INNOVATIVE COMPUTING LABORATORY

2021/22 REPORT

THE UNIVERSITY OF TENNESSEE KNOXVILLE

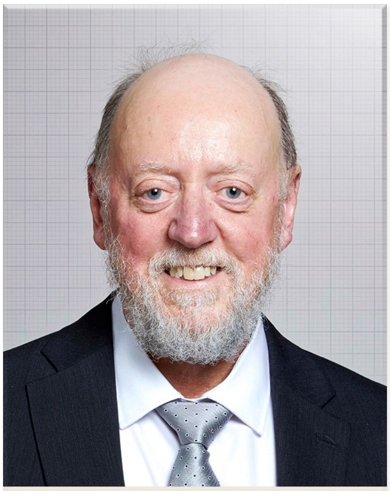
INNOVATIVE COMPUTING LABORATORY **2021/22 REPORT**

DESIGNED BY **David Rogers** EDITED BY **Rob Anderson**

INNOVATIVE COMPUTING LABORATORY

2021/22 REPORT

FROM THE DIRECTOR



Jack DONGARRA
DIRECTOR, ICL

Although it is difficult to write a letter that would give a fair description of the challenges we faced during the second year of the global COVID-19 pandemic, I am proud to have the opportunity to highlight the outstanding efforts of the ICL family. We had to continue adjusting our work dynamics and interactions to compensate for interruptions in our office environment and our families and communities. The continued creativity and flexibility of ICL's researchers, staff, students, and collaborators remains impressive. And while I have always been thankful and highly appreciative of everyone's effort, that is particularly true this year as we confronted these prolonged challenges.

ICL's research efforts during 2021 continued to provide valuable contributions to the Department of Energy's (DOE) Exascale Computing Project (ECP). Our research team developed innovative software products to meet DOE's mission in exascale needs. We advanced the growth of applications essential to contribute to a reliable software ecosystem, including PAPI, PaRSEC, heFFTe, OpenMPI, xSDK, SLATE, and the CEED center. Despite the disadvantages of working remotely and its consequences in decreased social interactions, we developed a collaboration for a newly funded DOE project focused on software development to model materials relevant in nuclear applications with exascale computing. This collaboration includes ICL, Stony Brook University, and Virginia Technological University. We continue to strengthen our relationship with the Joint Laboratory for Extreme Scale Computing (JLESC). This year, we began establishing initiatives to develop a strategic plan in diversity, equity, inclusion, and belonging for ICL. We established the MathWorks Professorship in the EECS department and are incredibly grateful to MathWorks for their generous contributions over the years. We hope to fill that position with someone aligned with the themes of ICL.

The pandemic continued to introduce challenges in hiring talented team members. However, we have been incredibly fortunate to include Wissam Lakhdar, Daniel Bielich, Kadir Akbudak, Giuseppe Congiu, and Anustuv Pal in the research team. Also, Robert Anderson joined our team as ICL's Communications Specialist. Rob comes to ICL from the Department of Information Sciences, pursuing a Ph.D. Furthermore, David Rogers is filling the role of Communications Services Leader, and Geri Ragghianti the position of Technical Services Leader. Although working remotely continued to cause issues due to social isolation, productivity, funding, and the pursuit of new funding opportunities and collaborations remain strong. I am thankful to Joan Snoderly, Tracy Lee, Teresa Finchum, Leighanne Sisk, Earl Carr, and Deborah Penchoff for their exceptional commitment and hard work; I am also honored to sustain our partnership with Michela Taufer and her group in the Global Computing Laboratory. I am confident that the strong commitment and extraordinary work ethic of the ICL family will further guarantee success even in challenging circumstances.

I am deeply thankful for the continued support of our partners and collaborators in industry, academia, and government facilities and for the outstanding staff at ICL, colleagues in the Tickle College of Engineering, the University of Tennessee, and Oak Ridge National Laboratory. Most of all, I am grateful to every member of the ICL family for their support, confidence, commitment, and friendship throughout the last 33 years. I may be ICL's founder, but it is all of you who have made ICL the exceptional center it has become.

This will be my last Director's statement, as most of you know. I began this season of my life 33 years ago, starting with just two people, and built an organization to develop algorithms and software for advanced high-performance systems. Today we are recognized worldwide for our innovations in high performance computing. It has been an exciting, rewarding, and fulfilling ride in no small part to the talents of the ICL staff. The current ICL family includes members who have been with the team for several decades and those who have recently joined the group. I am confident that their collaboration, commitment, and resilient nature guarantees that ICL will continue its global leadership in scientific computing for years to come. While I will retire at the end of June, I will not be closing the door on ICL as I plan to be emeritus, still interacting, working with, and offering my advice on the direction of ICL. With the endowed MathWorks Professorship, it is my hope and belief that the road being paved by ICL assures a dynamic and productive future.

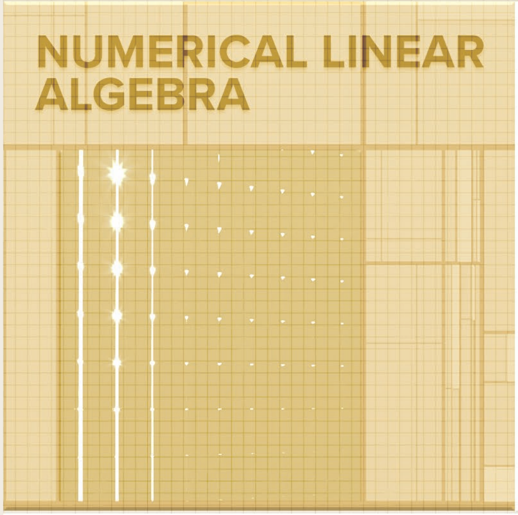
INTRODUCTION

Located in the heart of the University of Tennessee's Knoxville campus, ICL is a computer science research and development laboratory with a rich history that has spanned radical advancements in computing hardware and parallelism and the explosion of data-intensive science and computation wedged against the need for ever-increasing energy efficiency and resilience.

ICL's work, which has evolved and expanded to address these challenges, encompasses a solid understanding of the algorithms and libraries for multi-core, many-core, and heterogeneous computing, as well as performance evaluation and benchmarking for high-end computing. In addition, ICL's portfolio of expertise includes high-performance parallel and distributed computing—with keen attention to message passing and fault tolerance.

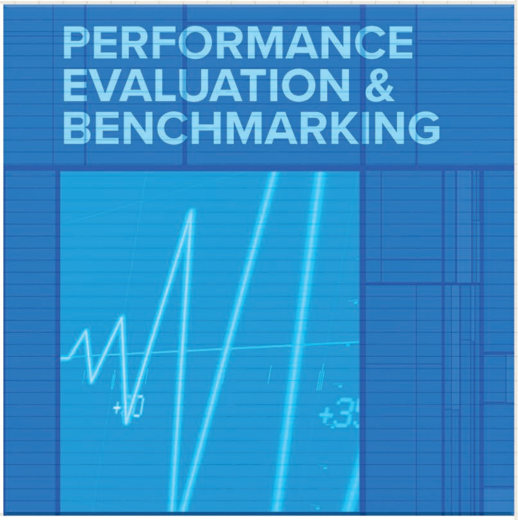
The tools and technologies that ICL designs, develops, and implements play a key role in supercomputing-based discoveries in areas like life sciences, climate science, earthquake prediction, energy exploration, combustion and turbulence, advanced materials science, drug design, and more.

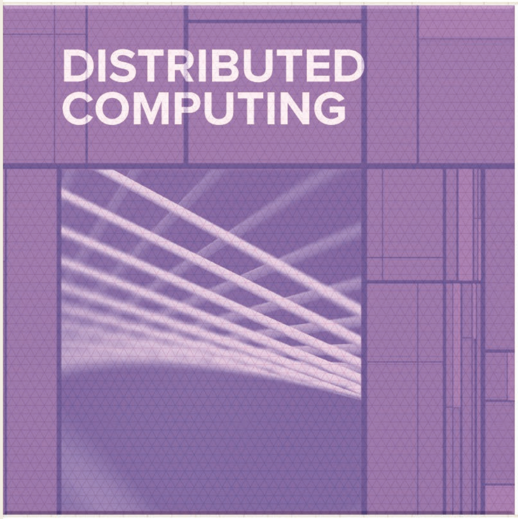
AREAS OF RESEARCH



NUMERICAL LINEAR ALGEBRA algorithms and software form the backbone of many scientific applications in use today. With the ever-changing landscape of computer architectures, such as the massive increase in parallelism and the introduction of hybrid platforms utilizing both traditional CPUs as well as accelerators, these libraries must be revolutionized in order to achieve high performance and efficiency on these new hardware platforms. ICL has a long history of developing and standardizing these libraries in order to meet this demand, and we have multiple projects under development in this arena.

PERFORMANCE EVALUATION AND BENCHMARKING are vital to developing science and engineering applications that run efficiently in an HPC environment. ICL's performance evaluation tools enable programmers to see the correlation between the structure of source/object code and the efficiency of the mapping of that code to the underlying architecture. These correlations are important for performance tuning, compiler optimization, debugging, and finding and correcting performance bottlenecks. ICL's benchmark software is widely used to determine the performance profile of modern HPC machines and has come to play an essential role in the purchasing and management of major computing infrastructure by government and industry around the world.





DISTRIBUTED COMPUTING is an integral part of the HPC landscape. As the number of cores, nodes, and other components in an HPC system continue to grow explosively, applications require runtime systems that can exploit all of this parallelism. Moreover, the drastically lower meantime to failure of these components must be addressed with fault-tolerant software and hardware, and the escalating communication traffic that they generate must be addressed with smarter and more efficient message passing standards and practices. Distributed computing research at ICL has been a priority for over two decades, and the lab has numerous projects in this arena under active development.

HISTORY

Prof. Jack Dongarra established ICL in 1989 when he received a dual appointment as a Distinguished Professor at UTK and as a Distinguished Scientist at Oak Ridge National Laboratory (ORNL). Over thirty years later, ICL has grown into an internationally recognized research laboratory specializing in numerical linear algebra, distributed computing, and performance evaluation and benchmarking.

1989

The Level-3 **Basic Linear Algebra Subprograms (BLAS)** specification was developed to perform assorted matrix-multiplication and triangular-system computations.

The **Parallel Virtual Machine (PVM)** was a parallel networking tool that enabled a user to leverage a network of heterogeneous Unix and Windows machines as a single distributed parallel processor.

1992

The **Basic Linear Algebra Communication Subprograms (BLACS)** project was created to make linear algebra applications easier to program and more portable.

Still developed today, the **Linear Algebra Package (LAPACK)** is a standard software library for numerical linear algebra.

1993

The **TOP500** was launched to improve and renew the Mannheim supercomputer statistics, which—at the time—had been in use for seven years.

1994

Version 1.0 of a standardized and portable message-passing system, called the **Message Passing Interface (MPI)**, was released. MPI has since become the de facto standard for communication in parallel distributed computing systems.

1995

Version 1.0 of the **Scalable LAPACK (ScaLAPACK)** library, which includes a subset of LAPACK routines redesigned for distributed memory multiple instruction, multiple data (MIMD) parallel computers, was released. Like LAPACK, ScaLAPACK is still under active development.

1997

Automatically Tuned Linear Algebra Software (ATLAS) was an instantiation of a new paradigm in high-performance library production and maintenance developed to enable software to keep pace with the incredible rate of hardware advancement inherent in Moore's Law.

NetSolve (GridSolve) was a client-server system that enabled users to solve complex scientific problems using remote resources.

