U.S. DEPARTMENT OF ENERGY

# Co-Design and You: Why Should Mathematicians Care About Exascale Computing

Karen Pao

Advanced Scientific Computing Research

Karen.Pao@science.doe.gov

# ASCR Mission

*The mission of the Advanced Scientific Computing Research (ASCR) program is to advance applied mathematics and computer science; deliver, in partnership with disciplinary science, the most advanced computational scientific applications; advance computing and networking capabilities; and develop, in partnership with U.S. industry, future generations of computing hardware and tools for science.*

## *Science First!*

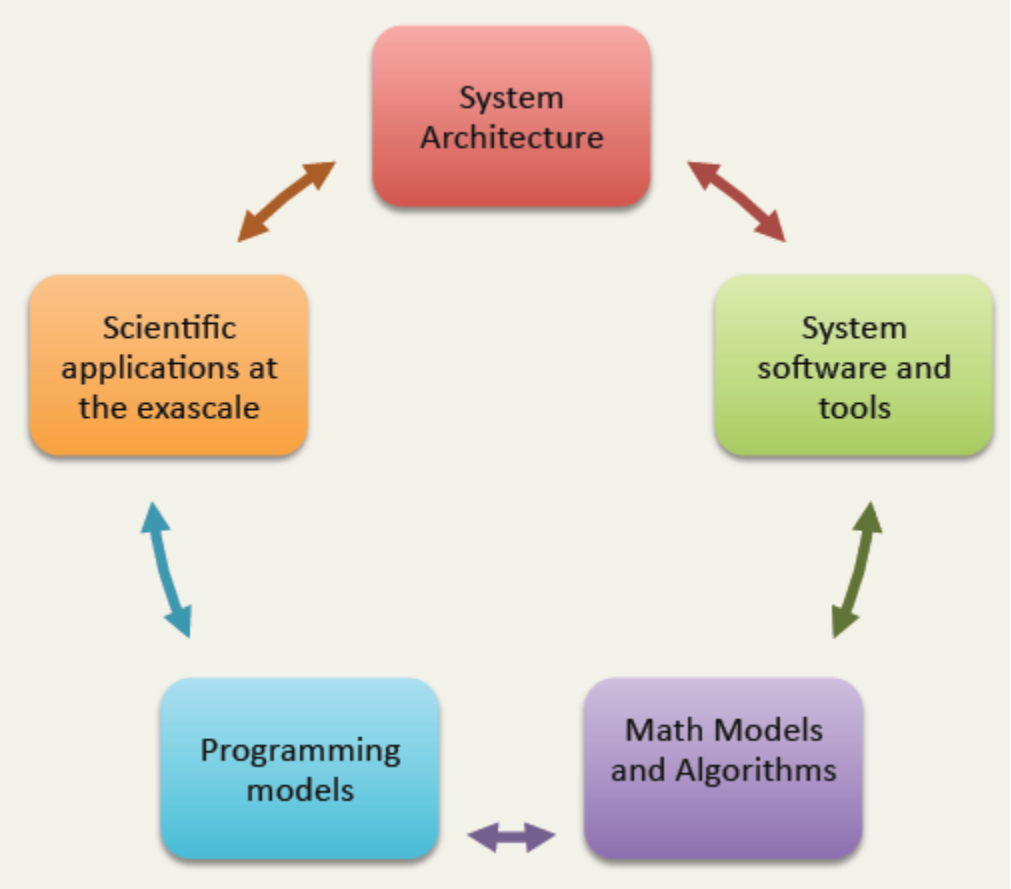
- Deliver science results on current and future Leadership computing facilities
- The need for greater energy efficiency will drive the architecture of all future computing systems, from desktops to exascale
- The entire spectrum of today's tools and techniques may need to be redesigned for future computational science requirements

# What is Co-Design?

From the FOA Lab10-07: "*Co-design refers to a computer system design process where scientific problem requirements influence architecture design and technology and constraints inform formulation and design of algorithms and software.*"

Co-Design will weigh *holistically* the key tradeoffs, such as

- Hardware and architecture
- Software stacks
- Numerical methods and algorithms
- Applications



# Three Exascale Co-Design Centers selected after intense competition

**Exascale Co-Design Center for Materials in Extreme Environments (ExMatEx)**

Director: Timothy Germann (LANL)

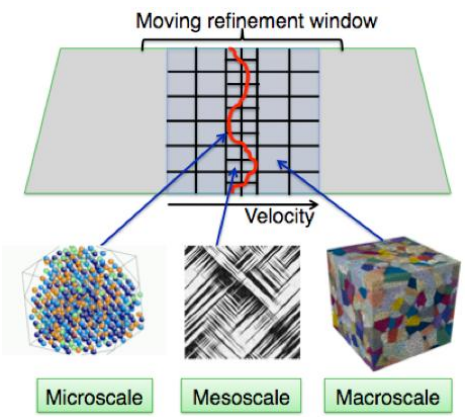
**Center for Exascale Simulation of Advanced Reactors (CESAR)**

