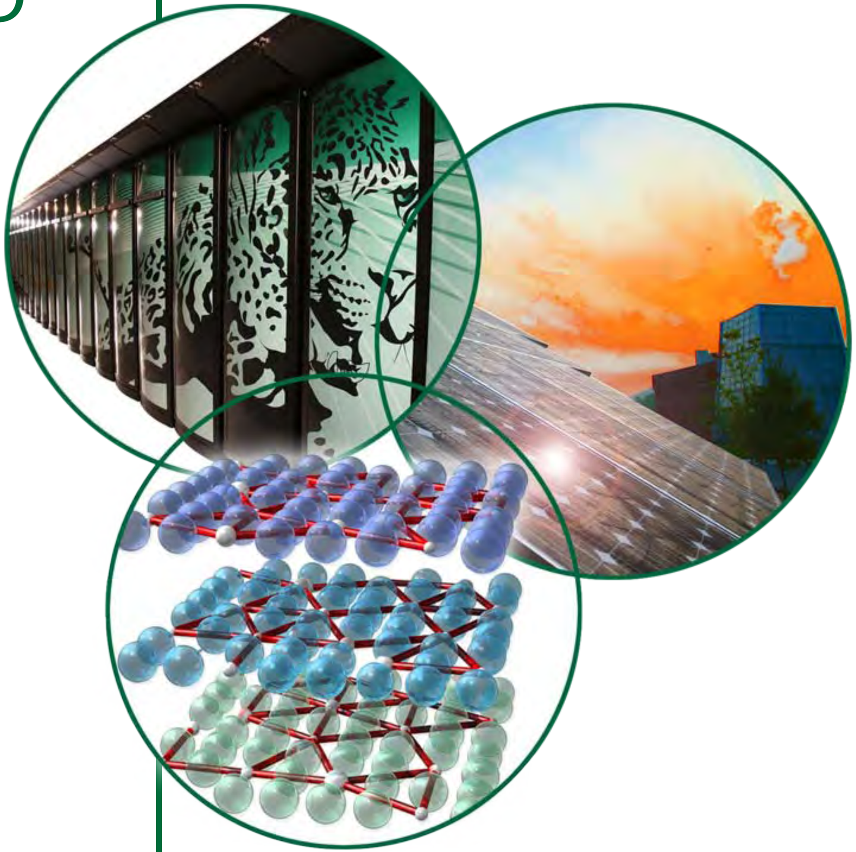
EASI MATH/CS INSTITUTE
PAVING THE ROAD TO EXASCALE



Al Geist

ORNL Corporate Fellow
Oak Ridge National Laboratory

ASCAC Meeting

Oak Ridge, TN
November 4, 2009

U.S. DEPARTMENT OF ENERGY

Research supported by DOE ASCR

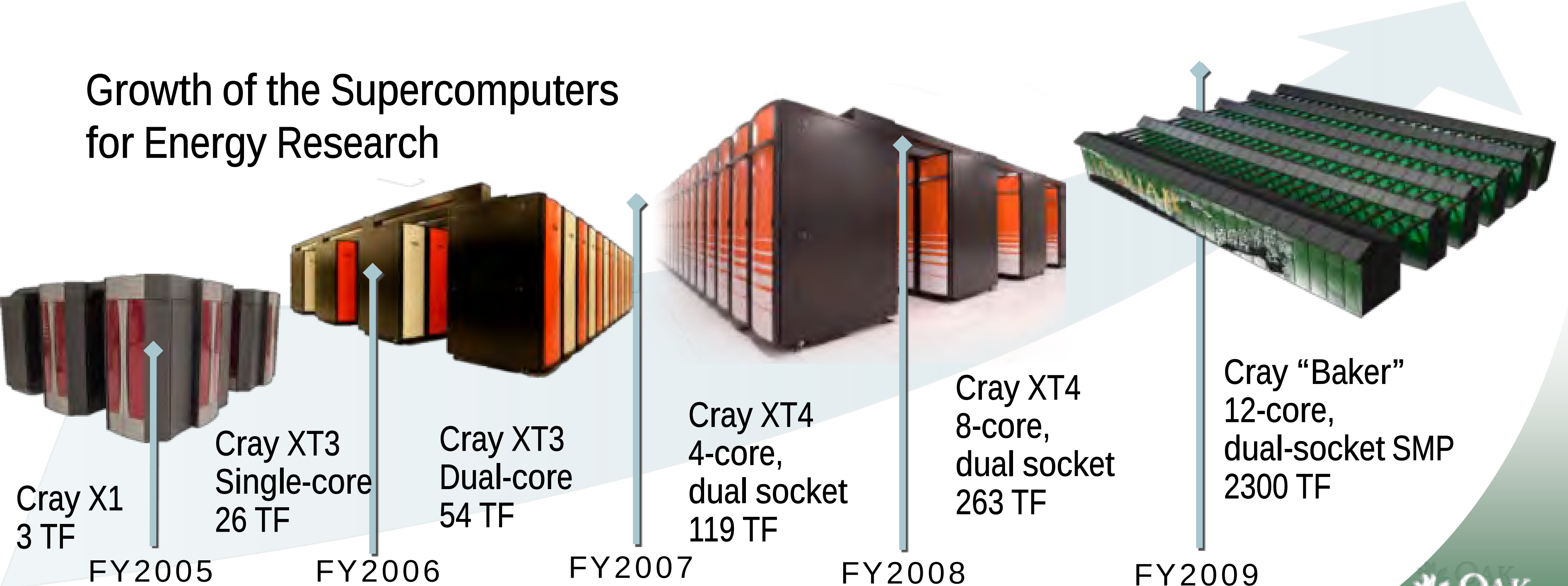
OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Petascale Roadmap
Oak Ridge increased computational capability by almost 1000X in half a decade.