1999

Heterogeneous Adaptable Reconfigurable Networked SystemS (HARNESS) was a pluggable, lightweight, heterogeneous, and distributed virtual machine environment.

Still in active development, the **Performance Application Programming Interface (PAPI)** is a standardized, easy-to-use interface that provides access to hardware performance counters on most major processor platforms.

2000

High-Performance Linpack (HPL) is a benchmark for distributed-memory computers that solves a (random) dense linear system in double-precision (64-bit) arithmetic. HPL is often one of the first programs to run on large HPC machines, producing a result that can be submitted to the TOP500 list of the world's fastest supercomputers.

2002

Fault Tolerant MPI (FT-MPI) was an MPI plugin for HARNESS that provided support for fault-tolerant applications crucial for large, long-running simulations.

2003

HPC Challenge was developed for the Defense Advanced Research Projects Agency (DARPA) and consisted of four benchmarks: HPL, Streams, RandomAccess, and PTRANS.

LAPACK for Clusters was developed in the framework of self-adapting numerical software to leverage the convenience of existing sequential environments bundled with the power and versatility of highly tuned parallel codes executed on clusters.

2006

Fault-Tolerant Linear Algebra (FT-LA) is a research effort to develop and implement algorithm-based fault tolerance in commonly used dense linear algebra kernels.

Four institutions merged efforts in the **Open Source Message Passing Interface (Open MPI)**: FT-MPI from UTK/ICL, LA-MPI from LANL, and LAM/MPI from Indiana University, with contributions from PACX-MPI at the University of Stuttgart.

2008

Matrix Algebra on GPU and Multi-core Architectures (MAGMA) is a linear algebra library that enables applications to exploit the power of heterogeneous systems of multi-core CPUs and multiple GPUs or coprocessors.

Parallel Linear Algebra Software for Multi-core Architectures (PLASMA) is a dense linear algebra package designed to deliver the highest possible performance from a system of multiple sockets of multi-core CPUs.

2009

The **International Exascale Software Project (IESP)** brought together representatives of the global HPC community to plan and create a new software infrastructure for the extreme-scale systems that represent the future of computational science.

2010

Distributed Parallel Linear Algebra Software for Multi-core Architectures (DPLASMA) is a linear algebra package that enables sustained performance for distributed systems, where each node features multiple sockets of multi-core CPUs and, if applicable, accelerators like GPUs or Intel Xeon Phi.

2011

The **Parallel Ultra Light Systolic Array Runtime (PULSAR)** project developed a simple programming model for large-scale, distributed-memory machines with multi-core processors and hardware accelerators to automate multithreading, message passing, and multi-stream, multi-GPU programming.

2012

The **Parallel Runtime Scheduling and Execution Controller (PaRSEC)** provides a generic framework for architecture-aware scheduling and management of microtasks on distributed, many-core heterogeneous architectures.

User Level Failure Mitigation (ULFM) is a set of new interfaces for MPI that enables message passing programs to restore MPI functionality affected by process failures.

2013

The **Big Data and Extreme-scale Computing (BDEC)** workshop was initiated to map out and account for the ways in which the major issues associated with "big data" intersect with national (and international) plans being laid out for achieving exascale computing.

The **High Performance Conjugate Gradients (HPCG)** benchmark is designed to measure performance that is representative of modern HPC capability by simulating patterns commonly found in real science and engineering applications.

2015

PAPI-Ex extends PAPI with measurement tools for changing hardware and software paradigms.

The **SparseKaffe** project establishes fast and efficient sparse direct methods for platforms with multi-core processors with one or more mance for scientific applications on massively parallel hybrid systems.

2016

The **Distributed Tasking for Exascale (DTE)** project will extend the capabilities of the PaRSEC framework.

The **Exascale Performance Application Programming Interface (Exa-PAPI)** project builds on PAPI-Ex and extends it with performance counter monitoring capabilities for new and advanced ECP hardware and software technologies.

The **Production-ready, Exascale-enabled Krylov Solvers for Exascale Computing (PEEKS)** project will explore the redesign of solvers and extend the DOE's Extreme-scale Algorithms and Solver Resilience (EASIR) project.

The **Software for Linear Algebra Targeting Exascale (SLATE)** project will converge and consolidate previous ICL efforts with LAPACK and ScaLAPACK into a dense linear algebra library that will integrate seamlessly into the ECP ecosystem.

2017

The **Batched BLAS (BBLAS)** effort will create an API for numerical computing routines that process batches of either uniformly sized or varying-size matrices or vectors and will serve as a working forum for establishing this strategy as the next official BLAS standard.

The **MAtrix, TEnsor, and Deep-learning Optimized Routines (MATEDOR)** team is performing the research required to define a standard interface for batched operations (BBLAS) and provide a performance-portable software library that demonstrates batching routines for a significant number of linear algebra kernels.

2018

The goal of **BDEC2**, a follow-on to BDEC and IESP, is to stage six international workshops to enable research communities in a wide range of disciplines to converge on a common platform in order to meet the daunting challenges of achieving exascale computing in the wake of a surging "data tsunami."

The main objective of the **ECP Fast Fourier Transform (ECP-FFT)** project is to design and implement a fast and robust 2-D and 3-D FFT library that targets large-scale heterogeneous systems with multi-core processors and hardware accelerators and to do so as a co-design activity with other ECP application developers.

2019

The **Ecosystem for Programming and Executing eXtreme Applications (EPEXA)**, aims to create a software framework that implements high-performance methods for irregular and dynamic computations that are poorly supported by current programming paradigms.

The **Scalable Run Time for Highly Parallel, Heterogeneous Systems (ScaRT)** project aims to increase the scientific throughput of existing and future cyberinfrastructure platforms by reducing communication overheads and better matching the functionality of communication libraries to modern communication adapters.

The **Development of Exascale Software for Heterogeneous and Interfacial Catalysis (DESC)** project focuses on understanding the relationship between algorithms and hardware platforms and on jointly optimizing the software and hardware to achieve efficient implementations of materials science, chemistry, and physics applications.

2020

The **Surrogate Benchmark Initiative (SBI)** aims to provide benchmarks and tools for assessing deep neural network "surrogate" models. A surrogate model can imitate part or all of a simulation and produce the same outcomes while requiring less resources. Tools developed under SBI will evaluate these surrogate models to measure progress and inform the codesign of new HPC systems to support their use.

Basic ALgebra LIbraries for Sustainable Technology with Interdisciplinary Collaboration (BALLISTIC) will create software components that deliver access to the most up-to-date algorithms, numerics, and performance via Sca/LAPACK interfaces; make available advanced algorithms, numerics, and performance capabilities; and provide a well-engineered conduit for new developments to be channeled to the science and engineering applications that depend on high-performance linear algebra libraries.

2021

oneAPI is an industry initiative aiming to create a single, unified, cross-architecture programming model for CPUs and accelerator architectures. Based on industry standards and its open development approach, the initiative will help streamline software development for high performance computers, increase performance and provide specifications for efficient and diverse architecture programming.

The **FFT Benchmarking project (FIBER)** provided an interim update on comprehensive testing and measuring various aspects of the FFT implementations for three-dimensional transforms on high performance systems. The results from the report revealed large differences in functionality, performance, and supported hardware and have given the ECP application community a much better appreciation of the available software choices.

YEAR IN REVIEW

UTK Mathworks Professorship in Scientific Computing

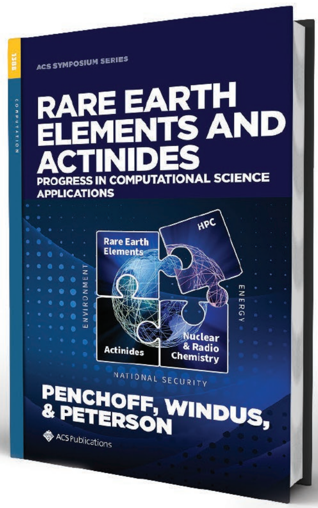
The Department of Electrical Engineering and Computer Science (EECS) at The University of Tennessee, Knoxville (UTK) invited candidates to apply for an Endowed MathWorks Professorship in Scientific Computing with a tenure track faculty position at the associate or full professor level. MathWorks has contributed financially to ICL for several years. MathWorks has made an additional contribution to endow the position. ICL's relationship with the MathWorks goes back to before the company began. MathWorks founder Cleve Moler worked with Jack Dongarra on LINPACK, a Fortran software library for performing numerical linear algebra on computers and supercomputers in the 1970s and early 1980s. ICL is hoping to fill the MathWorks Professorship with someone who can carry on the development of mathematical software by the summer of 2022 for a start in the fall 2022 semester.

Intel oneAPI Center of Excellence

In October, the University of Tennessee, Knoxville, announced the creation of a new Intel oneAPI Center of Excellence to provide solutions in high performance computing (HPC) and visualization using oneAPI. The center will focus on two projects: porting the open-source HPC Ginkgo library to oneAPI for cross-architecture support and expanding its Intel Graphics and Visualization Institute of XeLLENCE to enable high-end visualization as a service through oneAPI.

ICL consultant Hartwig Anzt, heads a project focused on porting the Ginkgo library to oneAPI. This work extends Ginkgo library support to other accelerators—including taking advantage of current and future Intel® Xe GPUs. Anzt will lead the oneAPI Center of Excellence with a team of experienced sparse linear algebra researchers. The center will conduct research and use oneAPI technology to contribute to the open specification and advance ecosystem adoption. The center's work will help prepare Ginkgo for the Aurora supercomputer at Argonne National Laboratory.

Additionally, Anzt participates on the oneAPI Math Kernel Library (oneMKL) Technical Advisory Board to provide Sparse Linear Algebra expertise and feedback on the oneAPI libraries, compilers, and tools to help support researchers and developers around the world to take advantage of the resulting improvements. Further leveraging the university's expertise, oneAPI curriculum modules will be developed and included in UT's coursework and open-sourced to bring oneAPI programming skills to students worldwide in 2022.



Rare Earth Elements and Actinides: Progress in Computational Science Applications

Rare earth elements (REEs) and actinides are critical to electronics, communication, military applications, and green energy systems. High Performance Computing (HPC) has had a direct impact on investigations of REEs and actinides. This work discusses recent advances in molecular and data driven modeling that are essential to the study of REEs and actinides, effects of computational science in nuclear and radiochemical applications, and advances and challenges in the exascale era of supercomputing.

Edited by ICL's Deborah A. Penchoff, with contributions by ICL faculty members Heike Jagode, Piotr Luszczek, Anthony Danalis, George Bosilca, and Jack Dongarra, *Rare Earth Elements and Actinides: Progress in Computational Science Applications* presents a unique glimpse into the combined contributions of the ICL team. Not only did an assembly of ICL staff contribute to authoring this work, but also ICL Communications Services Leader David Rogers created the cover art. Boasting contributions from a number of ICL researchers and support staff, this book exemplifies ICL's potential impact when utilizing the wealth of expertise present within the department.

In August, ICL Graduate Research Assitant Dong Zhong earned his PhD under the guidance and mentorship of Prof. Dongarra and ICL's research scientists. Dr. Zhong's dissertation was titled “Toward Reliable and Efficient Message Passing Software for HPC systems: Fault Tolerance and Vector Extension.”