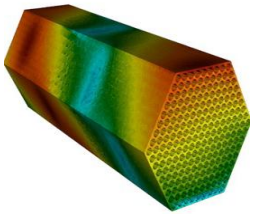
Director: Robert Rosner (ANL)

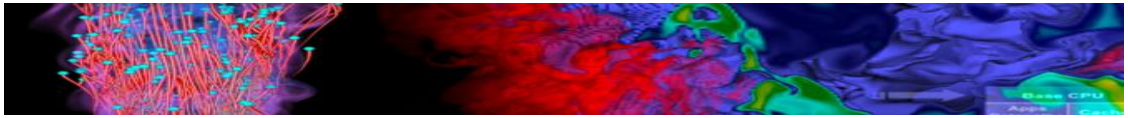
**Center for Exascale Simulation of Combustion in Turbulence (EXaCT)**

Director: Jacqueline Chen (SNL)

| | ExMatEx (Germann) | CESAR (Rosner) | EXaCT (Chen) |
|---|---|---|---|
| National Labs | LANL | ANL | SNL |
| | LLNL | PNNL | LBNL |
| | SNL | LANL | LANL |
| | ORNL | ORNL | ORNL |
| | | LLNL | LLNL |
| | | | NREL |
| University & Industry Partners | Stanford | MIT | Stanford |
| | CalTech | TAMU | GA Tech |
| | | Rice | Rutgers |
| | | U Chicago | UT Austin |
| | | IBM | Utah |
| | | TerraPower | |
| | | General Atomic | |
| | | Areva | |





*Each project is $4M/yr for 5 years, subject to satisfactory progress as gauged by frequent reviews*

# Expected Outcome of Co-Design Projects

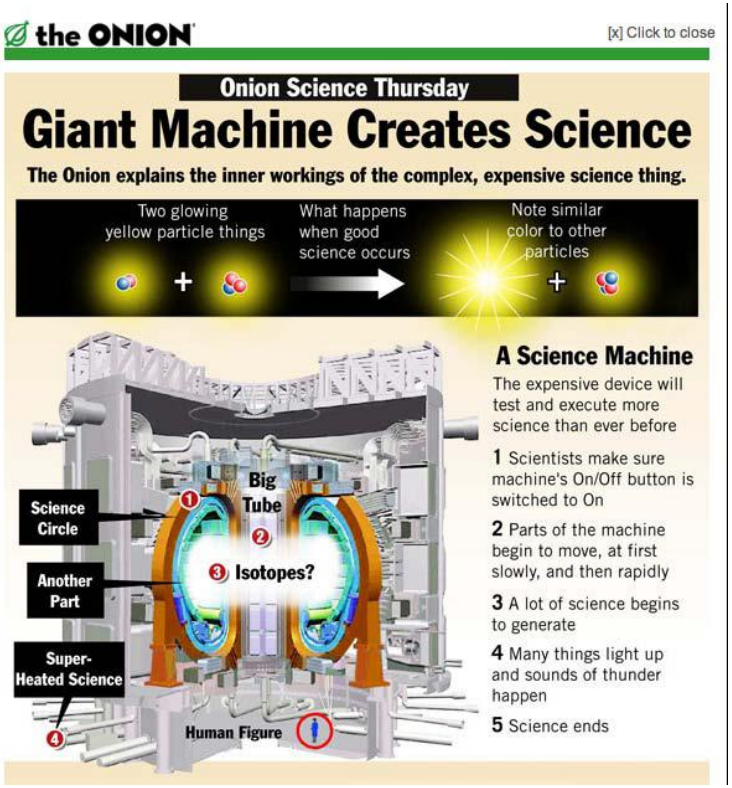
- Drive the exascale architectures
  - Collaborate with architecture researchers from multiple vendors without IP issues
  - Protocol for dealing with IP issues is being worked on
  - There is still time to influence processor & node design for the 2020 exascale machine
- Pioneer co-design methodology
  - HW/SW/alg/app collaboration on this scale is unprecedented in HPC history
  - Three *collaborative but independent* co-design efforts
- At the core: co-design is about doing great science
  - Project team members are the acknowledged and proven leaders of their respective fields of expertise
  - Production-quality code-base exists
  - Extreme-scale science & engineering *computational framework* to *enable* cutting-edge domain science research

# Funds for science will be finite, no matter how important it is to the Nation

http://www.theonion.com/articles/scientists-ask-congress-to-fund-50-billion-science,2291/

## Some cost considerations

- **Power**: at the usual scaling, power bill alone for the exascale machine could be $300M per year!
- **Performance**: memory is costly; we cannot afford the usual 1 byte/flops
- **Productivity**: will need to invest more R&D for all facets of application codes (integrated codes, physics models, algorithms, V&V and UQ, etc)




