *taskiter*, allowing users to annotate loops where the same task DAG is generated for every iteration. This task DAG is then processed and transformed into a cyclic format, allowing the reuse of the first iteration's tasks and dependencies, drastically reducing runtime overhead for successive iterations. Moreover, this cyclic transformation considers the fine-grained dependencies between tasks from each iteration, allowing the overlapped execution of tasks from multiple iterations.

The taskiter construct has already obtained significant performance benefits on OmpSs-2 benchmarks, especially when combined with the runtime's optimized scheduling features. This short talk aims to find collaboration opportunities to port real-world or proxy applications that can benefit from the low-overhead OmpSs-2 dataflow model.

JLESC topic:

Project Talks on AI/ML/DL / 43

Generating Efficient Neural Networks for Protein Diffraction Data

Authors: Georgia Wexler Channing[1]; Ria Patel[1]; Ariel Rorabaugh[1]; Silvina Caino-Lores[1]; Catherine Schumann[1]; Florence Tama[2]; Osamu Miyashita[2]; Michela Taufer[1]

[1] *University of Tennessee, Knoxville*

[2] *R-CCS*

Corresponding Author: gchannin@vols.utk.edu

Proteins and other biological molecules are responsible for many vital cellular functions, such as transport, signaling, or catalysis, and dysfunction can result in diseases. Information on the 3-dimensional (3D) structures of biological molecules and their dynamics is essential to understand mechanisms of their functions, leading to medicinal applications such as drug design. Different proteins have different structures; proteins in the same family share similar substructures and thus may share similar functions. Additionally, one protein may exhibit several structural states, also named conformations. X-ray Free Electron Laser (XFEL) beams are used to create diffraction patterns (images) that can reveal protein structure and function. The translation from diffraction patterns in the XFEL images to protein structures and functionalities is nontrivial.
We present A4NN (analytics for neural networks) applied to protein structure identification. In our previous talk, we reviewed our framework XPSI (XFEL-based Protein Structure Identifier). XPSI combines DL (autoencoder) and ML (kNN) to capture key information that allows the identification of structural properties, such as spatial orientation, protein conformation, and protein type from the diffraction patterns. In this talk, we will discuss improvements to protein structure identification with neural networks and neural architecture search. We will show improvements in accuracy, efficiency, and accessibility. In particular, we will demonstrate how the NSGA-Net workflow increases access to machine learning for domain scientists. We will also deliver a Jupyter Notebook.
As next steps, we are working on 1) testing the framework with additional neural architecture search workflows; and 2) understanding the qualities of successful neural architectures for classification and regression problems.
This project is collaborative research between RIKEN, GCLab, and ICL.

JLESC topic:

Short Talks on Applications / 44

Productive Large Scale QM Calculations

Authors: William Dawson[1]; Luigi Genovese[2]; Louis Beal[2]; Takahito Nakajima[1]

[1] *RIKEN R-CCS*

[2] *CEA Grenoble*

Density Functional Theory (DFT) is a popular Quantum Mechanical framework for computing the properties of molecules and materials. Recent advances in linear-scaling algorithms and computing power have made it possible to apply DFT to systems of an unprecedented size. This has significant consequences for the research paradigms employed by DFT users. In this talk, we will precent our research on practical calculations of large systems. In particular, we will give an overview of our high-level Python interface that is able to construct complex systems, launch calculations on remote supercomputers, and decompose complex systems into core building blocks. We hope that this work may stimulate some discussion about applications of large-scale QM modelling as well as general calculation frameworks for managing calculations across multiple computing resources.

Ratcliff, Laura E., William Dawson, Giuseppe Fisicaro, Damien Caliste, Stephan Mohr, Augustin Degomme, Brice Videau et al. "Flexibilities of wavelets as a computational basis set for large-scale electronic structure calculations." The Journal of chemical physics 152, no. 19 (2020): 194110.

JLESC topic:

Project Talks on further topics / 45

Scalable GPU-Accelerated Incremental Checkpointing of Sparsely Updated Data

Authors: Nigel Phillip Tan[1]; Bogdan Nicolae[2]; Jakob Luettgau[1]; Sanjukta Bhowmick[3]; Keita Teranishi[4]; Nicolas Morales[5]; Michela Taufer[6]; Franck Cappello[2]

[1] *University of Tennessee Knoxville*

[2] *Argonne National Laboratory*

[3] *University of North Texas*

[4] *Oak Ridge National Laboratory*

[5] *Sandia National Laboratories*

[6] *University of Tennessee*

Checkpointing large amounts of related data concurrently to stable storage is a common I/O pattern of many HPC applications in a variety of scenarios: checkpoint-restart fault tolerance, coupled workflows that combine simulations with analytics, adjoint computations, etc. This pattern is challenging because it needs to happen frequently and typically leads to I/O bottlenecks that negatively impact the performance and scalability of the applications.
Furthermore, checkpoint sizes are continuously increasing and overwhelm the capacity of the storage stack, prompting the need for data reduction. A large class of applications including graph algorithms such as graph alignment, perform sparse updates to large data structures between checkpoints. In this case, incremental checkpointing approaches that save only the differences from one checkpoint to another can dramatically reduce the checkpoint sizes, which reduces both the I/O bottlenecks and the storage capacity utilization. However, such techniques are not without challenges: it is non-trivial to transparently determine what data changed since a previous checkpoint and to assemble the differences in a compact fashion that does not result in excessive metadata. State-of-art deduplication techniques have limited support to address these challenges for modern applications that manipulate data structures directly on GPUs. Our approach builds a compact representation of the differences between checkpoints using Merkle-tree-inspired data structures optimized for parallel construction and manipulation.

Our previous talk introduced the project and focused on the challenge of making efficient incremental checkpoints on GPU-accelerated platforms. We presented our compact representation for representing incremental checkpoints. Our algorithm was implemented and initial testing was done with ORANGES, a graph alignment application with sparse update patterns.

For this project update, we have optimized and refactored our implementation and compared the performance of the following approaches.
Full Checkpoint: Copy all data from the GPU to the Host
Basic Incremental Checkpoint: Break data into chunks and save the chunks that have changed since the previous checkpoint
List Incremental Checkpoint: Identify and save a single copy of each new chunk along with a list of shifted duplicate chunks
Our approach: Expand on the List approach by storing shifted duplicates in a compact tree representation
We have analyzed the degree of deduplication for the checkpoint along with the runtime overhead for creating and saving the checkpoint to the Host. We have also examined various tradeoffs that affect checkpoint size and deduplication performance.

Our next steps are to compare performance with compression techniques, evaluate different applications or access patterns, and examine alternative hash functions. Locality-sensitive hash functions in particular are useful for lossy deduplication for floating-point data.

JLESC topic:

Resilience and fault tolerance

Short Talks on Workflows, I/O and Frameworks / **46**

On the Impact of Improving Runtime Estimates in HPC

Authors: Robin bozennec[1]; Guillaume Pallez[2]

[1] *INRIA*

[2] *Inria*

One of the information that HPC batch schedulers use to schedule jobs on the available resources is user runtime estimates: an estimation provide by the user of how long their job will run on the machine. These estimates are known to be inaccurate, hence many work have focused on improving runtime prediction.