RESEARCH

What originally began over 30 years ago as in-depth investigations of the numerical libraries that encode the use of linear algebra in software has grown into an extensive research portfolio, including eight projects sponsored by DOE's Exascale Computing Project (ECP). ICL has evolved and expanded our research agenda to accommodate the heterogeneous computing revolution and focus on algorithms and libraries for multi-core and hybrid computing. As we have gained a solid understanding of the challenges presented in these domains, we have further expanded our scope to include work in performance evaluation and benchmarking for high-end computers, as well as work in high-performance parallel and distributed computing, with efforts focused on message passing and fault tolerance.

AsyncIS

The Asynchronous Iterative Solvers for Extreme-Scale Computing (AsyncIS) project aims to explore more efficient numerical algorithms by decreasing their overhead. AsyncIS does this by replacing the outer Krylov subspace solver with an asynchronous optimized Schwarz method, thereby removing the global synchronization and bulk synchronous operations typically used in numerical codes.

AsyncIS, a DOE-funded collaboration between Georgia Tech, UTK, Temple University, and SNL, also focuses on the development and optimization of asynchronous preconditioners (i.e., preconditioners that are generated and/or applied in an asynchronous fashion). The novel preconditioning algorithms that provide fine-grained parallelism enable preconditioned Krylov solvers to run efficiently on large-scale distributed systems and many-core accelerators like GPUs.

FIND OUT MORE AT
www.icl.utk.edu/asyncis

BALLISTIC

Basic ALgebra LIbraries for Sustainable Technology with Interdisciplinary Collaboration (BALLISTIC) is an NSF-funded effort to create new software components capable of running at every level of the platform pyramid by delivering seamless access to the most up-to-date algorithms, numerics, and performance via familiar Linear Algebra PACKage (LAPACK) and Scalable Linear Algebra PACKage (ScaLAPACK) interfaces; by making advanced algorithms, numerics, and performance capabilities available through new interface extensions; and by providing a well-engineered conduit for channeling new developments to science and engineering applications that depend on high-performance linear algebra libraries.

Scientific software libraries have long provided a large and growing resource for high-quality, reusable software components upon which applications from science and engineering can be rapidly constructed. The BALLISTIC project, through the leading-edge research it channels into its software deliverables, will lead to the introduction of tools that will simplify the transition to the next generation of extreme-scale computer architectures. The main impact of the project will be to develop, release, and deploy software into the scientific community to make it competitive on a world-wide scale and to contribute to standardization efforts in the area.

FIND OUT MORE AT
icl.utk.edu/ballistic

Batched BLAS

The Batched Basic Linear Algebra Subprograms (BBLAS) effort, an international collaboration between INRIA, Rutherford Appleton Laboratory, Umeå University, the University of Manchester, and UTK, will create an application programming interface (API) for numerical computing routines that process batches of either uniformly sized or varying-size matrices or vectors. This will go beyond the original Basic Linear Algebra Subprogram (BLAS) standard by specifying a programming interface for modern scientific applications, which produce large numbers of small matrices at once.

Individually, the small sizes of the inputs obviate the potential benefits of using BLAS but are a perfect fit for BBLAS. The BBLAS project will also serve as a working forum for establishing the consensus for the next official standard that will serve the scientific community and ensure support from hardware vendors.

FIND OUT MORE AT
icl.utk.edu/bblas

CEED

The Lawrence Livermore National Laboratory (LLNL)–led Center for Efficient Exascale Discretizations (CEED) co-design effort will develop next-generation discretization software and algorithms—which deliver a significant performance gain over conventional low-order methods—to enable a wide range of DOE and National Nuclear Security Administration (NNSA) applications to run efficiently on future exascale hardware. CEED is a research partnership involving 30+ computational scientists from two DOE labs and five universities, including UTK.

For UTK's part, ICL is instrumental in identifying, developing, and optimizing tensor contractions that are essential building blocks for these kinds of DOE/NNSA applications. The ICL team will also play an integral role in co-designing application programming interfaces (APIs) with the LLNL scientists, external partners, and vendors, and will deliver a high-performance tensor contractions package through the Matrix Algebra on GPU and Multicore Architectures (MAGMA) library.

FIND OUT MORE AT
ceed.exascaleproject.org

CORES

The Convex Optimization for Real-time Embedded Systems (CORES) project aims to develop highly efficient, real-time convex optimization algorithms and toolsets for solving important engineering problems on hierarchical and heterogeneous embedded system architectures. Though recent advances in optimization solvers have enabled the solution of optimization problems on low-cost embedded systems, the size of the problems that can be solved in real time is still limited.

The CORES project, a collaboration between ICL and Michigan Technological University, works to address this limitation. The ICL team's main responsibility is the design and development of higher-performance, structure-aware linear solvers that would enable us to solve, in real time, the convex optimization problems that have significantly higher performance—and are orders of magnitude greater in size—compared to current state-of-the-art solvers.

AVAILABLE IN
MAGMA 2.6.2
FIND OUT MORE AT
www.icl.utk.edu/cores

RESEARCH

DESC

FIND OUT MORE AT
hetcat-ccs.github.io

The Development of Exascale Software for Heterogeneous and Interfacial Catalysis (DESC) project focuses on understanding the relationship between algorithms and hardware platforms and how to jointly optimize the software and hardware to achieve efficient implementations for applications in materials science, chemistry, and physics. DESC is a joint effort between ICL/UTK, DOE's Ames Laboratory, EP Analytics, Inc., Georgia Tech, Old Dominion University, and Virginia Tech and is funded by the DOE Computational Chemical Sciences project.

ICL's contribution focuses on expressing GAMESS computational chemistry algorithms in the form of a dataflow graph and subsequently mapping the DAG representation of the kernels to the hardware platforms. This representation allows for capturing the essential properties of the algorithms (e.g., data dependencies) and computation at extreme scale by utilizing the hardware components (e.g., CPU or GPU) best suited for the type of computational task under consideration. The dataflow-based form of these algorithms makes them compatible with next-generation task scheduling systems like PaRSEC, StarPU, and Legion.

DPLASMA

VERSION
3.0.0
FIND OUT MORE AT
icl.utk.edu/dplasma

The Distributed Parallel Linear Algebra Software for Multi-core Architectures (DPLASMA) package is the leading implementation of a dense linear algebra package for distributed heterogeneous systems. It is designed to deliver sustained performance for distributed systems, where each node features multiple sockets of multi-core processors and, if available, accelerators like GPUs or Intel Xeon Phi coprocessors. DPLASMA achieves this objective by deploying PLASMA algorithms on distributed-memory systems using the state-of-the-art PaRSEC runtime.

In addition to traditional ScaLAPACK data distribution, DPLASMA provides interfaces for users to expose arbitrary data distributions. The algorithms operate transparently on local data or introduce implicit communications to resolve dependencies, thereby removing the burden of initial data reshuffle and providing the user with a novel approach to address load balance.

DTE

FIND OUT MORE AT
icl.utk.edu/dte

The Distributed Tasking for Exascale (DTE) project will extend the capabilities of ICL's Parallel Runtime and Execution Controller (PaRSEC)—a generic framework for architecture-aware scheduling and management of microtasks on distributed, many-core, heterogeneous architectures. The PaRSEC environment also provides a runtime component for dynamically executing tasks on heterogeneous distributed systems along with a productivity toolbox and development framework that supports multiple domain-specific languages (DSLs) and extensions and tools for debugging, trace collection, and analysis.

PaRSEC also enables fast prototyping DSLs to express the dependencies between tasks and provides a stable, scalable, and efficient distributed runtime so they can run on any execution platform at any scale. The underlying dataflow paradigm attacks both sides of the exascale challenge: managing extreme-scale parallelism and maintaining the performance portability of the code. The DTE award is a vital extension and continuation of this effort and will ensure that PaRSEC meets the critical needs of ECP application communities in terms of scalability, interoperability, and productivity.

EPEXA

FIND OUT MORE AT
www.icl.utk.edu/epexa/

A collaborative project involving Virginia Tech, Stony Brook, and ICL/UTK, the Ecosystem for Programming and Executing eXtreme Applications (EPEXA) aims to create a software framework that implements high-performance methods for irregular and dynamic computations that are poorly supported by current programming paradigms. Employing science-driven codesign, the EPEXA team will harden a successful research prototype into an accessible, production-quality programming model that will leverage domain-specific languages (DSLs) to improve accessibility and accelerate the adoption of high-performance tools for computer scientists and domain scientists.

The project bridges the so-called “valley of death” between a successful proof of concept and an implementation with enough quality, performance, and community support to motivate application scientists and other researchers to adopt it and push for its community use. Specifically, the new powerful data-flow programming model and associated parallel runtime directly address multiple challenges faced by scientists as they leverage rapidly changing computer technologies—including current massively parallel, hybrid, and many-core systems.

Evolve

Evolve, a collaborative effort between ICL and the University of Houston, expands the capabilities of Open MPI to support the NSF’s critical software-infrastructure missions. Core challenges include: extending the software to scale to 10,000–100,000 processes; ensuring support for accelerators; enabling highly asynchronous execution of communication and I/O operations; and ensuring resilience. Part of the effort involves careful consideration of modifications to the MPI specification to account for the emerging needs of application developers on future extreme-scale systems.

So far, Evolve efforts have involved exploratory research for improving different performance aspects of the Open MPI library. Notably, this has led to an efficiency improvement in multi-threaded programs using MPI in combination with other thread-based programming models (e.g., OpenMP). A novel collective communication framework with event-based programming and data dependencies was investigated, and it demonstrated a clear advantage in terms of aggregate bandwidth in heterogeneous (shared memory + network) systems. Support for MPI resilience following the User-Level Failure Mitigation (ULFM) fault-tolerance proposal was released based on the latest Open MPI version and will soon be fully integrated into Open MPI.

Exa-PAPI

AVAILABLE IN
PAPI 6.0.0
FIND OUT MORE AT
icl.utk.edu/exa-papi/

The Exascale Performance Application Programming Interface (Exa-PAPI) project is developing a new C++ Performance API (PAPI++) software package from the ground up that offers a standard interface and methodology for using low-level performance counters in CPUs, GPUs, on/off-chip memory, interconnects, and the I/O system—including energy/power management. PAPI++ is building upon classic-PAPI functionality and strengthening its path to exascale with a more efficient and flexible software design—a design that takes advantage of C++’s object-oriented nature but preserves the low-overhead monitoring of performance counters and adds a vast testing suite.

In addition to providing hardware counter–based information, a standardizing layer for monitoring software-defined events (SDE), which exposes the internal behavior of runtime systems and libraries (e.g., communication and math libraries) to the applications, is being incorporated. As a result, the notion of performance events is broadened from strictly hardware-related events to also include software-based information. Enabling monitoring of both hardware and software events provides more flexibility to developers when capturing performance information.

FFT

AVAILABLE IN
heFFTe 2.2
FIND OUT MORE AT
icl.utk.edu/fft/

The fast Fourier transform (FFT) is used in many domain applications—including molecular dynamics, spectrum estimation, fast convolution and correlation, signal modulation, and wireless multimedia applications, but current state-of-the-art FFT libraries are not scalable on large heterogeneous machines with many nodes.

The main objective of the ECP FFT project is to design and develop a Highly Efficient FFTs for Exascale (heFFTe) library that provides fast and robust multidimensional FFTs for large-scale heterogeneous systems with multi-core processors and hardware accelerators. HeFFTe collects and leverages existing FFT capabilities while building a sustainable FFT library that minimizes data movements, optimizes MPI communications, overlaps computations with communications, and autotunes performance on various architectures and large scale-platforms. The current heFFTe v2.2 release achieves very good scalability on pre-exascale systems, and a performance that is close to 90% of the roofline peak.

