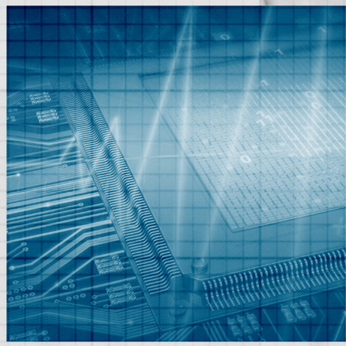
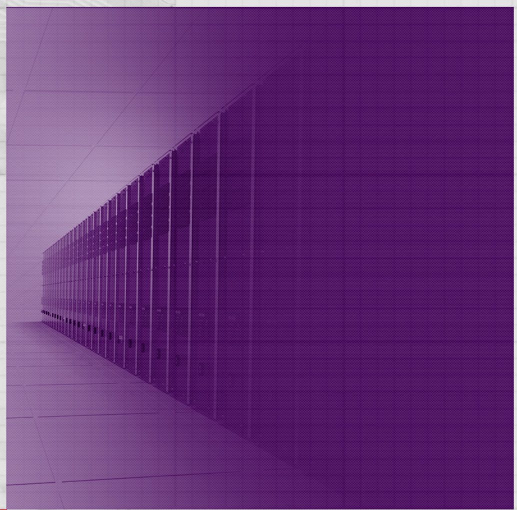
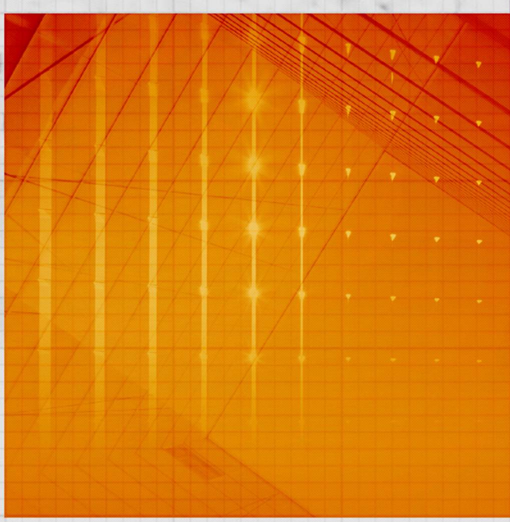






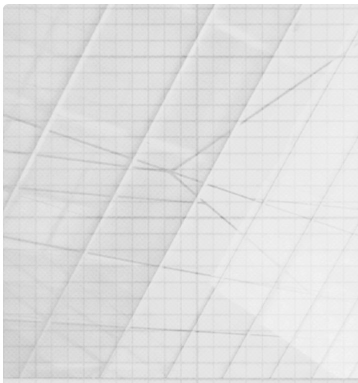
INNOVATIVE COMPUTING LABORATORY

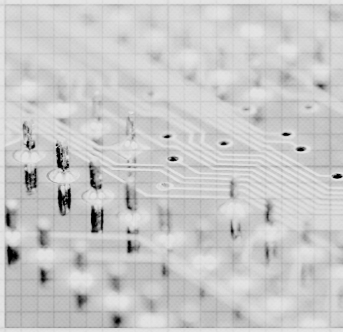
2017/2018 REPORT

THE UNIVERSITY OF TENNESSEE
KNOXVILLE

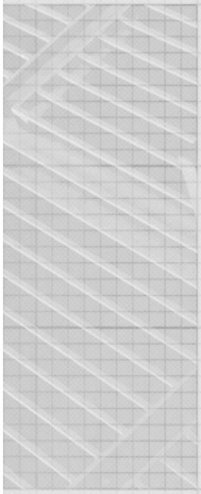
ICL 2017/2018 REPORT

DESIGNED BY **David Rogers** EDITED BY **Sam Crawford**





INNOVATIVE COMPUTING LABORATORY 2017/2018 REPORT



ICL/FROM THE DIRECTOR

I began my director's letter last year by highlighting the striking success that the Innovative Computing Laboratory (ICL) enjoyed with our funded proposals to the Department of Energy's (DOE's) Exascale Computing Project (ECP). Success, of course, has its consequences. This year, many of the highlights of 2017 revolve around our efforts to engage productively in ECP: staffing up to do the work of our seven ECP projects, engaging with ECP's complex management system, setting out our milestones, working to establish the necessary collaborations with other groups, and starting to design and actually write the software we proposed. As the review process that we recently underwent has shown, we are off to a very good start, and so I want to express my thanks to all of you who have contributed to that effort.

Last year I attributed our achievement with ECP funding to two factors: ICL's strong reputation for turning innovative software research into production-quality software, and the sustained improvement over time in our capacity to develop clear and compelling proposals. This year I think that actually working with ECP has brought out another impressive strength of our group—namely, the remarkable level of mature professionalism that we now bring to our collaborations. In the context of ECP, I have seen that professionalism in the way we have been able to scout, evaluate, and recruit the new people we need; in the thoroughgoing preparation for and execution of our management responsibilities; in the mentoring that our senior leadership has provided to our students and new research staff; and—simply—in the general way that ICL people comport themselves in all the different venues and activities in which we are involved. Naturally, I find the many positive comments I get from my colleagues and friends about the quality of our group's professionalism very gratifying.

But even though ECP demanded the lion's share of our attention in 2017, we still managed to have an excellent year in funding for creative new projects with the National Science Foundation (NSF). Our distributed computing group had an especially successful year, picking up two significant additions in the areas of runtime systems and software resilience: the Enhancing the Open MPI Software for Next Generation Architectures and Applications (Evolve) project, a collaboration with ICL alumnus Edgar Gabriel at the University of Houston; and the Cross-layer Application-Aware Resilience at Extreme Scale (CAARES) project, a collaboration with Barbara Chapman from Stony Brook University and Manish Parashar from Rutgers University. Our linear algebra team also made important progress in the critical new area of "batched" algorithms and routines by winning software infrastructure funding for the MAtrix, TEnsor, and Deep-learning Optimized Routines (MATEDOR) project. Finally, with the Highly Parallel Algorithms and Architectures for Convex Optimization for Real-time Embedded Systems (CORES) project, we established a valuable new collaboration with Saeid Nooshabadi of Michigan Tech in the area of performance optimization for embedded systems.

In 2017, we were also able to strengthen our foundation on the University of Tennessee, Knoxville (UTK) campus in two significant ways. First, five of ICL's Research Directors—Jakub Kurzak, George Bosilca, Heike Jagode, Piotr Luszczek, and Stan Tomov—were each raised to the level of Research Assistant Professor. This means, among other things, that they can be authorized to direct theses and dissertations, and we are currently working to establish that authorization for them in 2018. This new mode of productive engagement with UTK students is another sign of ICL's maturity and will clearly be a very positive development for us. Second, with the establishment of the "Jack Dongarra Endowed Professorship in High-Performance Computing" in UTK's Department of Electrical Engineering and Computer Science this year, we also helped the department recruit Professor Michela Taufer, an outstanding high-performance computing (HPC) researcher from the University of Delaware, to fill it. She will join the Department of Electrical Engineering and Computer Science faculty this summer, and I expect that we will be able to start collaborating with her almost immediately. Her presence at UTK is precisely the kind of development we need to move our work to the next level.

So, given our progress in 2017, we are very much on track to enhance that record and to accomplish great things in 2018 and beyond. Of course, as always, to achieve such goals we will need to work hard—especially by finding productive ways to engage with important ECP application groups. But my sense of our creativity, commitment, and solid professionalism gives me every confidence that we are well positioned to succeed—not only with ECP, but in all those projects that our many sponsors have chosen to support. And as ever, I continue to be amazed at and deeply appreciative of the skill, dedication, and tireless efforts that the researchers, staff, and students of ICL put into their work. They have helped to make ICL and UTK one of the premier centers in the world today for high-performance and scientific computing.

Jack Dongarra
DIRECTOR, ICL

ICL/INTRODUCTION

Situated in the heart of the University of Tennessee campus and at the nexus of academia, government, and industry, ICL impacts the world as a leader in advanced scientific computing and HPC through research, education, and collaboration.

The unique challenges of today's computational research are characterized by large datasets and the need for greater performance, energy conservation, and resilience. ICL's cutting-edge efforts, which now span more than 25 years, have evolved and expanded with the agility and focus required to address those challenges. ICL's work 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.

NUMERICAL LINEAR ALGEBRA

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 ANALYSIS AND BENCHMARKING

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

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 need runtime systems that can exploit all 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 two decades, and the lab has numerous projects in this arena under active development.

ICL / HISTORY

ICL's founder, Dr. Jack Dongarra, established the lab 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). Since then, 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 software tool for parallel networking of computers designed to allow a network of heterogeneous Unix and Windows machines to be used as a single distributed parallel processor.

1992

Basic Linear Algebra Communication Subprograms (BLACS) 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.

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.

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 remotely.

1999

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

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.

2008

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

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 Multicore Architectures (DPLASMA) is a linear algebra package that enables sustained performance for distributed systems, where each node features multiple sockets of multi-core processors 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 multistream multi-GPU programming.