In this work, we start by discussing bias and limitations of the most used optimization metrics and provide elements on how to evaluate performance when studying HPC batch scheduling,

Then we study qualitatively the impact of improving runtime estimates on these various optimization criteria.

JLESC topic:

scheduling in HPC

Poster Session / **47**

Monitoring mesoscale convection simulations with nekRS using JuMonC at Scale

Author: Christian Witzler[None]

Co-author: Mathis Bode [1]

[1] *Forschungszentrum Jülich GmbH*

With the increasing size and complexity of simulations, the need for interactions rises. JuMonC is a user controlled application, that runs parallel to the simulation and offers a REST-API for system monitoring, and is expandable through plugins to allow simulation monitoring and steering as well. This information can then be used multiple ways, for example to be displayed in Jupyter notebooks and dashboards.

In this case we are using it with the GPU enabled Navier Stokes solver nekRS, that is based on the spectral element method to run horizontally extended turbulent convection simulations. This so called mesoscale convection is particularly challenging in the case of stellar convection, because there are no comparable conditions on earth and increasing resolution requirements hinder Direct Numerical Simulations (DNS) as well. Scaling with and without JuMonC on JUWELS Booster has been studied at scale and will be discussed.

JLESC topic:

Poster Session / 48

Memory Visualization for Task-Based GEMM in PaRSEC

Authors: Daniel Mishler[1]; George Bosilca[None]; Thomas Herault[2]

[1] *Innovative Computing Laboratory*

[2] *UTK*

Tracing a task-based application is necessary to get an idea of what's going on, but a heavy-handed utility could be so expensive that the trace might tell a story that does not look close to what the hardware is really doing when the trace is off. Using DPLASMA and PaRSEC, we demonstrate with GEMM some of the memory patterns on local ICL machines, which result from code written to visualize PaRSEC's native tracing abilities. We provide graphs along with takeaways regarding what this knowledge of memory might gain developers seeking to maximize performance on a task-based application.

JLESC topic:

Poster Session / 49

Steering Large Scale Ensemble Simulations for Online DNN Training with Adaptive Sampling

Author: Sofya Dymchenko[1]

Co-author: Bruno Raffin [2]

[1] *INRIA and UGA*

[2] *Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000 Grenoble, France*

Simulation-based training of deep neural networks (DNN), such as surrogates and inference models, is technically challenging and expensive both memory- and computational-wise.

Large-scale deep learning applications for sciences (fluid dynamics, climate prediction, molecular structure exploration) demand novel approaches. One of them is online training, where the simulations are generated during the training process and used as soon as they are available. It benefits from (1) file-free processing and (2) ensemble steering. The first (1) overcomes the I/O bottleneck

and enables the generation of large datasets that couldn't be stored on disk. For example, in the context of sensitivity analysis, Melissa framework's [1] largest experiment processed 270 TB of data online. The goal of the second (2) is to accelerate the training process and improve data efficiency. By monitoring the training state, it controls the parameterization of the next set of simulations to run.

We investigate strategies for adaptive simulation sampling for DNN train data, which range from Bayesian Optimal Experimental Design (BOED) and Simulation-Based Inference (SBI) to reinforcement learning.

[1] T. Terraz, A. Ribes, Y. Fournier, B. Iooss, and B. Raffin. Melissa: large scale in transit sensitivity analysis avoiding intermediate files. In Proceedings of the international conference for high performance computing, networking, storage and analysis, pages 1–14, 2017.

JLESC topic:

Short Talks on Numerical Methods / 50

Parallel Scalable Domain Decomposition Methods in Pharmaco-Mechanical Fluid-Structure Interaction

Author: Lea Sassmannshausen[None]

Co-authors: Alexander Heinlein [1]; Axel Klawonn [2]

[1] *Delft Institute of Applied Mathematics*

[2] *Department of Mathematics and Computer Science*

Abstract

Today, cardiovascular diseases are among the leading causes of death worldwide.
With a special focus on the treatment of hypertension and the clinical consequences, thereof, the computational modeling of fluid-structure interaction with pharmaco-mechanical effects becomes vastly relevant.
Therefore, the state of the art of fluid-structure interaction is extended to reflect the influence of drugs on the structural properties of arterial walls leading to a fully coupled fluid-structure-chemical interaction, denoted as FSCI.
Highly-scalable parallel GDSW (Generalized Dryja–Smith–Wildund) preconditioners have been implemented in the solver framework FROSch (Fast and Robust Overlapping Schwarz), which is part of the software library Trilinos and can easily be applied to the geometry and structure blocks in an FSI simulation framework.
Furthermore, these methods have also been extended to monolithic GDSW-type preconditioners for fluid flow problems; the parallel implementation is also available in FROSch.
Ultimately, we plan to solve the resulting FSCI linearized system with a Krylov method preconditioned by the FaCSI preconditioner which was introduced by Deparis, Forti, Grandperrin, and Quarteroni in 2016. The inverses appearing in FaCSI will be approximated by GDSW-type overlapping Schwarz
preconditioners.

In this short talk, some first results and a brief summary of the methodology will be presented based on our software FEDDLib (Finite Element and Domain Decomposition Library) and Schwarz preconditioners from the Trilinos package FROSch.

JLESC topic:

Short Talks on Interative Tools and Monitoring / 51

Supercomputing in the Browser - Web-based interactive HPC-Access at JSC

Author: Jens Henrik Goebbert[1]

Co-authors: Alice Grosch [1]; Bernd Schuller [1]; Tim Kreuzer [1]

[1] *Forschungszentrum Jülich, Jülich Supercomputing Centre*

Interactive exploration and analysis of large amounts of data from scientific simulations, in-situ visualization and application control are convincing scenarios for explorative sciences. It is the task of High-Performance Computing (HPC) Centers to enable, support and, of course, simplify these workflows of our users of today's supercomputers. Especially technical work simplifications in dealing with HPC systems are of great importance to counteract the increasing complexity and requirements and to open up new application areas for HPC.

Based on the open source software JupyterLab, a way has been available for some time that combines interactive with reproducible computing and at the same time overcomes the challenges of supporting a wide variety of workflows. At the Jülich Supercomputing Center, users have interactive access to the HPC- and cloud resources via a pure browser-based web access based on JupyterHub + JupyterLab at https://jupyter-jsc.fz-juelich.de

The platform has been designed from the beginning so that future services/tools can be easily extended by both the HPC Center and the users themselves. New HPC systems and Kubernetes clusters can be added quickly and easily, and the service is designed to integrate in projects- and communities web-sites and -platforms. With the upstream secure authentication and authorization and the direct access to our compute resources (cloud and HPC, login- and also compute nodes), the possibilities to couple HPC with cloud techniques are given and used.

With hundreds of sessions per week Jupyter-JSC has been well received by users from different scientific domains and we are continuously working on improvements and new application areas. In this talk, we will introduce and show the technology behind the Jupyter-JSC web service, show the HPC-specific features and potential and venture a look into the future together in discussion.