FFT BENCHMARK (FIBER)

FIND OUT MORE AT
fiber.icl.utk.edu

The new benchmarking effort aims to benchmark a variety of implementations of parallel FFT implementations for the purposes of the Exascale Computing Project (ECP). With many scientific applications of strategic importance to DOE need multi-dimension Discrete Fourier transforms and among many they include ICL's own heFFTe project. Over the course of the CLOVER project and more specifically its heFFTe component, the need for comparing FFT implementations became apparent to understand how the tradeoffs made by the developers affect the floating-point performance and mitigate communication overheads.

In its current incarnation, the FFT Benchmarking project released a technical report ICL-UT-21-03 to provide interim update on comprehensive testing and measuring various aspects of the FFT implementations for three-dimensional transforms on high performance systems. The results from the report reveal large differences in functionality, performance, and supported hardware and give the ECP application community a much better appreciation of the available software choices.

Ginkgo

VERSION
1.4.0
FIND OUT MORE AT
ginkgo-project.github.io

In the Ginkgo project, we develop high performance numerical linear algebra functionality reflecting the parallelism of modern HPC platforms. The focus is on GPU-accelerated systems, and Ginkgo can currently be used on AMD GPUs, Intel GPUs, and NVIDIA GPUs using backends written in the respective vendor languages. Ginkgo features a variety of iterative Krylov solvers, sophisticated preconditioners exposing fine-grain parallelism including incomplete factorizations, incomplete sparse approximate inverses and algebraic multigrid technology, mixed precision algorithms, and preconditioned batched iterative solvers. Ginkgo is implemented in modern C++ and features interfaces to several popular simulation frameworks including MFEM, deal.ii, HyTeg, and OpenFOAM. The Ginkgo library is open source and licensed under BSD 3-clause. In the development of Ginkgo, we aim for industry-level code quality standards including Continuous Integration (CI) and Continuous Benchmarking (CB), comprehensive unit test coverage, and fulfilling the community policies of the extreme-scale Scientific Software Development Kit (xSDK) and the Extreme Scale Scientific Software Stack (E4S).

HPCG

VERSION
3.1
FIND OUT MORE AT
www.hpcg-benchmark.org

The High Performance Conjugate Gradients (HPCG) benchmark is designed to measure performance that is representative of modern scientific applications relying on discretizations of Partial Differential Equations (PDEs). It does so by exercising the computational and communication patterns commonly found in real science and engineering codes, which are often based on sparse iterative solvers with complex multi-level preconditioners. HPCG exhibits the same irregular accesses to the main memory and fine-grain recursive computations that dominate large-scale scientific workloads used to simulate complex physical phenomena.

The HPCG 3.1 reference code was released in March of 2019. In addition to bug fixes, this release positioned HPCG to even better represent modern PDE solvers and made it easier to run HPCG on production supercomputing installations. The reference version is accompanied by multiple binary or source code releases from AMD, ARM, Intel, and NVIDIA, which are carefully optimized for the these vendors' respective hardware platforms. The current HPCG performance list was released at SC21 and now features over 200 entries from across the supercomputing landscape. HPCG results have also been tracked by TOP500.org since June of 2017.

HPL

VERSION
2.3
FIND OUT MORE AT
icl.utk.edu/hpl

The High Performance LINPACK (HPL) benchmark solves a dense linear system in double precision (64 bit) arithmetic on distributed-memory computers. HPL is written in a portable ANSI C and requires an MPI implementation and either BLAS or the Vector Signal and Image Processing Library (VSIPL). HPL is often one of the first programs to run on large HPC machines, producing a result that can be submitted to the TOP500 list of the world's fastest supercomputers. Carefully optimized versions of HPL are available from major HPC hardware vendors.

The major focus of HPL 2.3, released in 2018, was to improve the accuracy of reported benchmark results and ensure easier configuration and building on modern HPC platforms. HPL is now hosted on Github and features more detailed reporting of the solution's scaled residual and of the achieved performance number. Another addition is a software configuration tool based on GNU Autotools and the removal of deprecated MPI functions. The LINPACK app for iOS achieved over 8 Gflop/s on the iPhone X. For the November 2021 TOP500 list, an optimized version of the HPL code achieved over 440 Pflop/s on the Fugaku supercomputer at RIKEN, Japan.

HPL-AI

VERSION
2021.05.02
FIND OUT MORE AT
icl.bitbucket.io/hpl-ai

The High Performance LINPACK for Accelerator Introspection (HPL-AI) benchmark seeks to highlight the convergence of HPC and AI workloads based on machine learning (ML) and deep learning (DL) by solving a system of linear equations using novel, mixed-precision algorithms that exploit modern hardware. While traditional HPC focuses on simulation runs for modeling phenomena in a variety of scientific disciplines, the mathematical models that drive these computations mostly require 64-bit accuracy. However, the ML/DL methods that fuel advances in AI achieve the desired results at 32-bit or lower precisions. This lesser demand for working precision fueled a resurgence of interest in new hardware accelerators that deliver unprecedented performance levels and energy savings to achieve the classification and recognition fidelity afforded by higher-accuracy formats on traditional hardware.

HPL-AI strives to unite these realms by connecting its solver formulation to the decades-old HPL framework of benchmarking supercomputers. A number of large-scale HPC installations have now been benchmarked with HPL-AI, starting with ORNL's Summit machine in 2019 and now including RIKEN's Fugaku supercomputer, which achieved 2 Eflop/s in mixed-precision performance. The growing list of machines with both HPL and HPL-AI results was 18-entries long as of the November 2021 release of the HPL-AI's bi-annual ranking.

RESEARCH

LAPACK/ ScaLAPACK

LAPACK VERSION
3.10
FIND OUT MORE AT
www.netlib.org/lapack/

ScaLAPACK VERSION
2.1.0
FIND OUT MORE AT
www.netlib.org/scalapack/

The Linear Algebra PACKage (LAPACK) and Scalable LAPACK (ScaLAPACK) are widely used libraries for efficiently solving dense linear algebra problems. ICL has been a major contributor to the development and maintenance of these two packages since their inception. LAPACK is sequential, relies on the BLAS library, and benefits from the multi-core BLAS library. ScaLAPACK is parallel, distributed, and relies on the BLAS, LAPACK, MPI, and BLACS libraries.

LAPACK 3.9.0, released in November 2019, adds a QR-preconditioned QR SVD method and an LAPACK Householder reconstruction routine. Since 2011, LAPACK has included LAPACKE, a native C interface for LAPACK developed in collaboration with Intel, which provides NAN check and automatic workspace allocation. ScaLAPACK 2.1.0, which includes a new robust ScaLAPACK routine for computing the QR factorization with column pivoting along with improved accuracy of the Frobenius norm, was released in November 2019.

MAGMA

VERSION
2.6.2
FIND OUT MORE AT
www.icl.utk.edu/magma

Matrix Algebra on GPU and Multi-core Architectures (MAGMA) is a collection of next-generation linear algebra libraries for heterogeneous computing. The MAGMA package supports interfaces for current linear algebra packages and standards (e.g., LAPACK and BLAS) to enable computational scientists to easily port any linear algebra–reliant software components to heterogeneous computing systems. MAGMA enables applications to fully exploit the power of current hybrid systems of many-core CPUs and multi-GPUs/coprocessors to deliver the fastest possible time to accurate solution within given energy constraints.

MAGMA features LAPACK-compliant routines for multi-core CPUs enhanced with NVIDIA or AMD GPUs. MAGMA 2.6.2 now includes more than 400 routines that cover one-sided dense matrix factorizations and solvers, two-sided factorizations, and eigen/singular-value problem solvers, as well as a subset of highly optimized BLAS for GPUs. A MagmaDNN package has been added and further enhanced to provide high-performance data analytics, including functionalities for machine learning applications that use MAGMA as their computational back end. The MAGMA Sparse and MAGMA Batched packages have been included since MAGMA 1.6.

MATEDOR

AVAILABLE IN
MAGMA 2.6.2
FIND OUT MORE AT
www.icl.utk.edu/matedor

The MAtrix, TEnsor, and Deep-learning Optimized Routines (MATEDOR) project is performing the research required to define a standard interface for batched operations and provide a performance-portable software library that demonstrates batching routines for a significant number of kernels. This research is critical, given that the performance opportunities inherent in solving many small batched matrices often yield more than a 10x speedup over the current classical approaches.

Working closely with affected application communities, along with ICL's Batched BLAS initiative, MATEDOR will define modular, optimizable, and language-agnostic interfaces that can work seamlessly with a compiler. This modularity will provide application, compiler, and runtime system developers with the option to use a single call to a routine from the new batch operation standard and would allow the entire linear algebra community to collectively attack a wide range of small matrix or tensor problems.

OMPI-X

FIND OUT MORE AT
www.icl.utk.edu/ompi-x

The Open MPI for Exascale (OMPI-X) project focuses on preparing the Message Passing Interface (MPI) standard—and its implementation in Open MPI—for exascale through improvements in scalability, capability, and resilience. Since its inception, the MPI standard has become ubiquitous in high-performance parallel computational science and engineering, and Open MPI is a widely used, high-quality, open-source implementation of the MPI standard. Despite their history and popularity, however, neither Open MPI nor the MPI standard itself is currently ready for the changes in hardware and software that will accompany exascale computing.

To mitigate this concern, OMPI-X will address a broad spectrum of issues in both the standard and the implementation by ensuring runtime interoperability for MPI+X and beyond, extending the MPI standard to better support coming exascale architectures, improving Open MPI scalability and performance, supporting more dynamic execution environments, enhancing resilience in MPI and Open MPI, evaluating MPI tools interfaces, and maintaining quality assurance.

oneAPI

oneAPI is an industry initiative to create a single, unified, cross-architecture programming model for CPUs and accelerator architectures. Based on industry standards and its open development approach, the initiative will help streamline software development for high performance computers, increase performance and provide specifications for efficient and diverse architecture programming.

In the oneAPI Center of Excellence, Intel teams up with the Innovative Computing Lab to spearhead the development of numerical linear algebra functionality in the oneAPI ecosystem by deploying a DPC++ kernels in the Ginkgo linear algebra library. This allows to run Ginkgo and applications that use Ginkgo's functionality on any hardware supporting the oneAPI industry standard, including embedded Intel GPUs and Intel's discrete GPUs that are expected to power the Aurora supercomputer.

Open MPI

VERSION
4.1.0
FIND OUT MORE AT
www.open-mpi.org

The Open MPI Project is an open-source Message Passing Interface (MPI) implementation developed and maintained by a consortium of academic, research, and industry partners. MPI primarily addresses the message-passing parallel programming model, in which data is moved from the address space of one process to that of another process through cooperative operations on each process. Open MPI integrates technologies and resources from several other projects (e.g., HARNESS/FT-MPI, LA-MPI, LAM/MPI, and PACX-MPI) in order to build the best MPI library available.

A completely new MPI 3.2–compliant implementation, Open MPI offers advantages for system and software vendors, application developers, and computer science researchers. ICL's efforts in the context of Open MPI have significantly improved its scalability, performance on many-core environments, and architecture-aware capabilities—such as adaptive shared memory behaviors and dynamic collective selection—making it ready for next-generation exascale challenges.

PAPI

VERSION
6.0.0
FIND OUT MORE AT
www.icl.utk.edu/papi

The Performance Application Programming Interface (PAPI) supplies a consistent interface and methodology for collecting performance counter information from various hardware and software components, including most major CPUs, GPUs and accelerators, interconnects, I/O systems, and power interfaces, as well as virtual cloud environments. Industry liaisons with AMD, Cray, Intel, IBM, NVIDIA, and others ensure seamless integration of PAPI with new architectures at or near their release. As the PAPI component architecture becomes more populated, performance tools that interface with PAPI automatically inherit the ability to measure these new data sources.