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 **Bench-testing Environment for Automated Software Tuning (BEAST)** project enables writing of tunable high-performance kernels by unleashing the power of heuristic autotuning.

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.

2014

Argo is an initiative to develop a new exascale operating system and runtime (OS/R) designed to support extreme-scale scientific computation.

2015

The **SparseKaffe** project establishes fast and efficient sparse direct methods for platforms with multi-core processors with one or more accelerators.

The **Task-based Environment for Scientific Simulation at Extreme Scale (TESSE)** used an application-driven design to create a general-purpose software framework focused on programmer productivity and portable performance for scientific applications on massively parallel hybrid systems.

2016

ICL won seven awards through DOE's Exascale Computing Project (ECP) during the fall of 2016 and is the lead institution on four of these projects:

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.

1993

Now on its 50th list, **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.

1999

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) Benchmark is a benchmark for distributed-memory computers that solves a (random) dense linear system in double-precision (64-bit) arithmetic.

2006

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

2008

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

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.

2014

The **Rapid Python Deep Learning Infrastructure (RaPyDLI)** delivered productivity and performance to the deep learning community by combining high-level Python, C/C++, and Java environments with carefully designed libraries supporting GPU accelerators and Intel Xeon Phi coprocessors.

2015

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

Data-driven Autotuning for Runtime Execution (DARE) provides application-level performance tuning capabilities to the end user.

2016

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.

SELECTED ICL ALUMNI

Throughout its history, ICL has attracted numerous postdoctoral researchers and professors from multiple disciplines. Many of these experts came to UTK specifically to work with Dr. Dongarra. This list features ICL alumni who played key roles in the establishment and growth of ICL and went on to build distinguished careers at other organizations and institutions.

Emmanuel Agullo
INRIA

Ed Anderson
Environmental Protection Agency

Thara Angskun
Suranaree University of Technology

Dorian Arnold
Emory University

Zhaojun Bai
University of California, Davis

Richard Barrett
Sandia National Laboratories

Adam Beguelin
Sensr.net

Susan Blackford
CSP, Inc.

Alfredo Buttari
Institut de Recherche en Informatique de Toulouse

Henri Casanova
University of Hawaii, Manoa

Zizhong Chen
University of California, Riverside

Jaeyoung Choi
Soongsil University

Andy Cleary
Amazon

David Cronk
Leidos

Frederic Desprez
INRIA

Victor Eijkhout
University of Texas, Austin

Graham Fagg
Microsoft

Mathieu Faverge
INRIA

Karl Fuerlinger
Ludwig Maximilian University of Munich

Edgar Gabriel
University of Houston

Sven Hammarling
University of Manchester

Greg Henry
Intel

Julien Langou
University of Colorado, Denver

Kevin London
Microsoft

Hatem Ltaief
King Abdullah University of Science and Technology

Bob Manchek
Akamai Technologies

Keith Moore
Independent Consultant

Shirley Moore
Oak Ridge National Laboratory

Antoine Petitet
Huawei Technologies

Roldan Pozo
National Institute of Standards and Technology

Yves Robert
École normale supérieure de Lyon

Kenneth Roche
University of Washington

Erich Strohmaier
Lawrence Berkeley National Laboratory

Martin Swany
Indiana University

Dan Terpstra
Living Waters for the World

Françoise Tisseur
University of Manchester

Bernard Tourancheau
Université Joseph Fourier de Grenoble

Sathish Vadhiyar
Indian Institute of Science

Robert van de Geijn
University of Texas, Austin

Reed Wade
Artella

Vince Weaver
University of Maine

Clint Whaley
Indiana University

Felix Wolf
Technische Universität Darmstadt

Haihang You
Chinese Academy of Sciences

ICL / 2017 HIGHLIGHTS

ACHIEVEMENTS AND ACCOLADES

JACK DONGARRA PROFESSORSHIP IN HIGH-PERFORMANCE COMPUTING

In 2017, Jack Dongarra created an endowment for the Jack Dongarra Professorship in High-Performance Computing within the Department of Electrical Engineering and Computer Science in UTK's Tickle College of Engineering. The professorship targets individuals with expertise and experience in high-performance and scientific computing.

Professor Michela Taufer, an Association for Computing Machinery Distinguished Scientist and J.P. Morgan Chase Scholar currently at the University of Delaware, will be the first Jack Dongarra Professor in High-Performance Computing. Dr. Taufer, who is an expert in HPC and director of the University of Delaware's Global Computing Lab, will join UTK's Department of Electrical Engineering and Computer Science in June 2018.

BEST PAPER

On June 2, 2017, ICL Research Scientist Ichitaro Yamazaki traveled to Orlando, Florida to present his paper, "Improving Performance of GMRES by Reducing Communication and Pipelining Global Collectives," at the 18th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC 2017), where it received the workshop's Best Paper Award.

The work outlined in the paper compared and combined two techniques to improve the parallel scalability of the Krylov subspace projection methods: a communication-avoiding technique and a communication-hiding technique. As the cost of communication continues to increase with scale, such techniques are expected to be critical on emerging extreme-scale supercomputers. These algorithms are being implemented in ICL's Production-ready, Exascale-Enabled Krylov Solvers for Exascale Computing (PEEKS) effort—part of DOE's ECP—to enable application scientists to take full advantage of exascale supercomputers.

Yamazaki, I., M. Hoemmen, P. Luszczek, and J. Dongarra, "Improving Performance of GMRES by Reducing Communication and Pipelining Global Collectives," The 18th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC 2017), Best Paper Award, Orlando, FL, June 2017.

INTEL PARALLEL COMPUTING CENTER

The Intel Parallel Computing Center (IPCC) program—composed of universities, institutions, and labs that are leaders in their field—focuses on modernizing applications to increase parallelism and scalability through optimizations that leverage the cores, caches, threads, and vector capabilities of microprocessors and coprocessors.

As part of this effort, ICL will develop and optimize numerical linear algebra libraries and technologies for emerging HPC applications while also tackling current challenges in Intel® Xeon Phi™ coprocessor-based HPC applications. In collaboration with Intel's Math Kernel Library (MKL) team, ICL will modernize the LAPACK and ScaLAPACK libraries to run efficiently on current and future many-core architectures and will disseminate the developments through the open-source MAGMA library.

NVIDIA GPU CENTER OF EXCELLENCE

ICL joins a very small and select group of labs given a GPU Center of Excellence (GPU COE) designation. ICL/UTK's GPU COE focuses on the development of numerical linear algebra libraries for CUDA-based hybrid architectures. ICL's work on the MAGMA project further enables and expands our CUDA-based software library efforts, especially in the area of high-performance scientific computing.

The GPU COE designation also led to the establishment of a productive long-term collaboration between ICL and NVIDIA. As part of the collaboration and GPU COE designation, ICL has continuously received hardware, financial support, and other resources from NVIDIA.

ICL/RESEARCH

What originally began nearly 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.

ICL has evolved and expanded our research agenda to accommodate the aforementioned evolution in HPC, which includes a focus on algorithms and libraries for multicore 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.

In the fall of 2016, ICL won an array of seven awards from the DOE's Exascale Computing Project (ECP). In doing so, ICL earned a place among an elite set of researchers from DOE laboratories who will create the software infrastructure for the nation's first exascale machines. On the following pages, we provide brief summaries of some of our efforts in these areas.

EXASCALE COMPUTING PROJECT

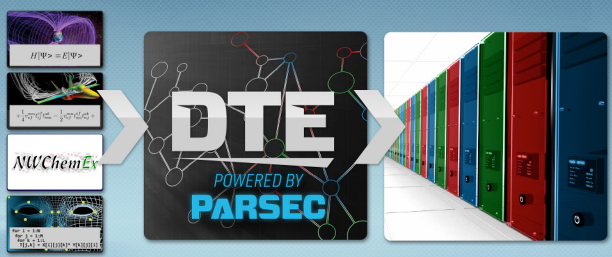
NUMERICAL LINEAR ALGEBRA

DISTRIBUTED COMPUTING

PERFORMANCE ANALYSIS AND BENCHMARKING

ECP

DTE



FIND OUT MORE AT http://icl.utk.edu/dte/

Distributed Tasking for Exascale

The Distributed Tasking for Exascale (DTE) project will extend the capabilities of ICL's Parallel Runtime and Execution Controller (PaRSEC) project—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 and extensions and tools for debugging, trace collection, and analysis.

With PaRSEC, applications are expressed as a direct acyclic graph (DAG) of tasks with edges designating data dependencies. This DAG 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.

ECP

EXA-PAPI

FIND OUT MORE AT http://icl.utk.edu/exa-papi/

Exascale Performance Application Programming Interface

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

Exa-PAPI extends this effort with performance counter monitoring capabilities for new and advanced ECP hardware and software technologies, fine-grained power management support, and integration capabilities for exascale paradigms like task-based runtime systems. Exa-PAPI also adds events that originate from the ECP software stack, extending the notion of performance events to include not only hardware but also software-based information—all through one consistent interface.