JLESC topic:

interactive hpc, service, jupyterlab

Short Talks on Workflows, I/O and Frameworks / **52**

CI in HPC: Working hard or hardly working?

Authors: Ivo Kabadshow[1]; Jakob Fritz[2]

[1] *Juelich Supercomputing Centre*

[2] *FZJ, JSC*

The growing complexity arsing in the development of HPC libraries and applications impedes speedy code development. To reel in this complexity, CI tools and workflows are a great way to automate large portions of test-driven development cycles.
In this short-talk we want to present the current impact of our CI-HPC tools to automate such workflows. Our FMM library FMsolvr will be used as a demonstrator to show - if started from scratch - how easy or hard such a setup can be.

JLESC topic:

RSE at work, Practical part of developed CI-HPC toolchain

Break-out Session: Heterogeneous and reconfigurable architectures for the future of computing / 53

Heterogeneous and reconfigurable architectures for the future of computing

Authors: Kazutomo Yoshii[1]; Kentaro Sano[2]; Xavier Martorell[3]

[1] *Argonne National Laboratory*

[2] *RIKEN*

[3] *BSC*

The end of Moore's law encourages us to challenge new approaches for the future of computing. One of the promising approaches is heterogeneous architecture with reconfigurable devices such as field-programmable gate arrays and coarse-grain reconfigurable architecture, which leverages hardware specialization and dataflow computing. In this break-out session, we will discuss subjects and opportunities related to specified hardware co-design, emerging accelerators/architectures, and programming paradigms with talks on recent research activities. We also plan to exchange research seeds between attendees and discuss the need for adjustment in the scope and direction of our JLESC collaboration.

JLESC topic:

Short Talks on Tasking / 54

Using coroutines in a task-based runtime system

Author: Joseph Schuchart[1]

Co-authors: George Bosilca [1]; Thomas Herault [1]

[1] *University of Tennessee, Knoxville*

This talk will focus on the design of device support in the Template Task Graph. Specifically, TTG employs C++ coroutines to suspend tasks during times of data motion and kernel execution. This design allows TTG to support transparent device memory oversubscription by delegating memory management to the underlying PaRSEC runtime system. TTG will also offer coroutines as a means for describing successor tasks. Open questions of this talk are on the general use and acceptance of coroutines and the treatment of memory oversubscription in task-based runtime systems.

JLESC topic:

Short Talks on AI/MD/DL / 55

Workflows for AI Model Curation and Comparison

Author: Justin Wozniak[1]

[1] *University of Chicago*

The new IMPROVE project at ANL is collecting and curating AI models for cancer and similar precision medicine problems. Comparing these models across a large configuration space of hyperparameters and data sets is a challenging problem. The IMPROVE team is building a scalable workflow

suite to answer a range of questions that arise when attempting to run diverse models developed by different teams on the same problem. This presentation will describe the problem in more detail and our approach using near-exascale or exascale computers.

JLESC topic:

Short Talks on Advanced Architectures / 56

Streaming hardware compressor co-design using the Chisel hardware construction language

Author: Kazutomo Yoshii[1]

[1] *Argonne National Laboratory*

Data compression is becoming a major topic in HPC, remote sensors, and scientific instrument communities. As a result, various software compression software has been developed. In addition, there is a huge interest in applying scientific data compression for real-time and streaming processing scenarios. However, software-only implementations may be challenging or impossible to meet the real-time streaming requirements. Therefore, we are seeking paths to hardware-only or software-hardware hybrid implementation. The opportunities are to study optimal hardware implementation strategies for scientific data compressors and implement/simulate hardware compressors efficiently in a software-developer-friendly manner. We employ emerging hardware construction language for exploring hardware compressor designs. In this short talk, I will briefly introduce Chisel and summarize the streaming hardware compressor blocks we have been designing.

JLESC topic:

Short Talks on Advanced Architectures / 59

HomE: Enabling Homomorphic Encryption of DL, a (recently started) ERC Consolidator Grant

Author: ANTONIO PENA[1]

[1] *Barcelona Supercomputing Center (BSC)*

Deep learning (DL) is widely used to solve classification problems previously unchallenged, such as face recognition, and presents clear use cases for privacy requirements. Homomorphic encryption (HE) enables operations upon encrypted data, at the expense of vast data size increase. RAM sizes currently limit the use of HE on DL to severely reduced use cases. Recently emerged persistentmemory technology (PMEM) offers larger-than-ever RAM spaces, but its performance is far from that of customary DRAMtechnologies.

This project aims to spark a new class of system architectures for encrypted DL workloads, by eliminating or dramatically reducing data movements across memory/storage hierarchies and network, supported by PMEM technology, overcoming its current severe performance limitations. HomE proposes a holistic approach yielding highly impactful outcomes that include novel comprehensive performance characterisation, innovative optimisations upon current technology, and pioneering hardware proposals.

JLESC topic:

Short Talks on Distributed Resources / 60

Seamless Heterogeneous Memory Management Via The EcoHMEM Methodology

Author: HATEM ELSHAZLY[1]

Co-author: ANTONIO PENA [1]

[1] *Barcelona Supercomputing Center (BSC)*

New memory technologies are an emerging to provide larger RAM sizes at reasonable cost and energy consumption. In addition to the conventional DRAM, recent memory infrastructures contain byte-addressable persistent memory (PMEM) technology that offers capacities higher than DRAM and better access times than Nand-based technologies such as SSDs.

In such hybrid infrastuctures, users have the choice to either manually manage allocations to different memory spaces or delegate such a task to system components where DRAM is used as a cache for PMEM. Nevertheless, recent research showed that both approaches have limitations. From the one hand, hardware-based mechanisms lack flexibility and are not always efficient, yielding inconsistent performance. From the other hand, software-based approaches pose management overheads and often require expert knowledge and intrusive code changes.

Hence, in order to fully take advantage of different capabilities offered by hybrid memory systems and remove the memory management burden from the user, we introduce the EcoHMEM methodology for automatic object-level placement. The methodology of the EcoHMEM frameowrk is able to transparently optimize applications memory allocations without making any modifications to the source codes and without requiring any modifications to the kernel. Compared to state–of–the–art solutions, our methodology can attain up to 2x runtime improvement in mini-benchmarks and up-to 6% improvement in complex production applications.

JLESC topic:

Poster Session / 61

Ginkgo — a High-Performance Portable Numerical Linear Algebra Software

Authors: Hartwig Anzt[1]; Terry Cojean[2]

[1] *ICL, University of Tennessee*

[2] *Karlsruhe Institute of Technology*

Numerical linear algebra building blocks are used in many modern scientific applications codes. Ginkgo is an open-source numerical linear algebra software that is designed around the principles of portability, flexibility, usability, and performance. The Ginkgo library is integrated into the deal.II, MFEM, OpenFOAM, HYTEG, Sundials, XGC, HiOp, and OpenCARP scientific applications, ranging from finite element libraries to CFD, power grid optimization, and heart simulations. The Ginkgo library grew from a math library supporting CPUs and NVIDIA GPUs to an ecosystem that has native support for GPU architectures from NVIDIA, AMD, and Intel, can scale up to hundreds of GPU. One of the keys to this success is the rapid development and availability of new algorithmic functionalities in the Ginkgo library such as, but not limited to, Multigrid preconditioner, advanced mixed-precision iterative solvers and preconditioners, a sparse iterative batched functionality, sparse direct solvers, and the distributed MPI-based backend. In this poster, we will expose Ginkgo's library design, performance results on a wide range of hardware, and integration within key applications.