In 2020, ICL and collaborators at the University of Maine worked on PAPI-Ex to build support for performance counters available in the latest generations of CPUs and GPUs, develop support for system-wide hardware performance counter monitoring, and strengthen the sampling interface in PAPI. PAPI 6.0 includes a new API for SDEs, a major revision of the "high-level API," and several new components, including ROCM and ROCM_SMI, powercap_ppc and sensors_ppc, SDE, and I/O. PAPI 6.0 also ships with a new Counter Analysis Toolkit (CAT) that assists with native performance counter disambiguation through micro-benchmarks.

PaRSEC

FIND OUT MORE AT
www.icl.utk.edu/parsec

The Parallel Runtime Scheduling and Execution Controller (PaRSEC) is a generic framework for architecture-aware scheduling and management of microtasks on distributed many-core heterogeneous architectures. Applications considered are expressed as a DAG of tasks with edges designating the data dependencies. DAGs are represented in a compact, problem-size independent format that can be queried to discover data dependencies in a totally distributed fashion—a drastic shift from today's programming models, which are based on sequential flow of execution.

PaRSEC orchestrates the execution of an algorithm on a particular set of resources, assigns computational threads to the cores, overlaps communications and computations, and uses a dynamic, fully distributed scheduler. PaRSEC includes a set of tools to generate the DAGs and integrate them into legacy codes, a runtime library to schedule the microtasks on heterogeneous resources, and tools to evaluate and visualize the efficiency of the scheduling. Many dense and sparse linear algebra extensions have been implemented, as well as chemistry and seismology applications, which produced significant speedup in production codes.

PLASMA

VERSION
21.8.29
FIND OUT MORE AT
www.bitbucket.org/icl/plasma

The Parallel Linear Algebra Software for Multi-core Architectures (PLASMA) implements a set of fundamental linear algebra routines using the latest updates to the OpenMP standard. PLASMA includes, among others, LAPACK-equivalent routines for solving linear systems of equations, linear least square problems, parallel BLAS, and parallel matrix norms.

Over the last decade, PLASMA has been used on a variety of systems using Intel CPUs and coprocessors as well as AMD, IBM POWER, and ARM processors. As a research vehicle, PLASMA continues as an example of modern design for new dense linear algebra algorithms. At the same time, PLASMA benefits from the continuous evolution of the OpenMP standard that now includes offload functionality and enables porting to hardware accelerators. The latest PLASMA release, version 21.8.29 from August 2021, added transpose option to xGETRS() and xGELS() functions, convenience scripts for C and Fortran examples, and a Python script for quickly launching tests in addition to fixing bugs for corner cases in LU and norm functions.

PULSE

AVAILABLE IN
PAPI 6.0.0
FIND OUT MORE AT
www.icl.utk.edu/pulse

The PAPI Unifying Layer for Software-defined Events (PULSE) project focuses on enabling cross-layer and integrated monitoring of whole application performance by extending PAPI with the capability to expose performance metrics from key software components found in the HPC software stack. Up to this point, the abstraction and standardization layer provided by PAPI has been limited to profiling information generated by hardware only. Information about the behavior of the underlying software stack had to be acquired either through low-level binary instrumentation or through custom APIs.

To overcome this shortfall, PULSE is extending the abstraction and unification layer that PAPI has provided for hardware events to also encompass software events. On one end, PULSE offers a standard, well-defined and well-documented API that high-level profiling software can utilize to acquire performance information about the libraries used by an application and present it to the application developers. On the other end, it provides standard APIs that library and runtime writers can utilize to communicate information about the behavior of their software to higher software layers.

REE-HPC

The Numerically-Exact Relativistic Many-Body Electronic Structure of Heavy Elements project is a collaborative effort focused on development of a completely new and novel computational tool that enables, for the first time, fully-predictive calculations on molecules containing f-block elements. This effort includes high-performance distributed and heterogenous computing to assist with tuning the implementation of novel methods for execution on large-scale HPC platforms, including exascale machines.

This effort enables advances in global needs involving Rare Earth Elements (REEs) and actinides. REEs are essential in applied technologies that include communications, computing, medical capabilities, green energy, and defense. Actinides are amongst the least studied elements, with many being essential in nuclear efforts relevant to defense, energy, and medical treatments.

ScaRT

FIND OUT MORE AT
www.icl.utk.edu/scart

The Scalable Run Time for Highly Parallel, Heterogeneous Systems (ScaRT) project aims to increase the scientific throughput of existing and future cyberinfrastructure platforms by reducing communication overheads; improving the match between modern, parallel-computing frameworks and the applications running on top; and by better matching the functionality of the underlying communication library to the capabilities of modern communication adapters.

To this end, SCaRT brings together a multidisciplinary team to (1) design and implement a communication library with new communication primitives; (2) accelerate multiple task-based runtimes (e.g., Legion and PaRSEC) and communication libraries (e.g., MPI and GasNET); (3) port key components to a programmable NIC; and (4) deliver improvements and extensions to mainstream communication libraries to provide the new functionality.

RESEARCH

SLATE

VERSION
2020.10.00
FIND OUT MORE AT
icl.utk.edu/slate/

For decades, ICL has applied algorithmic and technological innovations to the process of pioneering, implementing, and disseminating dense linear algebra software—including the Linear Algebra PACKage (LAPACK) and Scalable Linear Algebra PACKage (ScaLAPACK) libraries. The Software for Linear Algebra Targeting Exascale (SLATE) project will converge and consolidate that software into a dense linear algebra library that will integrate seamlessly into the ECP ecosystem.

For context, ScaLAPACK was first released in 1995, some 25 years ago. In the past two decades, HPC has witnessed tectonic shifts in the hardware technology, followed by paradigm shifts in the software technology, and a plethora of algorithmic innovations in scientific computing. At the same time, no viable replacement for ScaLAPACK emerged. SLATE is meant to be this replacement, boasting superior performance and scalability in the modern, heterogeneous, distributed-memory environments of HPC.

SMURFS

FIND OUT MORE AT
www.icl.utk.edu/smurfs

The Simulation and Modeling for Understanding Resilience and Faults at Scale (SMURFS) project seeks to acquire the predictive understanding of the complex interactions of a given application, a given real or hypothetical hardware and software environment, and a given fault-tolerance strategy at extreme scale.

SMURFS is characterized by two facets: (1) medium and fine-grained predictive capabilities and (2) coarse-grained fault tolerance strategy selection. Accordingly, ICL plans to design, develop, and validate new analytical and system component models that use semi-detailed software and hardware specifications to predict application performance in terms of time to solution and energy consumption. Also, based on a comprehensive set of studies using several application benchmarks, proxies, full applications, and several different fault tolerance strategies, ICL will gather valuable insights about application behavior at scale.

SBI

FIND OUT MORE AT
www.icl.utk.edu/sbi

The DOE-funded Surrogate Benchmark Initiative (SBI) is a collaborative effort involving Indiana University, UTK/ICL, Virginia University, and Rutgers University that aims to provide new benchmarks and tools for assessing deep neural network “surrogate” models. Trained on data produced by ensemble runs of a given HPC simulation, a surrogate model can imitate-with high fidelity-part or all of that simulation and produce the same outcomes for a given set of inputs while requiring far less time and energy.

At present, however, there are no accepted benchmarks to evaluate these surrogate models, and there is no easy way to measure progress or inform the codesign of new HPC systems to support their use. SBI aims to address this fundamental problem by creating a community repository and a Findable, Accessible, Interoperable, and Reusable (FAIR) data ecosystem for HPC application surrogate benchmarks, including data, code, and all relevant collateral artifacts that the science and engineering community needs to use and reuse these data sets and surrogates.

TOP500

FIND OUT MORE AT
www.top500.org

With over three decades of tracking the progress of high performance computing, the TOP500 list continues to provide a reliable historical record of supercomputers around the world. The list clearly lays out critical HPC metrics across all of its 500 machines and draws a rich picture of the state of the art in terms of performance, energy consumption, and power efficiency. The TOP500 now features an HPCG ranking, which measures a machine's performance using irregular accesses to memory and fine-grain recursive computations- the very factors that dominate real-world, large-scale scientific workloads.

In November 2021, the 58th TOP500 list was unveiled during the International Conference for High Performance Computing, Network, Storage, and Analysis (SC21), which was held in Saint Louis, Missouri, and online. Yet again, Japan took the crown with Fugaku, their new ARM-based machine built by Fujitsu. The system's 158,976 A64FX SoCs propelled Fugaku to 442 Pflop/s in the HPL benchmark, making it the fastest supercomputer in the world by a factor of 2.8x over the United States' Summit machine and the first ARM system to achieve the number one spot on the TOP500. Fugaku now held the first spot for 4 consecutive lists.

ULFM

VERSION
4.1.0
FIND OUT MORE AT
fault-tolerance.org

User Level Failure Mitigation (ULFM) is a set of new interfaces for MPI that enables message passing applications to restore MPI functionality affected by process failures. The MPI implementation is spared the expense of internally taking protective and corrective automatic actions against failures. Instead, it can prevent any fault-related deadlock situation by reporting operations wherein the completions were rendered impossible by failures.

Using the constructs defined by ULFM, applications and libraries drive the recovery of the MPI state. Consistency issues resulting from failures are addressed according to an application's needs, and the recovery actions are limited to the necessary MPI communication objects. A wide range of application types and middlewares are already building on top of ULFM to deliver scalable and user-friendly fault tolerance. Notable recent additions include the CoArray Fortran language and SAP databases. ULFM software is available in recent versions of both MPICH and Open MPI.

xSDK4ECP

VERSION
0.7
FIND OUT MORE AT
xsdk.info/ecp

The Extreme-Scale Scientific Software Development Kit for the Exascale Computing Project (xSDK4ECP) is a collaborative effort between Argonne National Laboratory (ANL), ICL, Kahrlsruhe Institute of Technology, Lawrence Berkeley National Laboratory (LBNL), Lawrence Livermore National Laboratory (LLNL), Sandia National Laboratory (SNL), and the University of California, Berkeley. The project aims to enable seamless integration and combined use of diverse, independently developed software packages in ECP applications.

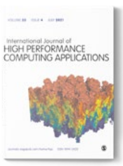
To ensure the consistency of naming conventions, runtime behavior, and installation procedures, xSDK informs the project development process by providing requirements and guidelines that are influential throughout the software development phase. xSDK lightens the burden on system administrators and application developers, because each xSDK package provides a Spack installation script that can be invoked independently or through the installation of the xSDK's Spack package. In addition, xSDK now ships with a set of curated examples that show potential integrations of packages into application exemplars. ICL's MAGMA, PLASMA, SLATE, and heFFTe libraries are now all included in the most recent release, xSDK 0.7.

PUBLICATIONS

Evidence of our contributions to the HPC community might be best exemplified by the numerous publications we produce every year. Here is a listing of our 2021 publications, including journal articles, book chapters, and conference proceedings. Many of these are available for download from our website.

