Joint Math/CS Institutes: Application Requirements

James B White III (Trey)
Scientific Computing Group
National Center for Computational Sciences
trey@ornl.gov

Chicago
October 7, 2008

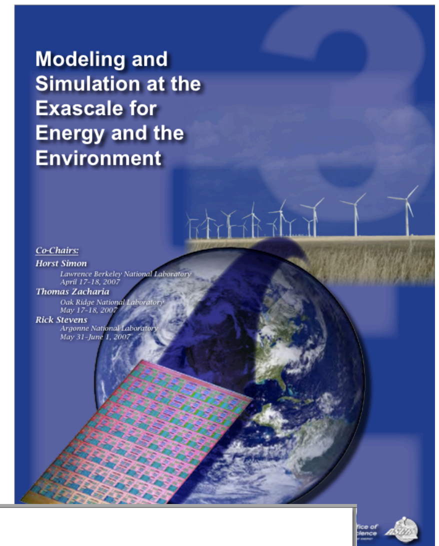
Challenges Facing Scalable Applications:

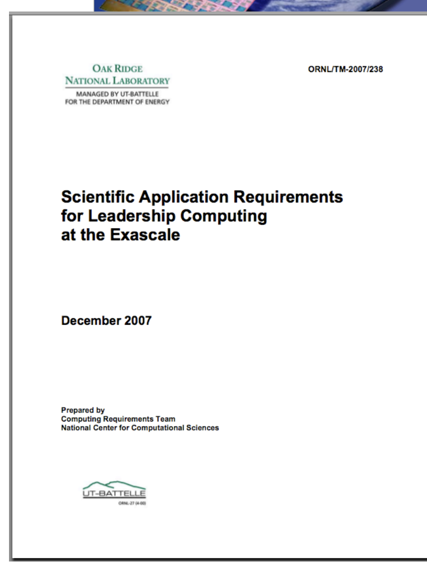
Where are the ‘gaps’ between potential and achieved performance?

- List some explicit examples that drive the discussion.
- What are the primary bottlenecks facing computational scientists?
- To what areas can a tightly integrated Math/CS effort contribute?

Challenges facing scalable applications

- ***Modeling and Simulation at the Exascale for Energy and Environment***
  - www.sc.doe.gov/ascr
  - Program Documents
- ***Scientific Application Requirements for Leadership Computing at the Exascale***
  - nccs.gov
  - Media Center → NCCS Reports




Oak Ridge National Laboratory
Managed by UT-Battelle for the Department of Energy

ORNL/TM-2007/238

Scientific Application Requirements for Leadership Computing at the Exascale

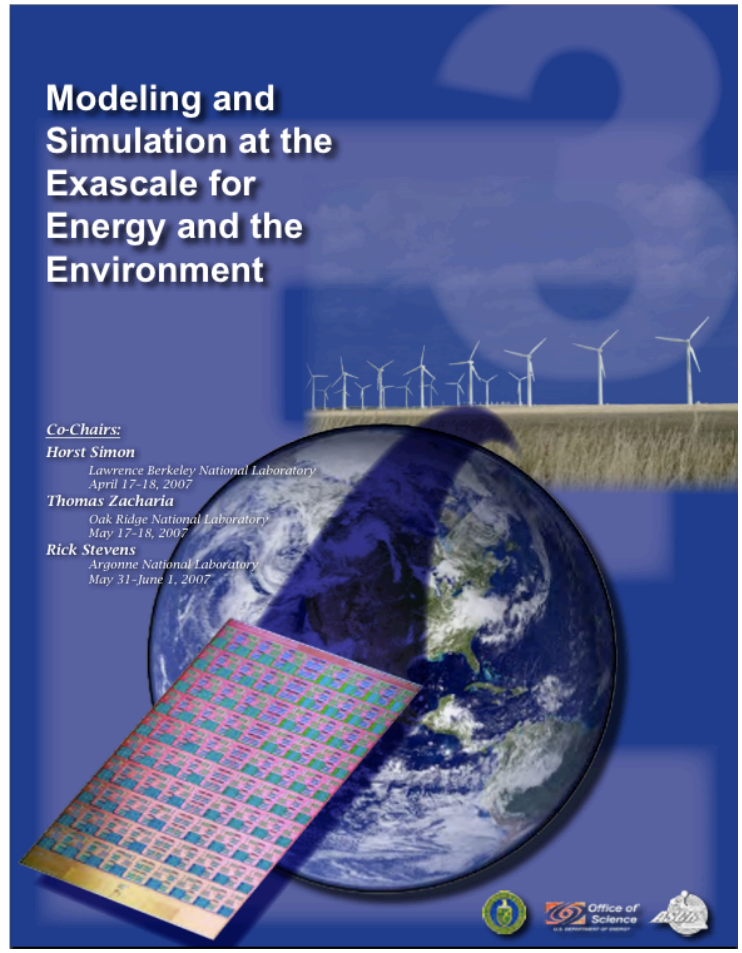
December 2007

Prepared by
Computing Requirements Team
National Center for Computational Sciences