JLESC topic:

Short Talks on Numerical Methods / 62

Batched Iterative Solvers in Plasma Fusion Simulations

Author: Hartwig Andreas Anzt[1]

[1] *University of Tennessee*

Batched linear solvers, which solve many small related but independent problems, are important in several applications. This is increasingly the case for highly parallel processors such as graphics processing units (GPUs), which need a substantial amount of work to keep them operating efficiently and solving smaller problems one-by-one is not an option. Because of the small size of each problem, the task of coming up with a parallel partitioning scheme and mapping the problem to hardware is not trivial. In recent history, significant attention has been given to batched dense linear algebra. However, there is also an interest in utilizing sparse iterative solvers in a batched form, and this presents further challenges. An example use case is found in a gyrokinetic Particle-In-Cell (PIC) code used for modeling magnetically confined fusion plasma devices. The collision operator has been identified as a bottleneck, and a proxy app has been created for facilitating optimizations and porting to GPUs. The current collision kernel linear solver does not run on the GPU-a major bottleneck. As these matrices are well-conditioned, batched iterative sparse solvers are an attractive option. A batched sparse iterative solver capability has recently been developed in the Ginkgo library. In this paper, we describe how the software architecture can be used to develop an efficient solution for the XGC collision proxy app. Comparisons for the solve times on NVIDIA V100 and A100 GPUs and AMD MI100 GPUs with one dual-socket Intel Xeon Skylake CPU node with 40 OpenMP threads are presented for matrices representative of those required in the collision kernel of XGC. The results suggest that GINKGO's batched sparse iterative solvers are well suited for efficient utilization of the GPU for this problem, and the performance portability of Ginkgo in conjunction with Kokkos (used within XGC as the heterogeneous programming model) allows seamless execution for exascale oriented heterogeneous architectures at the various leadership supercomputing facilities.

JLESC topic:

Short Talks on Distributed Resources / 63

Cloud-Bursting and Autoscaling for Python-Native Scientific and AI Workflows

Authors: Tingkai Liu[1]; Marquita Ellis[2]; Carlos Costa[2]; Claudia Misale[2]; Volodymyr Kindratenko[1]; Sara Kokkila-Schumacher[2]

[1] *University of Illinois at Urbana-Champaign*

[2] *IBM*

We have extended the Ray framework to enable automatic scaling of workloads on high-performance computing (HPC) clusters managed by SLURM© and bursting to a Cloud managed by Kubernetes®. Our implementation allows a single Python-based parallel workload to be run concurrently across an HPC cluster and a Cloud. The Python-level abstraction provided by our solution offers a transparent user experience, requiring minimal adoption of the Ray framework. Applications in Electronic Design Automation and Machine Learning are used to demonstrate the functionality of this solution in scaling the workload on an on-premises HPC system and automatically bursting to a public Cloud when running out of allocated HPC resources. The paper focuses on describing the initial implementation and demonstrating novel functionality of the proposed framework using three applications as well as identifying practical considerations and limitations for using Cloud bursting mode.

JLESC topic:

HPC+Cloud

Short Talks on Applications / 64

Running Native HPC Applications on the Cloud

Authors: Aditya Bhosale[1]; Kavitha Chandrasekar[1]; Pedro Bello-Maldonado[2]; Carlos Costa[2]; Claudia Misale[2]; Sara Kokkila-Schumacher[2]; Laxmikant Kale[1]; Volodymyr Kindratenko[1]

[1] *University of Illinois at Urbana-Champaign*

[2] *IBM*

In this project, we aim to enable Charm++ based HPC applications to run natively on a Kubernetes cloud platform. The Charm++ programming model provides a shrink/expand capability which matches well with the elastic cloud philosophy. We investigate how to enable running Charm++ applications with dynamic scaling of resources on Kubernetes. In order to run Charm++ applications on Kubernetes, we have implemented a Charm operator, very similar to Kubeflow's mpi-operator. The charm operator enables scaling of the number of pods in a job which isn't supported by the mpi operator since typically MPI applications do not support rescaling of resources at runtime. This operator also generates the nodelist in the correct format required by Charm++ programs for rescaling. The Charm++ application is launched in server mode to enable the injection of messages into the scheduler externally which is used to signal rescaling. The Charm operator handles allocation of resources and cleanup for all charm jobs on the Kubernetes cluster. For startup, it creates the launcher and worker pods for all jobs and performs monitoring for any change to a deployment configuration. We are implementing changes in the controller code which allow scaling of pods, i.e. shrinking or expanding the number of pods allocated to a Charm++ job. Currently, we have added support for making shrink/expand updates using the YAML file for the deployment. We use these shrink/expand updates to yaml script for testing our implementation. We are working on two modes for scaling, one where the pods are deleted on shrink and for expand new pods are created. In the second mode, we maintain a pool of worker pods where shrink releases worker pods to the pool of pods and these can be re-used for an expand request by another job in the context of the charm-operator.

JLESC topic:

HPC+Cloud

Short Talks on Distributed Resources / 65

Dynamic resources in MPI

Authors: ANTONIO PENA[1]; SERGIO ISERTE AGUT[2]

[1] *Barcelona Supercomputing Center (BSC)*

[2] *Barcelona Supercomputing Center*

Process malleability and dynamic resources have demonstrated, in several studies, to increase the productivity of HPC facilities, in terms of completed jobs per unit of time. In this regard, changing the number of resources assigned to an application during its execution accelerates global job processing. Furthermore, the users of malleable applications can also benefit from malleability when they are expected to execute large workloads since they will get their results faster.

Nevertheless, malleable applications are rather unusual, and commonly, they do not take part in production workloads. This side effect of malleability is mainly due to the difficulty of adopting malleability in already existent scientific applications, since the state-of-the-art solutions report complex APIs or even, a change of programming model.
In this work, we present the dynamic management of resources library (DMRlib), a malleability solution that poses to users a simple MPI-like syntax and provides support for job reconfiguration, data redistribution, process management, execution resuming, and dynamic resources.

JLESC topic:

Short Talks on Advanced Architectures / 66

DPU Offloading with OpenMP Programming Model

Author: MUHAMMAD USMAN[1]

Co-authors: ANTONIO PENA [2]; SERGIO ISERTE AGUT [1]

[1] *Barcelona Supercomputing Center*

[2] *Barcelona Supercomputing Center (BSC)*

Recent advancements in High-Speed NICs have gained a speed of 400 Gbps and achieved the status of SmartNICs by enabling offloads for cryptography and virtualization. Data Processing Units (DPUs) are taking this development further by integrating performant processing cores on the SmartNIC itself.