*Our challenge: world-class mission-relevant science at the exascale -- without bankrupting the Nation*

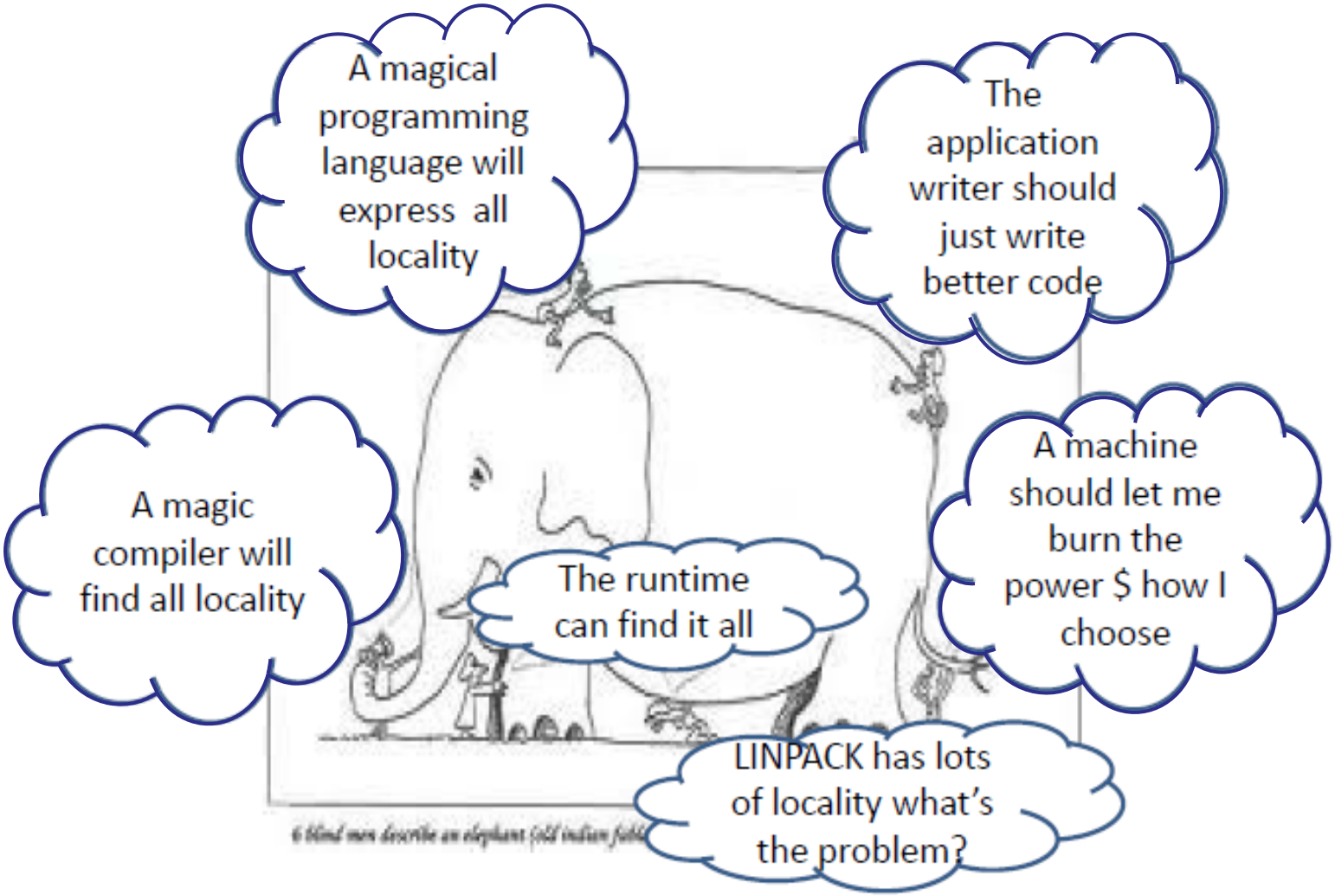
# Paradigm shift is coming

| Systems | 2009 | 2015 | 2020 |
|---|---|---|---|
| **System peak** | 2 Peta | 100 -- 300 Peta | **1 Exa** |
| **Power** | 6 MW | ~15 MW | **~20 MW** |
| **System memory** | 0.3 PB | 5 PB | **64 PB** |
| **System size (processors)** | 18,700 | 50,000 -- 500,000 | **O(100,000) -- O(1M)** |
| **Total concurrency** | 225,000 | billions | **10's -- 100's of billions** |
| **Storage** | 15 PB | 150 PB | **500-1000 PB** |
| **I/O bandwidth** | 0.2 TB/s | 10 TB/s | **60 TB/s** |
| **MTTI** | days | O(1day) | **O(1 day)** |

*Co-design of architecture, software, and algorithms is emerging as a necessity to negotiate the physical, power, and cost limitations of exascale systems.*