ORNL Leadership Computing Facility successfully executed its petascale roadmap plan on schedule and budget.	Mission: Delivering resources for science breakthroughs. Multiple science applications now running at over a sustained petaflop

Growth was driven by multi-core sockets and increase in the number of cores per node

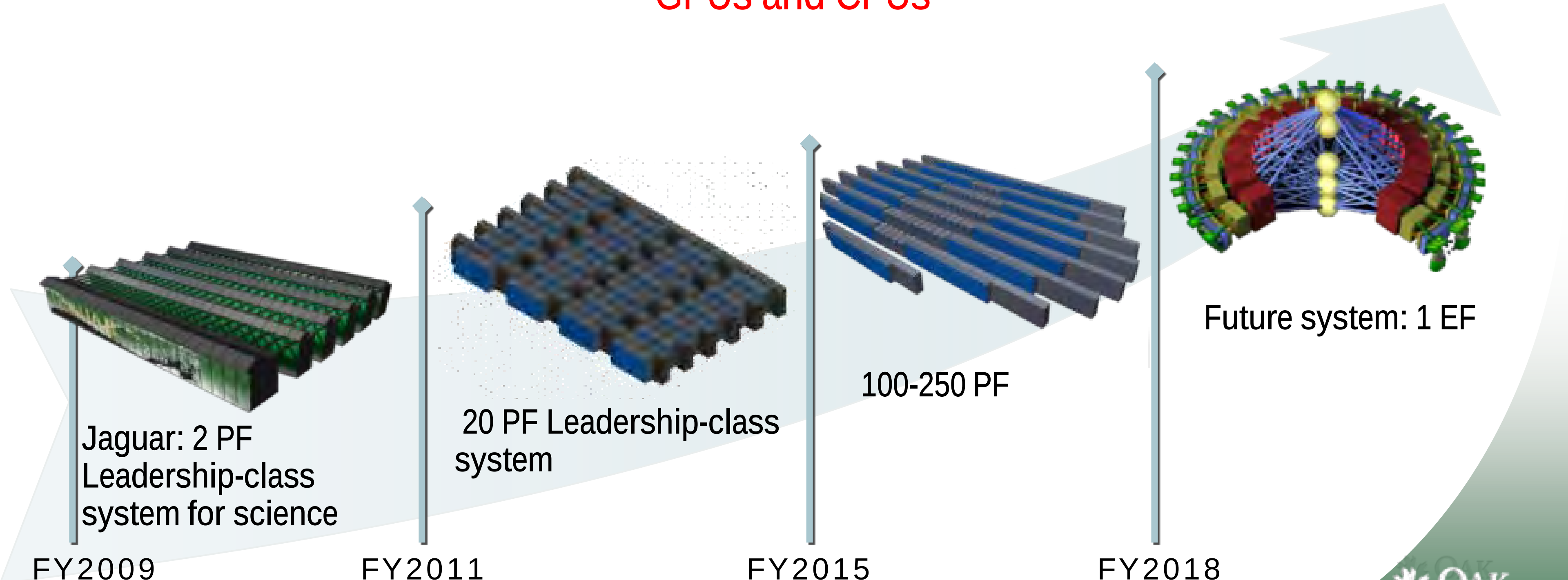
Growth of the Supercomputers for Energy Research