The DOCA API for programming BlueField DPUs requires proficiency in network technologies. We are enabling BlueField DPU capabilities in the OpenMP programming model. The enablement of OpenMP Target Offload features for BlueField DPUs will contribute to accessibility to a wider range of users. It will be an opportunity for the HPC community to leverage DPU features for a wider range of existing and emerging applications. We will be demonstrating DPU features by accelerating workload employing domain decomposition by offloading halo exchange operations to BlueField DPUs.

JLESC topic:

OpenMP, BlueField, DPU, In-Network Computing, Offloading

Short Talks on Distributed Resources / 67

Composition of Scheduling and Control-Theory Techniques

Authors: Eric Rutten[1]; Raphael Bleuse[2]

[1] *INRIA*

[2] *Univ. Grenoble Alpes, Inria*

The management and allocation of resources to users in HPC infrastructures often relies on the RJMS.
One key component for an optimized resource allocation, with respect to some objectives, is the scheduler.

Scheduling theory is interesting as it provides algorithms with performance guarantees.
These guarantees come at the cost of tedious and complex modeling effort.
The growing complexity of nowadays and future platforms (hardware heterogeneity, memory/bandwidth/energy constraints)
do push to its limit the scheduling approach.
Taking into account this new challenges either requires the design of new overly complex models, or exhibits the crudeness of the model.

In a sense, the scheduling approach fails to capture the dynamic aspects of the platforms.
From the control theory point of view, scheduling algorithms are open-loop systems: the actual state of the platform is not fed back into the decision process.
By closing the loop and using some control theory results/techniques, we propose to study how to combine both techniques.
This study would take place at various levels: from theory to actual applications.

JLESC topic:

control theory, scheduling

Short Talks on AI/MD/DL / 68

High-Dimensional Performance Modeling via Tensor Completion

Author: Edward Hutter[1]

Co-author: Edgar Solomonik [1]

[1] *University of Illinois at Urbana-Champaign*

Performance tuning, software/hardware co-design, and job scheduling are among the many tasks that rely on models to predict application performance. We propose and evaluate low rank tensor decomposition for modeling application performance. We use tensors to represent regular grids that discretize the input and configuration domain of an application. Application execution times mapped within grid-cells are averaged and represented by tensor elements. We show that low-rank canonical-polyadic (CP) tensor decomposition is effective in approximating these tensors. We then employ tensor completion to optimize a CP decomposition given a sparse set of observed runtimes. We consider alternative piecewise/grid-based (P/G) and supervised learning models for six applications and demonstrate that P/G models are significantly more accurate relative to model size. Among P/G models, CP decomposition of regular grids (CPR) offers higher accuracy and memory-efficiency, faster optimization, and superior extensibility via user-selected loss functions and domain partitioning. CPR models achieve a 2.18x geometric mean decrease in mean prediction error relative to the most accurate alternative models of size $\leq$10 kilobytes.

JLESC topic:

Performance Modeling

Break-out Session: Quantum Computing and HPC / 69

Quantum Computing and HPC

Author: miwako tsuji[1]

[1] *RIKEN R-CCS*

Quantum computing is the computation using the properties of quantum states, and considered to be an important block for the post-Moore Era. This break out session aims to introduce the researches and activities in the quantum computing area from different institutes. Especially, we would like to focus on the hybrid/cooperative computations by the quantum and classical computing.

There will be 3 talks followed by a short discussion session:
Miwako Tsuji and Mitsuhisa Sato (RIKEN)
Dennis Willsch and Madita Willsch (JSC)
Yuri Alexeev (ANL)
discussion

JLESC topic:

Short Talks on AI/MD/DL / 70

Training Deep Surrogate Models with Large Scale Online Learning

Author: Lucas Meyer[1]

Co-authors: Alejandro Ribes [2]; Bruno Raffin [3]; Marc Schouler [1]; Robert Caulk [1]

[1] *INRIA*

[2] *EDF*

[3] *Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000 Grenoble, France*

The spatial and temporal resolution of Partial Differential Equations (PDEs) plays important roles in the mathematical description of the world's physical phenomena. In general, scientists and engineers numerically solve PDEs by the use of computationally demanding solvers. Recently, deep learning algorithms have emerged as a viable alternative for obtaining fast solutions for PDEs. Models are usually trained on synthetic data generated by solvers, stored on disk and read back for training. We propose an online training framework for deep surrogate models implementing several levels of parallelism focused on simultaneously generating numerical simulations and training deep neural networks. This approach suppresses the I/O and storage bottleneck associated with disk loaded datasets, and opens the way to training on significantly larger datasets. The framework leverages HPC resources, traditionally used to accelerate solver executions, to parallelize the data generation with the training. Experiments compare the offline and online training of four surrogate models, including state-of-the-art architectures. Results indicate that exposing deep surrogate models to more dataset diversity, up to hundreds of GB generated on 300 nodes and 1200 cores, can increase model generalization capabilities. Fully connected neural networks, FNO, and Message Passing PDE Solver prediction accuracy is improved by 68%, 16% and 6%, respectively. Work in progress targets larger scale simulations.

JLESC topic:

Scientific applications (AI, deep surrogate models), Novel programming models (online training on data generated in parallel)

Short Talks on Interative Tools and Monitoring / 71

CharmTyles: Large-scale interactive Charm++ with Python

Authors: Aditya Bhosale[1]; Nikunj Gupta[1]; Zane Fink[1]; Laxmikant Kale[1]

[1] *University of Illinois Urbana-Champaign*

Python is emerging as a high-productivity language favored by many application scientists and engineers in simulation/modeling, data analytics, and machine learning. Interactive parallel computing is another related trend, especially for analyzing graphs in addition to the above. The CharmTyles project is aimed at addressing these needs while providing a highly efficient and adaptive parallel runtime.

CharmTyles provides abstractions based on a client-server model with a python frontend running in a Jupyter Notebook on the user's machine and a Charm++ backend server running on a parallel machine. The broad view is that of multiple collections of *tyles*, spread over an elastic parallel machine, either in the cloud, a cluster, or a supercomputer, orchestrated from the frontend and assisted by the Charm++ runtime system in the backend. In this work-in-progress talk, we introduce CharmTyles and describe preliminary implementations of the abstractions it supports: a NumPy-based dense linear algebra library, and a stencil library for structured grid computations. We discuss lazy evaluation and message coalescing, optimizations to reduce the communication cost between the frontend and

backend. We use JIT compilation on the backend to enable standard compiler optimizations such as vectorization and loop fusion.

The broad vision of this project is highly ambitious, and we seek collaborations to fulfill it.