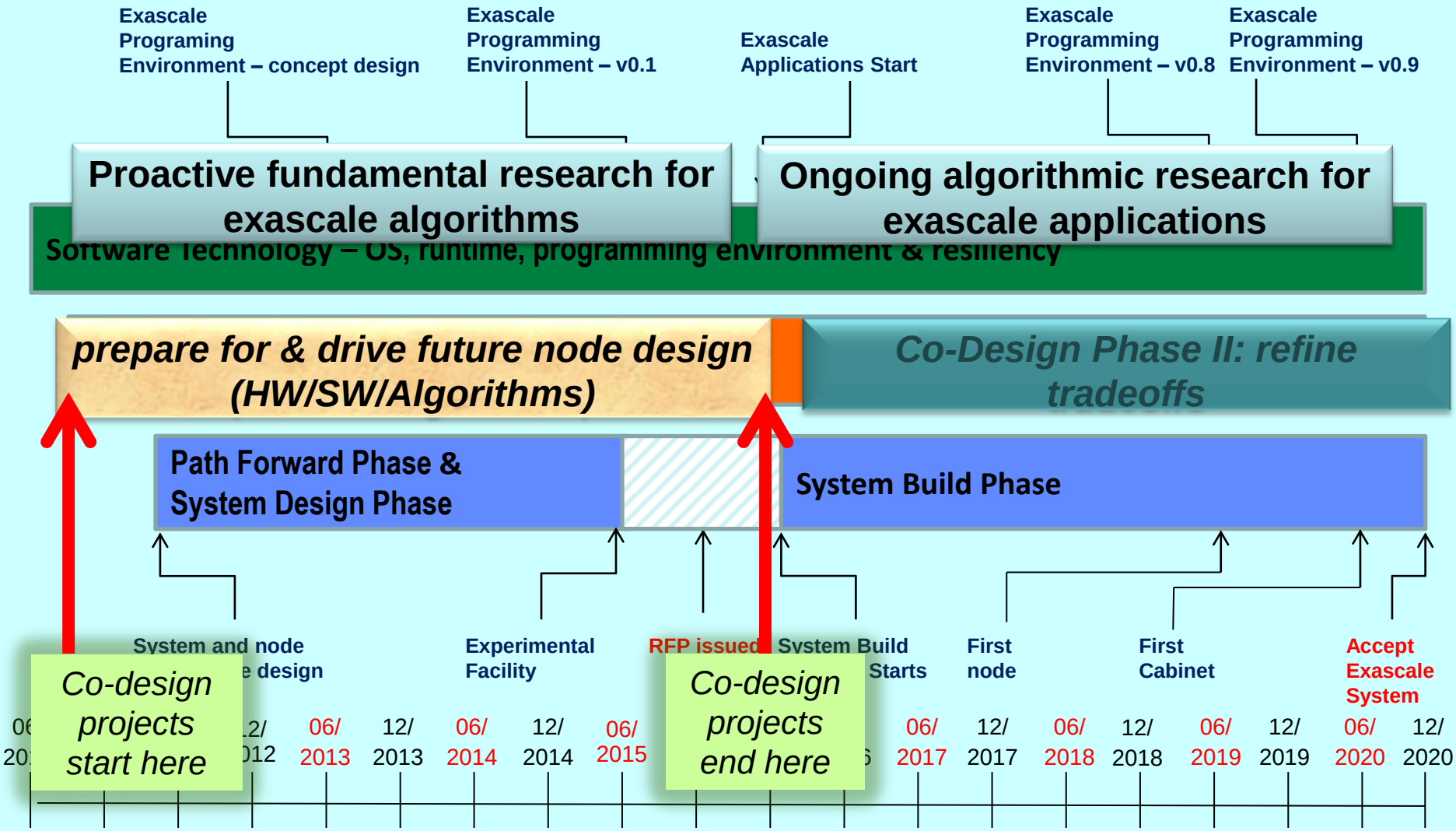
*– David Keyes, C. R. Mechanique 339 (2011)*

# One can wish for magic, but there will be no silver bullet




*(from Allen Snavely's talk at Arch I Workshop, Stanford, 8/2-3, 2011)*

# Co-design will contribute to exascale hardware and software technologies



# Co-Design Centers will explore pressing issues in the next 5 years

- Memory/Data Movement
  - The most important issue for energy efficiency
  - few tools are available today
  - Metrics are unclear
- Programming models
  - DSL, DSL-like, or evolutionary approaches
  - productivity is the main goal
- Application Architecture
  - At the end of the day, the teams will need to deliver a computational "framework" for domain science **and** drive hardware (node) design
- Scientific Data Management and I/O architecture
  - Moving, storing, and analyzing data without massive overhead to the actual computations

# Co-Design Centers need solutions in the next 5 years

- Abstract machine model
  - Co-Design centers will extract “proxy apps” from the full app as a vehicle for vendor collaboration, so the value of abstract machines is not immediately clear
- Operating system & system software
  - Co-Design Centers do not have the resources to do it alone and will need to leverage ongoing work
- Resilience and Fault Tolerance
  - Full of unknowns and uncertainties right now