Abdelfattah, A., M. Al Farhan, C. Brown, M. Gates, D. Sukkari, A. YarKhan, and J. Dongarra
SLATE Port to AMD and Intel Platforms
SLATE Working Notes, no. 16, ICL-UT-21-01, April 2021

Abdelfattah, A., T. Costa, J. Dongarra, M. Gates, A. Haidar, S. Hammarling, N. J. Higham, J. Kurzak, P. Luszczek, S. Tomov, et al.
A Set of Batched Basic Linear Algebra Subprograms and LAPACK Routines
ACM Transactions on Mathematical Software, vol. 47, no. 3, 2021



Abdelfattah, A., H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, A. Fox, M. Gates, N. J. Higham, X. S. Li, et al.
A survey of numerical linear algebra methods utilizing mixed-precision arithmetic
The International Journal of High Performance Computing Applications, vol. 35, no. 4, 2021

Abdelfattah, A., V. Barra, N. Beams, R. Bleile, J. Brown, J-S. Camier, R. Carson, N. Chalmers, V. Dobrev, Y. Dudouit, et al.
GPU algorithms for Efficient Exascale Discretizations
Parallel Computing, vol. 108, December 2021

Akbudak, K., P. Bagwell, S. Cayrols, M. Gates, D. Sukkari, A. YarKhan, and J. Dongarra
SLATE Performance Improvements: QR and Eigenvalues
SLATE Working Notes, no. 17, ICL-UT-21-02, April 2021

Ayala, A., S. Tomov, P. Luszczek, S. Cayrols, G. Ragghianti, and J. Dongarra
Interim Report on Benchmarking FFT Libraries on High Performance Systems
Innovative Computing Laboratory Technical Report, no. ICL-UT-21-03: University of Tennessee, July 2021

Ayala, A., S. Tomov, M. Stoyanov, and J. Dongarra
Scalability Issues in FFT Computation
International Conference on Parallel Computing Technologies: Springer, pp. 279–287, September 2021

Bak, S., O. Hernandez, M. Gates, P. Luszczek, and V. Sarkar
Task-graph scheduling extensions for efficient synchronization and communication
Proceedings of the ACM International Conference on Supercomputing, June 2021

Barry, D., A. Danalis, and H. Jagode
Effortless Monitoring of Arithmetic Intensity with PAPI's Counter Analysis Toolkit
Tools for High Performance Computing 2018/2019: Springer, pp. 195–218, 2021

Bosilca, G., A. Bouteiller, T. Herault, V. Le Fèvre, Y. Robert, and J. Dongarra
Revisiting Credit Distribution Algorithms for Distributed Termination Detection
2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW): IEEE, pp. 611–620, 2021

Brown, J., A. Abdelfattah, V. Barra, N. Beams, J-S. Camier, V. Dobrev, Y. Dudouit, L. Ghaffari, T. Kolev, D. Medina, et al.
libCEED: Fast algebra for high-order element-based discretizations
Journal of Open Source Software, vol. 6, no. 63, pp. 2945, 2021

Cao, Q., Y. Pei, K. Akbudak, G. Bosilca, H. Ltaief, D. Keyes, and J. Dongarra
Leveraging PaRSEC Runtime Support to Tackle Challenging 3D Data-Sparse Matrix Problems
35th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2021), Portland, OR, IEEE, May 2021

Dongarra, J., M. Gates, P. Luszczek, and S. Tomov
Translational process: Mathematical software perspective
Journal of Computational Science, vol. 52, pp. 101216, 2021

Herault, T., Y. Robert, G. Bosilca, R. Harrison, C. Lewis, E. Valeev, and J. Dongarra
Distributed-Memory Multi-GPU Block-Sparse Tensor Contraction for Electronic Structure
35th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2021), Portland, OR, IEEE, May 2021

Hoemmen, M., D. Hollman, C. Trott, D. Sunderland, N. Liber, L-T. Lo, D. Lebrun-Grandie, G. Lopez, P. Caday, S. Knepper, et al.
P1673R3: A Free Function Linear algebra Interface Based on the BLAS
ISO JTC1 SC22 WG22, no. P1673R3: ISO, April 2021

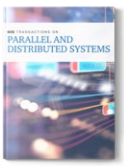
Hori, A., E. Jeannot, G. Bosilca, T. Ogura, B. Gerofi, J. Yin, and Y. Ishikawa
An international survey on MPI users
Parallel Computing, vol. 108, December 2021

Iqbal, Z., S. Nooshabadi, I. Yamazaki, S. Tomov, and J. Dongarra
Exploiting Block Structures of KKT Matrices for Efficient Solution of Convex Optimization Problems
IEEE Access, 2021

Jagode, H., H. Anzt, H. Ltaief, and P. Luszczek
Lecture Notes in Computer Science: High Performance Computing, vol. 12761
Springer International Publishing, 2021

Kolev, T., P. Fischer, M. Min, J. Dongarra, J. Brown, V. Dobrev, T. Warburton, S. Tomov, M. S. Shephard, A. Abdelfattah, et al.
Efficient exascale discretizations: High-order finite element methods
The International Journal of High Performance Computing Applications, vol. 35 issue 6, November 2021

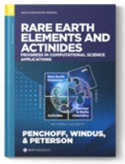
Kovalchuk, S. V., V. V. Krzhizhanovskaya, PMA. Sloot, G. Závodszky, M. H. Lees, M. Paszyński, and J. Dongarra
20 years of computational science: Selected papers from 2020 International Conference on Computational Science
Journal of Computational Science, vol. 53, July 2021



Lindquist, N., P. Luszczek, and J. Dongarra
Accelerating Restarted GMRES with Mixed Precision Arithmetic
IEEE Transactions on Parallel and Distributed Systems, vol. 33 issue 4, June 2021

Penchoff, D. A., E. Valeev, H. Jagode, P. Luszczek, A. Danalis, G. Bosilca, R. J. Harrison, J. Dongarra, and T. L. Windus
An Introduction to High Performance Computing and Its Intersection with Advances in Modeling Rare Earth Elements and Actinides
Rare Earth Elements and Actinides: Progress in Computational Science Applications, vol. 1388, Washington, DC, American Chemical Society, Oct 2021

Penchoff, D. A., C. B. Sims, and T. L. Windus
Rare Earth Elements and Critical Materials: Uses and Availability
Rare Earth Elements and Actinides: Progress in Computational Science Applications, vol. 1388, Washington, DC, American Chemical Society, pp. 63-74, October 2021

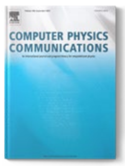


Penchoff, D. A., T. L. Windus, and C. C. Peterson
Rare Earth Elements and Actinides: Progress in Computational Science Applications
ACS Symposium Series, vol. 1388, Washington, DC, American Chemical Society, October 2021

Schuchart, J., P. Samfass, C. Niethammer, J. Gracia, and G. Bosilca
Callback-based completion notification using MPI Continuations
Parallel Computing, vol. 106, September 2021

Schuchart, J., C. Niethammer, J. Gracia, and G. Bosilca
Quo Vadis MPI RMA? Towards a More Efficient Use of MPI One-Sided Communication
EuroMPI'21, Garching, Munich Germany, 2021

Sharp, D., M. Stoyanov, S. Tomov, and J. Dongarra
A More Portable HeFFTe: Implementing a Fallback Algorithm for Scalable Fourier Transforms
ICL Technical Report, no. ICL-UT-21-04: University of Tennessee, August 2021



Spannaus, A., K. J. H. Law, P. Luszczek, F. Nasrin, C. Putman Micucci, P. K. Liaw, L. J. Santodonato, D. J. Keffer, and V. Maroulas
Materials fingerprinting classification
Computer Physics Communications, vol. 266, September 2021

Tsai, Y. M., P. Luszczek, and J. Dongarra
Mixed-Precision Algorithm for Finding Selected Eigenvalues and Eigenvectors of Symmetric and Hermitian Matrices
ICL Technical Report, no. ICL-UT-21-05, August 2021

EVENTS

Each year, members of our research staff attend national and international conferences, workshops, and seminars. These meetings provide opportunities to present our research, share our knowledge, and exchange ideas with leading computational science researchers from around the world. While a majority of these events were virtual in 2021, participation in the HPC community in this way remains an important aspect of ICL's activities.

FEBRUARY 22–25
MPI Forum Virtual Meeting

MARCH 1–5
SIAM Conference on Computational Science and Engineering (CSE21)

APRIL 5–9
The University of Arkansas Department of Mathematical Sciences 46th Annual Spring Lecture Series

APRIL 12–16
2021 ECP Annual Meeting

APRIL 12–16
GPU Technology Conference (GTC 2021)

APRIL 26–30
Salishan Conference on High Speed Computing

MAY 8–12
Asia Supercomputer Community Student Supercomputer Challenge Competition (ASC20-21)

MAY 13–17
IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID 2020)

MAY 17–21
SIAM Conference on Applied Linear Algebra (LA21)

MAY 17–21
35th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2021)

JUNE 7–9
MPI Forum Virtual Meeting

JUNE 24 – JULY 2
ISC High Performance 2021 Digital

JULY 5–9
The Platform for Advanced Scientific Computing (PASC) Conference

JULY 15–16
SAI Computing Conference 2021

JULY 19–21
2021 SIAM Annual Meeting

JULY 19–23
Supercomputing Frontiers Europe 2021

JULY 20–22
2021 Collegeville Workshop on Scientific Software

AUGUST 1–13
Argonne Training Program on Extreme-Scale Computing (ATPESC)

AUGUST 9–12
50th International Conference on Parallel Processing

AUGUST 25
American Chemical Society Fall Meeting 2021

SEPTEMBER 8–10
MPI Forum Virtual Meeting

SEPTEMBER 27–28
Russian Supercomputing Days 2021

OCTOBER 20
NJIT Institute for Data Science Seminar Series

OCTOBER 27–28
CnC 2021: The Thirteenth Annual Concurrent Collections Workshop

NOVEMBER 14–16
International Symposium on Benchmarking, Measuring and Optimizing (Bench'21)

NOVEMBER 14–19
Supercomputing 2021

NOVEMBER 19
ScalA21, the 12th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems

DECEMBER 6–9
MPI Forum Virtual Meeting

SC21

This year saw the first Hybrid experience for the International Conference for High Performance Computing Networking, Storage, and Analysis (SC21), held in St. Louis, MO on November 14 – 19. After a two-year hiatus from physical attendance the conference hosted more than 3,200 in-person attendees and 160 exhibitors with another 3,350 virtual attendees and 40 virtual exhibitors. Vaccine verification, face masks, and social distancing helped to ensure the safety of attendees and the mix of in-person and virtual events provided an environment that was every bit as creative, inspiring, and meaningful as past SC conferences.

The University of Tennessee groups represented this year were ICL, GCL (Global Computing Laboratory), and the UT Chattanooga Sim Center. Even with the Hybrid nature of the event, ICL was well represented in-person with faculty, research staff, and students giving booth talks, presenting papers and workshops, and leading "Birds of a Feather" sessions.

The dedicated ICL@SC webpage was once again running for the event. There, interested parties could keep tabs on ICL-related events during the conference. The page hosted a list of attendees, detailed schedule of talks, and the latest project handouts.

With the return to in-person attendance, members of ICL and other related attendees were able to meet up for dinner in St. Louis. The group included current and alumni ICL employees, as well as current GCL members, with roughly twenty-five attendees. Held at the Hilton 360 rooftop on Wed, Nov 17, the dinner provided an atmosphere of camaraderie and cooperation. It was a great opportunity for attendees to talk about future collaborations and to maintain relationships with those who are now working in the industry outside of ICL.

PARTNERSHIPS

ICL fosters relationships with many academic institutions and research centers and has proactively built enduring partnerships with HPC vendors and industry leaders in the United States and abroad.