UT-Battelle

Modeling and Simulation at the Exascale for Energy and Environment

- **Town Halls**
  - Berkeley, April 17-18, 2007
  - Oak Ridge, May 17-18, 2007
  - Argonne, May 31-June 1, 2007
- **Document**
  - Climate
  - Combustion
  - Fusion
  - Solar
  - Fission
  - Biology
  - Socioeconomic modeling
  - Astrophysics
  - Math and algorithms
  - Software
  - Hardware
  - Cyberinfrastructure



Climate

- **Quantification and reduction of uncertainty**
- **Attribution of errors**
- **Derivation of parameterizations**
- **Replacement of parameterizations with direct simulation**
- **Rapid software development**
- **Software verification**
  - Unit testing
  - Diagnostics
- **Assimilation of big data volumes**
- **Analysis tools for big data volumes**
- **Parallelism and scalability**
- **Unstructured and adaptive grids in the ocean**
- ***"closer collaboration with applied mathematicians"***

Combustion

- **Uncertainty quantification for validation**
- **Scalable nonlinear solvers for implicit systems**
- **Load balancing for heterogenous physics**
- **Discretization methods for basic physics and coupling**
- **Unstructured, adaptive, sometimes-moving meshes**
- **Software modularity**
- **Portable, scalable performance**
- **Managing huge data volumes**
- **Extracting knowledge from simulation data**
- **Coupling simulation data to design optimization**
- **Collaborative tools for sharing methods and data**
- ***"key combustion design issues will require a collaboration of computer scientists, applied mathematicians, and combustion scientists"***

Fusion

- Mathematical algorithms for coupling models across wide ranges of space and time scales
- Reduced models based on DNS
- Diagnostics and data management for development and debugging of coupled models
- Data analysis and visualization of large data volumes
- Algorithms with hierarchical parallelism
- Multi-core compilers
- Runtime systems for efficient fine-grain parallelism
- Hybrid MPI and UPC

Solar energy (materials science and nano-science)

- Linear-scaling algorithms for chemical structure
- Integrated simulations of complex systems
- Many-particle systems for long simulations
- Modular, composable software in a common framework

Fission

- **Requirements driven by industry and regulatory agencies**
- **Sensitivity and uncertainty analysis**
- **Verification and validation**
- **Fully coupled, nonlinearly consistent multi-physics time integration**
- **Generalized interpolation, integration, extrapolation**
- **Compatible geometry representations**
- **Milliseconds to years**
- **Molecules to full fuel cycle**
- **Workstations to extreme scale**
- **Interactive selection of models, levels of detail, and breadth of scope**
- ***"While there have been many advances in fundamental enabling technologies in mathematics and computer science in the past, additional research and development will undoubtedly be required to tackle a problem of this scale."***

Biology

- Long simulated times
- Multi-scale modeling from molecules to ecosystems
- Reusable software components
- Large, rich databases
- Integer computations
- Shared-memory programming models
- *"design of the tools and their application requires both biologists and mathematicians"*

Socioeconomic Modeling

- Uncertainty quantification in complex coupled models
- Validation of individual models and large model systems
- Large-scale nonlinear optimization
- Parameter-sweep methods for high dimensions
- Quality control of diverse data sources
- Data search and summary
- Fast comparison of measured and computed data
- Scalable statistical methods
- Physical constraints on "induced technical change"

Astrophysics

- **Long simulation times**
- **Higher resolution**
- **Memory-efficient algorithms**
- **Implicit methods**
- **Adaptive mesh refinement**
- **Dynamic load balancing**
- **Fault-tolerant algorithms**
- **Management and analysis of huge data volumes**
- **Scalable solvers for elliptic systems**
- ***"there are no obvious bottlenecks at present that suggest that an entirely new set of codes will have to be deployed"***
- ***"of primary importance to the computational astrophysics community will be collaboration with the applied mathematics community with an eye toward porting existing methods and codes to exascale platforms"***

Math and Algorithms (1 of 2)

- **Fully intrusive uncertainty quantification**
- **Adjoint methods for uncertainty quantification and control**
- **Formalized confidence measures for decision making**
- **Implicit methods + uncertainty quantification + integration-error control and estimation**
- **JFNK methods + automated adjoint models**
- **Preconditioners**
  - Physics based
  - Approximate block factorization
  - Multi-level, including algebraic multi-grid
- **Eigensolvers**
- **Mesh generation and adaptation**
- **Optimal rebalancing for AMR**
- **High-order finite-element methods**
- **System-of-systems approaches**
- **Hybrid deterministic/probabilistic approaches**

Math and Algorithms (2 of 2)

- Block iterative methods for multiple right-hand sides
- Out-of-core direct solvers
- Solvers for sparse indefinite systems
- Inexact subsystem solvers for nonlinear optimization
- Branch-and-cut solvers for discrete nonlinear optimization
- Sampling methods for high-dimensional spaces
- Combinatorial and discrete algorithms
- High-precision arithmetic
- Scalable data analysis
- Scalable, adaptive agent-based modeling
- Robust abstraction layers for portable performance
- Software engineering