FIND OUT MORE AT http://icl.utk.edu/peeks/

Production-ready, Exascale-Enabled Krylov Solvers for Exascale Computing

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. Many large-scale scientific applications rely heavily on preconditioned iterative solvers for large linear systems. For these solvers to efficiently exploit extreme-scale hardware, both the solver algorithms and the implementations must be redesigned to address challenges like extreme concurrency, complex memory hierarchies, costly data movement, and heterogeneous node architectures.

The PEEKS effort aims to tackle these challenges and advance the capabilities of the ECP software stack by making the new scalable algorithms accessible within the Trilinos software ecosystem. Targeting exascale-enabled Krylov solvers, incomplete factorization routines, and parallel preconditioning techniques will ensure successful delivery of scalable Krylov solvers in robust, production-quality software that can be relied on by ECP applications.

FIND OUT MORE AT http://icl.utk.edu/slate/

Software for Linear Algebra Targeting Exascale

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 23 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 innovation 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.

ECP

CEED

FIND OUT MORE AT
http://ceed.exascaleproject.org/

CEED Co-Design Center

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 will be 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.

ECP

OMPI-X

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

Open MPI for Exascale

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.

ECP

xSDK4ECP

FIND OUT MORE AT
https://xsdk.info/ecp/

Extreme-scale Scientific Software Development Kit

The Extreme-Scale Scientific Software Development Kit for the Exascale Computing Project (xSDK4ECP), a collaboration between Argonne National Laboratory (ANL), ICL, Lawrence Berkeley National Laboratory (LBNL), LLNL, Sandia National Laboratories (SNL), and the University of California at Berkeley (UCB), aims to enable seamless integration and combined use of diverse, independently developed software packages for ECP applications. Currently, this includes a wide range of high-quality software libraries and solver packages that address the strategic needs to fulfill the mission of DOE's Office of Science.

In the first year, ICL's MAGMA and Parallel Linear Algebra Software for Multicore Architectures (PLASMA) projects were integrated into the xSDK 0.3.0 release. To ensure consistency of naming conventions, runtime behavior, and installation procedure, xSDK informs the development process of large projects by providing sets of requirements and guidelines that are influential throughout the software development process. Additionally, as part of the inclusion process—and to lighten the burden on the system administrators and application developers—each package gets a Spack installation script that can be invoked independently or through the installation of the xSDK package itself.

AsyncIS

FIND OUT MORE AT
http://www.icl.utk.edu/research/asyncis/

Asynchronous Iterative Solvers for Extreme-scale Computing

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 many-core accelerators like GPUs.

BATCHED BLAS

FIND OUT MORE AT
http://icl.utk.edu/bblas/

Batched Basic Linear Algebra Subprograms

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 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 gathering ideas and working out a plan for establishing the consensus for the next official standard that will serve the scientific community and be supported by hardware vendors.

BONSAI

FIND OUT MORE AT
https://bitbucket.org/icl/bonsai/

BEAST OpeN Software Autotuning Infrastructure

The goal of the BEAST OpeN Software Autotuning Infrastructure (BONSAI) project is to develop a software infrastructure for using scalable, parallel hybrid systems to carry out large, concurrent autotuning sweeps in order to dramatically accelerate the optimization process of computational kernels for GPU accelerators and many-core coprocessors. BONSAI is based on Python 2.7 and the Python-based LANguage for Autotuning Infrastructure (LANAI) syntax.

As the name suggests, BONSAI builds on the experiences of the Bench-testing Environment for Automated Software Tuning (BEAST) project, which prototyped and validated an autotuning workflow consisting of generation and pruning of the parameter search space; compilation, benchmarking, and profiling of the kernels that pass the pruning; and collection, analysis, and visualization of the performance data.

CAARES

FIND OUT MORE AT
http://www.icl.utk.edu/research/caares/

Cross-layer Application-Aware Resilience at Extreme Scale

The Cross-layer Application-Aware Resilience at Extreme Scale (CAARES) project, a collaborative effort between ICL, Rutgers University, and Stony Brook, aims to provide a theoretical foundation for multi-level fault management and a clear understanding of existing obstacles that could obstruct generic and efficient approaches for fault management at scale. This effort is vital for large-scale science, because, as extreme-scale computational power enables new and important discoveries across all science domains, the current understanding of fault rates is casting a grim shadow, revealing a future where failures are not exceptions but are the norm.

By studying a combination of fault tolerance techniques not in isolation from each other, CAARES seizes the opportunity to identify moldable techniques at the frontier of known approaches, a composition of methodologies that will inherit their individual benefits but not exhibit their drawbacks, and techniques able to bridge the gap between fault tolerance ergonomics and efficiency.

CoDAASH

FIND OUT MORE AT
http://icl.utk.edu/codaash/

Cross-layer Application-Aware Resilience at Extreme Scale

The Co-design Approach for Advances in Software and Hardware (CoDAASH) 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. CoDAASH is a joint effort between UTK, Iowa State University, the University of Texas–El Paso, and the University of California–San Diego, and is funded by the United States Air Force Office of Scientific Research (AFOSR).

ICL's contribution focuses on expressing certain computational chemistry algorithms in the form of a data flow 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—in the era of many-core and highly heterogeneous platforms—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.

CORES

FIND OUT MORE AT
http://www.icl.utk.edu/research/cores/

Convex Optimization for Realtime Embedded Systems

The Convex Optimization for Realtime 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.

DARE

FIND OUT MORE AT
http://www.icl.utk.edu/research/dare/

Data-driven Autotuning for Runtime Execution

The objective of the Data-driven Autotuning for Runtime Execution (DARE) project is to provide application-level performance tuning capabilities to the end user. DARE's development motivation stems from the never-ending hurdles of performance tuning for the PLASMA and MAGMA linear algebra libraries. These hurdles motivated the development of a software architecture that combines three components: hardware analysis, kernel modeling, and workload simulation.

With DARE, the hardware analysis block builds a detailed model of the hardware, its computational resources, and its memory system. The kernel modeling block builds accurate performance models for the computational kernels involved in the workload, and the workload simulation block rapidly simulates a large number of runs to find the best execution conditions while relying on the information provided by the other two blocks. The ultimate objective of DARE is to arrange the blocks in a continuous refinement loop that can serve as a framework for optimizing applications beyond the field of dense linear algebra.

DPLASMA

FIND OUT MORE AT
http://icl.utk.edu/dplasma/

Distributed Parallel Linear Algebra Software for Multicore Architectures

The Distributed Parallel Linear Algebra Software for Multicore 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.

Evolve

FIND OUT MORE AT
http://www.icl.utk.edu/research/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.

For 2017, Evolve efforts revolved around exploratory research in improving the performance of multithreaded programs using MPI. Collective operations based on events have been investigated and have demonstrated a clear advantage in terms of aggregate bandwidth in heterogeneous (shared-memory + network) systems. User-Level Failure Mitigation (ULFM) fault-tolerance was released based on the latest Open MPI. Counters and performance profiling of internal Open MPI events are now exposed, which has enabled the team to discover and eliminate important performance limitations in the MPI implementation.

HPCG

FIND OUT MORE AT
http://www.hpcg-benchmark.org/

High Performance Conjugate Gradients

The High Performance Conjugate Gradients (HPCG) benchmark is designed to measure performance that is representative of modern scientific applications. 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. HPCG exhibits the same irregular accesses to memory and fine-grain recursive computations that dominate large-scale scientific workloads used to simulate complex physical phenomena.

The HPCG 3.0 reference code was released on November 11, 2015 for the SC15 conference in Austin, TX. In addition to bug fixes, this release positions HPCG to even better represent modern partial differential equation (PDE) solvers and aids in running HPCG on production supercomputing installations. The reference version is accompanied by binary releases from Intel and NVIDIA that are carefully optimized for the vendors' respective hardware platforms. The current HPCG performance list was released at SC17 and now features over 100 supercomputing sites. The HPCG score has also been tracked by the TOP500 list since June 2017.

HPL

FIND OUT MORE AT
http://icl.utk.edu/hpl/

High Performance LINPACK

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 installations, producing a result that can be submitted to the TOP500 list of the world's fastest supercomputers.

The major focus of HPL 2.2, released in 2016, was to improve the accuracy of reported benchmark results and ensure scalability of the code on large supercomputer installations with 100,000+ cores. HPL now features detailed time-of-run accounting to better assess power requirements at run time, a metric which is reported on the TOP500 and also highlighted on the Green500. The LINPACK app for iOS achieves over 4 gigaFLOP/s on the iPad Air and iPad Pro. For the November 2017 TOP500 list, an optimized version of the HPL code ran on a machine with over 10,000,000 cores.

LAPACK
ScaLAPACK

FIND OUT MORE AT
http://www.netlib.org/lapack/

The Linear Algebra PACKage

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 3.8.0, released in November 2017, includes level-3 BLAS communication-avoiding, symmetric-indefinite factorizations with Aasen's triangular tridiagonalization using the two-stage algorithm. 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 2.0.0, which includes the multiple relatively robust representations (MRRR) algorithm and new nonsymmetric eigenvalue problem routines, was released in November 2011. Two additional ScaLAPACK versions (2.0.1 and 2.0.2) were released in 2012 for minor bug fixes.