Exascale Roadmap
Delivering the next 1000x capability in a decade

Mission need: Provide the computational resources required to tackle critical national problems	Must also provide the expertise and tools to enable science teams to productively utilize exascale systems

Expectation is that systems will be heterogeneous with nodes composed of many-core GPUs and CPUs



Impediments to Useful Exascale Computing

Danger curves ahead

- **Scalability**
  - 10,000,000 nodes
  - 100,000,000 cores
  - 1,000,000,000 threads
- **Resilience**
  - Perhaps a harder problem than all the others
  - Do Nothing: an MTBI of 10's of minutes
- **Power Consumption**
  - Do Nothing: 100 to 140 MW
- **Programming Environment**
  - Data movement and heterogeneous architectures will drive new paradigms

- **Data Movement**
  - **Local**
    - node architectures
    - memory
  - **Remote**
    - Interconnect
    - Link BW
    - Messaging Rate
  - **File I/O**
    - Network Architectures
    - Parallel File Systems
    - Latency and Bandwidth

This talk describes two complementary ASCR Math/CS projects paving the way to exascale

They share a common goal:
Closing the "application-architecture performance gap"
The difference between the peak performance of a system and the performance achieved by real science applications

The IAA Algorithms Project begun in FY2009
Focused on homogeneous multi-core systems, and extreme scale system simulations. Hierarchical programming models

EASI Joint Math/CS Institute begun in FY2010
Focused on heterogeneous systems with accelerators and application resilience. Hybrid programming models

Both projects share a common approach to success

Integrated team of math, CS, and application experts working together to create new . . .

Architecture-aware algorithms and associated runtime to enable many science applications to better exploit the architectural features of DOE's petascale systems.

Applications team members immediately incorporate new algorithms providing **Near-term high impact on science**

Numerical libraries used to disseminate the new algorithms to the wider community providing **broader and longer-term impact.**

IAA - Institute for Advanced Architectures and Algorithms

- Begun in FY2009 as Joint effort between Sandia National Labs and Oak Ridge National lab, it has a steering committee, advisory board, and underlying project(s)
- Focused R&D on key impediments to high performance in partnership with industry and academia
- **Foster the integrated co-design of architectures and algorithms** to enable more efficient and timely solutions to mission critical problems
- Impact vendor roadmaps through partnership and joint research and development