Software

- **High-level domain-specific languages**
- **Modular, composable software components**
- **Model coupling through automated workflows**
- **Formal verification methods**
- **Fault-detection algorithms**
- **Automatic management of storage hierarchies**
- **Knowledge discovery**
  - Integrated into simulations
  - Integrated into data managment
  - Of software performance

Summary

- Big solvers
- Big data
- Big software
- Long times
- Uncertainty, sensitivity, verification, and validation
- Multi-scale in space and time
- Hierarchical parallelism
- Design optimization

Scientific Application Requirements for Leadership Computing at the Exascale

- **Survey of leading NCCS users: "Anticipating Requirements for Leadership Computing in the Next Decade"**
  - Science drivers and urgent problems
  - Looming computational challenges
  - Sample science objectives and outcomes
  - Improvement goals for science-simulation fidelity
  - Possible changes in physical model attributes
  - Major software-development projects
  - Major algorithm changes
  - Libraries and development tools
  - Priorities for system attributes
  - Workflow changes
  - Disruptive technologies
- **Summary document**
  - 6. Accelerated Development and Readiness


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

ORNL/TM-2007/238

Scientific Application Requirements for Leadership Computing at the Exascale

December 2007

Prepared by
Computing Requirements Team
National Center for Computational Sciences

UT-BATTELLE

Accelerated Development and Readiness

- Automated diagnostics
- Hardware latency
- Hierarchical algorithms
- Parallel programming models
- Solver technology and innovative solution techniques
- Accelerated time integration
- Model coupling
- Maintaining current libraries

Automated Diagnostics

- **Increasing complexity of software and systems**
- **Aggressive automation of diagnostic...**
  - Instrumentation
  - Collection
  - Analysis
- **Drivers**
  - Application verification
  - Performance analysis
  - Software debugging
  - Hardware-fault detection and correction
  - Failure prediction and avoidance
  - System tuning
  - Requirements analysis

Hardware Latency

- **Ever increasing computation rate, parallelism, and bandwidth**
- **Stagnant hardware latencies**
- **Targeted hardware improvements**
  - Fast synchronization on chip, in memory, over networks
  - Smart-network acceleration or offload
- **Latency hiding and avoidance in algorithms and software**

Hierarchical Algorithms

- **Latency stagnation → increasing depth of memory hierarchy**
- **Heterogeneous computing → process hierarchy**
- **Increasing fault rates → redundancy moves up hierarchy**
- **Current strategies**
  - Cache blocking
  - Hybrid data parallelism
  - File-based checkpointing
- **Emerging needs**
  - Dynamic decisions between recomputing and storing
  - Fine-scale task/data hybrid parallelism
  - In-memory checkpointing

Parallel Programming Models

- **Latency stagnation → minimize synchronization**
- **Memory hierarchies deepening**
- **Current programming models target one level at a time**
  - Source language for instruction-level parallelism
  - OpenMP for multi-processor intra-node parallelism
  - MPI for inter-node parallelism
- **New levels of memory hierarchy mapped to an existing level of parallelism or ignored**
- **Levels of parallelism must be mapped to specific, distinct levels of the programming model—*difficult to modify***
- **Models continue being coupled into large models**
- **Improved model**
  - Arbitrary levels of parallelism, mapped onto hardware at runtime, perhaps dynamically
  - Minimize synchronization, maximize asynchrony

Solver Technology and Innovative Solution Techniques

- **Higher resolution, longer time scales, more coupling**
- **Solver requirements**
  - Multi-level methods
  - Preconditioners
  - Adaptive mesh refinement
  - Irregular meshes
  - Newton-Krylov methods
  - Complex-mesh generation
- **Trade off flops for memory operations and communication**
- **New applications of stochastic methods?**

Accelerated Time Integration

- **Many apps need much-longer simulated times**
- **Single-process performance has stalled**
- **Increasing resolution shortens time step for explicit methods**
- **Implicit methods**
  - Longer time steps
- **Compact shape-preserving bases**
  - Greater accuracy for long steps
- **High-order time integration**
  - Greater accuracy for long steps
- **Pipelined-in-time and parallel-in-time algorithms**
  - Faster computation

Model Coupling

- Implementation, verification, validation
- Wide range of space and time scales
- Downscaling, upscaling, coupled nonlinear solving
- Data assimilation with huge data volumes
- Uncertainty and sensitivity analyses

Maintaining Current Libraries

- **Current HPC applications rely on libraries**
  - MPI
  - BLAS
  - LAPACK
  - FFTW
  - ScaLAPACK
  - PETSc
  - Trilinos
- **Must run well on new architectures**

Summary of summaries

- **Town Halls**
  - Big solvers
  - Big data
  - Big software
  - Long simulations
  - Uncertainty, sensitivity, validation, and verification
  - Multi-scale in space and time
  - Hierarchical parallelism
  - Design optimization
- **Survey Analysis**
  - Automated diagnostics
  - Hardware latency
  - Hierarchical algorithms
  - Parallel programming models
  - Solver technology and innovative solution techniques
  - Accelerated time integration
  - Model coupling
  - Maintaining current libraries

Joint Math/CS Institutes: Application Requirements

James B White III (Trey)
Scientific Computing Group
National Center for Computational Sciences
trey@ornl.gov

Chicago
October 7, 2008