MAGMA

FIND OUT MORE AT
http://icl.utk.edu/magma/

Matrix Algebra on GPU and Multicore Architectures

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

MAGMA 2.3 features LAPACK-compliant routines for multi-core CPUs enhanced with NVIDIA GPUs (including the Volta V100). MAGMA now includes more than 400 routines, covering one-sided dense matrix factorizations and solvers, and two-sided factorizations and eigen/singular-value problem solvers, as well as a subset of highly optimized BLAS for GPUs. A MagmaDNN package was launched to provide high-performance data analytics, including functionalities for machine learning applications that use MAGMA as their computational backend. The MAGMA Sparse and MAGMA Batched packages were added with the MAGMA 1.6 release and continuously extended and improved with each release.

MATEDOR

FIND OUT MORE AT
http://www.icl.utk.edu/research/matedor/

MAtrix, TEnsor, and Deep-learning Optimized Routines

The MAtrix, TEnsor, and Deep-learning Optimized Routines (MATEDOR) project will perform the research required to define a standard interface for batched operations and to 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 10× 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.

OPEN MPI

FIND OUT MORE AT
https://www.open-mpi.org/

Open-Source Message Passing Interface

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 (HARNESS/FT-MPI, LA-MPI, LAM/MPI, and PACX-MPI) in order to build the best MPI library available.

A completely new MPI 3.1–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 the next generation exascale challenges.

PAPI

FIND OUT MORE AT
http://icl.utk.edu/papi/

Performance Application Programming Interface

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 Bull, 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 2015, ICL, together with the University of Maine, began work on PAPI-EX to build support for performance counters available in recent generations of CPUs and GPUs, develop support for system-wide hardware performance counter monitoring, create a sampling interface in PAPI, and incorporate a user-facing command line tool for PAPI. PAPI-EX will also incorporate a counter inspection toolkit designed to improve understanding of low-level hardware events. At the end of 2016, ICL was awarded funding for the ECP Exa-PAPI project, which will augment PAPI-EX.

PaRSEC

FIND OUT MORE AT
http://icl.utk.edu/parsec/

Parallel Runtime Scheduling and Execution Controller

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

FIND OUT MORE AT
http://icl.utk.edu/plasma/

Parallel Linear Algebra Software for Multicore Architectures

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

Over the last decade, PLASMA—which has been deployed on a variety of systems using Intel processors (including Xeon Phi coprocessors), IBM POWER processors, and ARM processors—has served as a tremendous research vehicle for the design of new dense linear algebra algorithms and has paved the way for new developments, including the new ECP SLATE project, which will ultimately deliver these capabilities at exascale.

PULSE

PAPI Unifying Layer for Software-defined Events

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

FIND OUT MORE AT
http://www.icl.utk.edu/research/pulse/

SMURFS

Simulation and Modeling for Understanding Resilience and Faults at Scale

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.

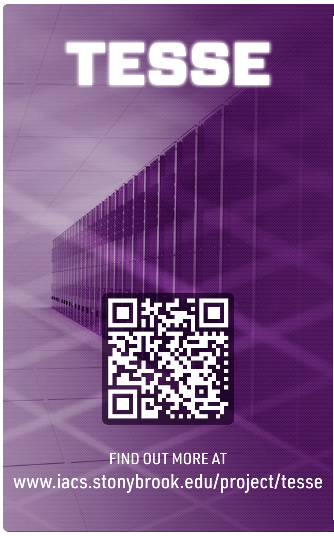
FIND OUT MORE AT
http://www.icl.utk.edu/research/smurfs/

SparseKaffe

The Sparse direct methods via Run-time Scheduling and Execution of Kernels with Auto-tunable and Frequency-scaling Features for Energy-aware computing on heterogeneous architectures (SparseKaffe) project creates fast and efficient sparse direct methods for platforms with multi-core processors with one or more accelerators (e.g., GPUs or Xeon Phi coprocessors). SparseKaffe spans the platform pyramid, from desktop machines to extreme-scale systems consisting of multiple heterogeneous nodes connected through a high-speed network, with the goal of achieving orders of magnitude gains in computational performance while also paying careful attention to energy requirements.

The SparseKaffe project is a collaboration between UTK, the University of Florida, and Texas A&M University. ICL's work on the project concentrates on kernel designs and performance tuning, as well as on dynamic runtime scheduling using a dataflow model. This work will leverage—and be a natural extension of—ICL's work on runtimes as part of the MAGMA, PLASMA, and PaRSEC projects. The autotuning of the algorithm-specific computational kernels will apply the principles behind ICL's DARE project.

FIND OUT MORE AT
www.icl.utk.edu/research/sparsekaffe



Task-based Environment for Scientific Simulation at Extreme Scale

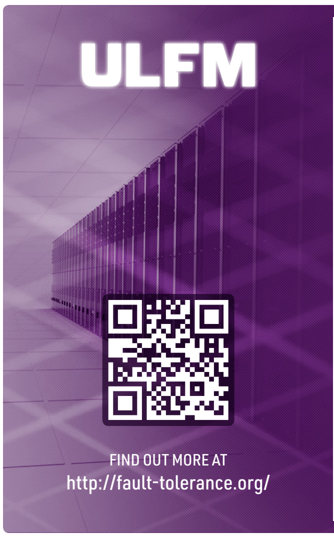
The goal of the Task-based Environment for Scientific Simulation at Extreme Scale (TESSE) is to use an application-driven design to create a general-purpose, production-quality software framework that attacks the twin challenges of programmer productivity and portable performance for advanced scientific applications on massively-parallel, hybrid, many-core systems of today and tomorrow.

The TESSE team is composed of researchers from Stony Brook, Virginia Tech, and UTK, who have designed a system that uses DAG-based data flow as the basis of the software. This capability, with the extensions being explored by the TESSE team, will provide significant potential advantages in ease of composition, performance, and ease of migration to future architectures for irregular parallel applications. The TESSE team's next major goal is the ubiquitous existence of a powerful DAG-based data flow tool that complements, and is completely interoperable with, mainstream standard parallel programming models such as OpenMP and MPI.

Ranking the 500 fastest computers in the world

With more than two decades of tracking supercomputing progress, 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 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 benchmarks a machine's performance using irregular accesses to memory and fine-grain recursive computations—factors which dominate real-world, large-scale scientific workloads.

In November 2017, the 50th TOP500 list showed China overtaking the United States in the number of ranked systems—with 202 Chinese machines to 143 US machines. With these numbers in play, China now has more machines in the TOP500 than they ever have previously, while the number of US machines drops to a 25-year low. As far as the rankings themselves, China's Sunway TaihuLight system, installed at the National Supercomputing Center in Wuxi, maintains its number one ranking for the fourth time, with an HPL run of 93.01 petaFLOP/s.



User Level Failure Mitigation

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 whose 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.

ICL/PUBLICATIONS

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

FIND OUT MORE AT
http://www.icl.utk.edu/publications/

Abalenkovs, M., N. Bagherpour, J. Dongarra, M. Gates, A. Haidar, J. Kurzak, P. Luszczek, S. Relton, J. Sistek, D. Stevens, et al., **"PLASMA 17 Performance Report**," *Innovative Computing Laboratory Technical Report*, ICL-UT-17-11: University of Tennessee, June 2017.

Abalenkovs, M., N. Bagherpour, J. Dongarra, M. Gates, A. Haidar, J. Kurzak, P. Luszczek, S. Relton, J. Sistek, D. Stevens, et al., "**PLASMA 17.1 Functionality Report**," *Innovative Computing Laboratory Technical Report*, ICL-UT-17-10: University of Tennessee, June 2017.

Abdelfattah, A., M. Baboulin, V. Dobrev, J. Dongarra, A. Haidar, I. Karlin, T. Kolev, I. Masliah, and S. Tomov, "**Small Tensor Operations on Advanced Architectures for High-Order Applications**," *Innovative Computing Laboratory Technical Report*, ICL-UT-17-10: University of Tennessee, April 2017.

Abdelfattah, A., A. Haidar, S. Tomov, and J. Dongarra, "**Fast Cholesky Factorization on GPUs for Batch and Native Modes in MAGMA**," *Journal of Computational Science*, vol. 20, pp. 85–93, May 2017.

Abdelfattah, A., H. Anzt, A. Bouteiller, A. Danalis, J. Dongarra, M. Gates, A. Haidar, J. Kurzak, P. Luszczek, S. Tomov, et al., "**Roadmap for the Development of a Linear Algebra Library for Exascale Computing: SLATE: Software for Linear Algebra Targeting Exascale**," *SLATE Working Notes*, no. 1, ICL-UT-17-02: Innovative Computing Laboratory, University of Tennessee, June 2017.

Abdelfattah, A., A. Haidar, S. Tomov, and J. Dongarra, "**Novel HPC Techniques to Batch Execution of Many Variable Size BLAS Computations on GPUs**," International Conference on Supercomputing (ICS '17), Chicago, Illinois, ACM, June 2017.

Abdelfattah, A., A. Haidar, S. Tomov, and J. Dongarra, "**Factorization and Inversion of a Million Matrices using GPUs: Challenges and Countermeasures**," *Procedia Computer Science*, vol. 108, pp. 606–615, June 2017.