IAA Algorithms Project is funded through this Institute

IAA Algorithms Project Overview

It all revolves around the science

Multi-core Aware
- Hierarchical, Parallel in time

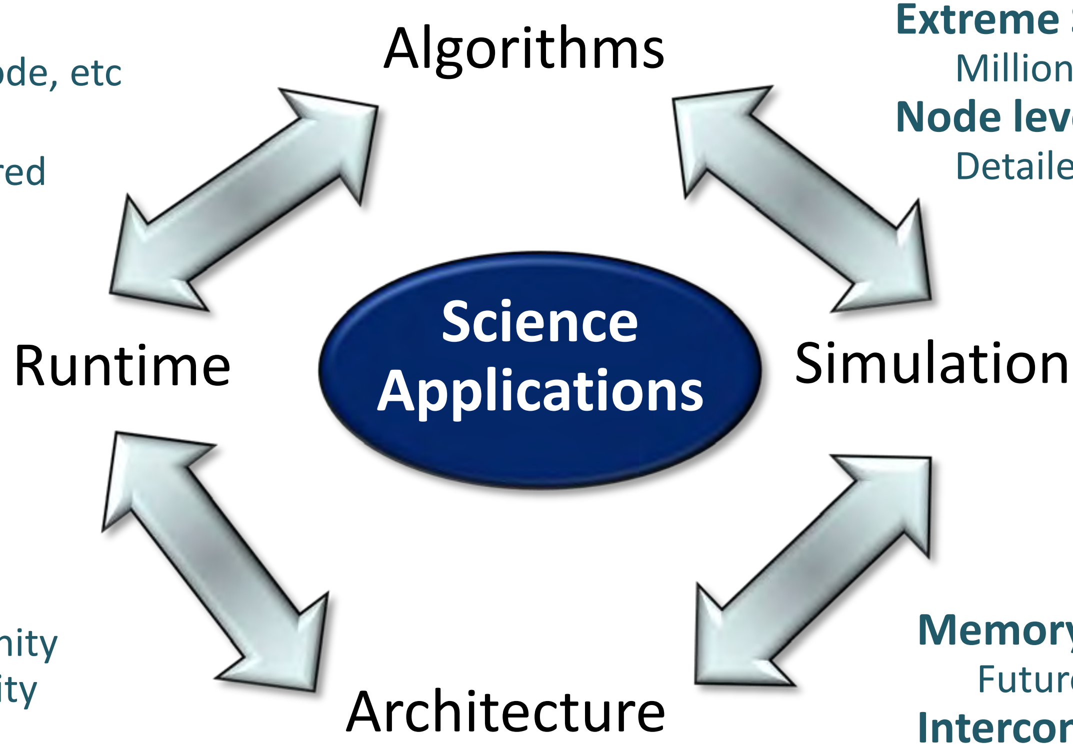
Multi-precision
- Krylov, Poisson, Helmholtz

Hierarchical MPI
- MPI_Comm_Node, etc

Shared memory
- MPI_Alloc_Shared

Extreme Scale
- Million node systems

Node level
- Detailed kernel studies



Multi-core
- Processor affinity
- Memory affinity
- Scheduling
- Threading

Memory hierarchy
- Future designs

Interconnect
- Latency/BW effects

Influence design

Technical Details
Architecture Aware Algorithms

Helps multi-core:
Doubles BW to socket
Doubles cache size
Doubles peak flop rate

Develop robust multi-precision algorithms:

- Multi-precision Krylov and block Krylov solvers.
- Multi-precision preconditioners: multi-level, smoothers.
- Multi-resolution, multi-precision solver fast Poisson and Helmholtz solvers coupling direct and iterative methods

Develop multicore-aware algorithms:

- Hybrid distributed/shared preconditioners.
- Develop hybrid programming support: Solver APIs that support MPI-only in the application and MPI+multicore in the solver.
- Parallel in time algorithms such as Implicit Krylov Deferred Correction

Develop the supporting architecture aware runtime:

- Multi-level MPI communicators (Comm_Node, Comm_Net).
- Multi-core aware MPI memory allocation (MPI_Alloc_Shared).
- Strong affinity - process-to-core, memory-to-core placement.
- Efficient, dynamic hybrid programming support for hierarchical MPI plus shared memory in the same application.