JLESC topic:

Short Talks on Tasking / **72**

Enhancing iteration performance on distributed task-based workflows

Author: ALEX BARCELO[1]

[1] *Barcelona Supercomputing Center*

Task-based programming models have proven to be a robust and versatile way to approach development of applications for distributed environments. The programming model itself feels natural and close to classical algorithms; the task-based distribution of tasks can achieve a high degree of performance. All this is achieved with a minimal impact on programmability. However, execution on this paradigm can be very sensitive to the granularity of tasks –i.e., the block size, or equivalently, the quantity and execution length of tasks. This is manifested during the iteration of the distributed datasets, a procedure that will yield tasks across the distributed computing resources. Identifying and setting this optimal block size is not trivial, requires inner knowledge of the computing environment, and is not an easy task for the domain expert –i.e. the application developer. Having the programming model performance be highly dependent on this block size is undesirable and a challenge to overcome.

Our proposal is to enhance the distributed iterations by including a new mechanism –a procedure that we call *split*. At its core, the *split* mechanism provides a transparent way to get *partitions* (which are logical groups of blocks, obtained without any transfers nor data rearrangement) of blocks. By doing so, performance is improved as the system produces fewer tasks, there is a cutback on the scheduling cost, and the invocation overhead is reduced. Our proposed implementation of the *split* goes one step further and also integrates with the storage framework, thus being able to attain those benefits while guaranteeing data locality.

The evaluation we have conducted shows that *split* mechanism is able to achieve performance improvements of over one order of magnitude. We have chosen different applications covering a wide variety of scenarios; those applications are representatives of a broader set of applications and domains (both memory-intensive and CPU-intensive applications, for applications widely used in Machine Learning, Data Analytics, etc.). The changes required in the source code of a task-based application are minimal, preserving the high programmability of the programming model.

JLESC topic:

Short Talks on Interative Tools and Monitoring / **73**

Data analysis, interactive development, and the Julia Language with HPC Distributed Systems.

Author: Aaron Saxton[1]

[1] *NCSA*

Data is messy. What's more, the most tantalizing data to study is often that which is new and has not attracted attention yet. This tends to be the messiest. One of the major driving forces in the popularity of interactive programming is the ability to be flexible with an unknown data-space. Environments such as Jupyter Notebooks have become ubiquitous in data analysis for this reason. And rightfully so. They allow a developer to rapidly build up code to adapt without having to recompile-reload program and data from scratch. But it often comes at a cost. For example, Python is notoriously slow in a variety of ways and resources allocated go unused while waiting for a developer to enter their instructions. Initially it is unclear if the need for rapid and serendipitous code development outweighs the need for raw processing speed and efficiency. Experience has shown that for desktop computing and data analysis, indeed, interactive development reigns supreme.

For a variety of practical reasons, HPC systems tend to employ a batch scheduled computing model. While this should be the major mode of operation, more space should be made for interactive development with distributed computing to replicate the success of interactive development seen in desktop computing. JuliaLang is a good candidate to fill this space and expand past it. Julia is interactive with a JIT compiler, inherently asynchronous (e.g. multithreading and more), and has varying degrees between dynamically and statically typed. Julia programs can be interactively and dynamically developed, optimized, and scaled to approach performance of a traditionally compiled program.

In this short talk I will start by describing high level features that make Julia an ideal candidate for interactive distributed computing. Then I will introduce a specific problem that involves a sufficiently messy dataset. It is too large for AI/ML analysis on a desktop computing enviroment. For this problem I've developed and will introduce two Julia packages, DistributedQuery, DistributedFluxML, that help with in memory distributed data hosting and distribute AI/ML training. I will finish the talk seeking collaboration to improve, harden, and find more novel data analysis problems that can capitalize on an interactive and distributed development environment.

JLESC topic:

Poster Session / 74

Memory Power Consumption on Heterogeneous Memory Systems

Author: Andrès RUBIO PROAÑO[1]

Co-author: Kento Sato [1]

[1] *Riken*

The architecture of supercomputers over the years has evolved to support different need in applications that seek to solve some human concerns. Heterogeneity role nowadays is important in processors and also in the memory-storage system. In processors, we can observe CPUs, GPUs and other accelerators coexisting. In the same fashion, different kinds of memory have appeared over the years, fulfilling some gaps in the memory-storage continuum. E.g., high bandwidth memory (HBM), that is embedded on the processor package, coexist mainly with dynamic random access memory (DRAM) into Intel Xeon Phi Processors or Knight Landing (KNL). Non-volatile memory (NVM), that can be found with DRAM into the 2nd Generation Intel Xeon Scalable Processors. Nowadays, the upcoming Intel Sapphire Rapids support HBM inside the processor package, DRAM through the memory bus, and also it supports disaggregated memory by the Compute Express Link (CXL) that in principle allows to connect HBM, NVM and DRAM on it.

The task of developers when programming new applications or adapting the existing ones requires full knowledge of the memory system and without a specific strategy it can be very complicated depending on the conditions in which the applications are required to run. Today, for developers is pertinent to prepare their applications so that they adequately face at least the main heterogeneous memory system (HMS) setups. For that reason we consider that every developer should at least understand HMSs in terms of simple and easy metrics such as: bandwidth, latency, capacity, data

persistence, power consumption, etc. Especially, it is essential to know how much memory power applications are going to use in a given memory system. It is vital in situations where executions need to be performed with minimal power consumption mode, or when we need to balance power consumption and performance.

In this poster presentation, we focuses on understanding and giving a perspective on how to analyse memory energy consumption metric over different HMS setups.
We consider, identifying and exposing the memory system in the simplest manner developers could access. E.g., a memory system with DRAM and NVM can be exposed as different NUMAs in some systems and their access implies binding the applications process to the kind of memory required. Then, we have selected a some memory-intensive applications that should be profiled. Profiling depends on the expertise of developers and also tools can give more or less information depending on their capabilities. In our case, Intel Performance Counter Monitor (PCM) enables the possibility to get some performance counters related to memory power consumption in between others related to bandwidth. Also we used Linux Perf profiler tool to retrieve relevant information related to cache misses and verify if applications are behaving as a memory-intensive application. The final objective when analysing the power consumption metric is to be able to give a certain ordering for which the developer can look for a memory with very low consumption, as she/he could look for one that allows her/him to have a balance between the performance of the applications and the consumption of memory power. In addition to this analysis, we have sought to provide developers with an early HMS memory power prediction model, which allows getting an idea of the possible consumption of their application towards a given HMS.

JLESC topic:

Break-out Session: Next-generation Numerical Linear Algebra Libraries / 75

BOS: Next-generation Numerical Linear Algebra Libraries

Author: Toshiyuki Imamura[1]

[1] *RIKEN R-CCS*

BoS: Next-generation Numerical Linear Algebra Libraries

Exa-class system development has achieved some successful results, and full-scale systems are in operation. RIKEN is currently conducting a feasibility study of technological trends for developing a successor to Fugaku. Based on the experience developing the numerical library for Fugaku, RIKEN is now studying library development trends in Fugaku NEXT and strengthening international development relationships. There is no doubt that sustainable development and functional enhancement of deliverables in the ECP and European HPC projects, in which many JLESC partners have been involved, also remain an issue. We want to provide a place to discuss such matters, especially trends in numerical linear algebra libraries and various other topics.