Abdelfattah, A., K. Arturov, C. Cecka, J. Dongarra, C. Freitag, M. Gates, A. Haidar, J. Kurzak, P. Luszczek, S. Tomov, et al., "**C++ API for Batch BLAS**," *SLATE Working Notes*, no. 4, ICL-UT-17-12: University of Tennessee, December 2017.

Anzt, H., G. Collins, J. Dongarra, G. Flegar, and E. S. Quintana-Orti, "**Flexible Batched Sparse Matrix-Vector Product on GPUs**," The 8th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (ScalA '17), Denver, Colorado, ACM Press, November 2017.

Anzt, H., J. Dongarra, M. Gates, J. Kurzak, P. Luszczek, S. Tomov, and I. Yamazaki, "**Bringing High Performance Computing to Big Data Algorithms**," *Handbook of Big Data Technologies*: Springer, 2017.

Anzt, H., J. Dongarra, G. Flegar, and E. S. Quintana-Ortí, "**Batched Gauss-Jordan Elimination for Block-Jacobi Preconditioner Generation on GPUs**," The 8th International Workshop on Programming Models and Applications for Multicores and Manycores, New York, NY, USA, ACM, pp. 1–10, February 2017.

Anzt, H., J. Dongarra, G. Flegar, E. S. Quintana-Ortí, and A. E. Thomas, "**Variable-Size Batched Gauss-Huard for Block-Jacobi Preconditioning**," International Conference on Computational Science (ICCS 2017), vol. 108, Zurich, Switzerland, *Procedia Computer Science*, pp. 1783-1792, June 2017.

Anzt, H., M. Gates, J. Dongarra, M. Kreutzer, G. Wellein, and M. Kohler, "**Preconditioned Krylov Solvers on GPUs**," *Parallel Computing*, June 2017.

Anzt, H., J. Dongarra, G. Flegar, and E. S. Quintana-Orti, "**Variable-Size Batched LU for Small Matrices and Its Integration into Block-Jacobi Preconditioning**," 46th International Conference on Parallel Processing (ICPP), Bristol, United Kingdom, IEEE, August 2017.

Anzt, H., E. Boman, J. Dongarra, G. Flegar, M. Gates, M. Heroux, M. Hoemmen, J. Kurzak, P. Luszczek, S. Rajamanickam, et al., "**MAGMA-sparse Interface Design Whitepaper**," *Innovative Computing Laboratory Technical Report*, ICL-UT-17-05, September 2017.

Aupy, G., A. Benoit, L. Pottier, P. Raghavan, Y. Robert, and M. Shantharam, "**Co-Scheduling Algorithms for Cache-Partitioned Systems**," 19th Workshop on Advances in Parallel and Distributed Computational Models, Orlando, FL, IEEE Computer Society Press, May 2017.

Aupy, G., Y. Robert, and F. Vivien, "**Assuming failure independence: are we right to be wrong?**," *3rd International Workshop on Fault Tolerant Systems (FTS)*, Honolulu, Hawaii, IEEE, September 2017.

Baboulin, M., J. Dongarra, A. Remy, S. Tomov, and I. Yamazaki, "**Solving Dense Symmetric Indefinite Systems using GPUs**," *Concurrency and Computation: Practice and Experience*, vol. 29, issue 9, March 2017.

Bell, G., D. Bailey, A. H. Karp, J. Dongarra, and K. Walsh, "**A Look Back on 30 Years of the Gordon Bell Prize**," *International Journal of High Performance Computing and Networking*, vol. 31, issue 6, pp. 469–484, 2017.

Benoit, A., A. Cavelan, Y. Robert, and H. Sun, "**Multi-Level Checkpointing and Silent Error Detection for Linear Workflows**," *Journal of Computational Science*, April 2017.

Benoit, A., L. Pottier, and Y. Robert, "**Resilient Co-Scheduling of Malleable Applications**," *International Journal of High Performance Computing Applications* (IJHPCA), May 2017.

Benoit, A., F. Cappello, A. Cavelan, Y. Robert, and H. Sun, "**Identifying the Right Replication Level to Detect and Correct Silent Errors at Scale**," 2017 Workshop on Fault-Tolerance for HPC at Extreme Scale, Washington, DC, ACM, June 2017.

Benoit, A., A. Cavelan, V. Le Fèvre, and Y. Robert, "**Optimal Checkpointing Period with replicated execution on heterogeneous platforms**," 2017 Workshop on Fault-Tolerance for HPC at Extreme Scale, Washington, DC, IEEE Computer Society Press, June 2017.

Benoit, A., A. Cavelan, V. Le Fèvre, Y. Robert, and H. Sun, "**Towards Optimal Multi-Level Checkpointing**," *IEEE Transactions on Computers*, vol. 66, issue 7, pp. 1212–1226, July 2017.

Bosilca, G., A. Bouteiller, A. Guermouche, T. Herault, Y. Robert, P. Sens, and J. Dongarra, "**A Failure Detector for HPC Platforms**," *The International Journal of High Performance Computing Applications*, July 2017.

Dong, T., A. Haidar, S. Tomov, and J. Dongarra, "**Optimizing the SVD Bidiagonalization Process for a Batch of Small Matrices**," International Conference on Computational Science (ICCS 2017), Zurich, Switzerland, Procedia Computer Science, June 2017.

Dongarra, J., S. Hammarling, N. Higham, S. Relton, P. Valero-Lara, and M. Zounon, "**The Design and Performance of Batched BLAS on Modern High-Performance Computing Systems**," International Conference on Computational Science (ICCS 2017), Zürich, Switzerland, Elsevier, June 2017.

Dongarra, J., S. Hammarling, N. Higham, S. Relton, and M. Zounon, "**Optimized Batched Linear Algebra for Modern Architectures**," Euro-Par 2017, Santiago de Compostela, Spain, Springer, August 2017.

Dongarra, J., "**Report on the TianHe-2A System**," *Innovative Computing Laboratory Technical Report*, ICL-UT-17-04: University of Tennessee, September 2017.

Dongarra, J., A. Haidar, O. Hernandez, S. Tomov, and M. Grentla Venkata, "**POMPEI: Programming with OpenMP4 for Exascale Investigations**," *Innovative Computing Laboratory Technical Report*, ICL-UT-17-09: University of Tennessee, December 2017.

Eberius, D., T. Patinyasakdikul, and G. Bosilca, "**Using Software-Based Performance Counters to Expose Low-Level Open MPI Performance Information**," EuroMPI, Chicago, IL, ACM, September 2017.

Fang, A., A. Cavelan, Y. Robert, and A. Chien, "**Resilience for Stencil Computations with Latent Errors**," International Conference on Parallel Processing (ICPP), Bristol, UK, IEEE Computer Society Press, August 2017.

Faverge, M., J. Langou, Y. Robert, and J. Dongarra, "**Bidiagonalization and R-Bidiagonalization: Parallel Tiled Algorithms, Critical Paths and Distributed-Memory Implementation**," IEEE International Parallel and Distributed Processing Symposium (IPDPS), Orlando, FL, IEEE, May 2017.

Fayad, D., J. Kurzak, P. Luszczek, P. Wu, and J. Dongarra, "**The Case for Directive Programming for Accelerator Autotuner Optimization**," *Innovative Computing Laboratory Technical Report*, ICL-UT-17-07: University of Tennessee, October 2017.

Gates, M., P. Luszczek, J. Kurzak, J. Dongarra, K. Arturov, C. Cecka, and C. Freitag, "**C++ API for BLAS and LAPACK**," *SLATE Working Notes*, no. 2, ICL-UT-17-03: Innovative Computing Laboratory, University of Tennessee, June 2017.

Gates, M., J. Kurzak, P. Luszczek, Y. Pei, and J. Dongarra, "**Autotuning Batch Cholesky Factorization in CUDA with Interleaved Layout of Matrices**," Parallel and Distributed Processing Symposium Workshops (IPDPSW), Orlando, FL, IEEE, June 2017.

Gates, M., S. Tomov, and J. Dongarra, "**Accelerating the SVD Two Stage Bidiagonal Reduction and Divide and Conquer Using GPUs**," *Parallel Computing*, November 2017.

Haidar, A., A. Abdelfattah, S. Tomov, and J. Dongarra, "**High-performance Cholesky Factorization for GPU-only Execution**," *Proceedings of the General Purpose GPUs* (GPGPU-10), Austin, TX, ACM, February 2017.

Haidar, A., H. Jagode, A. YarKhan, P. Vaccaro, S. Tomov, and J. Dongarra, "**Power-aware Computing: Measurement, Control, and Performance Analysis for Intel Xeon Phi**," 2017 IEEE High Performance Extreme Computing Conference (HPEC'17), Best Paper Finalist, Waltham, MA, IEEE, September 2017.

Haidar, A., K. Kabir, D. Fayad, S. Tomov, and J. Dongarra, "**Out of Memory SVD Solver for Big Data**," 2017 IEEE High Performance Extreme Computing Conference (HPEC'17), Waltham, MA, IEEE, September 2017.

Haidar, A., P. Wu, S. Tomov, and J. Dongarra, "**Investigating Half Precision Arithmetic to Accelerate Dense Linear System Solvers**," ScalA17: 8th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems, Denver, CO, ACM, 11/2017.