GOVERNMENT & ACADEMIC PARTNERSHIPS

- Argonne National Laboratory
- Ames Laboratory
- Emory University
- Georgia Tech
- Indiana University
- Iowa State University
- JICS Joint Institute for Computational Sciences
- Berkeley Lab
- Lawrence Livermore National Laboratory
- Los Alamos National Laboratory
- Michigan Technological University
- NCSA
- NICS
- Northwestern University
- Oak Ridge National Laboratory
- Old Dominion University
- Pacific Northwest National Laboratory
- Rice
- Rutgers, The State University of New Jersey
- SDSC San Diego Supercomputer Center
- Sandia National Laboratories
- Stony Brook University
- Texas A&M University
- Berkeley University of California
- UC Riverside
- The University of Chicago
- University of Colorado Denver
- University of Florida
- University of Houston
- Illinois University of Illinois at Urbana-Champaign
- The University of Maine
- The University of New Mexico
- USC University of Southern California
- The University of Utah
- University of Washington
- Virginia Tech

In June 2018, Prof. Michela Taufer joined UTK's Department of Electrical Engineering and Computer Science and relocated the Global Computing Laboratory (GCL) to the Min H. Kao Electrical Engineering and Computer Science Building. The Global Computing Laboratory focuses on various aspects of HPC and scientific computing—including computational chemistry and chemical engineering, pharmaceutical sciences, seismology, and mathematics.

INDUSTRY PARTNERSHIPS

AMD
Atos
CRAY THE SUPERCOMPUTER COMPANY
ENGILITY
EPAnalytics
IBM
IERUS TECHNOLOGIES
intel
MathWorks
nag
NVIDIA

INTERNATIONAL COLLABORATORS

Barcelona Supercomputing Center
BARCELONA, SPAIN

Central Institute for Applied Mathematics
JÜLICH, GERMANY

Doshisha University
KYOTO, JAPAN

École Normale Supérieure de Lyon
LYON, FRANCE

École Polytechnique Fédérale de Lausanne
LAUSANNE, SWITZERLAND

ETH Zürich
ZÜRICH, SWITZERLAND

European Centre for Research and Advanced Training in Scientific Computing TOULOUSE, FRANCE

European Exascale Software Initiative
EUROPEAN UNION

Forschungszentrum Jülich
JÜLICH, GERMANY

High Performance Computing Center Stuttgart
STUTTGART, GERMANY

INRIA FRANCE

Karlsruhe Institute of Technology
KARLSRUHE, GERMANY

Kasetsart University
BANGKOK, THAILAND

King Abdullah University of Science and Technology THUWAL, SAUDI ARABIA

Laboratoire d'Informatique de Paris 6 (LIP6)
PARIS, FRANCE

Moscow State University
MOSCOW, RUSSIA

National Institute of Advanced Industrial Science and Technology TSUKUBA, JAPAN

Parallel and HPC Application Software Exchange TSUKUBA, JAPAN

Prometeus GmbH
MANNHEIM, GERMANY

Regionales Rechenzentrum Erlangen (RRZE)
ERLANGEN, GERMANY

RIKEN
WAKŌ, JAPAN

Rutherford Appleton Laboratory
OXFORD, ENGLAND

Soongsil University
SEOUL, SOUTH KOREA

Technische Universität Wien
VIENNA, AUSTRIA

Technische Universität Dresden
DRESDEN, GERMANY

Tokyo Institute of Technology
TOKYO, JAPAN

Umeå University
UMEÅ, SWEDEN

Université Claude Bernard Lyon 1
LYON, FRANCE

University of Bordeaux
BORDEAUX, FRANCE

University of Cape Town
CAPE TOWN, SOUTH AFRICA

University of Manchester
MANCHESTER, ENGLAND

University of Paris-Sud
PARIS, FRANCE

University of Picardie Jules Verne
AMIENS, FRANCE

University of Tsukuba
TSUKUBA, JAPAN

LEADERSHIP

THE UNIVERSITY OF TENNESSEE KNOXVILLE

CENTER FOR INFORMATION TECHNOLOGY RESEARCH

FIND OUT MORE AT
https://citr.utk.edu/

CENTER FOR INFORMATION TECHNOLOGY RESEARCH

The Center for Information Technology Research (CITR) was established in 2001 to drive the growth and development of leading-edge information technology research at UTK. CITR's primary objective is to develop a thriving, well-funded community in basic and applied information technology research to help the university capitalize on the rich supply of opportunities that now exist in this area. As part of this goal, CITR staff members currently provide primary administrative and technical support for ICL, helping maintain the lab's status as a world leader in high-performance and scientific computing. CITR also provides administrative support for the Interdisciplinary Graduate Minor in Computational Science at UTK.

JLESC

FIND OUT MORE AT
https://jlesc.github.io/

JOINT LABORATORY FOR EXTREME SCALE COMPUTING

ICL is an Associate Partner of the Joint Laboratory for Extreme Scale Computing (JLESC). JLESC, founded in 2009 by the French Institute for Research in Computer Science and Automation (INRIA) and the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign, is an international, virtual organization that aims to enhance the ability of member organizations and investigators to overcome software challenges found in extreme-scale, high-performance computers.

JLESC engages computer scientists, engineers, application scientists, and industry leaders to ensure that the research facilitated by the joint laboratory addresses science and engineering's most critical needs and takes advantage of the continuing evolution of computing technologies. Other partners include Argonne National Laboratory, the Barcelona Supercomputing Center, the Jülich Supercomputing Center, and the RIKEN Center for Computational Science.

INTERDISCIPLINARY GRADUATE MINOR IN COMPUTATIONAL SCIENCE

Addressing the need for a curriculum in computational science, CITR worked with faculty and administrators from several departments and colleges in 2007 to help establish a university-wide program that supports advanced degree concentrations in this growing field. With the Interdisciplinary Graduate Minor in Computational Science (IGMCS), students pursuing advanced degrees in a variety of fields of science and engineering can enhance their education with special courses of study that teach them both the fundamentals and the latest ideas and techniques from this new era of information-intensive research. The IGMCS curriculum, requirements, and policies are governed by a program committee composed of faculty members from participating IGMCS academic units and departments.

2021 IGMCS PARTICIPATING UNITS & DEPARTMENTS

- ANTHROPOLOGY
- BIOCHEMISTRY AND CELLULAR AND MOLECULAR BIOLOGY
- BREDESEN CENTER
- CHEMICAL AND BIOMOLECULAR ENGINEERING
- CHEMISTRY
- CIVIL AND ENVIRONMENTAL ENGINEERING
- EARTH AND PLANETARY SCIENCES
- ECOLOGY AND EVOLUTIONARY BIOLOGY
- ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
- GENOME SCIENCE & TECHNOLOGY
- GEOGRAPHY
- INDUSTRIAL AND SYSTEMS ENGINEERING
- INFORMATION SCIENCES
- MATERIALS SCIENCE AND ENGINEERING
- MATHEMATICS
- MECHANICAL, AEROSPACE AND BIOMEDICAL ENGINEERING
- MICROBIOLOGY
- NUCLEAR ENGINEERING
- PHYSICS
- STATISTICS
- UT INSTITUTE OF AGRICULTURE
- UT SPACE INSTITUTE (UTSI)

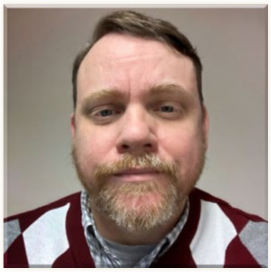
PEOPLE

STAFF AND STUDENTS

As the HPC landscape continues to evolve rapidly, remaining at the forefront of discovery requires great vision and skill. To address this evolution and to remain a leader in innovation, we have assembled a staff of top researchers from all around the world who apply a variety of novel and unique approaches to the challenges and problems inherent in world-class scientific computing.

Ahmad ABDELFATTAH
RESEARCH SCIENTIST II

Kadir AKBUDAK
RESEARCH SCIENTIST I



Robert ANDERSON
COMMUNICATIONS PROFESSIONAL II

Hartwig ANZT
CONSULTANT

Alan AYALA
RESEARCH SCIENTIST I

Paul BAGWELL
SOFTWARE ENGINEER

Daniel BARRY
GRADUATE RESEARCH ASSISTANT

Natalie BEAMS
RESEARCH SCIENTIST I

Daniel BIELICH
POST DOCTORAL RESEARCH ASSOCIATE

George BOSILCA
RESEARCH ASSISTANT PROFESSOR

Aurelien BOUTEILLER
RESEARCH ASSISTANT PROFESSOR

Qinglei CAO
GRADUATE RESEARCH ASSISTANT


Earl CARR
PROGRAM ADMINISTRATOR


Tony CASTALDO
RESEARCH SCIENTIST II


Sebastien CAYROLS
RESEARCH SCIENTIST I


Giuseppe CONGIU
RESEARCH SCIENTIST I


Anthony DANALIS
RESEARCH ASSISTANT PROFESSOR


Jack DONGARRA
DIRECTOR & DISTINGUISHED PROFESSOR


Hadeer FARAHAT
GRADUATE RESEARCH ASSISTANT


Teresa FINCHUM
ADMINISTRATIVE SPECIALIST II


Mark GATES
RESEARCH ASSISTANT PROFESSOR


Thomas HERAULT
RESEARCH ASSISTANT PROFESSOR


Heike JAGODE
RESEARCH ASSISTANT PROFESSOR


Julie LANGOU
RESEARCH LEADER


Tracy LEE
FINANCIAL SPECIALIST I


Jiali LI
GRADUATE RESEARCH ASSISTANT


Yicheng LI
GRADUATE RESEARCH ASSISTANT


Neil LINDQUIST
GRADUATE RESEARCH ASSISTANT

STAFF AND STUDENTS


Piotr LUSZCZEK
RESEARCH ASSISTANT PROFESSOR


Maksim MELNICHENKO
GRADUATE RESEARCH ASSISTANT


Daniel MISHLER
GRADUATE RESEARCH ASSISTANT


Terry MOORE
CONSULTANT


Anustuv PAL
RESEARCH SCIENTIST I

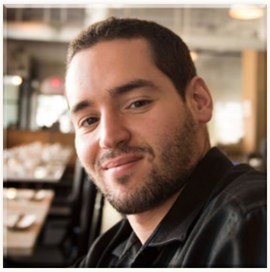

Yu PEI
GRADUATE RESEARCH ASSISTANT


Deborah PENCHOFF
ASSOCIATE DIRECTOR


Gerald RAGGHIANTI
RESEARCH LEADER


David ROGERS
COMMUNICATIONS SERVICES LEADER


Joseph SCHUCHART
POST DOCTORAL RESEARCH ASSOCIATE


Wissam SID LAKHDAR
RESEARCH SCIENTIST I


Leighanne SISK
ADMINISTRATIVE SPECIALIST I


Joan SNODERLY
ASSISTANT DIRECTOR


Dalal SUKKARI
POST DOCTORAL RESEARCH ASSOCIATE


Stanimire TOMOV
RESEARCH ASSISTANT PROFESSOR


Asim YARKHAN
RESEARCH SCIENTIST II

2021 VISITORS

ICL has a long-standing tradition of hosting visitors from all over the world. Some stay only briefly to give insightful seminars or presentations, while others remain with us for as long as a year to collaborate. Our connection to these researchers enables us to leverage an immense array of intellectual resources and work with the best and brightest people in the HPC community.