# Co-Design Centers will encounter unprecedented algorithmic challenges

*Select findings from the Workshop "Scientific Grand Challenges: Crosscutting Technologies for Computing at the Exascale"*

- Recast critical applied mathematics algorithms
  - New **PDE discretizations** reflecting shift from FLOP- to memory-constrained hardware
  - Take advantage of data-movement constraints to redesign **UQ** and **data analysis** algorithms and techniques
  - Need to reduced global communication in linear and nonlinear **solvers**
- Formulate new algorithms to take advantage of emerging architecture
  - New approach to solving conservation laws?
  - Stochastic solutions to counteract fault tolerance?
- Study numerical analysis issues associated with moving away from bulk-synchronous programming
  - Stability and accuracy of asynchronous multiphysics updates
- Need tools to help us understand effects of algorithms on performance
  - Data locality, data locality, data locality!
  - Joule per op?

*History suggests performance gain will mostly come from algorithmic advances!*

# Low-hanging algorithmic fruits abound

*A snapshot from a Nek5000 run:*

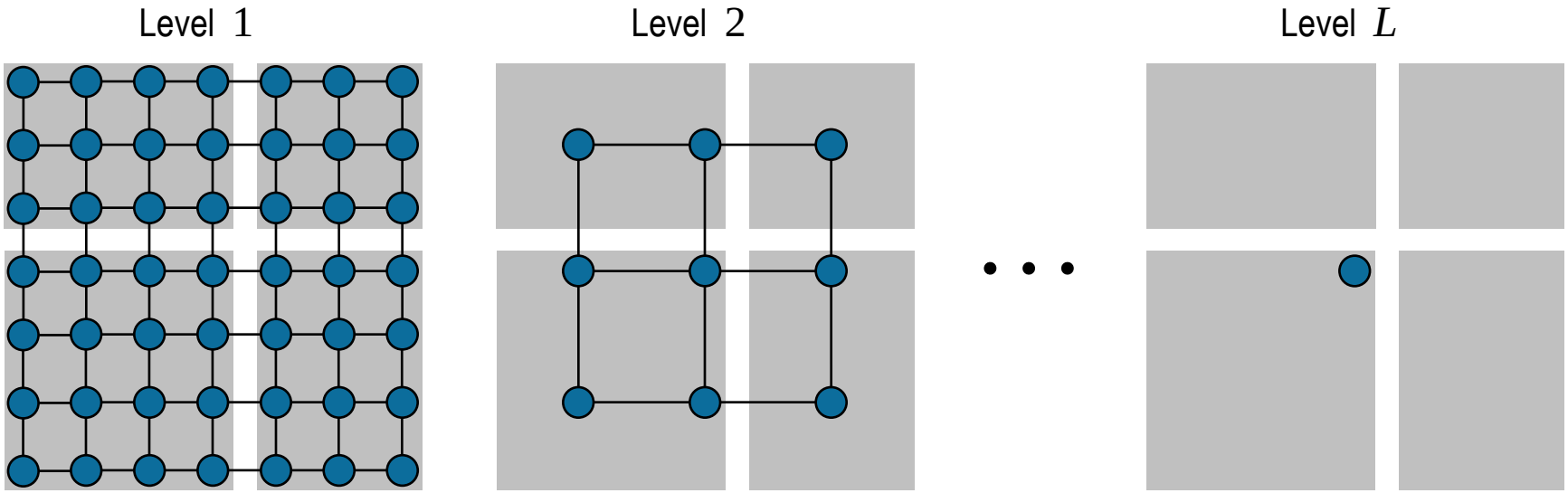
| | | | TIME (sec) | PERCENT | |
|---|---|---|---|---|---|
| inv3 | time | 3232 | 0.67736 | 0.2022895E-02 | |
| invc | time | 1000 | 0.25681 | 0.7669598E-03 | |
| mltd | time | 9996 | 25.21155 | 0.7529273E-01 | |
| cdtp | time | 9996 | 50.88502 | 0.1519649 | |
| pres | time | 100 | 194.64974 | 0.5813095 | <--- pressure solve |
| hmhz | time | 700 | 53.26729 | 0.1590798 | <--- velocity solve |
| usbc | time | 101 | 1.40704 | 0.4202047E-02 | |
| axhm | time | 7029 | 32.60166 | 0.9736284E-01 | <--- velocity mat-vec |
| gop | time | 30364 | 3.82274 | 0.1141632E-01 | <--- all_reduce time |
| vdss | time | 3241 | 9.05824 | 0.2705180E-01 | |
| dsum | time | 15740 | 13.89808 | 0.4150579E-01 | <--- nearest neighbor time |
| dadd | time | 0 | 8.67078 | 0.2589479E-01 | |
| ddsl | time | 3116 | 85.59198 | 0.2556159 | <--- preconditioner |
| crsl | time | 3116 | 18.16206 | 0.5423980E-01 | <--- coarse-grid solve time |
| solv | time | 3116 | 6.10630 | 0.1823611E-01 | |
| prep | time | 101 | 14.64240 | 0.4372867E-01 | |

elapsed time: 652.39 359.85 seconds

*Improvements in linear algebra will be pivotal to improvement of overall performance!*