Han, L., L-C. Canon, H. Casanova, Y. Robert, and F. Vivien, "**Checkpointing Workflows for Fail-Stop Errors**," IEEE Cluster, Honolulu, Hawaii, IEEE, September 2017.

Jagode, H., A. Danalis, and J. Dongarra, "**Accelerating NWChem Coupled Cluster through Dataflow-Based Execution**," *International Journal of High Performance Computing Applications*, pp. 1–13, January 2017.

Jagode, H., "**Dataflow Programming Paradigms for Computational Chemistry Methods**," *Innovative Computing Laboratory Technical Report*, ICL-UT-17-01: University of Tennessee, May 2017.

Kabir, K., A. Haidar, S. Tomov, A. Bouteiller, and J. Dongarra, "**A Framework for Out of Memory SVD Algorithms**," ISC High Performance 2017, pp. 158–178, June 2017.

Kurzak, J., P. Luszczek, I. Yamazaki, Y. Robert, and J. Dongarra, "**Design and Implementation of the PULSAR Programming System for Large Scale Computing**," *Supercomputing Frontiers and Innovations*, vol. 4, issue 1, 2017.

Kurzak, J., P. Wu, M. Gates, I. Yamazaki, P. Luszczek, G. Ragghianti, and J. Dongarra, "**Designing SLATE: Software for Linear Algebra Targeting Exascale**," *SLATE Working Notes*, no. 3, ICL-UT-17-06: Innovative Computing Laboratory, University of Tennessee, October 2017.

M. Lopez, G., V. Larrea, W. Joubert, O. Hernandez, A. Haidar, S. Tomov, and J. Dongarra, "**Evaluation of Directive-based Performance Portable Programming Models**," *International Journal of High Performance Computing and Networking (IJHPCN)*, vol. (In Press), 2017.

"**BDEC Pathways to Convergence: Toward a Shaping Strategy for a Future Software and Data Ecosystem for Scientific Inquiry**," I*nnovative Computing Laboratory Technical Report*, ICL-UT-17-08: University of Tennessee, November 2017.

Seo, S., A. Amer, P. Balaji, C. Bordage, G. Bosilca, A. Brooks, P. Carns, A. Castello, D. Genet, T. Herault, et al., "**Argobots: A Lightweight Low-Level Threading and Tasking Framework**," *IEEE Transactions on Parallel and Distributed Systems*, October 2017.

Yamazaki, I., S. Nooshabadi, S. Tomov, and J. Dongarra, "**Structure-aware Linear Solver for Realtime Convex Optimization for Embedded Systems**," *IEEE Embedded Systems Letters*, vol. 09, issue 03, pp. 61–64, May 2017.

Yamazaki, I., M. Hoemmen, P. Luszczek, and J. Dongarra, "**Improving Performance of GMRES by Reducing Communication and Pipelining Global Collectives**," *Proceedings of The 18th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC 2017)*, Best Paper Award, Orlando, FL, June 2017.

Yamazaki, I., S. Tomov, and J. Dongarra, "**Sampling Algorithms to Update Truncated SVD**," IEEE International Conference on Big Data, December 2017.

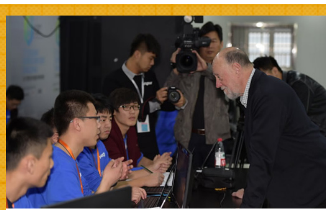
Zhao, Y., L. Wan, W. Wu, G. Bosilca, R. Vuduc, J. Ye, W. Tang, and Z. Xu, "**Efficient Communications in Training Large Scale Neural Networks**," ACM MultiMedia Workshop 2017, Mountain View, CA, ACM, October 2017.

ICL/EVENTS

Every 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. Participating in the intellectual life of the scientific community in this way is an essential part of the research process.

January 9 / Trieste, Italy
Master in HPC Advanced Programming and Software Optimization

Feb 27–Mar 2 / Portland, OR
MPI Forum



June 5–9 / Riverhead, NY
2017 GPU Hackathon

January 10 / Argonne, IL
ECP CEED Meeting

Feb 27–Mar 3 / Atlanta, GA
SIAM Conference on Computational Science and Engineering

April 28 / Wuxi, China
ASC17 Student Supercomputer Challenge

June 12–22 / Zurich, Switzerland
The International Conference on Computational Science (ICCS17)

January 24–26 / San Jose, CA
Open MPI Developers Meeting

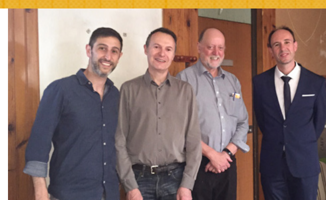
May 8–11 / San Jose, CA
NVIDIA's GPU Technology Conference (GTC)

June 19–22 / Frankfurt, Germany
ISC High Performance (ISC17)

Jan 30–Feb 2 / Knoxville, TN
ECP All-Hands Meeting

March 9–10 / Wuxi, China
Workshop on Big Data and Extreme-scale Computing (BDEC)

May 11–12 / Toulouse, France
CERFACS 30-Year Conference



February 4 / Austin, TX
Principles and Practice of Parallel Programming (PPoPP)

Mar 29–30 / Heidelberg, Germany
Parallel 2017

June 23 / Frankfurt, Germany
VI-HPS 10th Anniversary Workshop

February 4-5 / Austin, TX
GPGPU-10 Workshop

April 4–5 / Atlanta, GA
Future Online Analysis Platform

May 24–26 / Knoxville, TN
The 12th Scheduling for Large Scale Systems Workshop

July 5–8 / Schloss Ringberg, Germany
Power-Aware Computing (PACO2017)

February 23–25 / Atlanta, GA
Workshop on Batched, Reproducible, and Reduced Precision BLAS

May 29–Jun 1 / Orlando, FL
International Parallel & Distributed Processing Symposium (IPDPS)

Jun 26–28 / Lugano, Switzerland
Platform for Advanced Scientific Computing (PASC17)

July 10–14 / Pittsburgh, PA
2017 SIAM Annual Meeting

Jul 17–19 / Urbana, IL
7th JLESC Workshop

Aug 15–17 / Livermore, CA
CEED First Annual Meeting

Aug 27–30 / Dagstuhl, Germany
Dagstuhl Seminar 17352: Analysis and Synthesis of Floating-Point Programs

Aug 29–31 / Gatlinburg, TN
Smoky Mountains Computational Sciences & Engineering Conference (SMC17)

Sep 5–8 / Atlanta, GA
DOE Async IS Project Meeting

Sep 11–12 / Dresden, Germany
11th Parallel Tools Workshop

Sep 12–14 / Waltham, MA
IEEE High Performance Extreme Computing Conference (HPEC)

Sep 13–15 / Los Angeles, CA
Linux Plumbers Conference 2017

Sep 25–28 / Chicago, IL
EuroMPI/USA 2017

Nov 12–17 / Denver, CO
The International Conference for High Performance Computing, Networking, Storage and Analysis (SC17)

Dec 11–14 / Boston, MA
2017 IEEE International Conference on Big Data

The International Conference for High Performance Computing Networking, Storage, and Analysis (SC), established in 1988, is a staple of ICL's November itinerary. SC is vital to the growth and evolution of HPC in the United States because it is the only US event that elicits substantial participation from all segments of the HPC community, including hundreds of users, developers, vendors, research institutions, and representatives of government funding agencies. Such a talent-rich gathering enables participants to discuss challenges, share innovations, and coordinate relationships and collaborations with some of the best minds in scientific and high-performance computing.

SC17 was held in Denver, Colorado, on November 12–17. As usual, ICL had a significant presence at SC, with faculty, research staff, and students giving talks, presenting papers, and leading "Birds of a Feather" sessions. Since the University of Tennessee did not have a booth at SC this year, ICL leveraged an online "virtual booth" through which interested parties could keep tabs on ICL-related events at SC, including a list of attendees and detailed schedule of talks.

FIND OUT MORE AT
http://www.icl.utk.edu/sc17/

ICL / PARTNERSHIPS

Since its inception in 1989, ICL has fostered relationships with many academic institutions and research centers and has proactively built enduring collaborative partnerships with HPC vendors and industry leaders in the United States and abroad. In this section, we recognize many of those partners and collaborators, most of whom we continue to work with today.

GOVERNMENT AND ACADEMIC

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

INDUSTRY

AMD
Atos
Cray
Engility
Fujitsu
IBM
IERUS Technologies
Intel
MathWorks
NAG
NVIDIA

INTEL SCIENCE AND TECHNOLOGY CENTER FOR BIG DATA

ISTC BIG DATA

ICL joined the SciDB project of the Intel Science and Technology Center (ISTC) for Big Data, one of a series of research collaborations that Intel is establishing with universities in the United States to identify and prototype revolutionary technology opportunities, and exchange expertise in various fields of HPC.

In the case of ICL, the lab will help improve the efficiency of large-scale data analytics by providing efficient codes for linear algebra on the Intel Xeon Phi coprocessor. The lab will also provide expertise on fault tolerance to help make the compute intensive portion of data management more resilient, which is essential given the large databases used in Big Data applications. Finally, the distributed nature of large data processing calls for optimal data distribution and redistribution operations, which has long been one of ICL's core strengths.