IAA Algorithms Project Team
Mix of math, CS, and application experts

Climate (HOMME)
- Mike Heroux, Mark Taylor, Chris Baker (SNL)
- George Fann, Jun Jia, Kate Evans (ORNL)

Materials and Chemistry (MADNESS)
- George Fann, Judith Hill, Robert Harrison (ORNL)
- Mike Heroux, Curt Janssen (SNL)

Semiconductor device physics (Charon)
- George Fann, John Turner (ORNL)
- Mike Heroux, John Shadid, Paul Lin (SNL)

Project team includes key application developers

Runtime and Affinity
- Ron Brightwell, Kevin Pedretti, Brian Barrett (SNL)
- Al Geist, Geoffroy Vallee, Gregg Koenig (ORNL)

Excellent cross site teaming

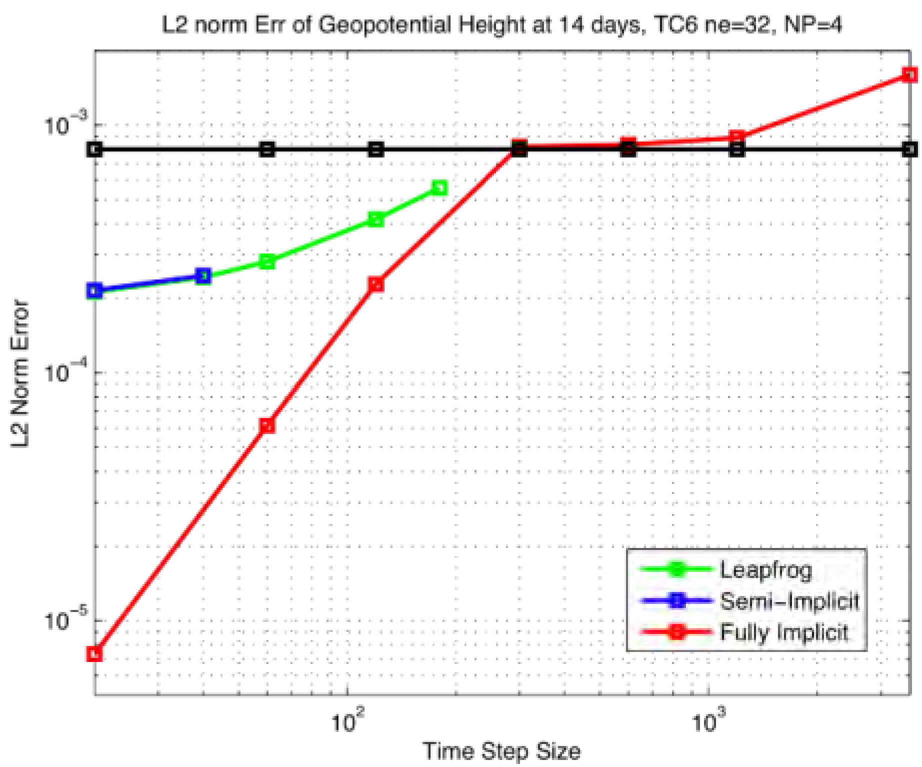
Simulation
- Arun Rodrigues, Scott Hemmert (SNL),
- Christian Engelmann, Kalyan Perumalla (ORNL)
- Bob Numrich (UM), Bruce Jacobs (U Maryland), Sudhakar (GaTech)

Highlight: Developed new algorithm for Climate application 20x faster solution

We have evaluated several numerical methods to scale HOMME to the next level.

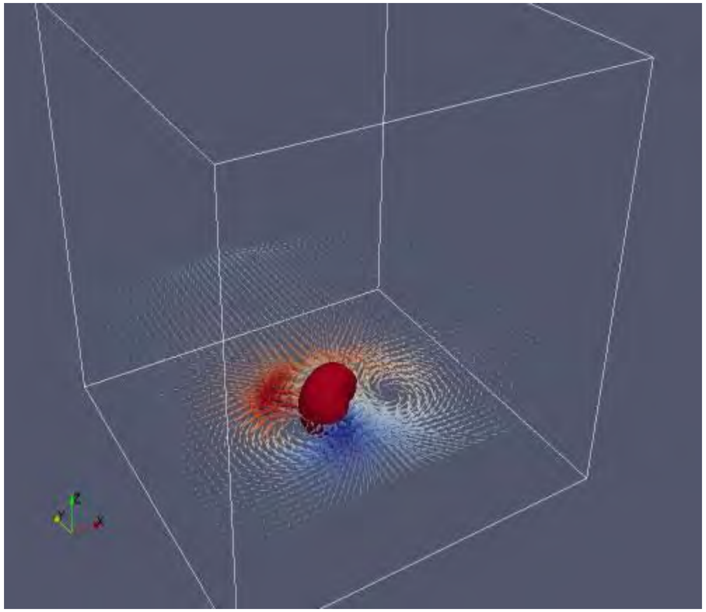
Jacobian-free Newton–Krylov method exhibits