Speakers:
Toshiyuki Imamura, and Atsushi Suzuki (R-CCS): Numerical Linear Algebra Libraries towards the Fugaku NEXT project
Edoardo Di Napoli (JSC): Massively parallel eigensolvers with spectral filters (remotely)
Sergi Laut, and Ramiro de Olazabal(BSC): Architecture-aware Sparse Patterns for the Factorized Approximate Inverse, and Parallel implementation of a Linelet preconditioner
Toby Isaac (ANL): Low-rank kernels in PETSc solvers (remotely)
Piotr Luszczek (ICL): Beyond classic linear algebra libraries for modern hardware platforms
Luc Giraud, and Emmanuel Agullo(INRIA and ANL): Variable accuracy GMRES and FGMRES

Thanks to the proposal:
Schleife Andre (UIUC)

JLESC topic:

Short Talks on Interative Tools and Monitoring / **76**

Advances on monitoring of supercomputers with LLview

Authors: Filipe Guimaraes[1]; Ilya Zhukov[1]; Vitor Silva[1]; Wolfgang Frings[1]; Yannik Müller[1]

[1] *Jülich Supercomputing Centre*

With the growth and evolution of supercomputers and the incorporation of diverse technologies, monitoring their usage has become a vital necessity for administrators and users altogether. In this context, LLview monitoring structure, developed by the Jülich Supercomputing Centre, stands out for providing extensive views on the system and job operations. The recently-released new version of LLview had its core completely restructured and its portal redesigned to enhance users interaction. Keeping its negligible overhead and role-controlled access, LLview has become more flexible and easier to configure, providing new insights and controls on the systems and jobs. This presentation will cover some of the new features, some of them implemented in the context of the European projects DEEP-SEA and IO-SEA, and the plans for future inclusions and collaborations.

JLESC topic:

Short Talks on Advanced Architectures / **77**

Life cycle environmental impacts of HPC systems

Author: Bill Kramer[1]

[1] *University of Illinois*

On a recent visit to NSF, I was asked about how Blue Waters was decommissioned. After describing the process, they asked me if I would write a paper/report on the process and the environmental impact. This expanded from the typical paper about the energy used by supercomputers to interest in the e-waste and other impacts. In fact, some recent decadal reports form science domains (e.g. astrophysics and astronomy) have noted the desire to decrease the use of HPC because of the impacts on the environments.

Thinking about e-waste was a new twist for me, and I don't understand why someone would think a supercomputer has more environmental impact than other things including hyperscaler systems and even cruise boats. Further, past energy only studies have only included energy costs and does not account for the positive impacts that are produced by use of HPC.

I would like to collaborate with others who may be able to add their sites' analysis to the report.

JLESC topic:

Short Talks on Interative Tools and Monitoring / **78**

Blue Waters Monitoring, Usage and Experience Data is Available

Author: Bill Kramer[1]

Co-author: Blue Waters Team Members [1]

[1] *University of Illinois*

The Blue Waters system and project was one of the measured and monitored system at scale. Now, over a Petabyte of monitoring, system activity, reliability, security, networking and performance data is available for the researchers to use for its entire operational period of over 9 years. This talk will summarize the types of data available and possible open questions that collaborators may want to consider investigating.

JLESC topic:

Short Talks on Numerical Methods / 79

SpMM, more computational intensive operation for sparse matrix in Krylov subspace methods

Author: Atsushi Suzuki[1]

[1] *R-CCS*

Krylov subspace methods are used to solve linear system with large sparse matrix, where main operations consist of the SpMV, operation of sparse matrix multiplication to vector and the inner product operation. The most common data structure to keep the sparse matrix is called CSR that only keeps nonzero entries in each row with compressed format. However, computational complexity of SpMV is very low because loaded coefficient data are only used once against element of the vector and it is known that the benchmark of HPCG only can achieve less than 5 percent of the peak performance of the modern super computer, even if the matrix data form a regular stencil pattern.
It is very important to enhance computational intensity of such operation with sparse matrix. If we need to solve linear system with several multiple right hand side, SpMV can be replaced by SpMM and we can expect substantial speed up thanks to recycling of coefficient data by keeping them in the cache memory. For the most standard linear system with single right hand side, we need to prepare several search vectors from a single residual vector. One candidate of such multiple search vectors is set of search vectors during convergence of local problems that are obtained by matrix decomposition. By combining restarting procedure and spectral analysis of the group of search vectors, convergence of such CG method can be accelerated.
Unfortunately numerical libraries for sparse matrix product is not intensively developed, and then I would to find a way to prepare optimized sparse BLAS and sparse linear solvers.

JLESC topic:

Short Talks on Interative Tools and Monitoring / 80

Process mapping on any topologies with TopoMatch

Author: Emmanuel Jeannot[1]

[1] *Inria*

Process mapping (or process placement) is a useful algorithmic technique to optimize the way applications are launched and executed onto a parallel machine. By taking into account the topology of the machine and the affinity between the processes, process mapping helps reducing the communication time of the whole parallel application. Here, we present TopoMatch, a generic and versatile library and algorithm to address the process placement problem. We describe its features and characteristics, and we report different use-cases that benefit from this tool. We also study the impact of different factors: sparsity of the input affinity matrix, trade-off between the speed and the quality of the mapping procedure as well as the impact of the uncertainty (noise) onto the input.

JLESC topic:

Topology-aware execution

Short Talks on Applications / 81

Femtoscale Imaging of Nuclei Using High-performance Computing

Author: Anshu Dubey[1]

[1] *Argonne National Laboratory*

Subatomic particles have a size of about one femtometer and are studied through measurement of scattering events at various particle accelerator facilities around the world. An experimental event is a particle collision that triggers a detector response, which then collects various signals that allows the properties of the measured final state particles to be reconstructes. For imaging quarks and gluons at the femtoscale the challenge is they never reach a detector. This is a unique challenge in all of science, because the elementary degrees of freedom (quarks and gluons) are not those directly accessible in experiment. Our project aims to develop a framework that can extract the maximum amount of information on a quark and gluon tomography of nucleons and nuclei from high-energy scattering data. To achieve this goal of maximal information it is essential to compare theory and experiment at the most fundamental level. We are developing a workflow for the extraction of QCFs from an event-level analysis of experimental data with four connected modules. Module 1 generates QCFs using a deep neural network. Module 2 constructs particle momentum distributions (PMDs) and generates idealized theory events using Markov chain Monte Carlo. Module 3 incorporates detector effects to create simulated events. Module 4 compares the simulated and measured events using a discriminator. This process repeats until the simulated and experimental events correspond to the same theory by a given measure. The complexity of this workflow can increase dramatically because module 2 can represent many processes giving different PMDs and idealized events that correspond to the same QCFs. Then module 3 can represent many detectors from different experiments generating a larger set of simulated events that must be compared with experimental events from different sources. This is a new computational paradigm for the field and several possibilities of collaboration and innovation exist.