INTERNATIONAL COLLABORATIONS

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 Federale de Lausanne
Lausanne, Switzerland

ETH Zurich
Zurich, 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

Hokkaido University
Sapporo, Japan

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
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
Erlangen, Germany

RIKEN
Wako, Japan

Rutherford Appleton Laboratory
Oxford, England

Soongsil University
Seoul, South Korea

Technische Universitaet 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

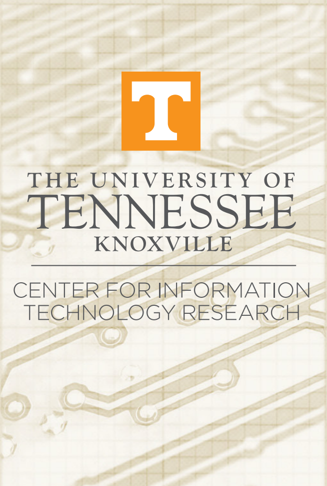
University of Paris-Sud
Paris, France

University of Picardie Jules Verne
Amiens, France

University of Tsukuba
Tsukuba, Japan

ICL/LEADERSHIP

While leading-edge research and high-impact software are hallmarks of the ICL mission, the lab also cultivates a strong core of leadership. To this end, ICL actively engages the HPC and computational research communities through impactful efforts like those outlined below.



FIND OUT MORE AT
http://citr.cs.utk.edu/

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 research 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 research. CITR has also provided secondary support for other UTK research centers.

CITR's second objective is to grow an interdisciplinary computational science program as part of the university curriculum. To this end, CITR helped establish the Interdisciplinary Graduate Minor in Computational Science (IGMCS) to offer UTK graduate students an opportunity to acquire the balanced package of knowledge and skills required for today's computationally intensive research methods. CITR is also the sole provider of administrative support for the IGMCS.

FIND OUT MORE AT
http://igmcs.utk.edu/

INTERDISCIPLINARY GRADUATE MINOR IN COMPUTATIONAL SCIENCE

Addressing the need for a new educational strategy in computational science, CITR worked with faculty and administrators from several departments and colleges in 2007 to help establish a new university-wide program that supports advanced degree concentrations in this critical new field across the curricula. Under the IGMCS, students pursuing advanced degrees in a variety of fields of science and engineering are able to extend 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. Through the IGMCS, graduate students can augment their graduate work in their chosen field with courses from other disciplines that are specifically tailored to round out their education in computational science. The IGMCS curriculum, requirements, and policies are governed by a program committee composed of faculty members from participating IGMCS academic units and departments.

BIG DATA AND EXTREME-SCALE COMPUTING

BDEC China Group Photo / MARCH 2017

In the past several years, the United States, the European Union, Japan, and China have each moved aggressively to develop their own plans for achieving exascale computing in the next decade. Such concerted planning by the traditional leaders of HPC speaks eloquently about both the substantial rewards that await the success of such efforts, and about the unprecedented technical obstacles that apparently block the path upward to get there. But while these exascale initiatives, including the International Exascale Software Project (IESP), have understandably focused on the big challenges of exascale for hardware and software architectures, the relatively recent emergence of the phenomena of "big data" in a wide variety of scientific fields represents a tectonic shift that is transforming the entire research landscape on which all plans for exascale computing must play out.

The workshop series on Big Data and Extreme-scale Computing (BDEC) marks a distinctly new phase for the work of the IESP community and is premised on the idea that we must begin to systematically map out and account for the ways in which the major issues associated with big data intersect with, impinge upon, and potentially change the national (and international) plans that are now being laid out for achieving exascale computing.

In 2017, ICL was instrumental in organizing and staging the fifth BDEC workshop in Wuxi, China. Along with Jack Dongarra, and following through with work they began with the IESP and the first four BDEC meetings in Charleston, Fukuoka, Barcelona, and Frankfurt, several members of ICL's CITR staff—including Terry Moore, Tracy Rafferty, and David Rogers—played essential roles in making the fifth BDEC workshop a major success.

A follow-up BDEC BoF was held in Denver, Colorado during SC17. This meeting marked the release of the BDEC "Pathways to Convergence" report, which examines the progress toward (or potential for) convergence of HPC and emerging high-end data analysis at three different levels: (1) large-scale scientific applications, (2) large-scale HPC platforms, and (3) next generation distributed services.

The next BDEC workshop is planned for the spring of 2018. Given the ever-increasing emphasis that science, government, and industry continue to place on both big data and extreme-scale computing, this example of ICL's community leadership seems likely to become more and more prominent.

FIND OUT MORE AT
http://www.exascale.org/

ICL / PEOPLE

ICL Group / FALL 2017

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.

As part of an engineering college at a top 50 public research university, we have a responsibility to combine exemplary teaching with cutting-edge research. As such, we regularly employ bright and motivated graduate and undergraduate students. We have been, and will continue to be, very proactive in securing internships and assistantships for highly motivated and hardworking students.

In 2017, five members of ICL earned their degrees from UTK under the guidance and mentorship of Prof. Dongarra and ICL's research scientists.

- Chongxiao Cao, PhD in Computer Science, May 2017
- Heike Jagode, PhD in Computer Science, May 2017
- Khairul Kabir, PhD in Computer Science, May 2017
- Sangamesh Ragate, MS in Computer Science, May 2017
- Wei Wu, PhD in Computer Science, May 2017

WEI WU, CHONGXIAO CAO, GEORGE BOSILCA, AND HEIKE JAGODE

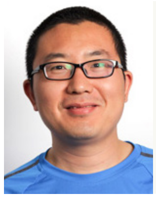
Ahmad Abdelfattah
RESEARCH SCIENTIST I

Hartwig Anzt
RESEARCH SCIENTIST I

Matthew Bachstein
GRADUATE RESEARCH ASSISTANT

George Bosilca
RESEARCH ASST. PROFESSOR

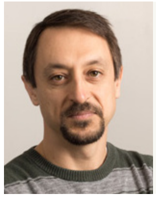
Aurelien Bouteiller
RESEARCH DIRECTOR



Qinglei Cao
GRADUATE RESEARCH ASSISTANT

Earl Carr
PROGRAM ADMINISTRATOR

Sam Crawford
INFORMATION SPECIALIST II



Anthony Danalis
RESEARCH DIRECTOR

Jack Dongarra
UNIVERSITY DISTINGUISHED PROFESSOR / DIRECTOR OF ICL

David Eberius
GRADUATE RESEARCH ASSISTANT

Teresa Finchum
ADMINISTRATIVE SPECIALIST II

Jamie Finney
SOFTWARE ENGINEER

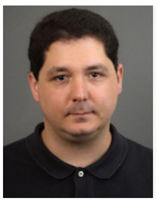
Mark Gates
RESEARCH SCIENTIST II

Damien Genet
RESEARCH SCIENTIST I

Azzam Haidar
RESEARCH SCIENTIST II

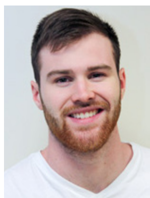
Hanumantharayappa
GRADUATE RESEARCH ASSISTANT

John Henry
RESEARCH LEADER I



Thomas Herault
RESEARCH DIRECTOR

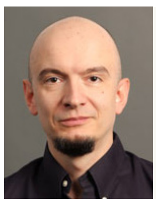
Reazul Hoque
GRADUATE RESEARCH ASSISTANT



Harry Hughes
GRADUATE RESEARCH ASSISTANT

Heike Jagode
RESEARCH ASST. PROFESSOR

Cindy Knisley
FINANCIAL SPECIALIST



Jakub Kurzak
RESEARCH ASST. PROFESSOR

STAFF AND STUDENTS
CONTINUED

Julie Langou
RESEARCH LEADER

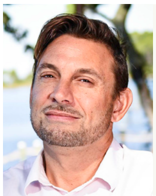
Jiali Li
GRADUATE RESEARCH ASSISTANT

Xi Luo
GRADUATE RESEARCH ASSISTANT

Piotr Luszczek
RESEARCH ASST. PROFESSOR

Tyler McDaniel
GRADUATE RESEARCH ASSISTANT

Terry Moore
ASSOCIATE DIRECTOR



Phil Mucci
RESEARCH CONSULTANT

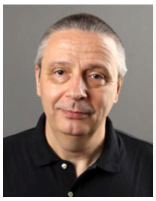
Thananon Patinyasakdikul
GRADUATE RESEARCH ASSISTANT

Yu Pei
GRADUATE RESEARCH ASSISTANT

Tracy Rafferty
PROGRAM MANAGER

Gerald Ragghianti
RESEARCH LEADER

Stephen Richmond
GRADUATE RESEARCH ASSISTANT



Yves Robert
VISITING SCHOLAR

David Rogers
IT SPECIALIST III

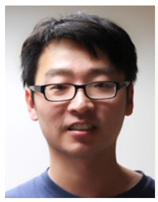
Leighanne Sisk
ADMINISTRATIVE SPECIALIST I

Stanimire Tomov
RESEARCH ASST. PROFESSOR

Yaohung Tsai
GRADUATE RESEARCH ASSISTANT

Phil Vaccaro
GRADUATE RESEARCH ASSISTANT

Frank Winkler
CONSULTANT



Panruo Wu
POST DOCTORAL RESEARCH ASSOCIATE

Ichitaro Yamazaki
RESEARCH SCIENTIST I

Asim YarKhan
RESEARCH SCIENTIST II

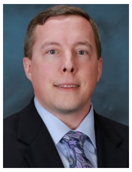
Dong Zhong
GRADUATE RESEARCH ASSISTANT

ICL/VISITORS

Since its founding, ICL has had a tradition of hosting many visitors from around the world. Some stay only briefly to give insightful seminars or presentations; others remain with us for as long as a year to collaborate, teach, and learn. 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.