- Time step converged to the discretized order of accuracy, 2nd order
- **Can run 20x above the explicit stability limit with accuracy equal to the level of uncertainly of subscale effects** (black line)



Highlight: Krylov Deferred Correction Parallel in Time, Time-stepping Algorithm

- Successful test Krylov Deferred Correction (KDC) on 3-D Navier-Stokes equation with periodic boundary condition in **MADNESS**



- Working on blackbox KDC for **HOMME**
  - Using Helmholtz for backward Euler with similar CFL condition as in climate code
- Interfacing KDC to use Trilinos solvers
- Scale testing of KDC on Cray XT Jaguar
  - 353,000 unknowns, 4 levels of refinement
- Improved strategy for scaling **MADNESS** to large processor counts 140K cores on Cray XT-5

Runtime Progress
Overcoming key MPI limitations on multi-core processors

Building on Open MPI – a highly portable, widely used MPI package

- Our extensions should work across a wide range of platforms
- The extensions are needed by the architecture aware algorithms
- Our focus is the Cray XT, which SNL and ORNL have large systems

Hierarchal MPI programming

- MPI_COMM_NODE
- MPI_COMM_SOCKET
- MPI_COMM_NETWORK
- MPI_COMM_CACHE

done

In development

Shared memory

- MPI_ALLOC_SHARED_MEM

Design phase

This feature will allow algorithm developers to avoid significant complexity associated with using MPI and threads

New Joint Math/CS Institute
Extreme-scale Algorithms & Software Institute

Architecture-aware Algorithms for Scalable Performance and Resilience on Heterogeneous Architectures

EASI Project Team

PI: **Al Geist (ORNL)**

Co-PIs:

- **Michael Heroux and Ron Brightwell (SNL)**
- **George Fann (ORNL)**
- **Bill Gropp (U ILL)**
- **Jack Dongarra (UTK)**
- **Jim Demmel (UC Berkeley)**

EASI Project Overview
Addressing Heterogeneity and Resilience

Research Areas in Institute

Community Outreach

Applications

HOMME
MADNESS
Charon

Algorithms

Krylov
Poisson
Helmholtz
Dense NLA
BLAS

Heterogeneous programming API
Robust multi-precision algorithms
Hybrid programming
Resilient algorithms
Communication optimal algorithms
Auto-tuned BLAS (API)
New parallelization methods

Runtime

MPI
Shared-memory
Processor affinity
Memory affinity

Task placement and scheduling
Memory management
Architecture-aware MPI

Deliver codes to community through:

ScaLAPACK
Trilinos
Open MPI
MPICH2
MADNESS
HOMME

Workshops
Training
Publications

Architecture **Heterogeneous, multi-core, extreme-scale**

EASI Highlight:
Developed heterogeneous programming API

- Completed a portable API for multicore CPUs and GPUs.
- Allows writing portable parallel linear algebra software that can use pthreads, OpenMP, CUDA, or Intel TBB (even more than one within the same executable)
- API is extensible to other programming models as needed.
- Using the API, we demonstrated compiling and running the same software kernel using pthread, Intel Threading Building Blocks and CUDA.
- The Trilinos Tpetra and Kokkos packages will incorporate this API in Trilinos 10.0.
- The API is documented in http://www.cs.sandia.gov/~maherou/docs/TrilinosNodeAPI.pdf

Thank You