Anouar BENALI
Argonne National Laboratory

Michele BENZI
Scuola Normale Superiore in Pisa

Katie CAHILL
Howard H. Baker Jr. Center for Public Policy

Jamie COBLE
UTK Department of Nuclear Engineering

Sajal DASH
Oak Ridge National Laboratory

Tu MAI ANH DO
USC Information Sciences Institute

Azzam HAIDAR
NVIDIA

Stephen HERBEIN
Lawrence Livermore National Laboratory

Kate KEAHEY
Argonne National Laboratory

Scott KLASKY
Oak Ridge National Laboratory

Grzegorz KWASNIEWSKI
ETH Zurich

Pedro VALERO LARA
Oak Ridge National Laboratory

Mohsen MAHMOUDI-AZNAVEH
Texas A&M

Seetharami SEELAM
IBM

Ed VALEEV
Virginia Tech

Fan ZHANG
Georgia Tech

ALUMNI

ICL has attracted many research scientists and students from a variety of backgrounds and academic disciplines. Many of these experts came to UTK specifically to work with Prof. Dongarra—beginning a long list of research talent to pass through ICL and move on to make exciting contributions at other institutions and organizations.

Maksims ABALENKOVS
Carolyn AEBISCHER
Sudesh AGRAWAL
Bivek AGRAWAL
Emmanuel AGULLO
Mohammed AL FARHAN
Rabab AL-OMAIRY
Jennifer ALLGEYER
Wes ALVARO
Ed ANDERSON
Daniel ANDRZEJEWSKI
Thara ANGSKUN
Papa ARKHURST
Dorian ARNOLD
Rizwan ASHRAF
Cedric AUGONNET
Marc BABOULIN
Matthew BACHSTEIN
Zhaojun BAI
Ashwin BALAKRISHNAN
Richard BARRETT
Alex BASSI
John BATSON
David BATTLE
Micah BECK
Daniel BECKER
Dulceneia BECKER
Adam BEGUELIN
Annamaria BENZONI
Tom BERRY
Vincent BERTHOUX
Scott BETTS
Nikhil BHATIA
Laura BLACK
Noel BLACK
Susan BLACKFORD
Wesley BLAND
Kartheek BODANKI
David BOLT
Fernando BOND
Carolyn BOWERS
Vincent BRIDONNEAU
Barry BRITT
Randy BROWN
Cade BROWN
Bonnie BROWNE
Murray BROWNE
Cynthia BROWNE
Giuseppe BRUNO
Antonin BUKOVSKY
Greg BUNCH
Alfredo BUTTARI
Anthony CANINO
Domingo GIMENEZ CANOVAS
Chongxiao CAO
Henri CASANOVA
Cedric CASTAGNEDE
Ramkrishna CHAKRABARTY
Sharon CHAMBERS
Ali CHARARA
Zizhong CHEN
Jaeyoung CHOI
Wahid CHRABAKH
Eric CLARKSON
Andy CLEARY
Michelle CLINARD
Vincent COHEN-ADDAD
Matthias COLIN
Charles COLLINS
Rephael CONGMON
Stéphanie COOPER
Tom CORTESE
Camille COTI
Jason COX
Sam CRAWFORD
David CRONK
Javier CUENCA
Manoel CUNHA
Yuanshun (SHAUN) DAI
Cricket DEANE
Remi DELMAS
Frederic DESPREZ
Ying DING
Jin DING
Jun DING
Martin DO
Simplice DONFACK
Tingxing DONG
Leon DONG
Nick DONGARRA
David DOOLIN
Joe DORRIS
Andrew DOWNEY
Mary DRAKE
Julio DRIGGS
Brian DRUM
Peng DU
David EBERIUS
Eduardo ECHAVARRIA
Victor EIJKHOUT
Brett ELLIS
Shawn ERICSON
Zachary EYLER-WALKER
Lisa EZZELL
Christoph FABIANEK
Graham FAGG
Mathieu FAVERGE
Diana FAYAD
Shengzhog FENG
Don FIKE
Salvatore FILIPPONE
Anna FINCHUM
Mike FINGER
Jamie FINNEY
Markus FISCHER
Len FREEMAN
Xiaoquan FU
Erika FUENTES
Karl FUERLINGER
Megan FULLER
Edgar GABRIEL
Lynn GANGWER
Tracy GANGWER
Kelley GARNER
Nathan GARNER
Tina GARRISON
Adriana GARTIES
Peter GAULTNEY
Christoph GEILE
Jean PATRICK GELAS
Boris GELFEND
Damien GENET
Jonathan GETTLER
Scott GIBSON
Eric GREASER
Stan GREEN
Alice GREGORY
Amina GUERMOUCHE
Jason GURLEY
Bilel HADRI
Hunter HAGEWOOD
Azzam HAIDAR
Christian HALLOY
Sven HAMMARLING
J. Mike HAMMOND
Hanumantharayappa
Satomi HASEGAWA
Hidehiko HASEGAWA
Chris HASTINGS
Blake HAUGEN
David HENDERSON
John HENRY
Greg HENRY
Julien HERRMANN
Holly HICKS
Alexandra HICKS-HARDIMAN
Sid HILL
Tomoyuki HIROYASU
George HO
Josh HOFFMAN
Reazul HOQUE
Jeff HORNER
Mitch HORTON
Yan HUANG
Harry HUGHES
Aurelie HURAULT
Chris HURT
Paul JACOBS

Emmanuel JEANNOT
Weizhong JI
Yulu JIA
Weicheng JIANG
Song JIN
Patrick JOHANSSON
Aral JOHNSON
Matt JOHNSON
Sean JOLLY
Jan JONES
Kim JONES
Vijay JOSHI
Khairul KABIR
Venkata KAKANI
Ajay KALHAN
Balajee KANNAN
Madhuri KASAM
Kiran Kumar KASICHAYANULA
Ajay KATTA
David KATZ
Joshua KELLY
Supriya KILAMBI
Youngbae KIM
Myung HO KIM
Jenya KIRSHTEIN
Cindy KNISLEY
Hannah KNOCH
Michael KOLATIS
Anara KOZHOKANOVA
Moritz KREUTZER
Chandra KRINTZ
Tilman KUESTNER
Jakub KURZAK
Krerkchai KUSOLCHU
Coire KYLE
Amanda LAAKE
Xavier LACOSTE
Julien LANGOU
Jeff LARKIN
Brian LAROSE
Frank LAUER
Valentin LE FEVRE
DongWoo LEE
Pierre LEMARINIER
Todd LETSCHE
Sharon LEWIS
Yinan LI
Weiran LI
Xiang LI
Chaoyang LIU
Kevin LONDON
Matt LONGLEY
Florent LOPEZ
Nuria LOSADA
Hatem LTAIEF
Yuechao LU
Daniel LUCIO
Richard LUCZAK
Xi LUO
Teng MA
Sticks MABAKANE
Robert MANCHEK
Gabriel MARIN
Tushti MARWAH
Theo MARY
Ian MASLIAH
Donald MCCASLAND
Tyler MCDANIEL
Paul MCMAHAN
Eric MEEK
James MEYERING
Jeremy MILLAR
Michelle MILLER
Cindy MITCHELL
Stuart MONTY
Shirley MOORE
Erik MOORE
Keith MOORE
Robert MORGAN
Kishan MOTHERAMGARI
Steven MOULTON
Phil MUCCI
Daichi MUKUNOKI
Matthew NABITY
Shankar NARASIMHASWAMI
Rajib NATH
Fernando NAVARRO
John NELSON
Donnie NEWELL
Peter NEWTON
Jonas NILSSON
Jakob OESTERGAARD
Caroline PAPADOPOULOS
Leelinda PARKER
Thibault PARPAITE
Thananon PATINYASAKDIKUL
Dilip PATLOLLA
Andy PEARSON
Paul PELTZ
Theresa PEPIN
Antoine PETITET
Peter PHAM
Gregoire PICHON
Vlado PJESIVAC
Jelena PJESIVAC-GRBOVIC
James S. PLANK
Tim POORE
Roldan POZO
Farzona PULATOVA
Martin QUINSON
Tammy RACE
Tracy RAFFERTY
Sangamesh RAGATE
James RALPH
Ganapathy RAMAN
Kamesh RAMANI
Mei RAN
Arun RATTAN
Sheri REAGAN
Mike REYNOLDS
George RHINEHART
Jon RICHARDSON
Stephen RICHMOND
Ken ROCHE
Andrew ROGERS
Tom ROTHROCK
Tom ROWAN
Narapat (Ohm) SAENGPATSA
Kiran SAGI
Evelyn SAMS
Daniel SCHULTZ
Ken SCHWARTZ
Keith SEYMOUR
Farial SHAHNAZ
Hejer SHAIEK
Daniel SHARP
Brian SHEELY
Zhiao SHI
Sergei SHINKAREV
Majed SIDANI
Shilpa SINGHAL
Matt SKINNER
Peter SOENDERGAARD
Raffaele SOLCA
Gwang SON
Fengguang SONG
Thomas SPENCER
Jeffrey STEILL
Erich STROHMAIER
Xiaobai SUN
Martin SWANY
Daisuke TAKAHASHI
Judi TALLEY
Ronald TAM
Chunyan TANG
Yuan TANG
Yusuke TANIMURA
Keita TERANISHI
Dan TERPSTRA
Joe THOMAS
John THURMAN
Francoise TISSEUR
Jude TOTH
Bernard TOURANCHEAU
Yaohung TSAI
Volodymyr TURCHENKO
Lauren VACA
Phil VACCARO
Sathish VADHIYAR
Robert VAN DE GEIJN
Chad VAWTER
Eugene VECHARYNSKI
Scott VENCKUS
Antoine VERNOIS
Reed WADE
Michael WALTERS
Robert WALTZ
Mike WALTZ
Xiaoyang WANG
Bob WARD
Jerzy WASNIEWSKI
Vince WEAVER
Aaron WELCH
Scott WELLS
David WEST
R. Clint WHALEY
Jody WHISNANT
James WHITE
Scotti WHITMIRE
Frank WINKLER
Susan WO
Felix WOLF
Stephen WOOD
Jiayi WU
Panruo WU
Wei WU
Qiu XIA
Tinghua XU
Ichitaro YAMAZAKI
Tao YANG
Erlin YAO
Kevin YE
Jin YI
Haihang YOU
Lamia YOUSEFF
Brian ZACHARY
Dmitry ZAITSEV
Omar ZENATI
Yuanlei ZHANG
Junlong ZHAO
Yong ZHENG
Dong ZHONG
Luke ZHOU
Min ZHOU

ACKNOWLEDGMENTS

For over 30 years, our knowledge and hard work have earned the trust and support of many agencies and organizations that have funded and continue to fund our efforts. Without them, we simply would not be able to conduct leading-edge research.

The main source of support has been federal agencies that are charged with allocating public research funding, and we acknowledge the following agencies for supporting our efforts:

AFOSR — Air Force Office of Scientific Research

U.S. Department of Energy — Office of Science

U.S. Department of Energy — NNSA National Nuclear Security Administration

National Science Foundation

Oak Ridge National Laboratory

In addition to the support of the federal government, we have solicited strong support from private industry, which has also played a significant role in our success and growth. We gratefully acknowledge the following vendors for their generosity and support:

AMD

Atos

ENGILITY

intel

MathWorks

NVIDIA

UT-BATTELLE

Administrative and technical support for ICL is provided by the Center for Information Technology Research (CITR).

The University of Tennessee Knoxville

Center for Information Technology Research

ICL INNOVATIVE COMPUTING LABORATORY

FIND OUT MORE AT **https://www.icl.utk.edu/**