Dorian Arnold
EMORY UNIVERSITY

Camille Coti
UNIVERSITE PARIS 13, FRANCE



Chris Davis
OAK RIDGE NATIONAL LABORATORY

Markus Eisenbach
OAK RIDGE NATIONAL LABORATORY

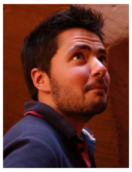
Mathieu Faverge
CENTRE DE RECHERCHE INRIA BORDEAUX - SUD-OUEST, FRANCE

Diana Fayad
UNIV. OF VERSAILLES SAINT-QUENTIN-EN-YVELINES, FRANCE

Ryan Glasby
JOINT INSTITUTE FOR COMPUTATIONAL SCIENCES

Amina Guermouche
UNIVERSITY OF PARIS-SUD, FRANCE

Benjamin Hernandez
OAK RIDGE NATIONAL LABORATORY



Julien Herrmann
GEORGIA TECH

Scott Klasky
OAK RIDGE NATIONAL LABORATORY

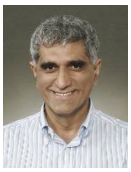
Jeff Larkin
NVIDIA

Pierre Lemarinier
IBM DUBLIN, IRELAND

Nuria Losada
UNIVERSITY OF A CORUÑA, SPAIN

Hatem Ltaief
KING ABDULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Samuel Mabakane
CENTRE FOR HIGH PERFORMANCE COMPUTING, SOUTH AFRICA



Saeid Nooshabadi
MICHIGAN TECHNOLOGICAL UNIVERSITY

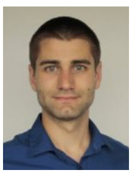
George Ostrouchov
OAK RIDGE NATIONAL LABORATORY

Lynne Parker
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE, UTK

Charles Phillips
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE, UTK

Sushil Prasad
NATIONAL SCIENCE FOUNDATION

Arvind Ramanathan
OAK RIDGE NATIONAL LABORATORY



Joseph Schuchart
HIGH PERFORMANCE COMPUTING CENTER, STUTTGART, GERMANY

Frédéric Vivien
INRIA, FRANCE

Sophie Voisin
OAK RIDGE NATIONAL LABORATORY

Kwai Wong
JOINT INSTITUTE FOR COMPUTATIONAL SCIENCES

Dmitry Zaitsev
INTERNATIONAL HUMANITARIAN UNIVERSITY UKRAINE

ICL/ALUMNI

Since its inception, 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 Dr. Dongarra, beginning a long list of top 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
Jennifer Allgeyer
Wes Alvaro
Ed Anderson
Daniel Andrzejewski
Thara Angskun
Papa Arkhurst
Dorian Arnold
Cedric Augonnet
Marc Baboulin
Zhaojun Bai
Ashwin Balakrishnan
Richard Barrett
Alex Bassi
David Battle
Micah Beck
Daniel Becker
Dulceneia Becker
Adam Beguelin
Annamaria Benzoni
Tom Berry
Vincent Berthoux
Scott Betts
Nikhil Bhatia
Noel Black
Laura Black
Susan Blackford
Wesley Bland
Kartheek Bodanki
David Bolt
Fernando Bond
Carolyn Bowers
Barry Britt
Randy Brown
Bonnie Browne
Cynthia Browne
Murray Browne
Giuseppe Bruno
Antonin Bukovsky
Greg Bunch
Alfredo Buttari
Anthony Canino
Domingo Gimenez Canovas
Chongxiao "Shawn" Cao
Henri Casanova
Cedric Castagnede
Ramkrishna Chakrabarty
Sharon Chambers
Zizhong Chen
Jaeyoung Choi
Wahid Chrabakh
Eric Clarkson
Andy Cleary
Michelle Clinard
Vincent Cohen-Addad
Matthias Colin
Charles Collins
Stephanie Moreaud Cooper
Tom Cortese
Camille Coti
Jason Cox
David Cronk
Javier Cuenca
Manoel Cunha
Yuanshun (Shaun) Dai
Cricket Deane
Remi Delmas
Frederic Desprez
Ying Ding
Jun Ding
Jin Ding
Martin Do
Simplice Donfack
Tingxing "Tim" Dong
Leon Dong
Nick Dongarra
David Doolin
Joe Dorris
Andrew Downey
Mary Drake
Julio Driggs
Brian Drum
Peng Du
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
Markus Fischer
Len Freeman
Xiaoquan Fu
Erika Fuentes
Karl Fuerlinger
Megan Fuller
Edgar Gabriel
Tracy Gangwer
Lynn Gangwer
Nathan Garner
Kelley Garner
Tina Garrison
Adriana Garties
Peter Gaultney
Christoph Geile
Jean Patrick Gelas
Boris Gelfend
Jonathan Gettler
Scott Gibson
Eric Greaser
Stan Green
Alice Gregory
Amina Guermouche
Jason Gurley
Bilel Hadri
Hunter Hagewood
Christian Halloy
Sven Hammarling
J. Mike Hammond
Satomi Hasegawa
Hidehiko Hasegawa
Chris Hastings
Blake Haugen
David Henderson
Greg Henry
Julien Herrmann
Holly Hicks
Alexandra Hicks-Hardiman
Sid Hill
Tomoyuki Hiroyasu
George Ho
Josh Hoffman
Jeff Horner
Mitch Horton
Yan Huang
Aurelie Hurault
Chris Hurt
Paul Jacobs
Emmanuel Jeannot
Weizhong Ji

Yulu Jia
Weicheng Jiang
Song Jin
Patrick Johansson
Matt Johnson
Aral 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
Myung Ho Kim
Youngbae Kim
Jenya Kirshtein
Michael Kolatis
Moritz Kreutzer
Chandra Krintz
Tilman Kuestner
Krerkchai Kusolchu
Coire Kyle
Amanda Laake
Xavier Lacoste
Julien Langou
Jeff Larkin
Brian LaRose
Frank Lauer
Valentin Le Fevre
Tracy Lee
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
Daniel Lucio
Richard Luczak
Teng Ma
Sticks Mabakane
Robert Manchek
Gabriel Marin
Tushti Marwah
Theo Mary
Ian Masliah
Donald McCasland
Paul McMahan
Eric Meek
James Meyering
Jeremy Millar
Michelle Miller
Cindy Mitchell
Stuart Monty
Erik Moore
Keith Moore
Shirley Moore
Robert Morgan
Kishan Motheramgari
Steven Moulton
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
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
Sangamesh Ragate
James Ralph
Ganapathy Raman
Kamesh Ramani
Mei Ran
Arun Rattan
Sheri Reagan
Mike Reynolds
George Rhinehart
Jon Richardson
Ken Roche
Andrew Rogers
Tom Rothrock
Tom Rowan
Narapat (Ohm) Saengpatsa
Kiran Sagi
Evelyn Sams
Ken Schwartz
Keith Seymour
Farial Shahnaz
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
Volodymyr Turchenko
Lauren Vaca
Sathish Vadhiyar
Robert van de Geijn
Chad Vawter
Eugene Vecharynski
Scott Venckus
Antoine Vernois
Reed Wade
Michael Walters
Mike Waltz
Robert Waltz
Jerzy Wasniewski
Vince Weaver
Scott Wells
David West
R. Clint Whaley
Jody Whisnant
James White
Scotti Whitmire
Susan Wo
Felix Wolf
Stephen Wood
Jiayi Wu
Wei Wu
Qiu Xia
Tinghua Xu
Tao Yang
Erlin Yao
Kevin Ye
Jin Yi
Haihang You
Lamia Youseff
Brian Zachary
Dmitry Zaitsev
Omar Zenati
Yuanlei Zhang
Junlong Zhao
Yong Zheng
Min Zhou
Luke Zhou

ACKNOWLEDGMENTS

For nearly 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. Therefore, we acknowledge the following agencies for supporting our efforts, both past and present:

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:

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

CENTER FOR INFORMATION
TECHNOLOGY RESEARCH

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

CITR is supported in part by the University of Tennessee Office of Research and Engagement.

INNOVATIVE COMPUTING LABORATORY

Suite 203 Claxton
1122 Volunteer Blvd
Knoxville, TN 37996-3450

INNOVATIVE
COMPUTING LABORATORY

FIND OUT MORE AT
http://www.icl.utk.edu/