*Slide material courtesy of Paul Fischer (ANL) - CESAR*

# Approach for parallelizing multigrid is straightforward data decomposition




- Basic communication pattern is “nearest neighbor”
- Different neighbor processors on coarse grids
- Many idle processors on coarse grids (100K+ on BG/L)
  - Algorithms to take advantage have had limited success

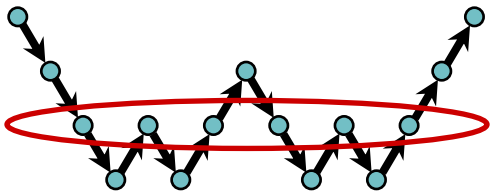
*The idle processor problem seems severe, but standard parallel V-cycle multigrid performance has optimal order*

*Slide material courtesy of Rob Falgout (LLNL) - EXaCT*

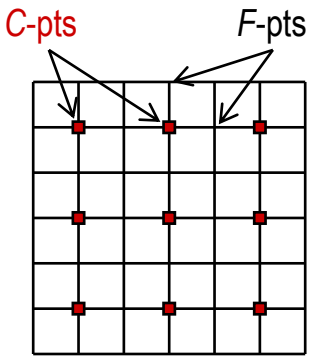
# Where the difficulties lie for parallel multigrid

- W-cycles scale poorly:

$$T_W \approx (2^{\log N})4\alpha + (\log N)4n\beta + (2)5n^2\gamma$$



- Lexicographical Gauss-Seidel is too sequential
  - Use red/black or multi-color GS
  - Use weighted Jacobi, hybrid Jacobi/GS, L1
  - Use *C*-*F* relaxation (Jacobi on *C*-pts then *F*-pts)
  - Use Polynomial smoothers




- Parallel smoothers are often less effective

- Even if computations were free, doing extra local work can actually degrade convergence (e.g., block smoothers)

*Slide material courtesy of Rob Falgout (LLNL) - EXaCT*

# EASI: Proving communication lower bounds

## *Objectives*

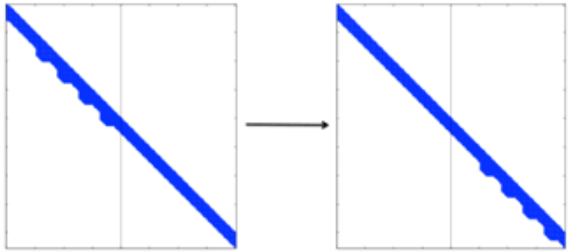
- **Prove communication lower bounds for *matrix multiplication, LU, QR, SVD*, and Krylov subspace methods for Ax=b, Ax=λx**
- **Develop algorithms that attain the minimum data movement, in some cases by using more memory**

## *Impact*

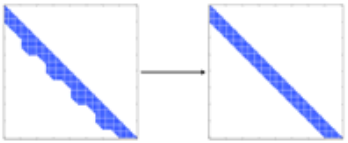
- Cost of communication (moving data between layers of memory or between processors) greatly exceeds cost of arithmetic by 100X
- Order of magnitude speedups have been demonstrated

Performance of new communication minimizing Successive band reduction algorithm is up to 30 times faster than existing implementations

**Parallel SBR Algorithm**



**Attains lower bound for #words moved. Reduces #messages from O(n) to O( $p^{1/2}$ log(n/$p^{1/2}$))**



## *2011 Progress*

- Best Paper Prize at SPAA'11 “Graph expansion and communication costs of fast matrix multiplication” for communication lower bounds for Strassen-like algorithms
- Distinguished Paper Award at Euro-Par'11 for “Communication-optimal parallel 2.5D matrix multiplication and LU factorization algorithms “
- New communication-avoiding successive band reduction (SBR) up to 30X speedup over ACML and 17X over MKL.

**Contact: Jim Demmel, UC Berkeley**

U.S. DEPARTMENT OF ENERGY

ASCR- Joint Math/CS Institute Highlight

# EASI - Resilience to Soft Error via Selective Reliability

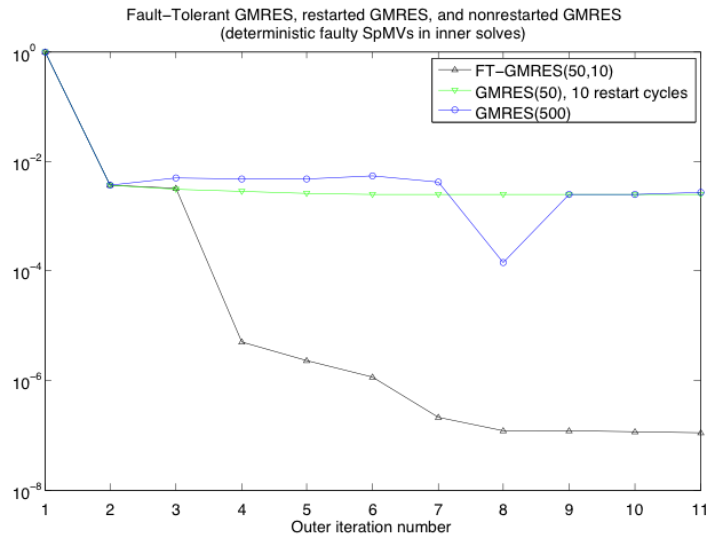
## *Objectives*

**Develop fault-resilient algorithms:**

- **FT-GMRES: Soft-error resilient version of GMRES.**
- **Framework for broad set of resilient methods.**

## *Impact*

- New approach to soft errors demonstrated – with ramifications to extreme-scale computing resilience challenges
- Add a "thin layer" of extra reliability, with most of the calculations still done in standard libraries.
- Instead of causing failure, soft errors can be made to only cause delay in convergence




Numerical error in FT-GMRES continues to be reduced after fault, while the error flattens out – problem fails to converge -- in conventional GMRES schemes

## *2011 Accomplishments*

- Established a new framework and methodology for the broad development of a new class of resilient algorithms, necessary for reliable computation at extreme scales.
- Formulated a version of preconditioned GMRES that does not fail in the presence of soft errors.
- Demonstrated the value of co-design: Requires collaboration of algorithm, library, programming models and runtime developers.
- Papers submitted to SIAM SISC and Europar 2011.
- Demonstration version of FTGMRES available in Trilinos/Belos iterative solver package.

**Contact: Michael Heroux , Sandia National Labs**

U.S. DEPARTMENT OF ENERGY

ASCR- Joint Math/CS Institute Highlight

# CACHE: Generating High-Performance Linear Algebra Code from DSLs

## Objectives

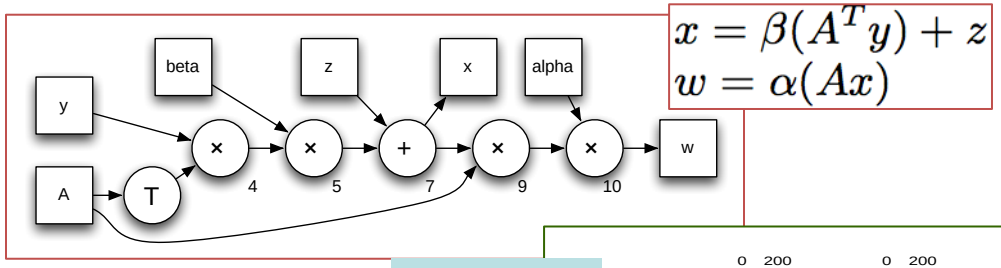
**Generate high-performance implementations of linear algebra kernels defined using a domain specific language (DSL) to**

- **Optimize cache performance**
- **Exploit multicore architectures**

## Impact

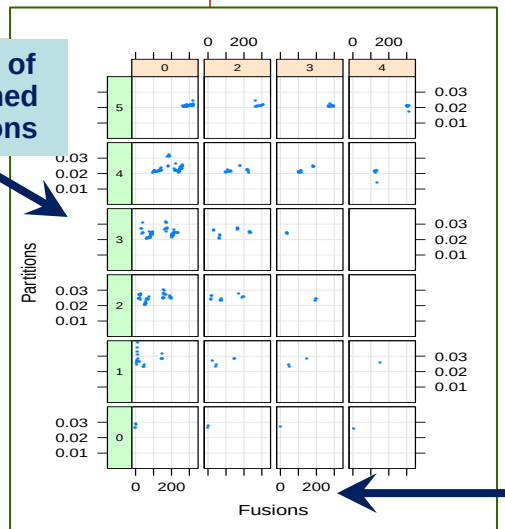
- Define algorithms using the Build to Order BLAS (BTO) compiler's readable, high-level DSL for linear algebra
- Generate optimized linear algebra computations with up to 180% improvement over vendor-tuned libraries

***Data flow representation of computation expressed in the BTO linear algebra language***




$$x = \beta(A^T y) + z$$
$$w = \alpha(Ax)$$

**Number of partitioned operations**




**Number of fused loops**

*The space of all fusion and partitioning decisions. The inner x-axis is our internal version IDs and the inner y-axis is the runtime in seconds (lower is better).*

*Images: G. Belter, T. Nelson*

## Accomplishments FY2011

- New representation of the search space for loop fusion and parallel partitioning in BTO
- Comparison of four different empirical search strategies
- Simplified code generation for additional autotuning with Orio
- "Exploring the Optimization Space for Build to Order Matrix Algebra?", G. Belter, E. Jessup, I. Karlin, T. Nelson, B. Norris, J. Siek, ANL Tech. Report ANL/MCS-P1890-0511, 2011

**Contact: B. Norris and T. Nelson, Argonne National Lab**

U.S. DEPARTMENT OF ENERGY

ASCR- Joint Math/CS Institute Highlight

# CACHE: Advancing Search in Autotuning

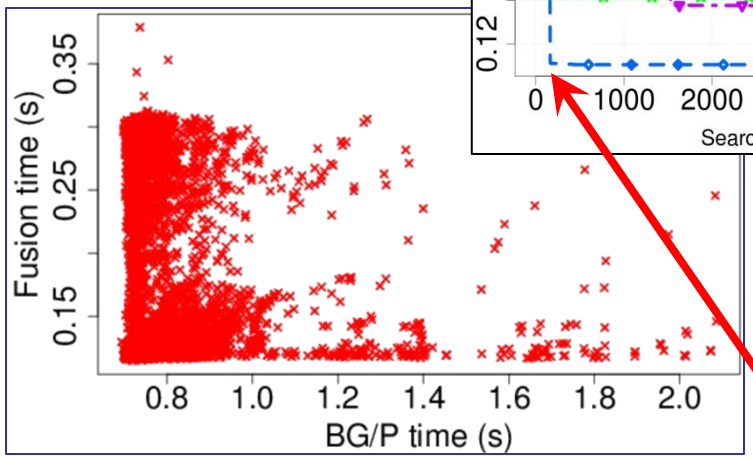
## *Objectives*

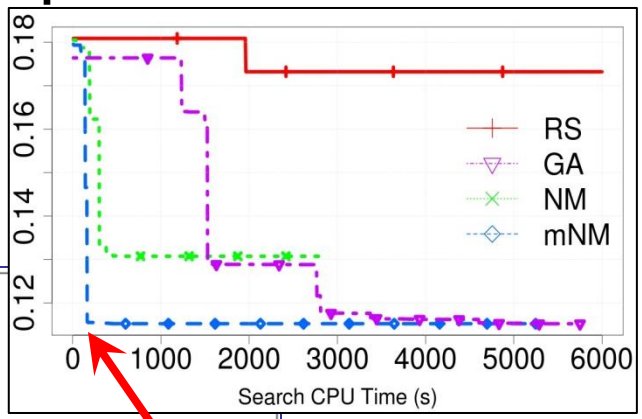
**Use modern mathematical optimization algorithms to find high-performance tuning parameterizations rapidly, by examining only a small part of the parameter search space**

## *Impact*

- Autotuning essential as scientists move from one computer architecture to another
- Complete enumeration impossible and sampling inefficient for large-scale problems
- Portability -- tune any code on any machine

***Run times for the same code configurations can vary greatly from machine to machine***




***Search times for 4 different popular mathematical optimization algorithms***

**1.5X speedup evaluating 40 of $10^7$ possible variants**

*Images: P. Balaprakash*

## *2011 Accomplishments*

- Released SPAPT (Search Problems in Automatic Performance Tuning) test-set to advance numerical optimization algorithms for empirical-based search
- Introduced new search problem specifications in the autotuning tool ORIO
- "Can search algorithms save large-scale automatic performance tuning?" P. Balaprakash, S. Wild, and P. Hovland. Proc.Comp.Sci. 4, pp. 2136-2145, 2011

**Contact: P. Hovland & S. Wild, Argonne National Lab**

# Co-design presents a golden opportunity for basic research

- The co-design projects will need to leverage ongoing research in applied mathematics and computer science, such as
  - UQ
  - Solvers and other numerical algorithms
  - Software stack
  - Hardware simulation
  - Programming model/compilers/languages
  - I/O provisioning
  - Performance engineering
  - Resilience
  - Data analytics
- We envision mutually beneficial collaborations between co-design teams and base program

U.S. DEPARTMENT OF ENERGY

# ASCR at-a-glance

## Office of Advanced Scientific Computing Research

Associate Director – Daniel Hitchcock (acting)
Phone: 301-903-7486
E-mail: Daniel.Hitchcock@science.doe.gov

### Research

Division Director – William Harrod
Phone: 301-903-5800
E-mail: William.Harrod@science.doe.gov

### Facilities

Division Director – Daniel Hitchcock
Phone: 301-903-9958
E-mail: Daniel.Hitchcock@science.doe.gov

## Relevant Websites

ASCR: science.energy.gov/ascr/
ASCR Workshops and Conferences:
science.energy.gov/ascr/news-and-resources/workshops-and-conferences/
SciDAC: www.scidac.gov
INCITE: science.energy.gov/ascr/facilities/incite/
Exascale Software: www.exascale.org
DOE Grants and Contracts info: science.doe.gov/grants/