JLESC topic:

Numerical methods and algorithms

Short Talks on Workflows, I/O and Frameworks / 82

Perspectives on the Versatility of a Searchable Lineage for Scalable HPC Data Management

Author: Bogdan Nicolae[1]

[1] *ANL*

Checkpointing is the most widely used approach to provide resilience for HPC applications by enabling restart in case of failures. However, coupled with a searchable lineage that records the evolution of intermediate data and metadata during runtime, it can become a powerful technique in

a wide range of scenarios at scale: verify and understand the results more thoroughly by sharing and analyzing intermediate results (which facilitates provenance, reproducibility, and explainability), new algorithms and ideas that reuse and revisit intermediate and historical data frequently (either fully or partially), manipulation of the application states (job pre-emption using suspend-resume, debugging), etc.

This talk advocates a new data model and associated tools (DataStates, VELOC) that facilitate such scenarios. Avoid direct use of a data service to read and write datasets; instead, during runtime, users should tag datasets with properties that express hints, constraints, and persistency semantics. Doing so will automatically generate a searchable record of intermediate data checkpoints, or data states, optimized for I/O. Such an approach brings new capabilities and enables high performance scalability, and FAIR-ness through a range of transparent optimizations. The talk will introduce DataStates and VELOC, will underline several vital technical details, and will conclude with several examples of where they were successfully applied.

JLESC topic:

Short Talks on Interative Tools and Monitoring / 83

Understanding the relation between monitoring events and topology of exascale architectures for HPC applications

Author: Idriss Daoudi[1]

[1] *Argonne National Laboratory*

With an increasing workload diversity and hardware complexity in HPC, the boundaries of today's runtimes are pushed to their limits. This evolution needs to be matched by corresponding increases in the capabilities of system management solutions.

Power management is a key element in the upcoming exascale era. First to allow us to stay within the power budget, but also for the applications to make the most of the available power in order to make progress. Therefore, our objective is to balance complex applications requirements while keeping power consumption under budget.

To achieve this goal, the Argo group is working on the Node Resource Manager (NRM) tool, which allows us to centralize node management activities such as resource and power management. The latter is achieved by getting information (monitoring) from various sensors (power, temperature, fan speed, frequency…) and adjusting actuators (CPU p-states, Intel RAPL) according to the application needs. The next step in our power management strategy is to improve NRM monitoring to more easily identify the location (within the topology) and scope (range of devices) that monitoring events are related to.

To evaluate our implementation, we are looking for JLESC members willing to extend this work with more complex applications with dynamic resource balancing problems, on which we first can observe such imbalance, and then address it with a better power management strategy relying on precise identification of the relation between the gathered monitoring events, the devices present, and the inner components of applications. We are aiming to get a better understanding of the behavior of such applications under various scenarios of power management, as well as studying the possibility of characterizing applications' power needs in order to develop an automated resource management policy.

JLESC topic:

Keynote: Vector operations, tiled operations, distributed execution, task graphs, what next?

In the past decades, we have made dramatic changes in the way we express HPC computations. Vector architectures made us employ vector operations, architectures with caches made use leverage tiled kernels, and distributed systems made us express communications. More recently, notably GPU architectures made us try to embrace task-based parallelism so as to efficiently distribute work among a heterogeneous set of resources, and automatically optimize the entailed flurry of data transfers. It is thus intriguing to try to imagine what could be the next programming paradigm shift.

In this talk, I will explore one the current candidates: recursive task graphs. We can indeed notice that they are currently being proposed in various task-based runtime systems. While the details differ, a lot of commonalities arise, and similar performance benefits are observed. On the application side, expressing computation recursively is also a pattern that arises commonly for expressing e.g algorithms on compressed data such as H-matrices. We will thus consider the current proposals and results, and discuss the kinds of benefits that we can expect in the long run. This can include making it easier to express complex algorithms with complex data structures, and improve the efficiency of the execution. It also opens for new perspectives in optimizing execution, which will be part of the just-starting NumPEx PEPR French project.

Session chairperson: Emmanuel Jeannot

85

Keynote Talk 2

Chairperson: Franck Cappello

86

Keynote: Co-designing Self-Service Digital Twin Workflows with DIY Cluster Toolbox and DRI Leasing Federation

Per industry definition, objectives of Digital Twins (DTs) include facilitating real-time or on-demand investigation, impact studies, and recalibrating for monitoring, diagnostics and prognostics of virtual ecosystems representing real world scenarios. Efficient and secure implementation of complex workflows encompassing a wide range of experimental, observational and simulation computational and data science methods have been identified among the underpinning requirements for DTs. This talk covers two foundational building blocks for enabling complex and distributed workflows as the underlying technologies span from edge to supercomputing ecosystems. The first one is technical and the second one relates to business models. A do-it-yourself (DIY) cluster toolbox by the University of Bristol called Cluster-in-the-Cloud (CitC) has been co-designed for diverse workflows across different software defined, public cloud, IT infrastructure. In addition, these workflows rely on distributed data and compute resources, spanning from supercomputing facilities to cloud to the edge instruments. CitC enables flexible mapping and orchestration of software stacks pipelines and data-driven workflows. On the business model side, specifically for institutional, national, and internationally funded tier-n digital research infrastructure (DRI), there needs to be a resource allocation scheme managed in a federated manner across resource and service providers. This could be compared to a leasing model that can be deployed on a multi-tenant DRI. Domain-specific examples exist within global-scale science experiments and service providers consortia but not across them. A discussion on DT workflows as co-design driver use cases will be presented for realising a self-service and secure edge-cloud-supercomputing continuum.

Short Talks on AI/MD/DL / 88

Waggle AI@Edge Computing: NSF Sage and Beyond.

Author: Rajesh Sankaran[1]

[1] *Argonne National Laboratory*

From the sensor to the laptop, from the telescope to the supercomputer, from the microscope to the database, scientific discovery is part of a connected digital continuum that is dynamic and fast. In this new digital continuum, Artificial intelligence (AI) is providing tremendous breakthroughs, making data analysis and automated responses possible across the digital continuum. Sage is a National Science Foundation funded project to build a national cyberinfrastructure for programmable edge computing, leveraging DOE funded Waggle AI@Edge platform. This new edge computing programming framework gives scientists a new tool for exploring the impacts of global urbanization, natural disasters such as flooding and wildfires, and climate change on natural ecosystems and city infrastructure. The Sage infrastructure allows scientists to write "software-defined sensors" by analyzing the data in situ, at the edge, at the highest resolution of data. Data from the edge computation are then transmitted to a cloud computing infrastructure where they can be archived and provided to the community as data products or used in real time to trigger computational models or dynamically modify subsequent edge computation. This new edge computing programming framework gives scientists a new tool for exploring the impacts of global urbanization, natural disasters such as flooding and wildfires, and climate change on natural ecosystems and city infrastructure. Sage is deploying cyberinfrastructure in environmental test-beds in California, Montana, Colorado, and Kansas, in the National Ecological Observatory Network, and in urban environments in Illinois and Texas. In this talk, beyond Sage and other DOE efforts that utilize Waggle, we will discuss our vision for edge computing from applications to capabilities, and outline the research and development challenges.

JLESC topic: