

SANDIA REPORT

SAND2004-3244
Unlimited Release
Printed July 2004

Validating DOE's Office of Science "Capability" Computing Needs

Edwin H. Barsis, Peter L. Mattern, William J. Camp, Robert W. Leland

Prepared by Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.osti.gov/bridge>

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



SAND2004-____
Unlimited Release
Printed July 2004

Validating DOE's Office of Science "Capability" Computing Needs

William J. Camp
Computation, Computers, Information and Mathematics

Robert W. Leland
Computer & Software Systems

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0321

Edwin H. Barsis
Peter L. Mattern
BMV Associates, LLC
Albuquerque, NM 87123

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Background

- Ray Orbach commissioned an external review of the Office of Science Capability computing needs.
- The effort was started mid-April, 2004
- Table of Contents
 - I. Acknowledgments
 - II. Abstract
 - III. Assessment Process
 - IV. Related Issues
 - V. Summary by Discipline
 - VI. Written analysis, detailed descriptions, and additional documentation.

I. Acknowledgments

The authors offer special thanks to the following persons ...

In alphabetical order:

David Keyes, Columbia University, for granting access to the unpublished SCaLeS Vol. 2, and for introductions to the report's discipline coordinators.

Jeffrey Nichols, ORNL, for hosting on-site discussions with key staff.

Ed Oliver, DOE/HQ, for offering a framework for the requirements of the study.

Horst Simon, NERSC, for making available a wealth of information about NERSC.

Rick Stevens, ANL, for sharing perspectives on key computing issues.

Ray Stults, LANL, for arranging discussions at LANL.

Andy White, LANL, for organizing and participating in valuable technical discussions.

David Womble, SNL, for perspectives on DOE computing needs.

Thomas Zacharia, ORNL, for making available the significant expertise at ORNL.

SCaLeS coordinators provided a large volume of valuable information — excerpts in Section VI.

In alphabetical order:

John Bell, LBNL, Combustion
Peter Cummings, ORNL, Nanoscience
John Drake, ORNL, Climate
Robert Harrison, ORNL, Chemistry
Steve Jardin, PPPL, Plasma Science
Phillip Jones, LANL, Climate
Kwok Ko, SLAC, Accelerators
Anthony Mezzacappa, ORNL, Astrophysics
William Nevins, LLNL, Plasma Science
Larry Rahn, SNL, Combustion
Robert Ryne, LBNL, Accelerators
Malcolm Stocks, ORNL, Materials
Robert Sugar, QCD, UCSB;
Lin-wang Wang, LBNL, Nanoscience
Mary Wheeler, UT-Austin, Environment
Theresa Windus, PNNL, Chemistry
Steve Yabusaki, PNNL, Environment

And others provided important perspectives and examples ...

Some of their material is included in this report for reference – it was not possible to include all examples. In alphabetical order:

Pratul Agarwal

John Aidun

Richard Alexander

Don Batchelor

Tanmoy Bhattacharya

Buddy Bland

Mark Boslough

Matt Challacombe

Jackie Chen

Bill Daughton

Ed D'Azevedo

Mark Fahey

Al Geist

Andrey Gorin

Rajan Gupta

Grant Heffelfinger

Bill Kramer

Daniel Livescu

Phil Locascio

Robert Lowrie

Thomas Maier

Vincent Meunier

Normand Modine

Greg Newman

Phani Nukala

Philip Pebay

Steve Plimpton

Ken Roche

Manjit Sahota

George Samara

Nagiza Samatova

Kevin Sanbonmatsu

William Shelton

Thomas Schulthess

Robert Silvia

Srdan Simunovic

Srini Srivilliputhur

Roger Stoller

Bobby Sumpter

Mark Taylor

Aidun Thompson

Ed Uberbacher

Jeffrey Vetter

Mike Warren

Trey White

Pat Worley

II. Abstract

A study was undertaken to validate the “capability” computing needs of DOE’s Office of Science. More than seventy members of the community provided information about algorithmic scaling laws, so that the impact of having access to Petascale capability computers could be assessed.

We have concluded that the Office of Science community has described credible needs for Petascale capability computing.

III. Assessment Process

We solicited responses from the SCaLeS coordinators & Lab PIs

Questions we asked:

- What are the programmatic impacts of having a capability machine in the range of 100TF/s to 1PF/s peak?
- What are the scientific challenges that will be met?
- Using algorithmic scaling laws or operation counts, show how the application would use 100TF/s- 1PF/s.

Responses: [Assembled in Section VI. of this report]

- All but one of the eleven SCaLeS disciplines provided details
- ORNL and LANL hosted on-site visits
- Sandia staff were generous with their time

Example of a Scaling-Law Response from “SCaLeS-Accelerators”

Quasi-static and fully explicit Particle-In-Cell codes

- In the particle-dominated regime, these codes scale linearly with the number of macroparticles, N_p . In the simulation of a beam in a circular machine, the scaling is $N_p * N\{\text{bunches}\} * N\{\text{turns}\} * N\{\text{kicks-per-turn}\}$, where N_p is the number of macroparticles per bunch, $N\{\text{bunches}\}$ is the number of bunches, and $N\{\text{turns}\}$ is the number of turns simulated, and $N\{\text{kicks-per-turn}\}$ is the number of kicks per turn.
- For enhanced programmatic & scientific impact (see Section VI.) the following parameter values must be increased: N_p by 8x, $N\{\text{turns}\}$ by 8x, $N\{\text{bunches}\}$ by 4x, $N\{\text{kicks-per-turn}\}$ by 4x. The resultant total increase in computing power required is $8 \times 8 \times 4 \times 4 = \sim 1000$.
- Current computer use is 12 - 24 hours on a 3-TF/s (peak) machine. On a 1-PF/s (peak) computer, this simulation will require 1.5 to 3 days per run, assuming no loss in efficiency.

Example From a Lab PI: Biology

Modeling a cell's metabolic, regulatory & signaling networks:

- A particle-based method has been developed for cellular response.
- The model runs at 54,000 operations per particle, per time step (measured).
- Elapsed time on a 1 PF/s machine for
 - 50 million atoms
 - 180 million time steps
$$= 54,000 * 50 \text{ million} * 180 \text{ million} / 10^{15} = 6 \text{ days}$$



IV. Related Issues

There is confusion in the OS community between “capability” and “capacity”

- A “capability” application:
 - Uses most of the machine in a single run
 - Has a turn-around time for a single run of hours to weeks
- PIs and SCaLeS contributors ...
 - Tend not to distinguish between the two,
 - Sometimes treat “capability” as a generalized statement of goodness
- Concerns were voiced about the narrow focus of this assessment – *i.e.*, capacity needs are not addressed

A key issue emerged during the discussions

Many PIs are not aware of a major shift in the semiconductor industry strategy:

- *Increased processor throughput will be accomplished by providing more computing elements (“cores”), rather than by increasing the operating frequency. Thus, petascale architectures in the next five years will have many times the number of processors (cores) than current designs, thereby increasing the burden of parallelization.*
- The 5-year Semiconductor Industry Roadmap shows a factor of two increase in operating frequency from 3.2 GHz (2004) to 6.4 GHz (2009), whereas Intel, AMD, and others are migrating to multiple (2x, 4x, 8x ...) cores within a single “processor.”

Experience has shown that balance on multi-core processors can be a significant issue.

This change in processor design has important implications for Petascale computing

The baseline architectures we've assumed for this study:

100 TF/s:

5 GF/s each "processing element"

20,000 processing elements

1 PF/s:

10 GF/s each processing element

100,000 processing elements

Not all applications can fully utilize these designs at the present time:

- While there are many applications in fusion, biology, & materials that can use these designs, some applications that are multi-scale in time require significant algorithmic advances.
- A single climate simulation using the spectral transform method for atmospheric modeling (100 yrs, 10km res.) would require more than one year of wall-clock time. (The SEAM method applied to the atmosphere, however, could reduce that time to a few months or less.)

V. Summary by Discipline

Applications Summary — Examples

[Please see Section VI. for comprehensive discussions]

- **Accelerators:** Through advances in modeling non-linear and collective effects, Petascale computing could be used to enhance the understanding of beam behavior and to optimize accelerator performance.
- **Astrophysics:** Petascale computing could be used to greatly improve the understanding of core-collapse supernovae, and to resolve galaxy formation (cosmology) and planet formation (accretion disks).
- **Biology:** Petascale capability computing could be used to model cell function, ribosome machine function, docking, In addition, needs/desires for capacity computing (image analysis and data searching) have been expressed.

Applications Summary — Examples

- **Chemistry:** Petascale computing could make possible high-accuracy calculations that are capable of replacing experiment in terms of reliability and precision. For instance, it will be possible to compute accurate thermodynamics for all the hydrocarbons and intermediates important to combustion.
- **Climate and Earth Science:** Some aspects of Climate modeling could take advantage of Petascale computing: for example, adding the carbon cycle, ocean biogeochemistry, and other new physics to climate simulations, and resolving ocean eddies. However, due to algorithmic costs and scaling, the Holy Grail of achieving a 100-year atmospheric simulation at 10 km resolution will not be possible in a reasonable amount of time on foreseeable Petascale computers using the currently dominant method of Spectral Transforms. (Another approach, Spectral Elements, would not be constrained in this regard.)

Applications Summary — Examples

- **Combustion:** Petascale computing could be used to predict pollutant emissions, to simulate autoignition with realistic fuels, to model the growth and oxidation of soot particles, and to model laboratory-scale turbulent combustion experiments in detail (3-d, sufficient chemistry).
- **Environmental Remediation and Processes:** Petascale computing could be used to simulate the Hanford “leak event,” probably used to simulate other regional ecological impacts requiring long-term, large-scale, 3-d, high-resolution, 3-phase, multi-fluid flow and multi-component reactive transport, and to approach the elusive goal of real time multi-sensor data inversion. However, detailed scaling estimates are available only for the Hanford event. (One of the SCaLeS discipline coordinators notes a current lack of priority in accessing high-performance computer resources).

Applications Summary — Examples

- **Materials Science:** Petascale computing could be used to advantage for problems such as high temperature superconductivity, magnetism, and toughening ceramics, but many applications (and PIs) in the field are focused more on capacity rather than capability.
- **Nanoscience:** Petascale computing could be used to carry out molecular dynamics simulation of early key steps in the growth of colloidal quantum dots, the calculation of the electron transport properties of organic molecules, and the characterization of a 1000-atom FePt particle (perhaps applicable to future storage devices).

Applications Summary — Examples

- **Plasma Science/Fusion:**

Petascale computing could be used for Tokamak modeling that does not require stressful multi-scale time estimates (such as increasing the simulation time from 1 msec to 1 sec, approaching the confinement time). The biggest programmatic advance will probably be a comprehensive, integrated simulation and 10x resolution. Estimates which distinguish between 100Tf/s and 1 PF/s peak are not available for Tokamak modeling. Petascale computing could also be used for resolving magnetic reconnection through 3-d simulations using realistic mass ratios.

- **QCD:** Petascale computing could be used to compute the weak interaction matrix elements of strongly interacting particles in support of DOE's effort to make precise tests of the Standard Model, to calculate the temperature and order of quark-gluon plasma phase transition, and to elucidate the quark & gluon structure of nucleons.

Conclusions

- In general, the Office of Science community has described — with varying degrees of insight and specificity — credible needs for Petascale capability computing and the ability to take advantage of such machines. Issues of op-counts, algorithmic scaling, and processor scaling to thousands of nodes have not been addressed uniformly across the community, but appear reasonable when discussed.
- There are numerous examples requiring Petascale computing in the disciplines of Materials, Chemistry, and Biology. However, significant elements of these communities appear tilted toward capacity needs (materials, chemistry) and data-base needs (biology) rather than capability.
- Because of shifting technology trends within the semiconductor industry, some applications will require significant algorithmic improvements to fully utilize the 10,000s of processors on Petascale architectures that will become available in the next five years.

VI. Written analysis, detailed descriptions, and additional documentation.

Input from the persons providing analysis, descriptions, and examples has been re-produced verbatim, including any clarifying dialogue with the authors.

Petascale Applications —

Accelerators

Impact of Petaflop-scale Computing: Application — Accelerators

	Accelerators
<p>Programmatic impact to be gained by access to capability Petaflop-scale computing</p>	<p>Please indicate a few bullets which indicate the potential impact of Petaflop/s-scale computing as defined in the email cover letter.</p> <ul style="list-style-type: none"> • Getting the most Science from the Nation’s particle accelerators – using petascale modeling in concert with theory and experiments to optimize performance and expand operational envelopes • Improved designs for future accelerators – using petascale modeling to reduce cost & risk • Development of novel, groundbreaking methods for particle acceleration – using petascale modeling, in concert with theory and experiment, to explore, optimize, and implement laser- and plasma-based accelerators
<p>Major scientific challenges to be addressed</p>	<p>Indicate the scientific challenges that are associated with the entries in the box above.</p> <ul style="list-style-type: none"> • Optimizing the performance of an accelerator is an extremely challenging task: the beam behavior is governed by a combination of nonlinear effects and collective effects that can degrade beam quality and beam intensity and can lead to beam instabilities. Using petascale computing to improve accelerator performance will require a combination of petascale hardware and software resources (to perform and analyze the simulations), beam measurements, and mathematical methods for code validation, code calibration, uncertainty analysis, and prediction. <p>An example is provided by the Large Hadron Collider, which is expected to come on line at the end of the decade. This is a multi-billion dollar facility, in which the US investment is approximately 1 billion dollars. When this machine comes on line, high-end computing will play an important role in commissioning, understanding beam behavior, and optimizing the accelerator performance. An important collective phenomenon known as the electron-cloud effect will be a key issue, and it is now being vigorously studied using terascale resources. A complete, high-fidelity simulation will require the use of near-petascale resources.</p> <ul style="list-style-type: none"> • Accelerators are among the largest and most complex scientific instruments ever built, and future accelerators will “push the envelope” even further, particularly with regard to beam intensity. Because of their size, small changes in the design of large accelerator facilities can have huge financial consequences. “Over-designing” a machine (i.e. using an extremely conservative design) can cost hundreds of millions of dollars in capital costs; conversely, accelerator system optimization and better decision-making through high-fidelity, end-to-end petascale simulations can lead to designs that save hundreds of millions of dollars. • The successful development of ultra-high gradient accelerators through the laser or particle beam driven approach would have huge consequences for science, industry, and medicine. But, though experiments have already demonstrated gradients 100x to 1000x larger than conventional technology,

	<p>it is extremely challenging to control and stage plasma sections into usable, production-capable particle accelerators. The systems themselves involve the simultaneous interaction of beams, plasmas, and radiation under extreme conditions, making diagnostics difficult. As a result, petascale simulations, used in concert with theory and experiment, provide one of the most powerful tools to understand these complex systems, and to ultimately design and implement plasmas-based accelerators.</p>
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p> <p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p>[We are assessing the needs for <i>capability</i> – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)]</p>	<p>Shown below are four examples of accelerator simulations performed on 2048 processors of the NERSC IBM SP3 computer:</p> <p>Quasi-static Particle-In-Cell (PIC) code: 208 GFLOPS, equivalent to 7% of peak. Fully explicit electromagnetic PIC code: 300 GFLOPS, equivalent 10% of peak Beam-Beam code (“weak-strong” model): 167 GFLOPS, equivalent to 5% of peak. Nonlinear beam optics code: 304 GFLOPS, equivalent to 10% of peak.</p> <p>Typical execution time for these codes is currently 12-24 hours per run. Here we describe 3 factors that are currently impacting our simulation capability, and which will be alleviated by petascale resources:</p> <ol style="list-style-type: none"> 1. Accelerator design codes are often used in parameter studies and error studies involving tens to hundreds of runs. As a result, the time-to-solution for a single study can be as much as several thousand hours on present hardware. 2. Even on terascale systems, some problems, such as modeling beam dynamics in accumulator rings, involve simplifications and limitations in order to have acceptable execution time. For example, an accumulator may contain on the order of 100 microbunches, but we typically use only a few microbunches in the simulation. Using petascale resources, it will be possible to model all the microbunches. Similarly, beam-beam simulations of hadron colliders are now typically performed for on the order of 100,000 turns, equal to about a second of beam time; but in order to accurately extract the predicted beam lifetime from the simulation, it is desirable to simulate a few minutes of beam time, which would require approximately 100 times more computation. Again, petascale resources will make this possible. 3. Accelerator modelers have begun developing tools to simulate beams in circular machines for hundreds of thousands or millions of turns in the presence of weak space-charge effects and machine resonances. In such a situation, the issue of numerical collisionality is much more stringent than in other types of accelerator simulations, because, due to the long simulation time, the disparity of longitudinal motion and transverse motion, and the weakness of the space-charge, the numerical collisionality may overwhelm the physics being studied. Petascale resources are essential, because the simulations are <i>both</i> very long and require very low noise.
<p>What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors</p>	<p>Typical beam dynamics runs are performed on 256-1024 processors. The largest number of processors used by our team to date is 4096.</p>

used to-date?	
What is the Operations Count/Scaling from other computers?	<p>1. Quasi-static and fully explicit Particle-In-Cell codes: In the particle-dominated regime, such codes scale linearly with the number of macroparticles, N_p. Furthermore, in the simulation of a beam in a circular machine, the scaling is $N_p * N_{\text{bunches}} * N_{\text{turns}}$, where N_p is the number of macroparticles per bunch, N_{bunches} is the number of bunches, and N_{turns} is the number of turns simulated.</p> <p>2. Beam-Beam code (weak-strong model): Scaling varies as $N_p * N_{\text{turns}} * N_{\text{collpts}}$, where N_p is the the number of simulation particles, N_{turns} is the number of turns simulated, and N_{collpts} is the number of collision points. Due to the nature of the calculation, parallel efficiency is nearly 100%.</p> <p>3. Beam-Beam code (strong-strong model): Scaling varies as $N_p * N_{\text{turns}} * N_{\text{collpts}} * N_{\text{slices}}^2 * N_{\text{bunches}}^2$, where N_p is the number of macroparticles per bunch, N_{turns} is the number of turns, N_{collpts} is the number of collision points around the ring, N_{slices}^2 is the square of the number of slices used in the simulation, and N_{bunches}^2 is the square of the number of effective bunches circulating around the ring.</p> <p>4. Self-consistent Langevin code: This type of code is thousands of times more compute-intensive than a quasi-static PIC code, because the equivalent of thousands of Poisson solves are required at every time step. Since the calculation is analogous to a quasi-static PIC calculation, but many more operations are required per time-step and per processor, such codes are expected to have much longer run times, but they are likely to have sustained performance similar to quasi-static PIC codes, and to scale at least as well as they scale or better.</p> <p>5. Direct Vlasov Codes: Scaling varies as N^6 for a 3D code, where N is the # of grid points in each phase space dimension. Due to the 6th order scaling, it is certain that such methods, to be successful, will have to use adaptive gridding in phase space, or to uses bases (e.g. wavelet bases) that allow for significant information compression.</p>
Projected increase in software efficiency?	<p>Many types of accelerator modeling codes require significant interprocessor communication; as a result, our community needs “balanced” systems. We are projecting that petascale systems will provide a balance of processor speed, latency, and bandwidth, and include optimized mathematical software for the system, so that our codes, when scaled up to tens of thousands of processors, will perform with an efficiency at least equal to that of our current codes on present-day platforms. In addition, we intend to exploit a new simulation methodology for a certain class of problems in order to achieve high parallel efficiency on petascale platforms: namely, for problems involving <i>design optimization</i>, we will perform multiple terascale simulations simultaneously in a single petascale run.</p>
Other	

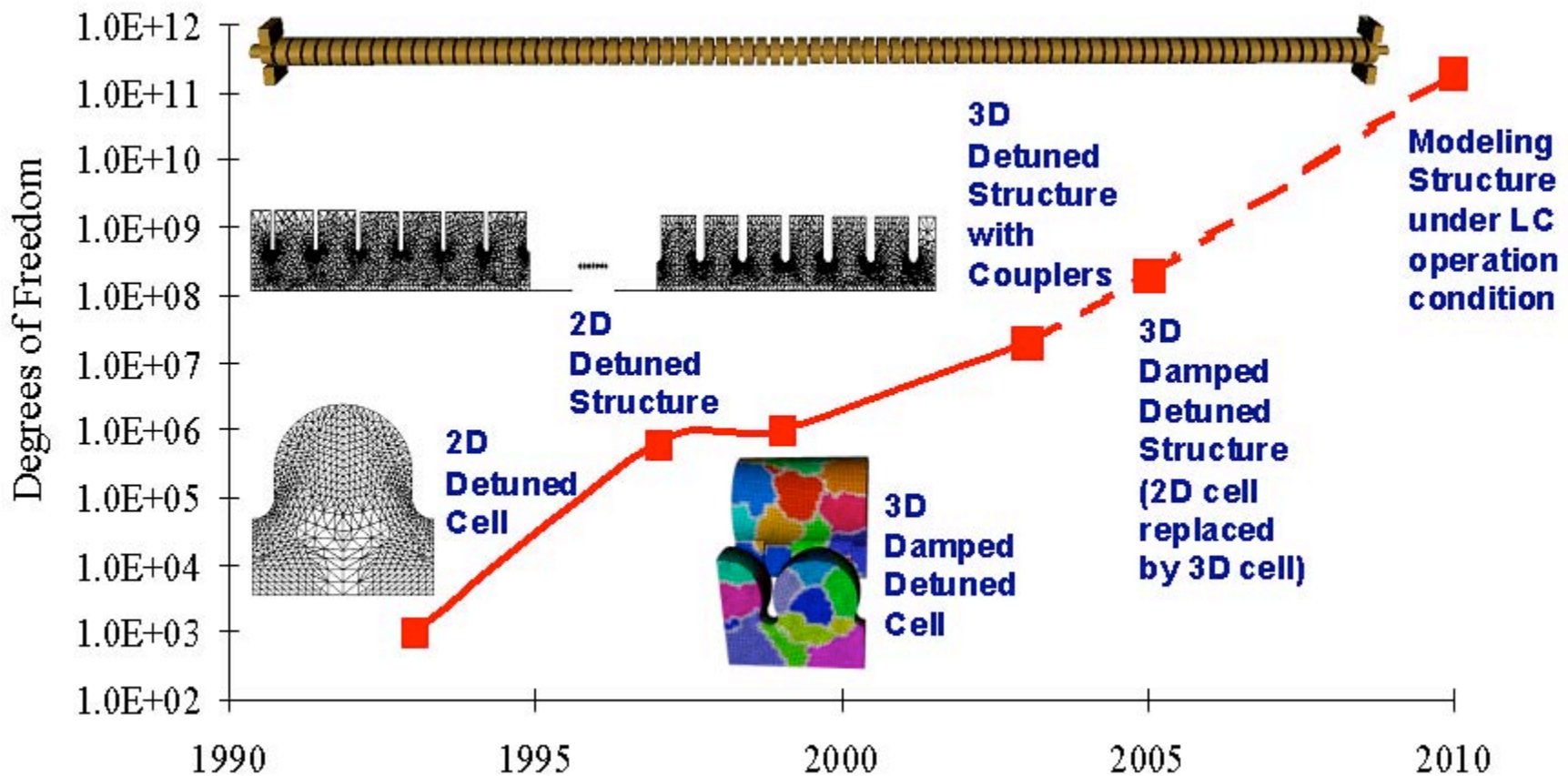
Impact of Petaflop-scale Computing: Application — Accelerators

	Accelerators
<p>Programmatic impact to be gained by access to capability Petaflop-scale computing</p>	<p>Please indicate a few bullets which indicate the potential impact of Petaflop/s-scale computing as defined in the email cover letter.</p> <ul style="list-style-type: none"> • <i>Half of the scientific instruments in DOE SC's 20-year Strategic Plan is accelerator based</i> • <i>Terascale computing is already playing an essential role in Accelerator Modeling to improve existing accelerators, design future machines, and advance accelerator science</i> • <i>Petascale computing will make possible the next level of high fidelity, high resolution simulations with major impact on DOE's science portfolio by ensuring the success of operating and constructing accelerator facilities at lower cost and risk</i>
<p>Major scientific challenges to be addressed</p>	<p>Indicate the scientific challenges that are associated with the entries in the box above.</p> <ul style="list-style-type: none"> ⟨ <i>High performance computing is used to address three main accelerator areas: Electromagnetic modeling, Beam-beam Interactions, and Advanced Acceleration</i> ▪ <i>In electromagnetic modeling, challenge is to virtually prototype 3D, complex accelerating Systems essential to existing and future facilities which include PEP-II, NLC and RIA</i> ⟨ <i>In beam-beam interactions, challenge is to understand beam behavior in realistic to predict and optimize performance of accelerators such as the Tevatron, LHC and LCLS</i> ⟨ <i>In advanced acceleration, challenge is to realize novel, compact accelerating schemes like Laser plasma and plasma wakefield accelerators</i> • <i>PEP-II and the Tevatron are accelerators currently in operation; LCLS, RIA and the NLC are high Priority items on the 20-year Strategic Plan</i>
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p> <p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p>[We are assessing the needs for <i>capability</i> – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be</p>	<p>These type of throughput data are not directly indicated in the report. However, the kind of answer we are lookinor might be something like this: "We did calculations on a 5 Tflops/s (peak) machine, achieving sustained throughput of 0.5 Tflops/s (or 10% efficiency). The turn-around time is about days."</p> <p>(Ed – We also have numbers from beam-beam simulations but you may already have some from Rob) (- For numbers on Advanced Acceleration, Warren is a good source)</p> <ul style="list-style-type: none"> • <i>In electromagnetic modeling, both speed and memory are important. We have two types of Codes – frequency domain and time domain. Both types have efficiency in the 5-10% range Depending on the application. Frequency codes require more memory and runtime is dictated by number of modes or frequency points. For example, our largest eigensolver run</i>

done with many runs.]	<p><i>took close to 750 GB, 1024 CPUs (1.5 Tflops/s peak) and 24 hours on the NERSC's IBM/SP with a throughput of 150 Gflops/s. What we'd like to do is to increase the resolution of the model which at least doubles the memory and calculates 10 times more modes which at least runs 10 times longer. Certainly a 100X capability will put this level of simulations within reach.</i></p> <ul style="list-style-type: none"> • <i>SLAC's beam-beam codes also have efficiency in the 5-10% range also using the IBM/SP at NERSC.</i>
What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?	<p>Credible architectures for achieving Petaflop-scale capability within 5 years will contain tens of thousands of processors. The ability for codes to run efficiently with large numbers of processors will be critical. Running today's codes with large numbers of processors can give useful insights into projected scaling behavior. Please provide us with your experience.</p> <ul style="list-style-type: none"> • <i>Our electromagnetic codes typically use from 256 to 1024 processors and 2048 is the largest number to date</i> • <i>Our beam-beam codes use 256 processors mostly and 1024 is the maximum used.</i>
What is the Operations Count/Scaling from other computers?	<p>To scale performance from today's machines to larger capability machines requires either:</p> <ul style="list-style-type: none"> • An operations count, or • A scaling law based on current performance on current machines <p>If you have used a scaling law to characterize Petaflop-scale performance, please provide the logic used (e.g. compute time scales as n^4, where n is a linear cell dimension), along with the current computer capability used in the scaling. In both cases please provide the required turn-around time for the longest runs.</p> <ul style="list-style-type: none"> ○ <i>For the eigensolver mentioned above, the operation count scales as $N^{*(1,5)}$ where N is the number of degrees of freedom (DOF). In the largest run described, N was 93 million.</i>
Projected increase in software efficiency?	<p>If you are counting on an increase from better algorithms (historically, algorithm improvements have approximately matched improvements in hardware), please indicate the factor you've used.</p> <ul style="list-style-type: none"> • <i>We are expecting a gain of at least ten fold in efficiency from our SciDAC efforts in computer science and applied mathematics to develop better algorithms</i>

Large-Scale Electromagnetic Simulation

More than 10^3 fold increase in problem size over a decade



Subject: two examples (Re: Table_Accel.doc (Re: help in justifying future DOE "capability" systems foraccelerators))

Date: Friday, May 21, 2004 12:39 PM

From: Robert D. Ryne <RDryne@lbl.gov>

To: Ed Barsis ebarsis@bmv.com

Cc: "Ko, Kwok" kwok@slac.stanford.edu, "David E. Keyes" kd2112@columbia.edu, "Peter L. Mattern" pmattern@bmv.com, "Viktor K. Decyk" decyk@physics.ucla.edu, More...

Ed,

Here are two examples, as you requested.

Note that, in the second example, I mistakenly omitted a parameter, $N_{\text{kicksperturn}}$, when I emailed you the original writeup.

(1) beam-beam modeling (weak-strong regime):

Increase N_{turns} by 200x
Increase N_p by 5x
Total increase = $200 \times 5 = 1000$.

(2) quasi-static PIC (circular machine w/ space charge effects):

Increase N_p by 8x
Increase N_{turns} by 8x
Increase N_{bunches} by 4x
Increase $N_{\text{kicksperturn}}$ by 4x
Total increase is $8 \times 8 \times 4 \times 4 = \sim 1000$

Rob

Ed Barsis wrote:

>Rob,

>

>Thanks for the very useful information. Would you pick two examples from
>the five scaling laws that you gave, and indicate what increases in
>parameters are needed (eg N_p increases by a factor of 10,.....) to get from
>the computers used today to Petascale computers (eg 100 Tflop/s sustained).
>For example the beam-beam code sustains 167 Gflop/s (run time 12-24 hours).
>If you did the problem you want to do how would the parameters noted for
>beam-beam scaling change if you were using a Petascale computer (and still
>get run times of 12-24 hours)?

>

>Thanks again.

>

>Ed

>

>-----Original Message-----

>From: Robert D. Ryne [mailto:RDryne@lbl.gov]

>Sent: Wednesday, May 19, 2004 5:27 PM

>To: Ed Barsis

>Cc: Ko, Kwok; 'David E. Keyes'; 'Peter L. Mattern'; Viktor K. Decyk;

>Warren Bicknell Mori; Ji Qiang; Panagiotis Spentzouris

>Subject: Table_Accel.doc (Re: help in justifying future DOE "capability"
>systems foraccelerators)

>

>Ed,

>Here is the write-up from me and my colleagues [thanks Viktor, Warren,
>Ji, and Spentz].
>This covers several of the types of codes in use. Combined with the info
>that Kwok sends you,
>you should have a good overview of the field.
>
>It is possible that we will send you some minor revisions tomorrow, but
>whether or not
>we do, I wanted to make sure that you could start looking at this now as
>you prepare your
>viewgraphs for Friday. Please call me if you need any clarification or
>follow-up info.
>I can be reached by cell, (510)847-3089. I will be at LANL starting on
>Thursday,
>and I cannot always use my phone there. But I can of course be reached
>by email too.
>
>Regards,
>Rob
>
>
>

Petascale Applications —

Astrophysics

Impact of Petaflop-scale Computing: Application — Astronomy and Astrophysics

	Astronomy and Astrophysics
<p>Programmatic impact to be gained by access to Petaflop-scale computing</p>	<p>The general Scientific Opportunities have been well laid out in this chapter. However, the opportunities that are within the reach of Petascale capability computing of a few tenths of a Petaflop/s peak to a few Petaflop/s peak (for this study) have not been delineated. In this box, we have made some guesstimates from your write-up which we assume you will correct: some are probably out of the range of the reduced level of capability we are considering.</p> <ul style="list-style-type: none"> • Large-Scale Structure and Cosmology. Predict the formation of galaxies by luminosity, morphology... • Galaxy Formation and Interactions. Model stellar evolution from birth to death. • Star Formation. Model the formation of stars and planets from the star forming clouds of interstellar gas. • Stellar Evolution. Model the evolution of stars. • Stellar Death. Correctly describe the explosion mechanism. • Numerical Relativity. Simulate black hole and neutron star mergers to predict gravitational wave emission. • Astrophysical Data. Programmatic impact and scientific challenge of using capability machines is not clear from the write-up.
<p>Major scientific challenges to be addressed</p>	<ul style="list-style-type: none"> • Large-Scale Structure and Cosmology. Current models which treat stars as point masses must be supplanted by modeling details of star formation. • Galaxy Formation and Interactions. Include feedback such as stellar winds, supernovae ... to the interstellar and intergalactic media, correctly treat magnetic fields, and include energetic particle origins and dynamics. • Star Formation. Couple the multi-physics evolution equations involving turbulence, MHD, self-gravity, chemical networks, multi-dimensional radiation transport, "dusty" plasmas and interstellar gas. • Stellar Evolution. Develop 3-D models incorporating convection, interior rotation, pulsation, nuclear chemistry, photon and neutrino radiation and magnetic fields. • Stellar Death. Develop models of turbulent deflagration and deflagration-to-detonation transitions in the conditions of a supernova, and incorporate these processes into 3-d simulations of the explosions. • Numerical Relativity. Similar challenges to Stellar Death with the added complexity that the applied math for the solution of the Einstein field equations lags that for PDEs. • Astrophysical Data. Programmatic impact and scientific challenge of using capability machines is not clear from the write-up.
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made? Please indicate the code efficiency and/or the computer</p>	<p>These type of throughput data are not directly indicated in the report. However, the kind of answer we are looking for might be something like this: "We do our longest calculations on a 5 Tflops/s (peak) machine, achieving sustained throughput of 0.5 Tflops/s (or 10% efficiency). The turn-around time is about days."</p>

<p>peak performance.</p> <p>Please also indicate the turn-around time.</p> <p>[We are emphasizing <i>capability</i> – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)]</p>	
<p>What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?</p>	<p>These data are not mentioned in the Report. Credible architectures for achieving Petaflop-scale capability within 5 years will contain tens of thousands of processors. The ability for codes to run efficiently with large numbers of processors will be critical, as suggested in your report and comments on interconnect speed. Running today’s codes with large numbers of processors can give useful insights into projected scaling behavior. Please provide us with your experience.</p>
<p>What is the Operations Count/Scaling from other computers?</p>	<p>These numbers do not appear in the Report.</p> <p>To scale performance from today’s machines to larger capability machines requires either:</p> <ul style="list-style-type: none"> • An operations count, or • A scaling law based on current performance on current machines <p>If you have used a scaling law to characterize Petaflop-scale performance, please provide the logic used (e.g. compute time scales as n^4, where n is a linear cell dimension), along with the current computer capability used in the scaling. In both cases please provide the required turn-around time for the longest runs.</p>
<p>Projected increase due to better algorithms?</p>	<p>If you are counting on an increase in throughput from better algorithms (historically, algorithm improvements have approximately matched improvements in hardware), please indicate the factor you’ve used.</p>
<p>Other</p>	

Estimating the computational cost (dominated by dense block solve)...

$$\text{FLOPS}_{\text{lower}} \sim N_t N_i N_s N_m^2 \sim 1.58 \times 10^{20}$$

$$\text{FLOPS}_{\text{upper}} \sim N_t N_i N_s N_m^3 \sim 2.43 \times 10^{23}$$

$$N_t = \text{number of time steps} \sim 5 \times 10^4$$

$$N_i = \text{number of iterations per time step} \sim 10$$

$$N_s = \text{number of spatial zones} \sim 512^3$$

$$N_m = \text{number of neutrino momentum zones}$$

$$N_m = N_E \times N_p \times N_a$$

$$N_E = \text{number of neutrino energy groups} \sim 24$$

$$N_p = \text{number of neutrino polar direction angles} \sim 8$$

$$N_a = \text{number of neutrino azimuthal direction angles} \sim 8$$

@ 2 TF (20% of a 10 TF machine) \sim 3 yrs.

@ 20 TF (20% of a 100 TF machine) \sim 3 mos.

@ 200 TF (20% of a 1 PF machine) \sim 1 week

$$\text{FLOPS}_{\text{actual}} \sim \text{factor} \times N_m^2$$

The algorithms used will set this factor.

$$\text{factor} \ll N_m ?$$

$$\text{factor} \sim N_m ?$$

For lower FLOPS limit!



TeraScale Supernova Initiative



TeraScale Supernova Initiative

www.tsi-scidac.org

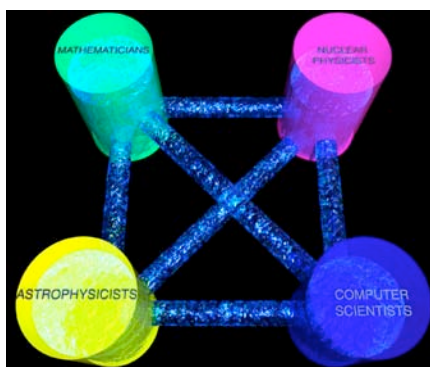
Explosions of Massive Stars

Relevance:

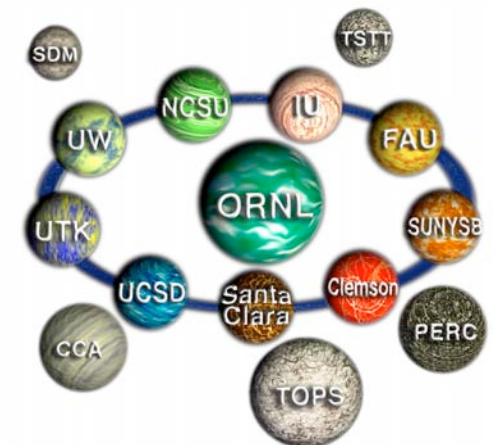
- ⇒ **Element Production**
- ⇒ **Cosmic Laboratories**
- ⇒ **Driving Application**

*11 Institution, 21 Investigator, 34 Person, **Interdisciplinary Effort***

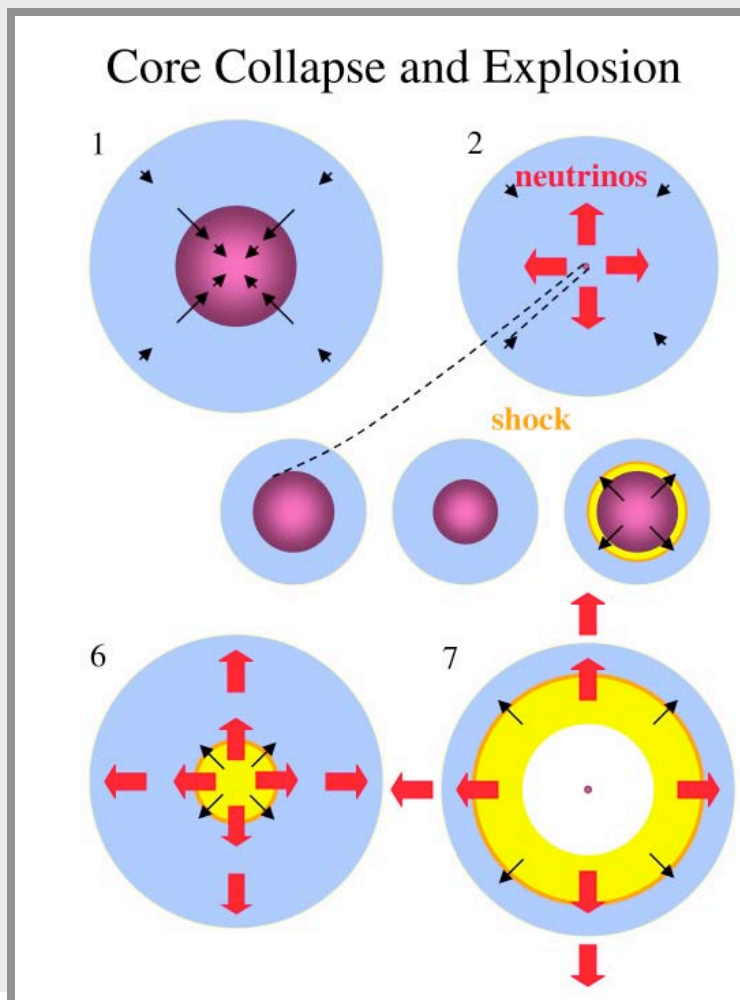
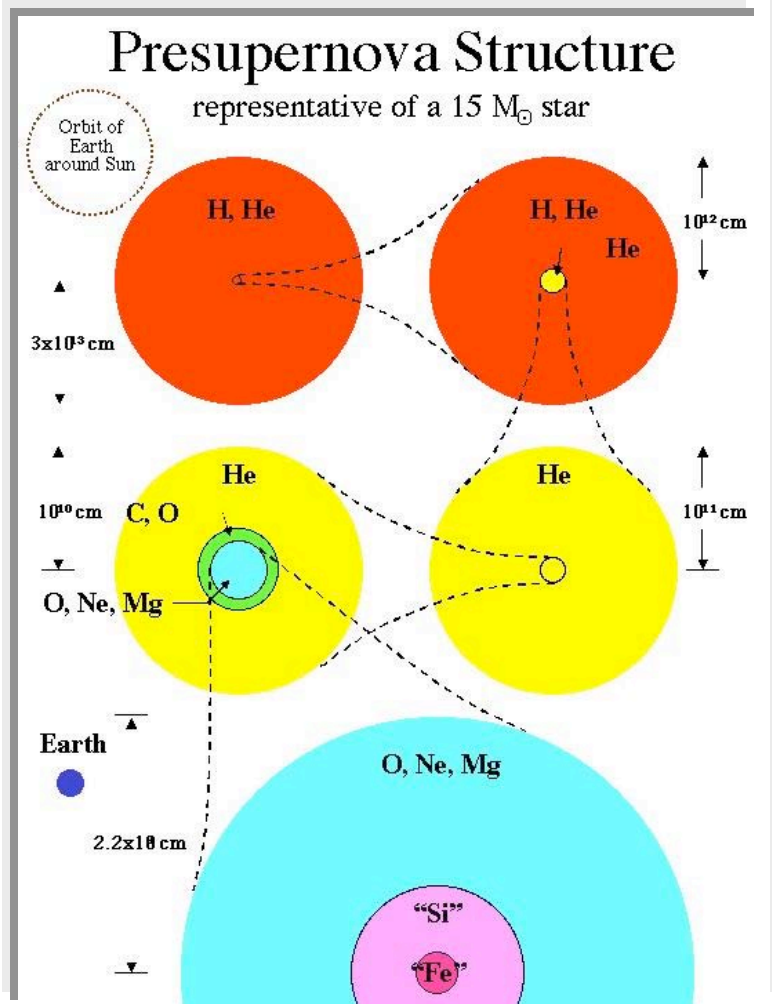
- ⇒ *ascertain the core collapse supernova mechanism(s)*
- ⇒ *understand supernova phenomenology*
 - *e.g.: (1) element synthesis, (2) neutrino, gravitational wave, and gamma ray signatures*
- ⇒ *provide theoretical foundation in support of OS experimental facilities*
- ⇒ *develop enabling technologies of relevance to many applications*
 - *e.g. 3D, multifrequency, precision radiation transport*
- ⇒ *serve as computational science testbed*
 - *drive development of technologies in simulation “pipeline” (data management, networking, data analysis, and visualization)*



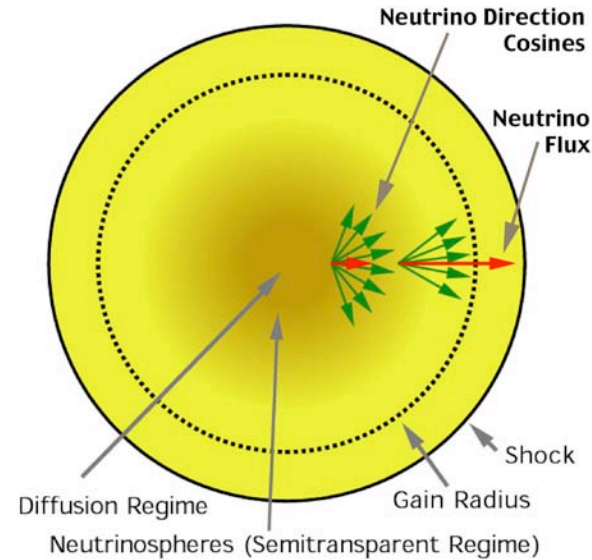
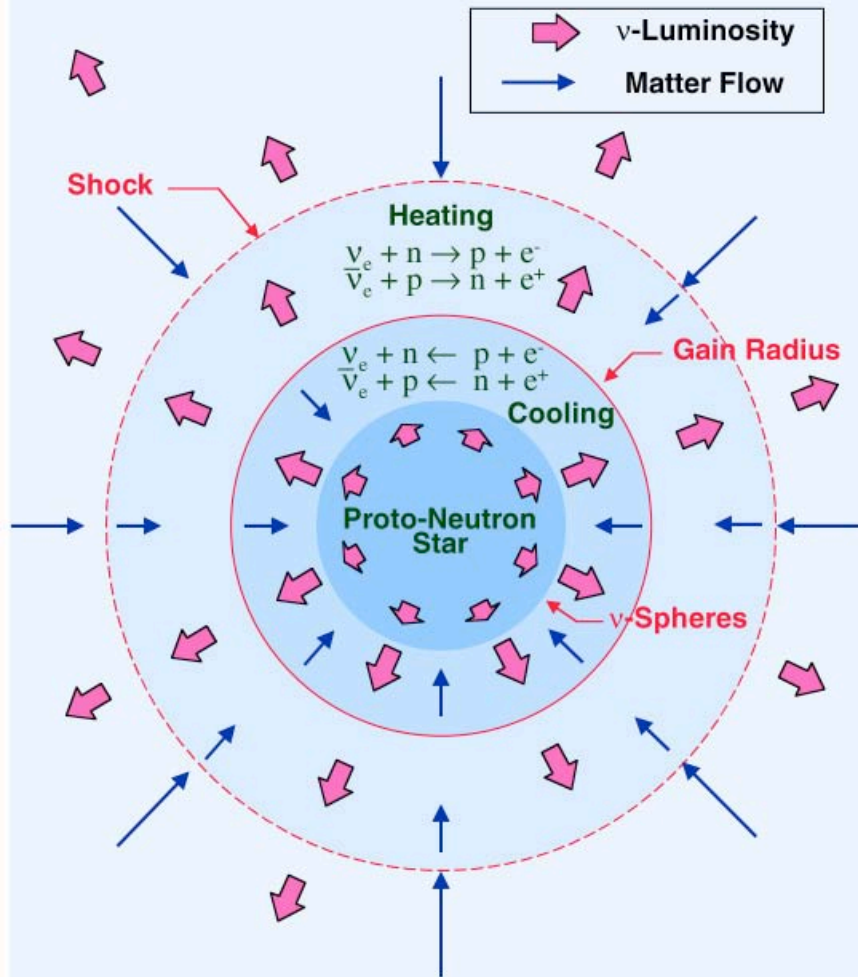
*With ISIC and other collaborators:
89 people from 28 institutions involved.*



Core Collapse Supernova Paradigm



Anatomy of a Supernova



Need Boltzmann Solution

Need Angular Distribution

Need Spectrum

Need Neutrino Distribution



$\sim 10^{53}$ erg radiated in neutrinos

$\sim 10^{51}$ erg explosion energy

**Will need to conserve total energy to
0.1% over the entire simulation of**

10^{5-6}

cycles!

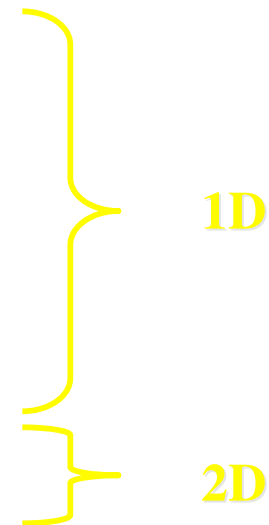
The Case for Accurate Neutrino Transport



Ten Years, Ten Studies

With the exception of Wilson's models, no models with multigroup transport explode.

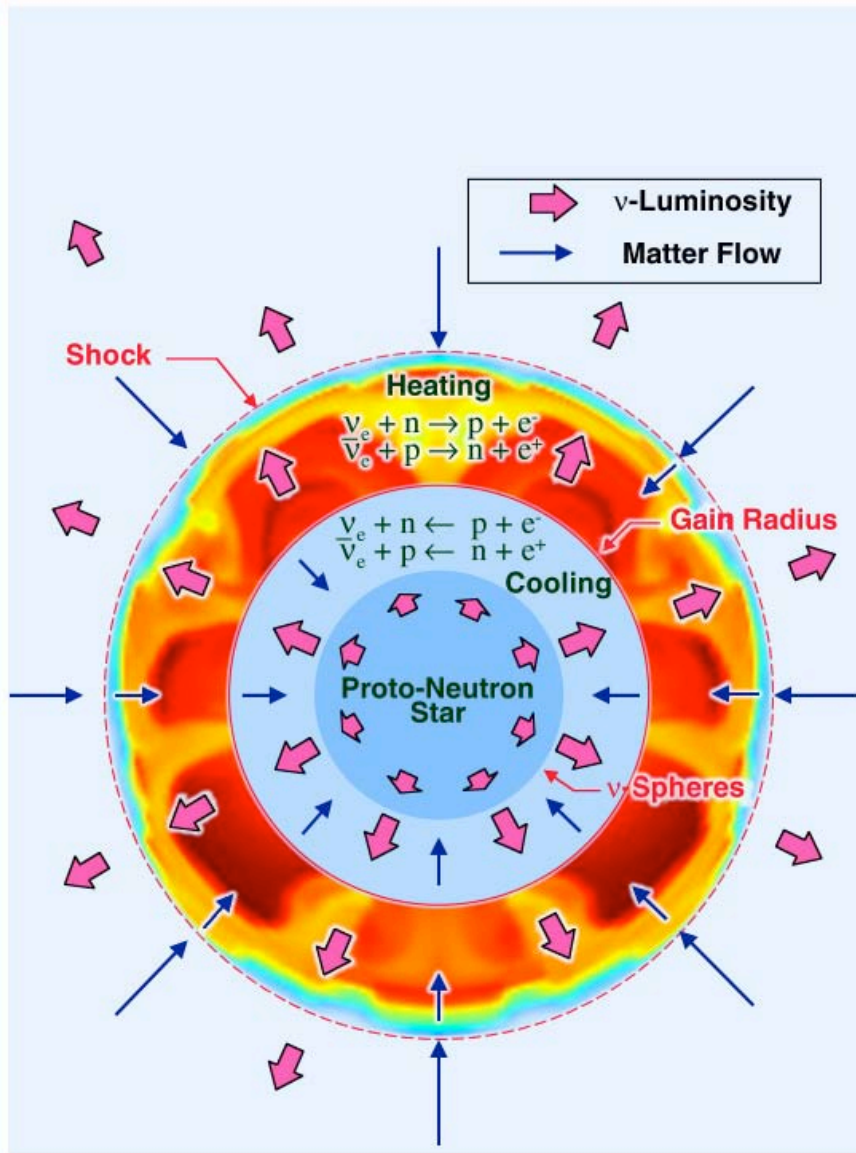
- Bruenn (1993)
- Wilson and Mayle (1993)
- Swesty et al. (1994)
- Rampp and Janka (2000)
- Bruenn, DeNisco, and Mezzacappa (2001)
- Mezzacappa et al. (2001)
- Liebendoerfer et al. (2001)
- Thompson and Burrows (2002)
- Mezzacappa et al. (1998)
- Buras et al. (2003)



Wilson's models invoke neutron fingers. Without them, his models do not explode either.

- Existence of neutron fingers is a matter of debate (Bruenn and Dineva, 1996).
- Wilson does not “get” neutron fingers when Lattimer-Swesty EOS is used.

Is the story that “simple”?



Possible Instabilities:

⇒ Convection (e.g., Ledoux)

Negative gradients in entropy, lepton fraction, or both.

⇒ Doubly Diffusive Instabilities (e.g., Neutron Fingers, LEF)

Crossed gradients in entropy and lepton fraction.

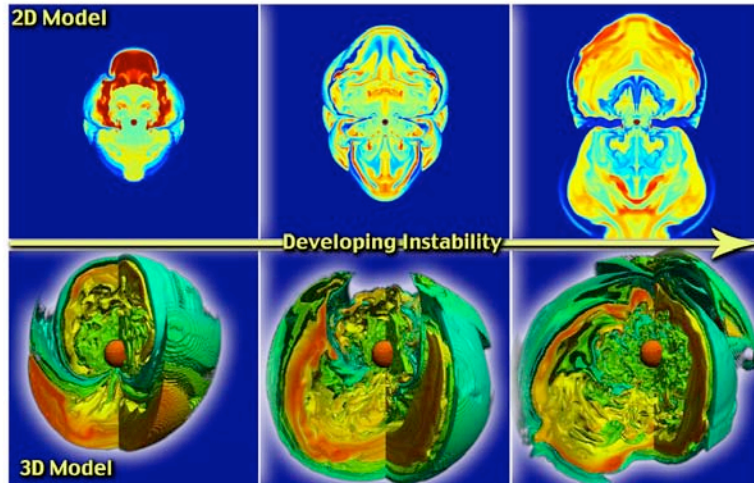
⇒ Shock Wave Instability

Something completely different.

2 fundamentally new instabilities discovered by TSI (computationally):



Stationary Accretion Shock Instability (SASI)



Supernova shock wave may become unstable.

Instability will

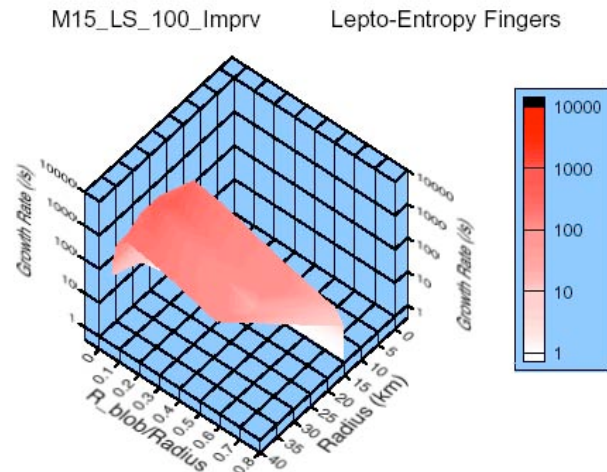
⇒ help drive explosion,

⇒ define explosion's shape.

Operates between the proto-neutron star and supernova shock wave.

Blondin, Mezzacappa, and DeMarino (2003)

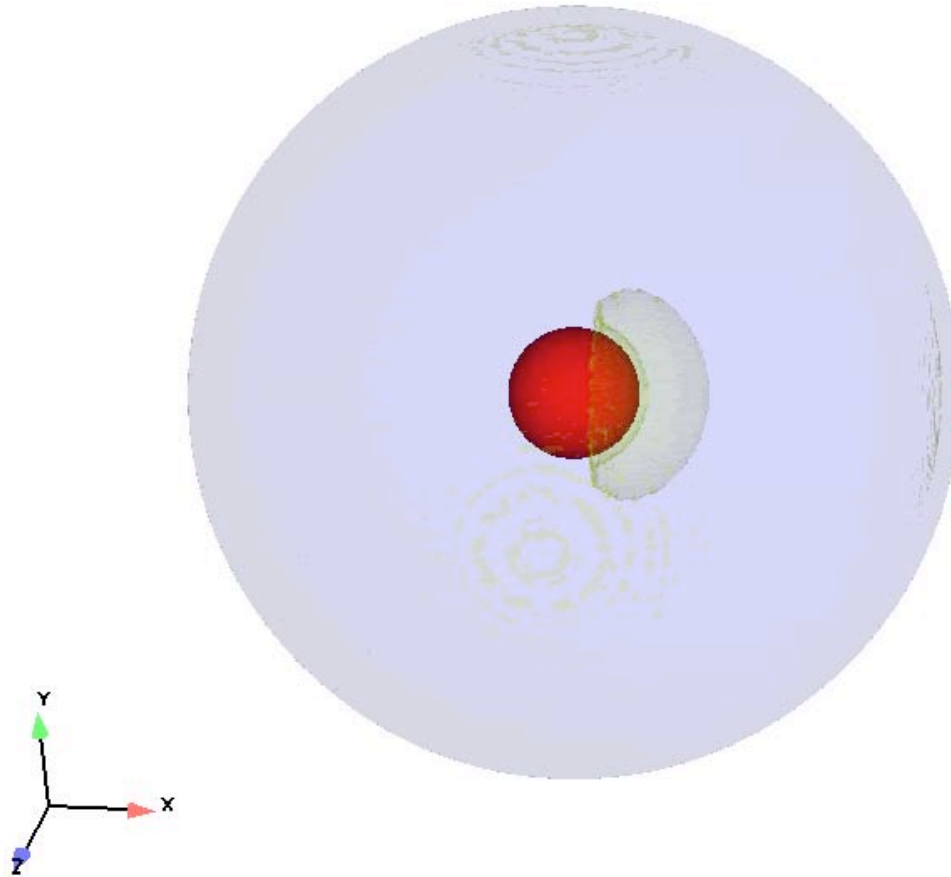
Lepto-Entropy Fingers



A new doubly diffusive instability in the proto-neutron star. Instability may help boost neutrino luminosities, which power the explosion.

Bruenn, Raley, and Mezzacappa (2004)

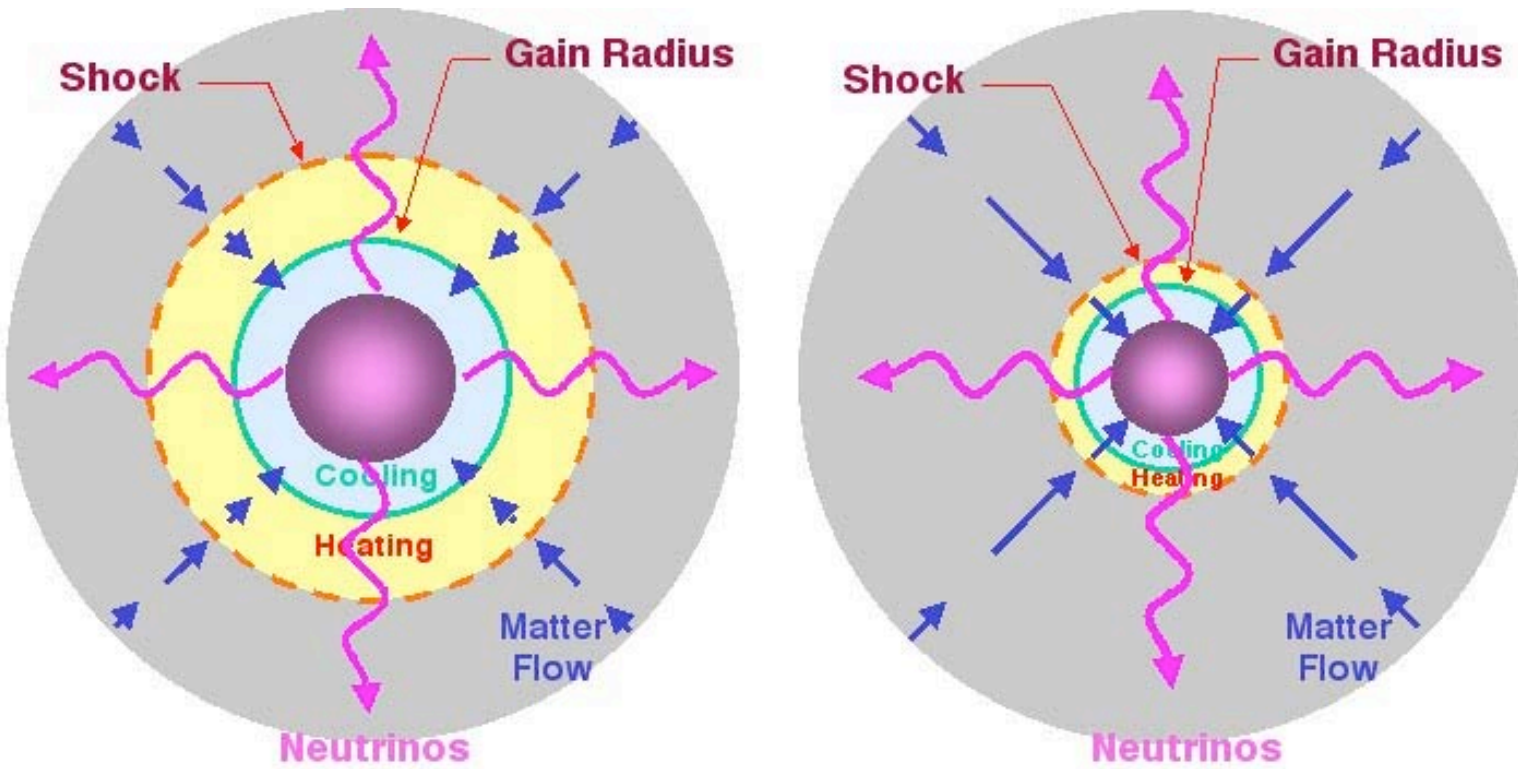
SASI Visualized with EnSight



Is a Newtonian model sufficient?

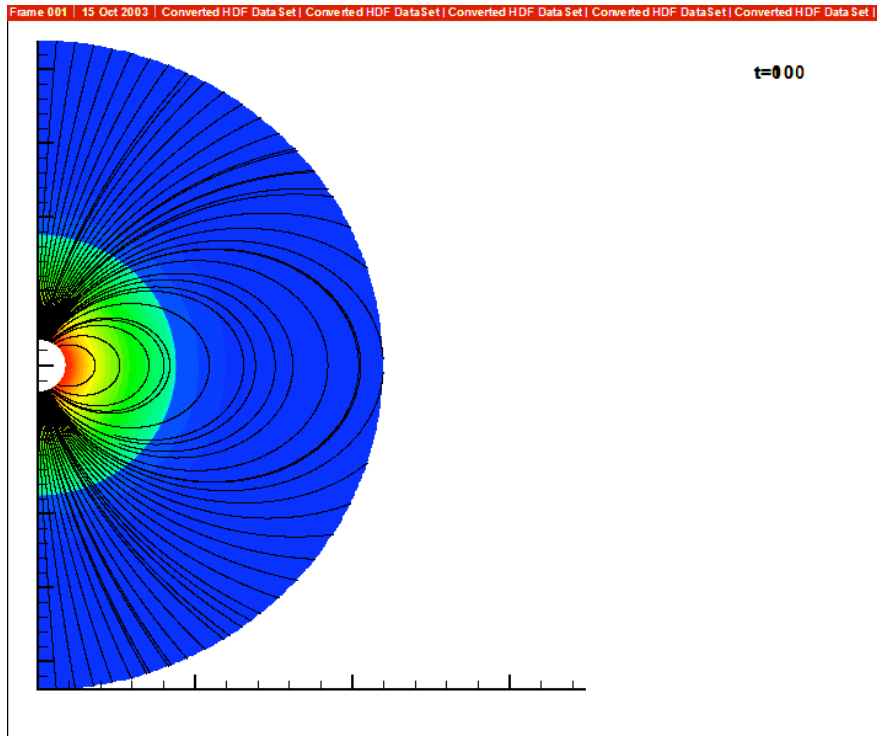


A comparison of key radii in a Newtonian versus a general relativistic model:



Bruenn, DeNisco, and Mezzacappa (2001)

And what about rotation and magnetic fields?



ud-Doula, Blondin, and Mezzacappa (2004)

If the fields are amplified sufficiently, the MHD luminosity may be sufficient to drive an explosion.

⇒ Akiyama et al. (2004)

* **Whether core collapse supernovae are neutrino powered, MHD powered, or both...both the neutrinos and the magnetic fields will have a significant impact on the supernova dynamics.**



Components of a Supernova Model

1. **Accurate** (Boltzmann) Neutrino Transport
2. Turbulent, Rotating Stellar Core Flow
3. Stellar Core Magnetic Fields
4. Gravity (Einsteinian)
5. Nuclear (Stellar Core) and Weak Interaction (Neutrino) Physics

What are we up against?



Dominant Computation:

Nonlinear, integro-partial differential equations for the radiation distribution functions.

Spherical Symmetry	$f(r, \mu, E)$	$R(r, \mu, E, \mu', E')$
Axisymmetry	$f(r, \theta, \mu_1, \mu_2, E)$	$R(r, \theta, \mu_1, \mu_2, E, \mu_1', \mu_2', E')$
No Symmetry	$f(x, y, z, \mu_1, \mu_2, E)$	$R(x, y, z, \mu_1, \mu_2, E, \mu_1', \mu_2', E')$

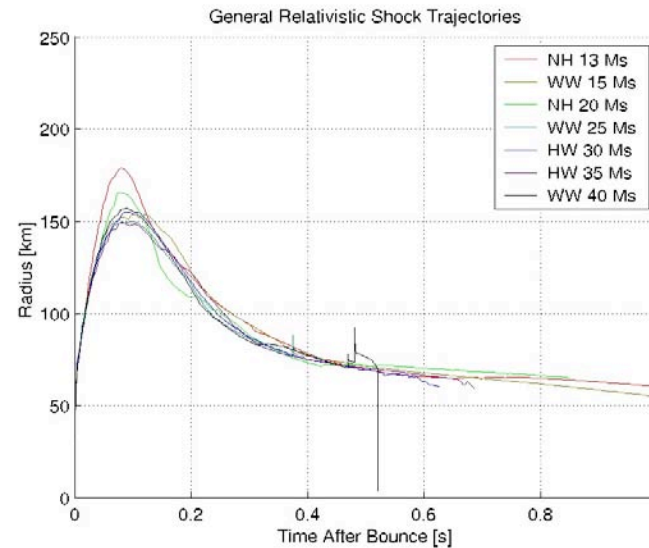
**Boltzmann
Equation in
Spherical
Symmetry**

$$\begin{aligned}
 & \frac{1}{c} \frac{\partial F}{\partial t} + 4\pi\mu_0 \frac{\partial(r^2 \rho_0 F)}{\partial r} \\
 & + \frac{1}{r} \frac{\partial[(1 - \mu_0^2)F]}{\partial \mu_0} \\
 & + \frac{1}{c} \left(\frac{\partial \ln \rho_0}{\partial t} + \frac{3v}{r} \right) \frac{\partial[\mu_0(1 - \mu_0^2)F]}{\partial \mu_0} \\
 & + \frac{1}{c} \left[\mu_0^2 \left(\frac{\partial \ln \rho_0}{\partial t} + \frac{3v}{r} \right) - \frac{v}{r} \frac{1}{E_0^2} \frac{\partial(E_0^3 F)}{\partial E_0} \right] \\
 & = \frac{j}{\rho_0} - \tilde{\chi} F \\
 & + \frac{1}{c h^3 c^3} E_0^2 \int d\mu'_0 R_{IS}(\mu_0, \mu'_0, E_0) F(\mu'_0, E_0) \\
 & - \frac{1}{c h^3 c^3} E_0^2 F \int d\mu'_0 R_{IS}(\mu_0, \mu'_0, E_0) \\
 & + \frac{1}{h^3 c^4} \left(\frac{1}{\rho_0} - F(\mu_0, E_0) \right) \int dE'_0 E_0'^2 d\mu'_0 \tilde{R}_{NES}^m(\mu_0, \mu'_0, E_0, E'_0) F(\mu'_0, E'_0) \\
 & - \frac{1}{h^3 c^4} F(\mu_0, E_0) \int dE'_0 E_0'^2 d\mu'_0 \tilde{R}_{NES}^{out}(\mu_0, \mu'_0, E_0, E'_0) \left(\frac{1}{\rho_0} - F(\mu'_0, E'_0) \right)
 \end{aligned}$$

Completed: Spherical Models with Boltzmann Transport

Newtonian

General Relativistic



Messer et al. (2002)

Liebendoerfer et al. (2002)

TSI will explore both!

No Explosions!

New Microphysics?

High-Density Stellar Core Thermodynamics

Neutrino-Matter Interactions

New Macrophysics? (2D/3D Models)

Fluid Instabilities, Rotation, Magnetic Fields



- TSI members first to perform 1D models with Boltzmann transport.
Mezzacappa and Bruenn (1993)
Mezzacappa et al. (2001)
- Progress on 2D (*and* 3D) Boltzmann transport progressing rapidly.

- ⇒ Development of formalism for
 - ⇒ **conservative** (energy *and* lepton number)
 - ⇒ general relativistic neutrino transport (*analytical tour de force*).
Cardall and Mezzacappa (2003)

Without this, supernova simulations difficult to interpret.

- ⇒ Development of finite differencing.
- ⇒ Construction of GenASiS.
- ⇒ Completion of test problems.
- ⇒ Initiation of realistic 2D supernova studies.

What makes neutrino transport difficult?

1. Difficult to develop number- *and* energy-conservative differencing for these “observer corrections (aberration, frequency shift).”
2. Difficult to handle “advection” terms when neutrinos and matter are in equilibrium.
3. Memory and CPU requirements.

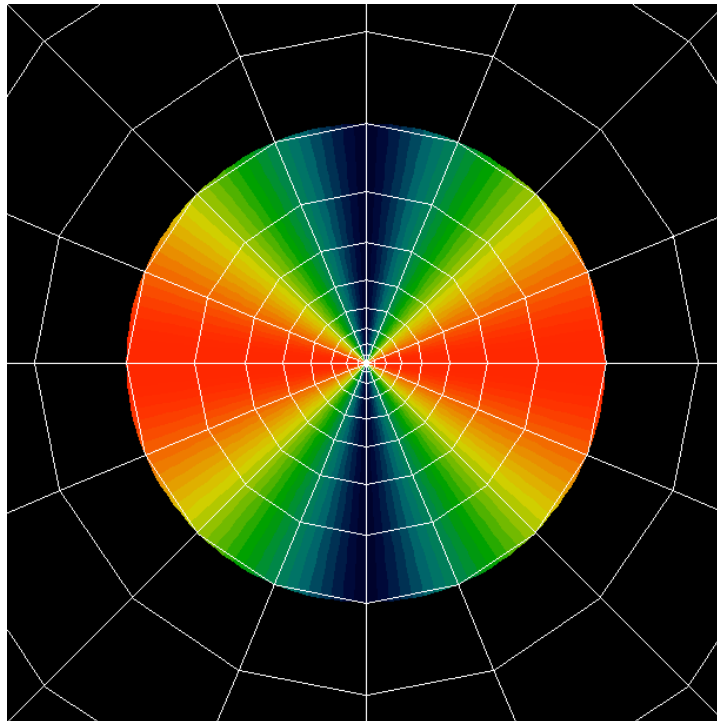
$$\begin{aligned}
 & \frac{1}{c} \frac{\partial F}{\partial t} + 4\pi \mu_0 \frac{\partial(r^2 \rho_0 F)}{\partial m} \\
 & + \frac{1}{r} \frac{\partial[(1 - \mu_0^2) F]}{\partial \mu_0} \\
 & + \frac{1}{c} \left(\frac{\partial \ln \rho_0}{\partial t} + \frac{3v}{r} \right) \frac{\partial[\mu_0(1 - \mu_0^2) F]}{\partial \mu_0} \\
 & + \frac{1}{c} \left[\mu_0^2 \left(\frac{\partial \ln \rho_0}{\partial t} + \frac{3v}{r} \right) - \frac{v}{r} \frac{1}{E_0^2} \frac{\partial(E_0^3 F)}{\partial E_0} \right] \\
 & = \frac{j}{\rho_0} - \tilde{\chi} F \\
 & + \frac{1}{c} \frac{1}{h^3 c^3} E_0^2 \int d\mu'_0 R_{IS}(\mu_0, \mu'_0, E_0) F(\mu'_0, E_0) \\
 & - \frac{1}{c} \frac{1}{h^3 c^3} E_0^2 F \int d\mu'_0 R_{IS}(\mu_0, \mu'_0, E_0) \\
 & + \frac{1}{h^3 c^4} \left(\frac{1}{\rho_0} - F(\mu_0, E_0) \right) \int dE'_0 E_0'^2 d\mu'_0 \tilde{R}_{NES}^{\text{in}}(\mu_0, \mu'_0, E_0, E'_0) F(\mu'_0, E'_0) \\
 & - \frac{1}{h^3 c^4} F(\mu_0, E_0) \int dE'_0 E_0'^2 d\mu'_0 \tilde{R}_{NES}^{\text{out}}(\mu_0, \mu'_0, E_0, E'_0) \left(\frac{1}{\rho_0} - F(\mu'_0, E'_0) \right)
 \end{aligned}$$

2D Boltzmann Neutrino Transport Test Problem

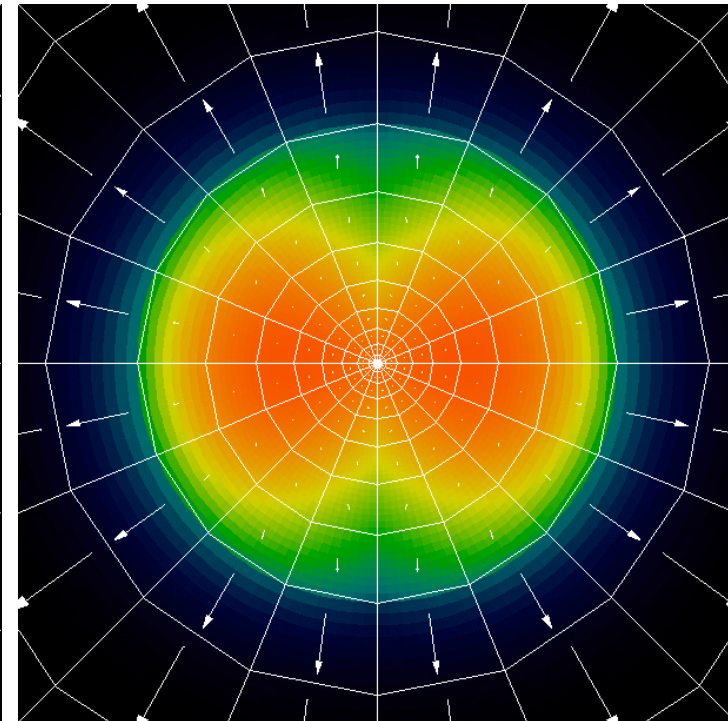


Development of radiation field stationary state in nonspherical fixed medium:

Density Distribution



Radiation Field Energy Density and Flux





WHAT IS MULTIGROUP FLUX-LIMITED DIFFUSION?

$$\frac{1}{2} \int d\mu_0 \text{ (BOLTZMANN EQUATION)} \Rightarrow \text{EQUATION FOR } \psi^{(0)}$$

$$\frac{1}{2} \int d\mu_0 \mu_0 \text{ (BOLTZMANN EQUATION)} \Rightarrow \text{EQUATION FOR } \psi^{(1)}$$

$$\psi^{(0)} \equiv \frac{1}{2} \int d\mu_0 f$$

$$\psi^{(1)} \equiv \frac{1}{2} \int d\mu_0 \mu_0 f$$

FLUX-LIMITED DIFFUSION APPROXIMATION

$$\frac{\partial \psi^{(1)}}{\partial t} \equiv 0$$

DROP VELOCITY DEPENDENT TERMS IN $\psi^{(1)}$ EQUATION

$$\Rightarrow \psi^{(1)} = -A^{(1)} \left(\frac{\partial \psi^{(0)}}{\partial r} + \dots \right)$$

WHERE

$$A^{(1)} \equiv 1 / \left(\frac{3}{\lambda^{(1)}} - \frac{1}{\psi^{(0)}} \frac{\partial \psi^{(0)}}{\partial r} \right)$$

Solve for first moment of neutrino distribution (truncation of 2N-1 moments obtained with Boltzmann solution).

2D MGFLD Equations

Advection Terms

Observer Corrections

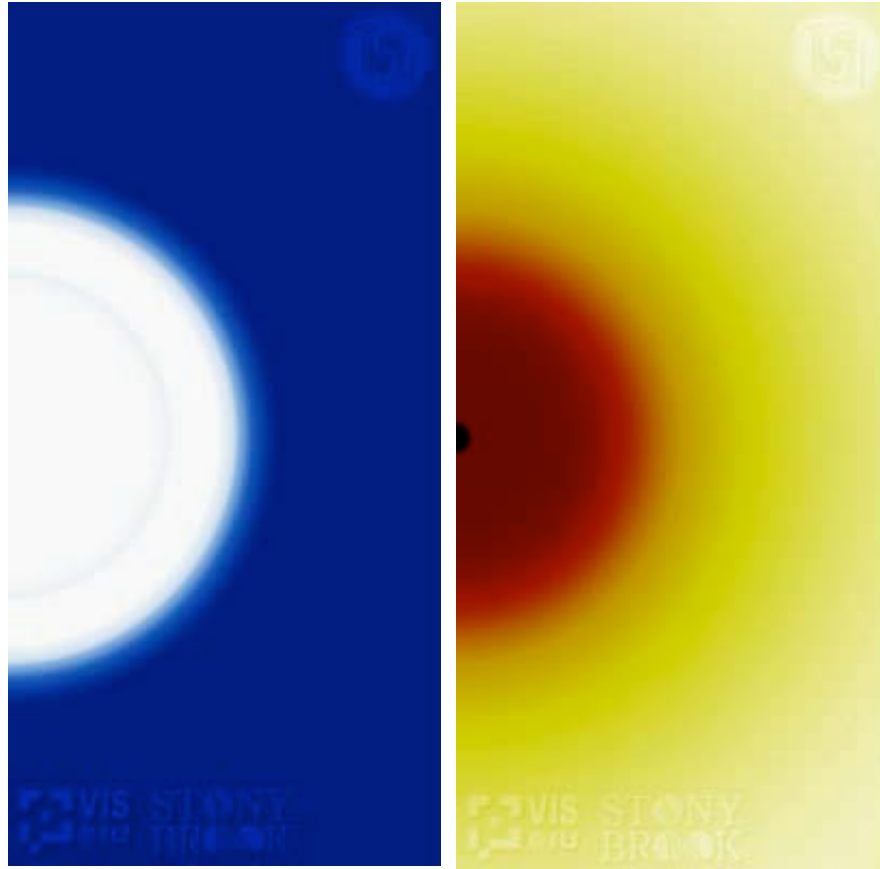
$$\left\{ \begin{aligned} & \frac{\partial E_\epsilon}{\partial t} + \nabla \cdot (E_\epsilon \mathbf{v}) + \nabla \cdot \mathbf{F}_\epsilon - \epsilon \frac{\partial}{\partial \epsilon} (P_\epsilon) : \nabla \mathbf{v} = S_\epsilon \\ & \frac{\partial \bar{E}_\epsilon}{\partial t} + \nabla \cdot (\bar{E}_\epsilon \mathbf{v}) + \nabla \cdot \bar{\mathbf{F}}_\epsilon - \epsilon \frac{\partial}{\partial \epsilon} (\bar{P}_\epsilon) : \nabla \mathbf{v} = \bar{S}_\epsilon \end{aligned} \right.$$

First 2D simulations with multifrequency neutrino transport, advection terms, and observer corrections:



Scientific Target: Development of Proto-Neutron Star Instabilities

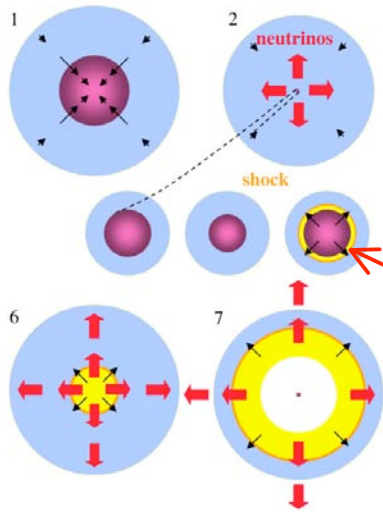
- ⇒ Close coupling of matter and neutrinos requires fully 2D transport for an accurate assessment.
- ⇒ What impact do the neutrinos have on the development of these instabilities?



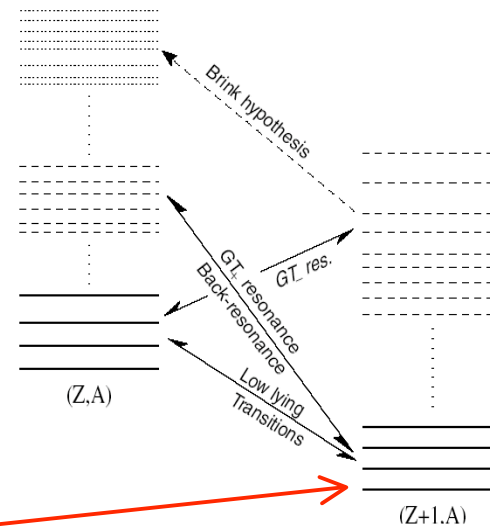
- * Running on 1024 processors at NERSC.
- * Scaling now to 2048.
- * Fully implicit solve.

Swesty and Myra (2004)

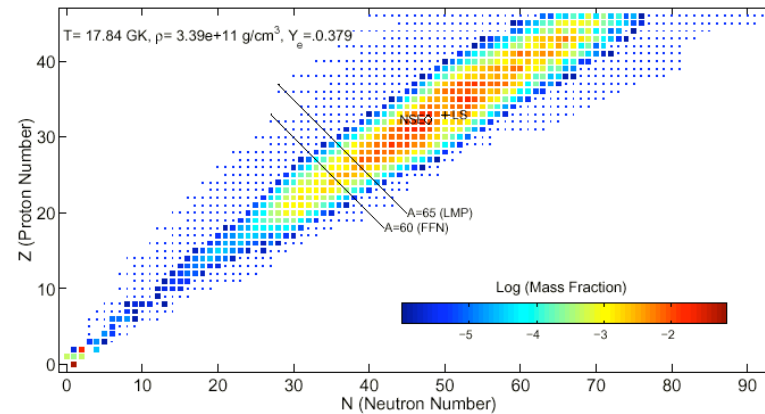
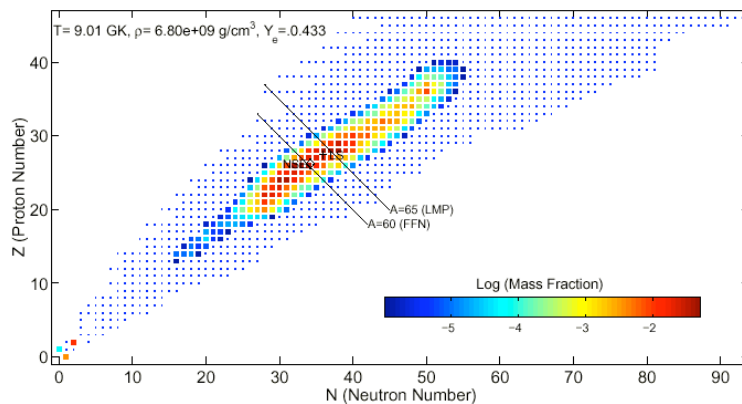
Core Collapse and Explosion



Initial shock location/strength depend on knowledge of nuclear states and their occupation during core collapse.



This is a challenge in nuclear computation being addressed by TSI's nuclear theorists.



This challenge is exacerbated by the fact that nuclei increase in size (neutron and proton number) /complexity (population of states, collective excitations) during collapse.

Computational Astrophysics:

(1.) Cosmological N-body simulations;

(2.) Core-Collapse Supernovae

Michael S. Warren Chris Fryer
msw@lanl.gov fryer@lanl.gov

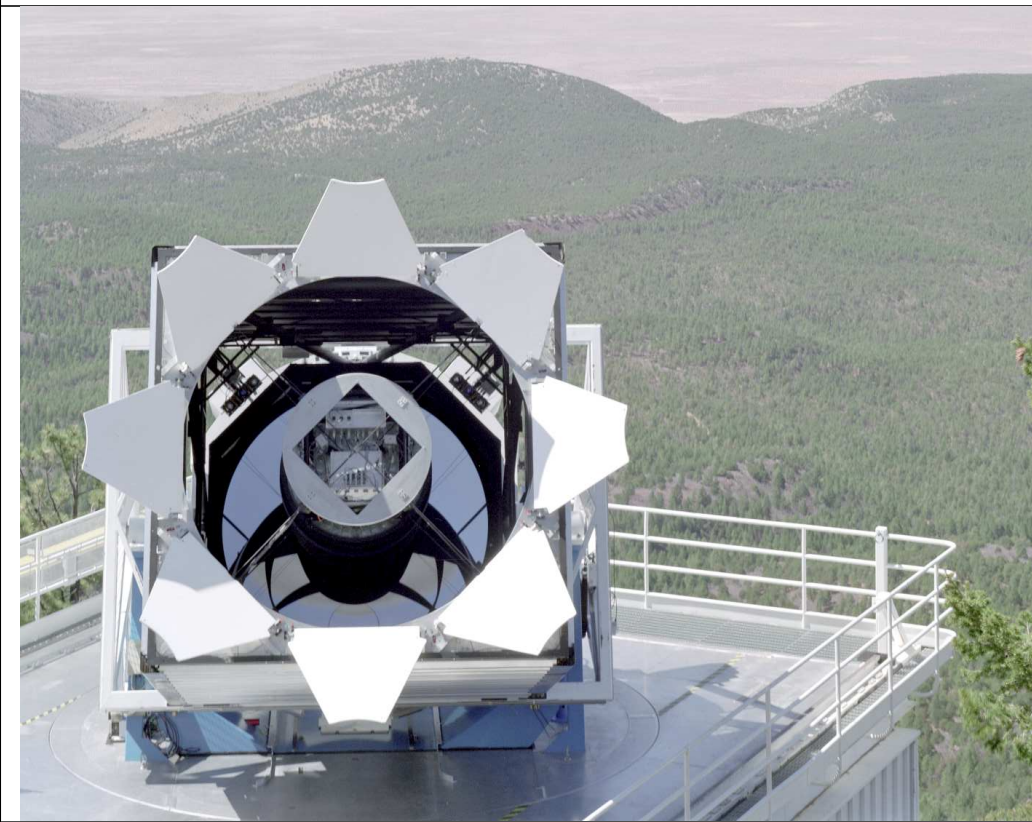
<http://t6-www.lanl.gov/msw/qb>
<http://space-simulator.lanl.gov/>



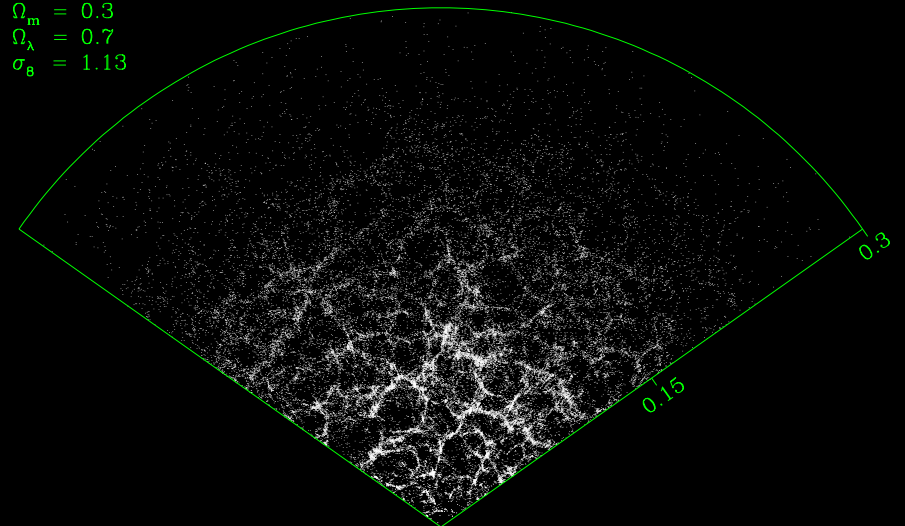
Astrophysics is now data-driven

The density of CCD pixels in the world's major telescopes has followed a Moore's Law type scaling for the past 15 years, resulting in an increase in data volume and quality which has paralleled the development of supercomputing.

The SDSS telescope in southern New Mexico (shown below) has mapped the positions of nearly 1 million galaxies. We are currently unable to simulate an equivalent volume of the Universe with sufficiently high precision to make valid tests of the current standard model of cosmology.



$$\begin{aligned}\Omega_m &= 0.3 \\ \Omega_\Lambda &= 0.7 \\ \sigma_8 &= 1.13\end{aligned}$$



Programmatic Impact

- The Nature of Dark Energy
- Probing Fundamental Physics at Extreme Energies and Scales
- Precision Cosmology — Testing the Standard Model

5

Major Scientific Challenges

- Cosmology – Dark Energy / Dark Matter / Resolving galaxies
- Core-collapse Supernovae / Gamma Ray Bursts – Nuclear physics, Neutrino physics
- Accretions Disks – Black Holes / Planet Formation

6

Methods

- Treecodes - scale as $N \log N$
- Fourier based - scales as $N \log N$
- Direct - scale as N^2 , has very limited utility
- Hybrid - AMR $N \log N$

7

Why Petaflops?

For precision cosmology, the volume simulated must be large enough to keep the largest modes in the box well within the linear regime. The particle mass and force resolution must also be small enough to accurately simulate the dynamics on galactic scales. With a box size of 500 Megaparsecs and 100 billion particles, the particle mass would be 6×10^7 solar masses, which would accurately resolve galaxies down to a mass of about 10^{10} solar masses, which is sufficient to model the galaxy catalogs which will exist in 5-10 years.

For core-collapse supernova, the models with accurate neutrino transport physics are still only 2-d. We currently have the only existing 3-d code, and improving the transport method in that code will add at least a factor of 10 to the operation count. Improving the resolution by a factor of 3 in space and time in order to fully resolve the convective region will add another factor of 100.

8

The Space Simulator

Qty.	Price	Ext.	Description
294	280	82,320	Shuttle SS51G mini system (bare)
294	254	74,676	Intel P4/2.53GHz, 533MHz FSB, 512k cache
588	118	69,384	512Mb DDR333 SDRAM (1024Mb per node)
294	95	27,930	3com 3c996B-T Gigabit Ethernet PCI card
294	83	24,402	Maxtor 4K080H4 80Gb 5400rpm Hard Disk
294	35	10,290	Assembly Labor/Extended Warranty
		4,000	Cat6 Ethernet cables
		3,300	Wire shelving/switch rack
		1,378	Power strips
1		186,175	Foundry FastIron 1500+800, 304 Gigabit ports
Total		\$483,855	\$1646 per node 5.06 Gflops peak per node

Space Simulator architecture and price (September, 2002).

9



Typical Beowulf cluster run, Fall 2003

- Started Nov. 14 at 23:51, ended Nov. 17 at 00:57 (48 hours, 6 minutes = 176760 seconds).
- Generated 512^3 initial conditions, ran 735 timesteps, identified galaxy halos.
- Stored 50 Gbytes of data.
- Evolution required 1.3×10^{17} flops.
- Averaged 746 Gflops (845 if you count redundant work due to hardware failure).

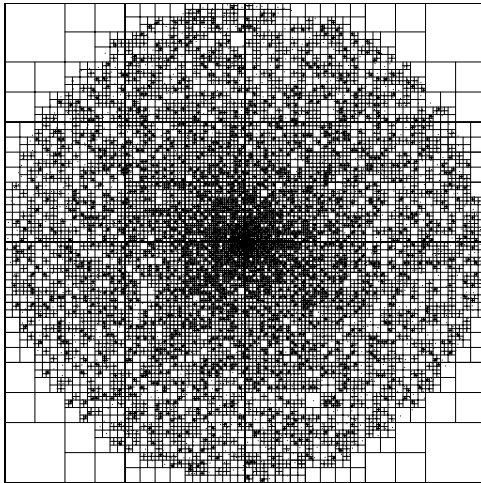
11

QB Simulations, January 2003

- Nearly 10^{18} flops
- Over 100 cosmological models
- Largest complete runs, 1.4 billion particles
- Largest partial run, 2.04 billion particles

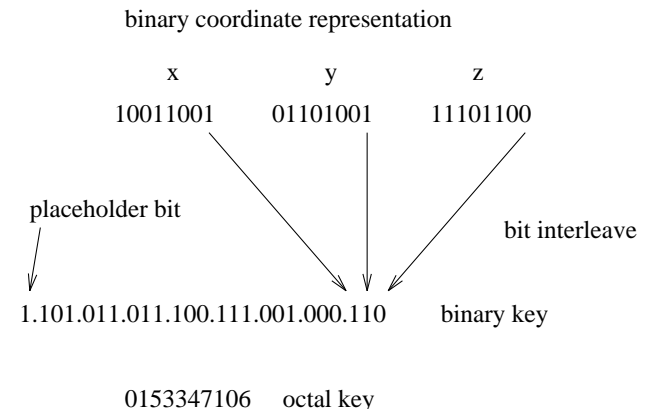
12

Tree Data Structures



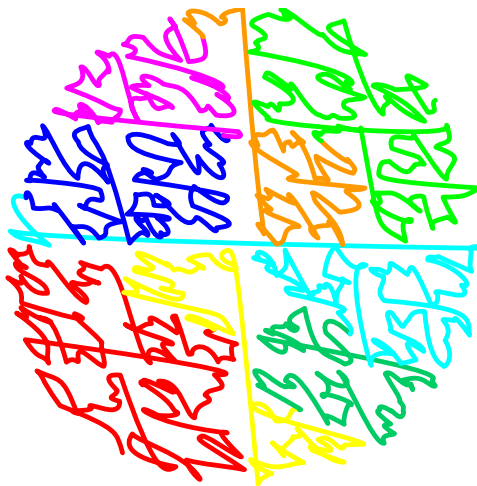
13

Bit interleaving



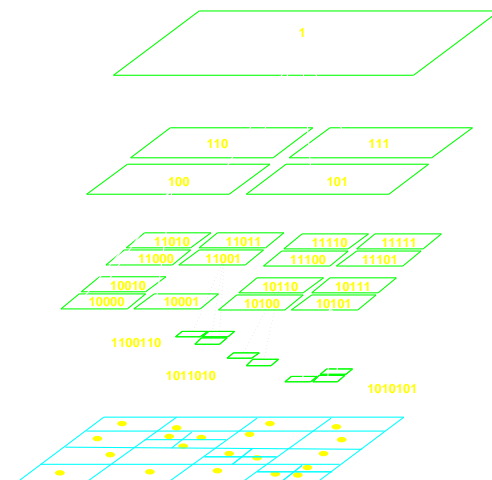
14

Domain Decomposition



15

Topology



16

Gravity Kernel Performance

Processor	libm	Karp
533-MHz Alpha EV56	76.2	242.2
667-MHz Transmeta TM5600	128.7	297.5
933-MHz Transmeta TM5800	189.5	373.2
375-MHz IBM Power3	298.5	514.4
1133-MHz Intel P3	292.2	594.9
1200-MHz AMD Athlon MP	350.7	614.0
2530-MHz Intel P4	779.3	792.6
1800-MHz AMD Athlon XP	609.9	951.9
1250-MHz Alpha 21264C	935.2	1141.0
2530-MHz Intel P4 (icc)	1170.0	1357.0
2530-MHz Intel P4 (SSE)	6514.0	

Table 1: Mflop/s obtained on our gravitational micro-kernel benchmark.

17

Historical Performance of the Treecode

Year	Site	Machine	Procs	Gflop/s	Mflops/proc
2003	LANL	Space Simulator (SSE)	288	1166	4050.0
2003	LANL	ASCI QB	3600	2793	775.8
2003	LANL	Space Simulator	288	179.7	623.9
2002	NERSC	IBM SP-3(375/W)	256	57.70	225.0
2002	LANL	Green Destiny	212	38.9	183.5
2000	LANL	SGI Origin 2000	64	13.10	205.0
1998	LANL	Avalon	128	16.16	126.0
1996	LANL	Loki	16	1.28	80.0
1996	SC '96	Loki+Hyglac	32	2.19	68.4
1996	Sandia	ASCI Red	6800	464.9	68.4
1995	JPL	Cray T3D	256	7.94	31.0
1995	LANL	TMC CM-5	512	14.06	27.5
1993	Caltech	Intel Delta	512	10.02	19.6

18

QB Performance (2 billion particles, 512 procs)

<i>computation stage</i>	<i>time per timestep (sec)</i>
Domain Decomposition	6.9
Tree Build	78
Tree Traversal	186
Data Communication During Traversal	57
Force Evaluation	1053
Load Imbalance	181
Total (453 Gflops)	1562

19

Moore's Law Applied to Beowulf Clusters

It is interesting to note there have been exactly six years between the completion of the Loki and Space Simulator clusters, which results in four "Moore's Law" doublings. Comparing the Loki architecture and price to the Space Simulator, we can see that Moore's Law scaling has actually been greatly exceeded in some aspects of the architecture.

For instance, in 1996, Loki's disks cost \$111 per Gigabyte. For the SS, they are close to \$1 a Gigabyte, which is a factor of seven beyond the factor of 16 from Moore's Law over six years. For memory, in the Loki days it was \$7.35 per Megabyte, and is now 23 cents per Megabyte, 2x lower than Moore's Law would have predicted.

20

Moore's Law Applied to Beowulf Clusters

These factors of improvement over Moore's Law are also realized in the NPB performance results. For a given cost, the NPB performance exceeds Moore's Law scaling by 25% for BT, and close to a factor of two for LU and MG.

For the N-body code, Loki obtained performance of 1.28 Gflop/s for the N-body code, while the whole SS obtains 1160 Gflops, an improvement of a factor of 900. The price ratio between the machines is 9.4, which when multiplied by 16 for four 18-month Moore's Law doublings, results in a ratio of 150. Thus, N-body performance has exceeded Moore's Law by a factor of 4 over the past 6 years.

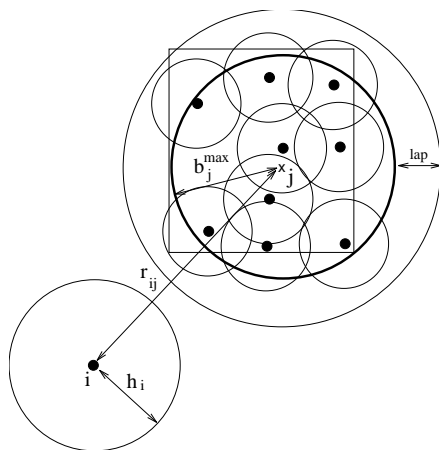
21

Smoothed Particle Hydrodynamics

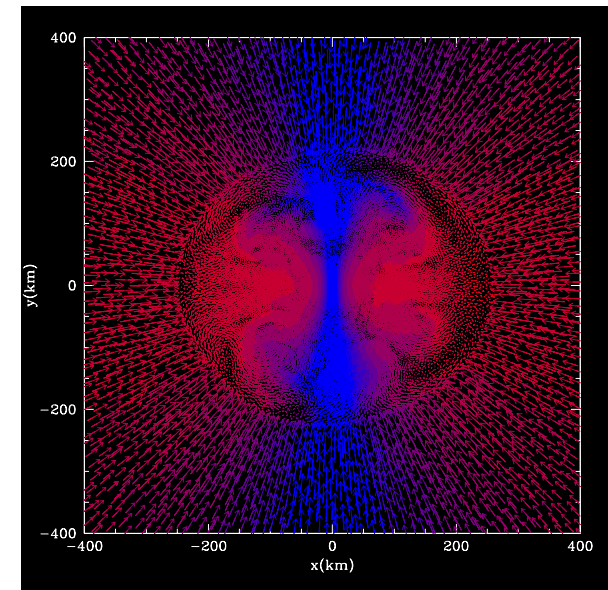
- Supernova simulations use the smoothed particle hydrodynamics (SPH) method, invented by Lucy and Gingold & Monaghan in the mid-70s.
- SPH uses particles to carry hydrodynamical quantities like mass and energy, instead of using a computational grid.
- Additional routines to calculate neutrino transport and equation of state information were added by Benz, Herant and Fryer for the 2-d code in the mid-90s.
- Modifications to the code to support parallel computers and three dimensions have been underway since 1996.

22

Neighbor Finding



23



The Simulations

- First major 3-d run on Avalon, started May 1999
- Initial paper “Modeling Core-Collapse Supernovae in 3-Dimensions” has been accepted for publication by the Astrophysical Journal. Available at qso.lanl.gov/~clf
- Initial three runs of 300k, 1 million and 3 million particles completed Sep. 2001—Apr. 2002 at the National Energy Research Scientific Computing Center (NERSC) in Oakland, California on the IBM SP system.
- Runs of 1 million and 5 million particles completed on QB.

25

Petaflop Cluster Prediction

Qty.	Price	Ext.	Description
65536	280	18M	Enclosure/mainboard
65536	250	17M	Intel/AMD 64-bit Dual-core 5GHz, 2Mb cache
131072	120	15M	4Gb DDR1000 SDRAM (8Gb per node)
65536	200	13M	Infiniband 8-Gigabit NIC
131072	100	13M	500Gb 10000rpm Hard Disk
65536	35	2M	Assembly
		1M	cables
		1M	Rack infrastructure
		30M	Network switches
Total		\$110M	\$1700 per node 50.0 Gflops peak per node

Petaflop cluster architecture and price (projected late 2007).

26

Petascale Applications —

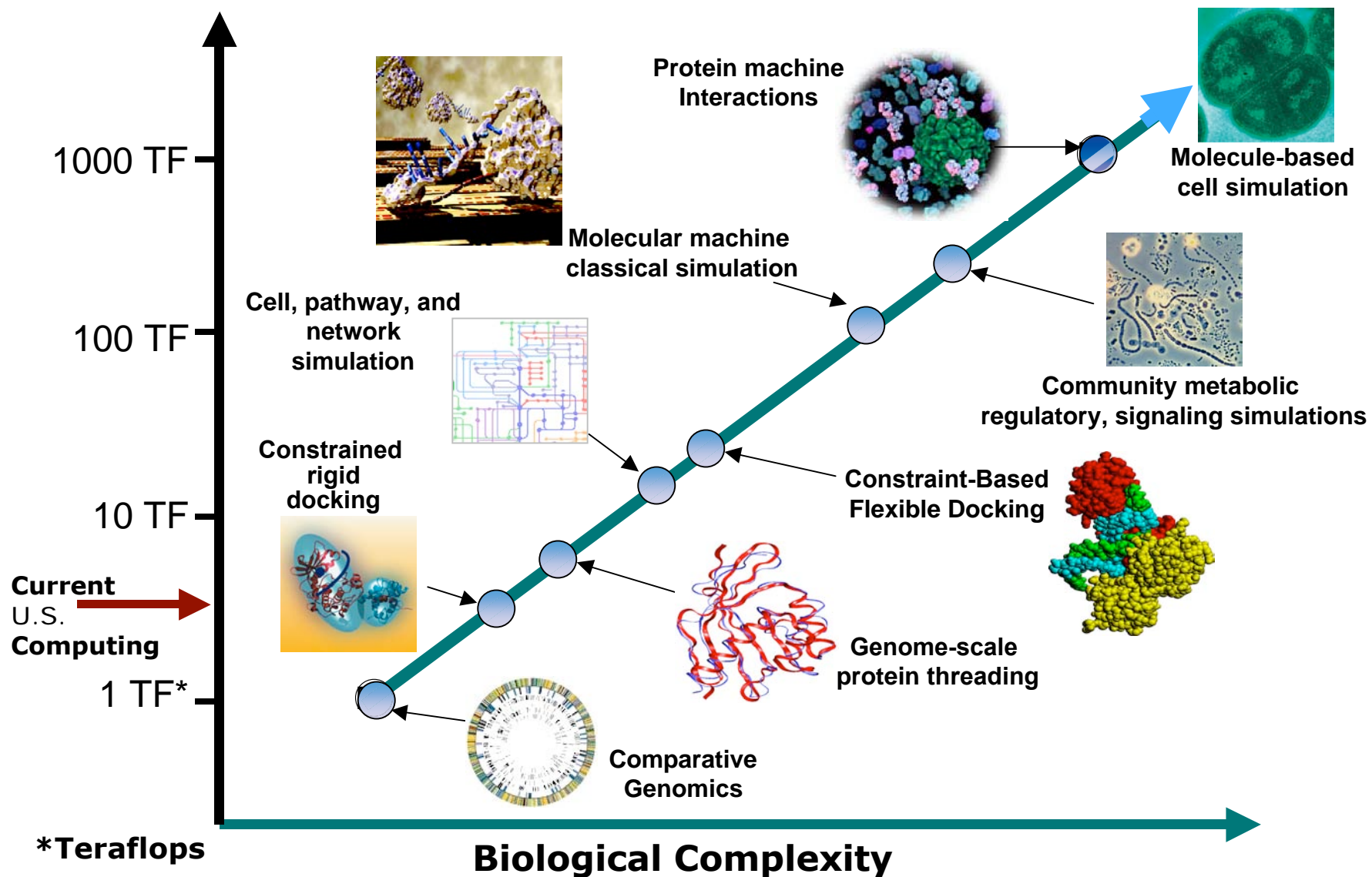
Biology

Impact of Petaflop-scale Computing: Application — Computational Biology

	Computational Biology
<p>Programmatic impact to be gained by access to Petaflop-scale computing</p>	<p>As noted in the cover email, achieving a factor of 100x - 1000x increase in capability over today's biggest machines would result in Petaflop-scale computing. By this we mean the the ability to obtain from a few tenths of a Petaflop/s peak speed to a few Petaflops/s peak speed in the not-too-distant future. The only reference to capability driven improvements in this range appears to be on page 8, where reference is made to 100-Teraflop/s-class computers which are at the lower end of the range of Petaflop-scale computing. We have made one entry in this box on the advancement obtained with 100-Teraflop/s-class computers, and would like you to add a few improvements that would be achievable with Petaflop-scale capability. The same applies to the next box on Scientific Challenges.</p> <ul style="list-style-type: none"> • "Major progress in the predictive power of classical MD and fpMD techniques." Would you please try to make this description more specific (for example -- this example is made up -- it would be possible to perform a ribosome simulation using xx million atoms and time steps of xxx femtosec in about one month using a 1 Petaflop/s (peak) computer) - thanks. <p>Please also include informatics advances here if they can be achieved with capability machines.</p>
<p>Major scientific challenges to be addressed</p>	<p>Referring to the box above, please add a few scientific challenges, corresponding to the programmatic impacts listed above, that will be met with Petaflop-scale computing.</p> <ul style="list-style-type: none"> • •
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p> <p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p>[We are emphasizing <i>capability</i> – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)]</p>	<p>These type of throughput data are not directly indicated in the report. However, the kind of answer we are looking for might be something like this: "We did calculations on a 5 TFlops/s (peak) machine, achieving sustained throughput of 0.5 Tflops/s (or 10% efficiency). The turn-around time is about days."</p>
<p>What is the <i>typical</i> number of processors used for your codes today? What is the <i>largest</i> number of processors used to-date?</p>	<p>These data are not mentioned in the Report. Credible architectures for achieving Petaflop-scale capability within 5 years will contain tens of thousands of processors. The ability for codes to run efficiently with large numbers of processors will be critical. Running today's codes with large numbers of processors can give useful insights into projected scaling behavior. Please provide us with your experience.</p>

<p>What is the Operations Count/Scaling from other computers?</p>	<p>These numbers do not appear in the Report. To scale performance from today's machines to larger capability machines requires either:</p> <ul style="list-style-type: none"> • An operations count, or • A scaling law based on current performance on current machines <p>If you use a scaling law to characterize Petaflop-scale performance, please provide the logic used (e.g. compute time scales as n^4, where n is a linear cell dimension), along with the current computer capability used in the scaling. In both cases please provide the required turn-around time for the longest runs.</p>
<p>Projected increase in software efficiency?</p>	<p>If you are counting on an increase from better algorithms (historically, algorithm improvements have approximately matched improvements in hardware), please indicate the factor you've used.</p>
<p>Other</p>	

GTL High-Performance Computing Roadmap

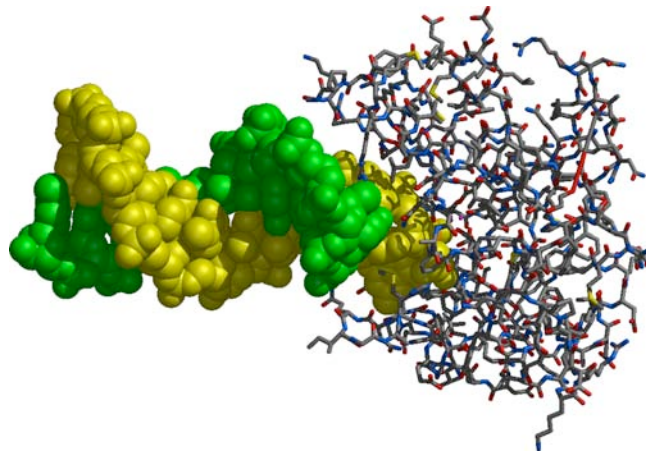
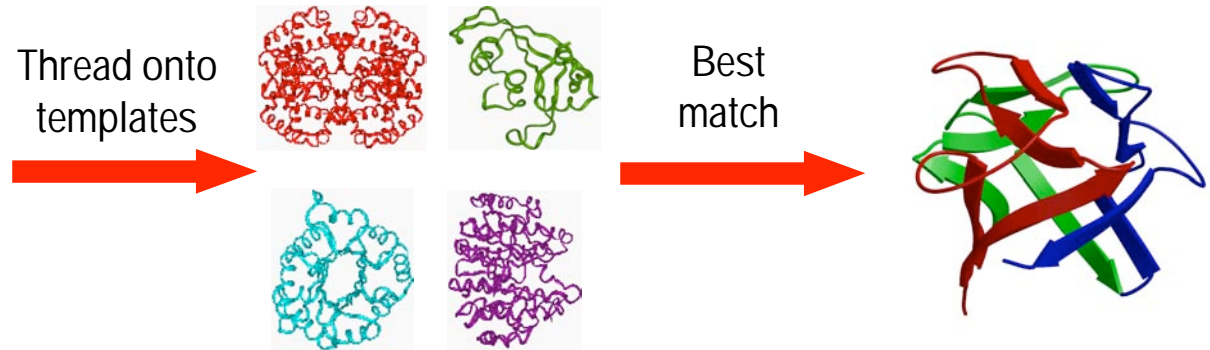


GTL facilities will Require High Performance Computing for Both *Capacity* and *Capability*

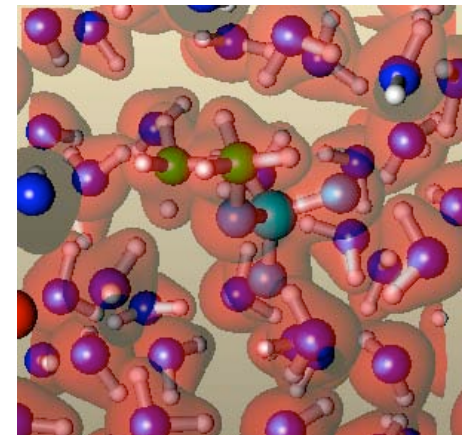
```
ATCGTAGCAATCGACCGT...
CGGCTATAGCCGTTACCG...
TTATGCTATCCATAATCGA...
GGCTTAATCGCATACGAC...
```

Capacity: e.g., High-throughput protein structure predictions, data analysis, sequence comparison

Capability: e.g., Large scale biophysical simulations, stochastic regulatory simulations:



Large size and timescale classical simulations



Highly accurate quantum mechanical simulations

Petascale *Capacity* Problems in Biology

Microbial and Community Genome Annotation

Now

Analyze and annotate 20 microbial genomes - (720,000 processor hours)

In 5 years

Assemble, Analyze and annotate community of 200 Microbes and phage (10,000,000 Processor hours)

Compare genome sequences (200 megabases)
To previous genomes (4 gigabases)
(5,000,000 processor hours)

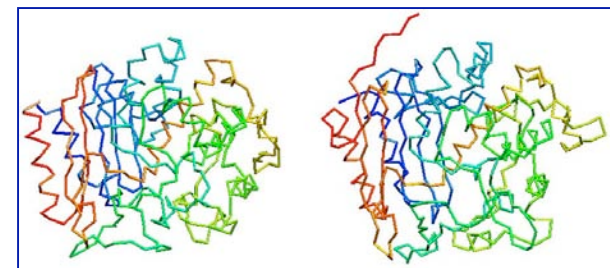


Petascale *Capacity* Problems in Biology

Protein Fold Prediction using Knowledge based potentials

Now

Protein fold prediction of 2000 proteins in a microbial genome using knowledge-based potentials - (100,000 processor hours)

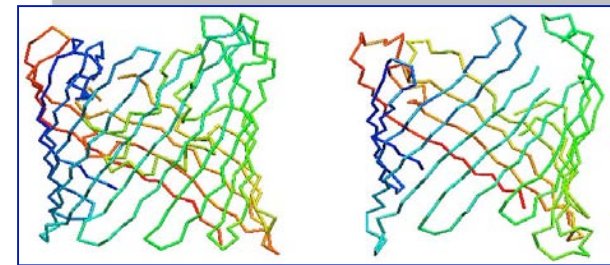


actual

Predicted

In 5 years

Protein fold prediction for 200 microbes (400,000 proteins) in a microbial community (20,000,000 processor hours)



Petascale Capability Problems in Biology

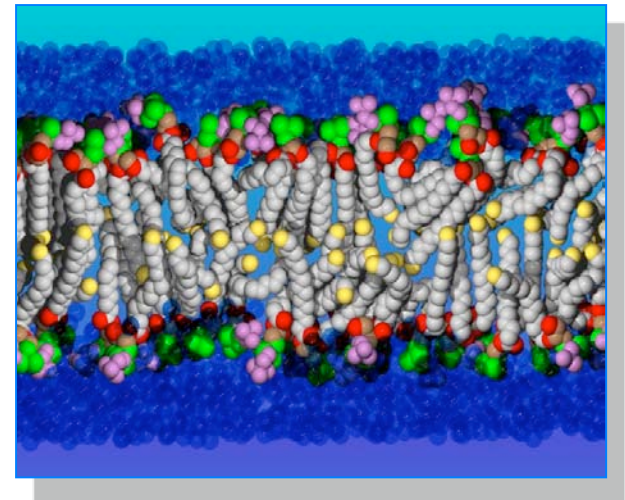
Membrane simulation using classical potentials

Now

Observe heterogeneous lipid segregation (patching) - (600,000 processor hours for 200 Nanosecond simulation)

In 5 years

Simulate membrane protein Association and lipid interactions (7,000,000 processor hours for 1 millisecond simulation)



Petascale Capability Problems in Biology

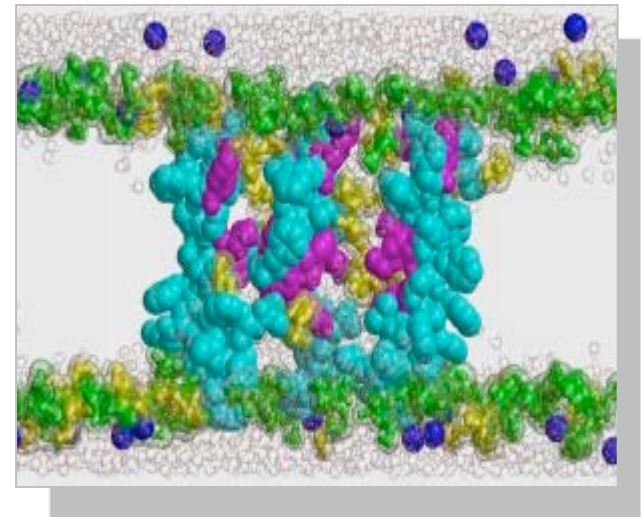
Membrane channel simulation

Now

Simulate non-flexible protein ion channel
K⁺ flow using quantum methods
(2,200,000) processor hours for
4 second simulation

In 5 years

Simulate flexible protein ion pump
for producing ATP from K⁺ gradient
(15,000,000 processor hours for 200
nanosecond simulation



Petascale Capability Problems in Biology

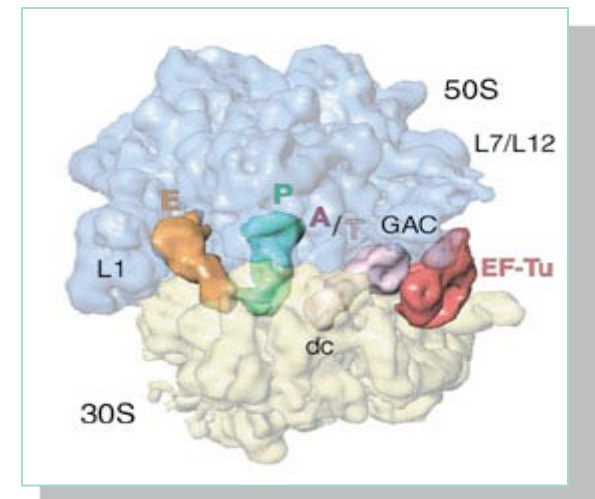
Ribosomal Interactions and Dynamics

Now

Simulate Ribosome EF-tu interaction
Using classical molecular dynamics
(400,000) processor hours for 20
Nanosecond simulation

In 5 years

Simulate an individual component step
of amino acid translation process in
the Ribosome (25,000,000 processor hours
for 1 millisecond simulation)



Petascale Capability Problems in Biology

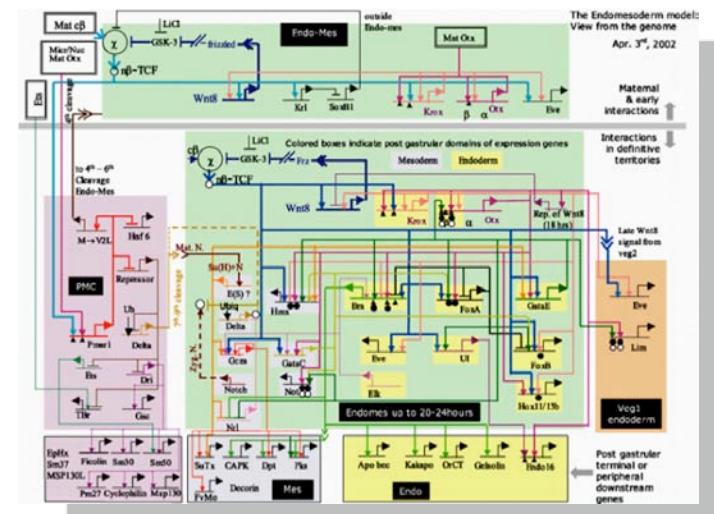
Regulatory and Protein Interaction Networks

Now

Simulate moderately complex network using ordinary differential equations (10,000) processor hours for 1 millisecond simulation

In 5 years

Multiscale stochastic and stochastic differential equation simulation of complex regulatory network (1,000,000 processor hours for 1 millisecond simulation)



Petascale Capability Problems in Biology

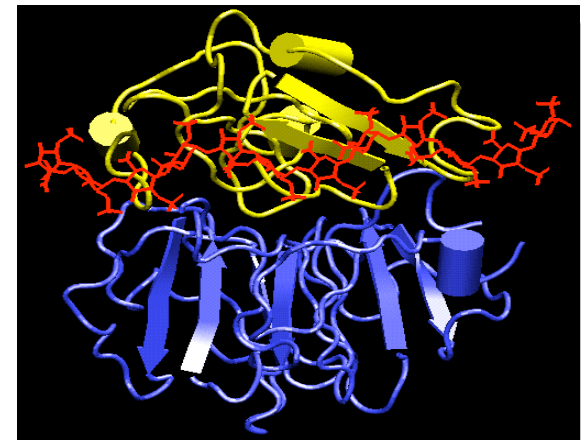
Molecular Machine Interactions

Now

Simulation of microbial gene regulatory factor binding (300,000 processor hours for 50 nanosecond simulation)



Rigid docking of multiple components of Protein complex and ligand (500,000 Processor hours for orientation search)

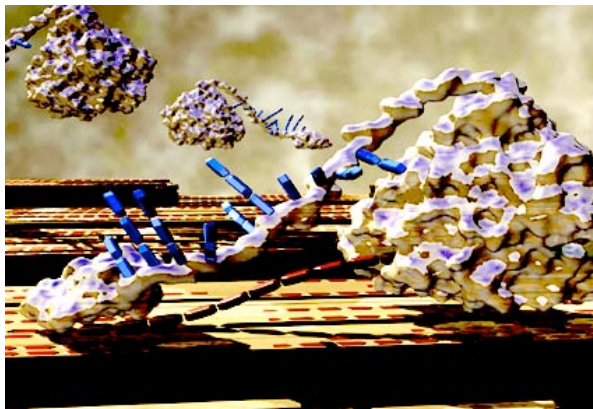


Petascale Capability Problems in Biology

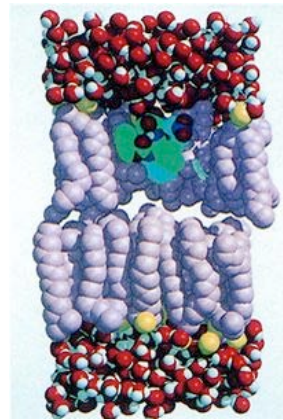
Molecular Machine Interactions and Dynamics

In 5-10 years

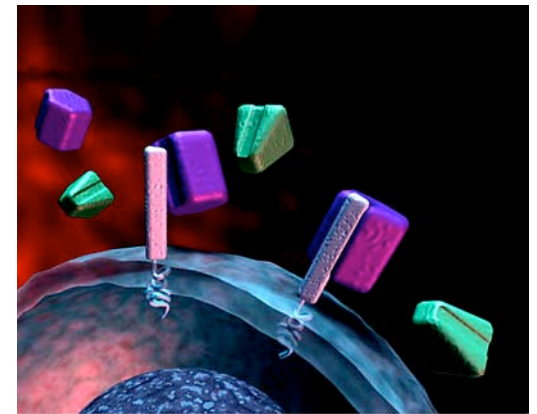
Simulate dynamics and chemistry of
Cellulase, cell surface receptors,
bionano structures



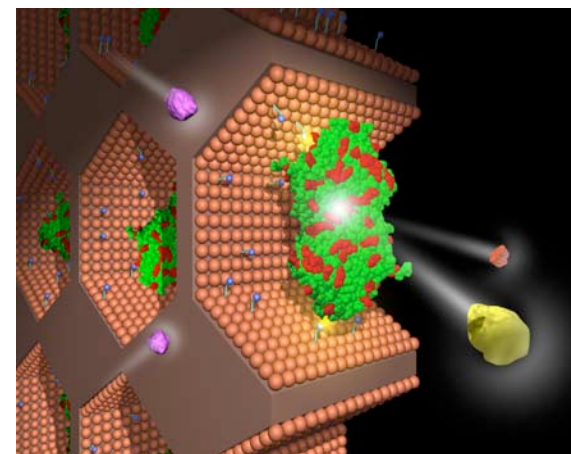
cellulase



Membrane
complexes

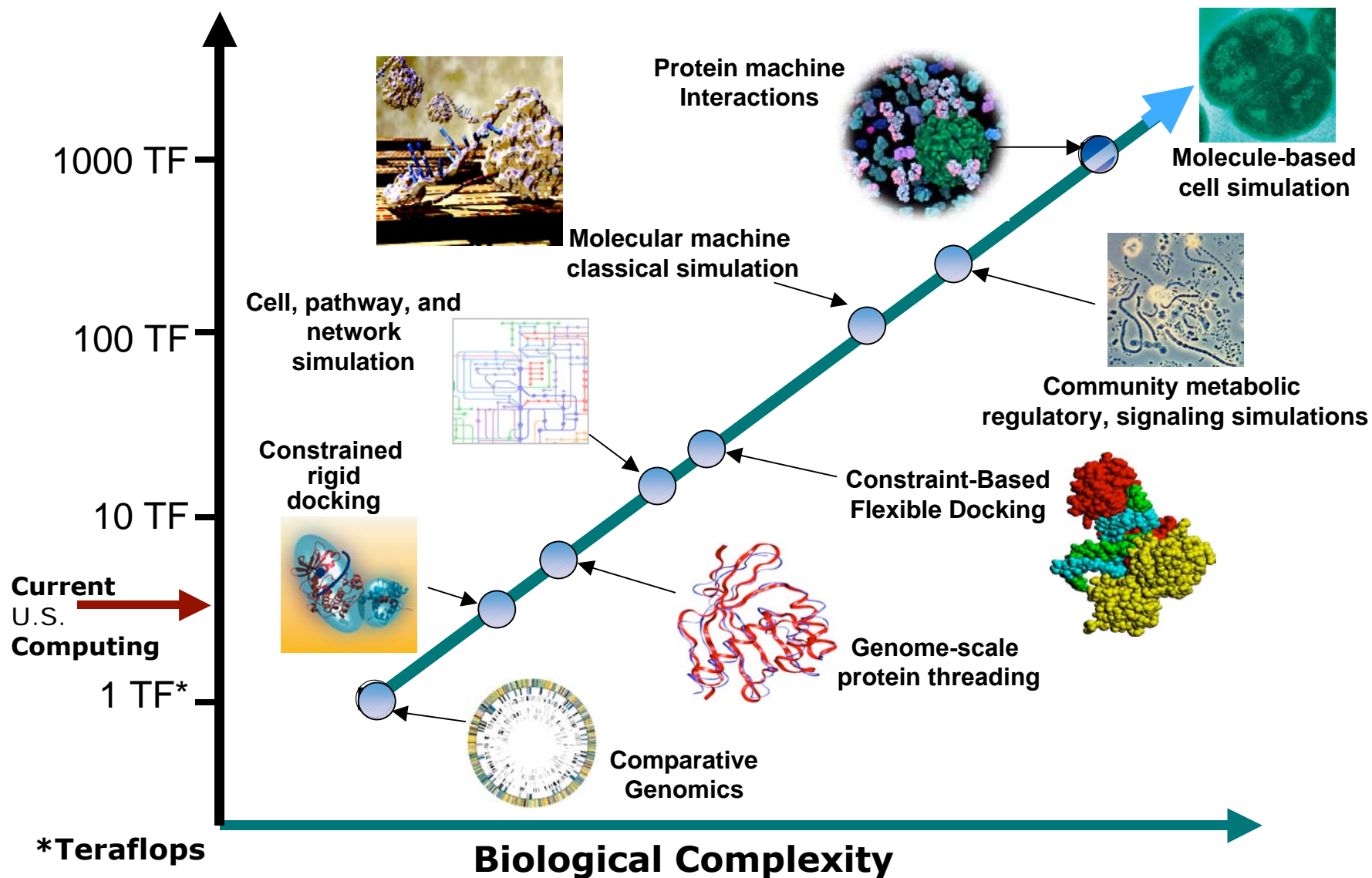


Cell-environment sensors



Bionano chemistries

GTL High-Performance Computing Roadmap



Petaflop Computing in Biology

LSD: Ed Uberbacher, Phyl LoCascio

CSMD: Pratul Agarwal, Al Geist, Andrey
Gorin, Nagiza Samatova

Oak Ridge National Laboratory

Mass Spectrometry Analysis

Ultrascale Challenge

Now

Analyze proteome of *Rhodopseudomonas palustris* 35,000 MS/MS spectra against 4000 proteins-.01 TF for a day

In 2-5 years

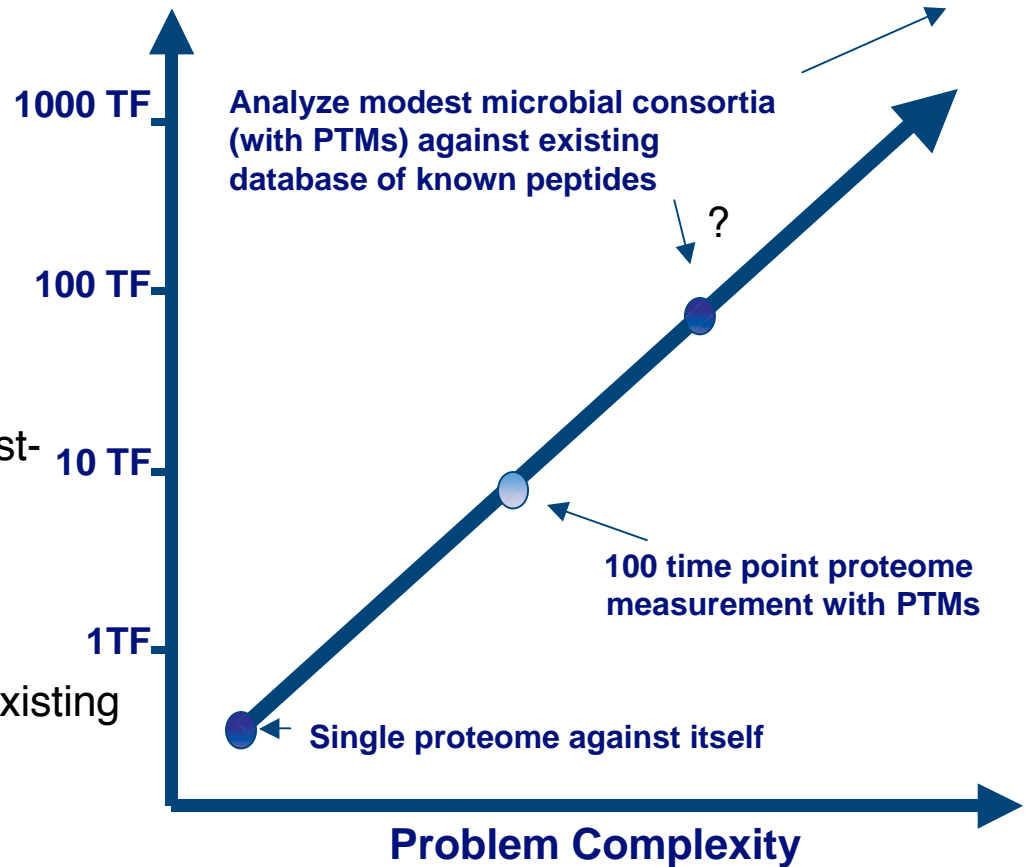
Measure 100 time points with protein post-translational modifications – 1000X

Examine microbial consortia with 100 proteomes (P) with Post-translation modifications (PTM). Compare against existing proteomes DB (4×10^6 proteins)

$100 \times 10 \times 1 \times 1000 = 1,000,000X$

$C \sim P \times PTM \times t \times DB$

Also need capability to move Petabyte datasets to computing site (collaboration with George Michaels)



□ Need significant algorithm improvements
Plus >100 TF shared memory capability

Molecular Interaction Image Analysis

Ultrascale Challenge

Now

Single event detection
(find face in picture)
Recognize FRET event
in a microbial cell- single processor
problem

In 5 years

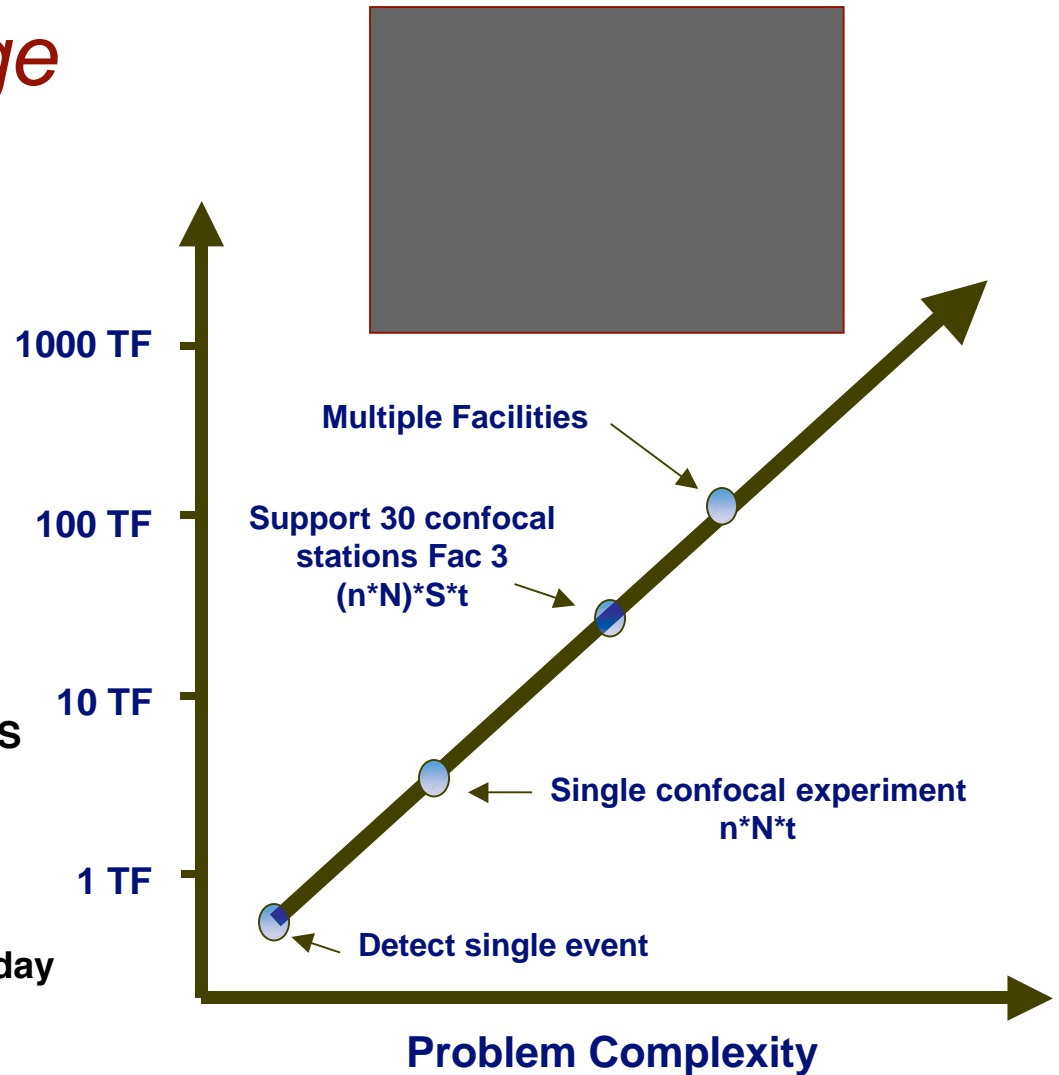
Track 100 proteins in microbe = n
Dynamic segment cell ultrastructure = N
Track for 100 time points = T
Interpret output of 30 confocal stations= S

$$C \sim (N \cdot n) \cdot S \cdot T$$

$(100 \times 100) \times 100 \times 30 = 3 \cdot 10^7$

Facility will generate 10,000 images per day
@ 4 megabyte each

Query and retrieval issues
Infrastructure for moving Image Datasets



- Need sustained 100 TF capacity
- Need significant algorithm development

Machine Docking and Dynamics

Docking

Now

Rigid docking of Protein complex with ligand

$C \sim n * (t + F)$ where

n is number of possible orientation

t is time to find a possible orientation

F is time to fit ligand

1 TF day

In 5 years

Multi-component docking

$C \sim (N-1)! * n * (t + F)$ for $N=6$

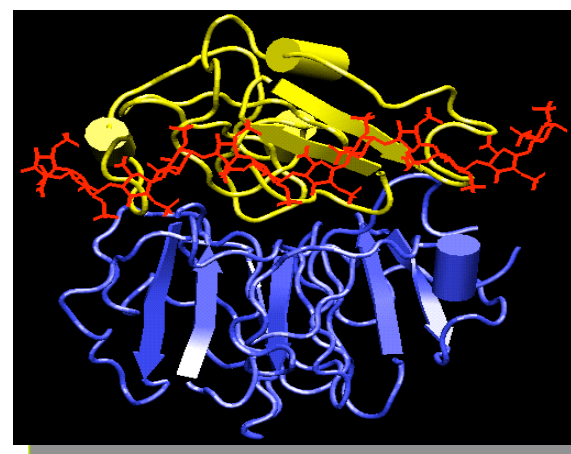
-> 120 TF day

GTL goal - do a microbe (>200 complexes) in year 1

Points to the need to use experimental constraints to

limit $(N-1)!$ and n terms

Constraints come from mass spectrometry and imaging



Atomistic Modeling of Proteins using MD

Protein/DNA complexes: structure, folding, dynamics
biochemical function & biomolecular recognition

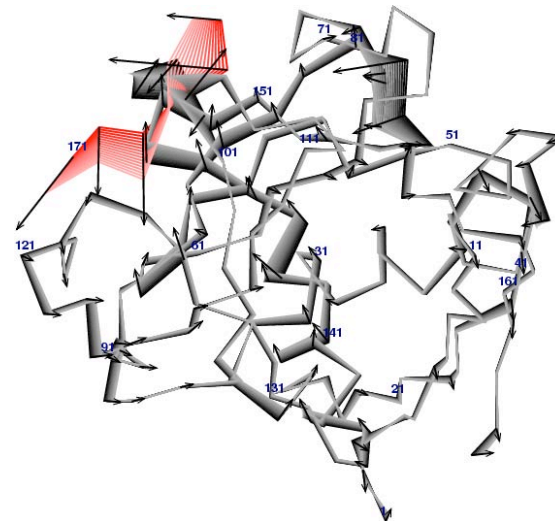
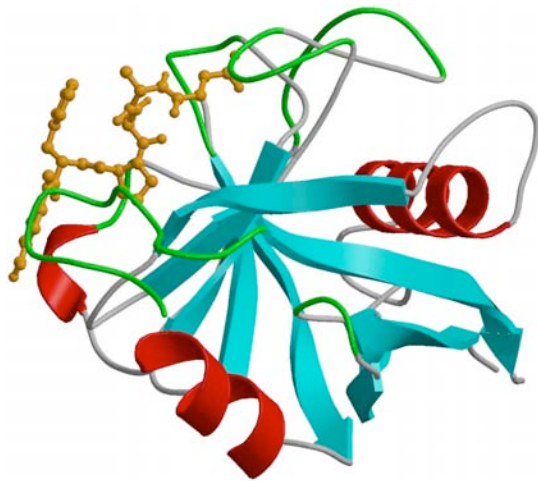
Current: 100,000 atoms

Real systems: >1,000,000 atoms

Current: $10^{-9} \sim 10^{-6}$ seconds

Real activity: $10^{-3} \sim 1$ seconds

> 10^6 difference in computing power available & required



Source: P. Agarwal, ORNL

Scalability of the MD programs

AMBER

In past, on Cheetah Supercomputer (IBM Power4 1.3GHz)

- 100 x 10⁻⁹s simulation took ~3600 hours (32 CPU/node)

- speed up ~8 on 32 CPUs

Present: Better time-to-solution and scaling on Cray X1

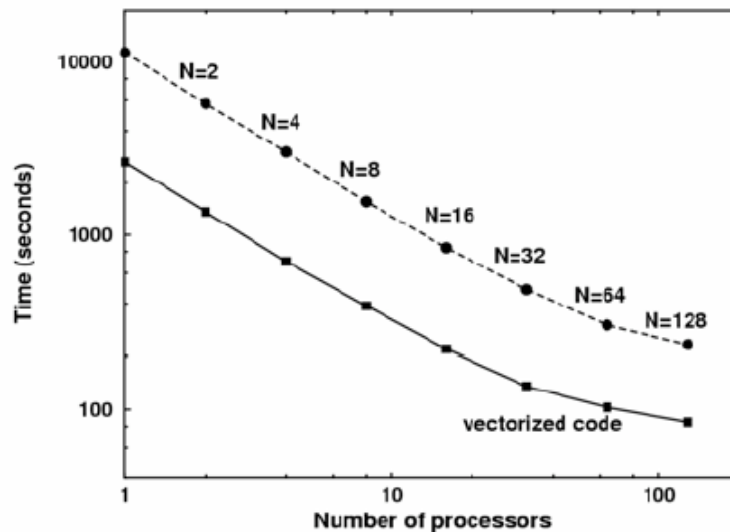


Figure 11: AMBER Time-to-solution on Cray X1.

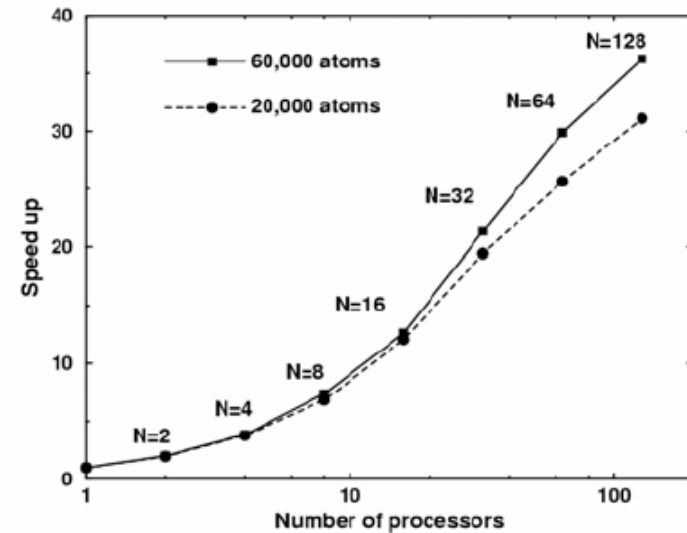


Figure 12: AMBER speedup on Cray X1.

Source: P. Agarwal, ORNL

Protein Folding

Ab initio protein folding for ~32,000 atoms will require 1000 teraFLOP processors, with a bandwidth of 10^9 and a latency of $1/10^9$

Description	Count*	Comment
Atoms	~32,000	300 amino acid protein + water
Force evaluations / time step	10^9	Pairwise atom - atom interactions
FLOPs / force evaluation	150	Typical molecular dynamics
FLOPs / time step	1.5×10^{11}	
Each time step	$\sim 10^{-15}$ s	1 - 5 femto second
Total simulation time	10^{-3} s	Protein folds in ~1 milli second
Total time steps	2×10^{11}	
FLOPs / simulation	3×10^{22}	Total FLOP/s to fold a protein
Execution time	3×10^7 s	1 year
Required FLOPS	$\sim 1 \times 10^{15}$	1 Petaflop/s

Estimate is conservatively based on quadratic algorithm.

Better algorithms will reduce (somewhat) running time, but usual surprises will increase it!

And good science will require multiple simulations

Source: David Klepacki, IBM

Multiple Sequence Alignment

Complexity

Run-time of dynamic programming solution
= $O(2^k * n^k)$

where n = length of each sequence
 k = number of sequences

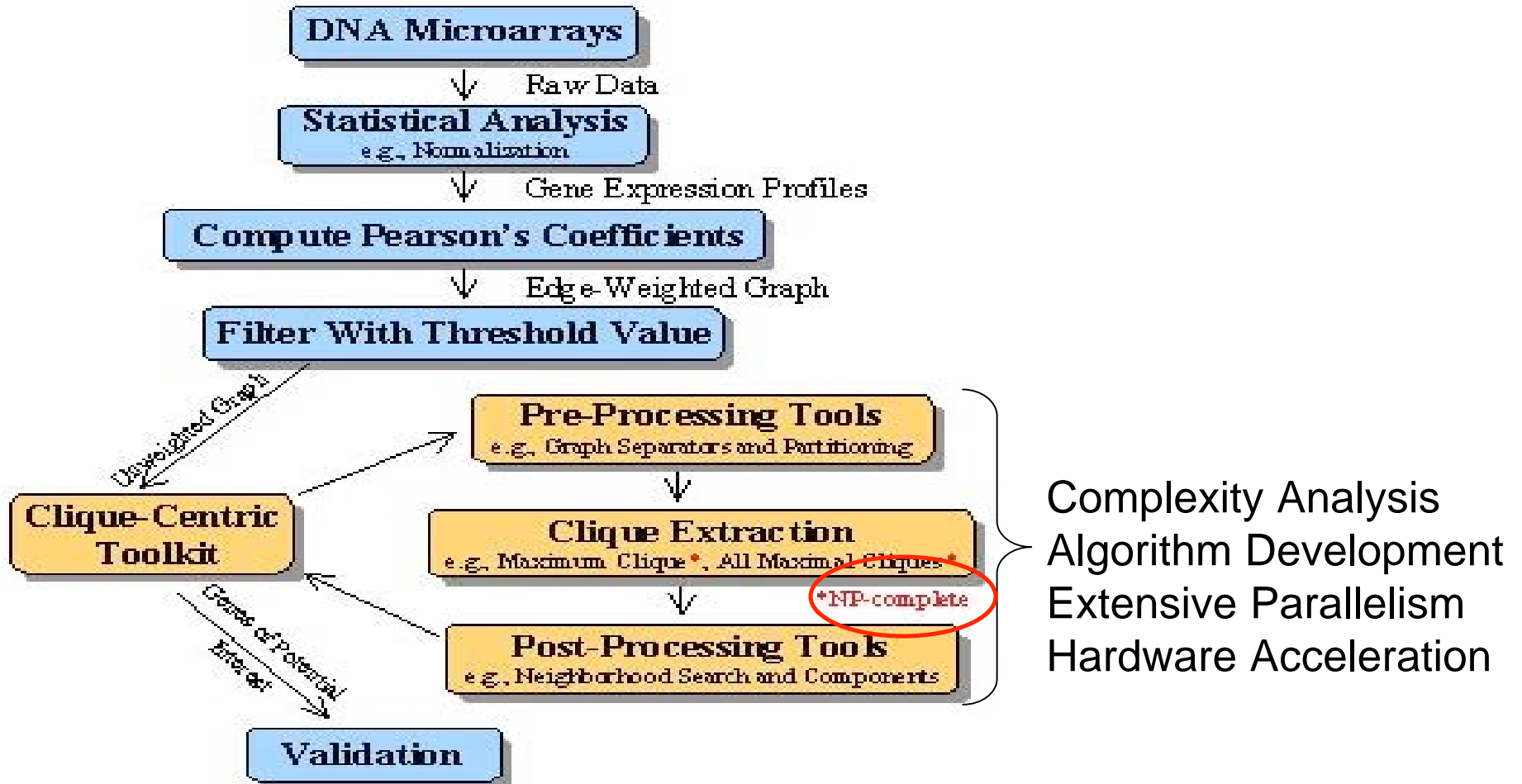
Space, $O(n^k)$, is prohibitively large!

Example:

6 sequences of length 100 _
 6.4×10^{13} calculations!

```
ARAGEEGRGFSVIADDEVRS LAAQSAEATAAMEALIVTI 120
ARAGEQGKGFSSVVAEEVRKLAQDSQAATQQVNAILGDI 373
ARAGEHKGKGFVVADEVKRLAEQSRQSSSEVSNIVKNI 260
ARAGESGKGFSSVVAEIRKLATNSKENVSQINDITNTI 235
SRAGEKKGKGFVVADEVKRLADQTKASTNTVSQLIEKT 351
:**** *:***:***:***:***:*.** ::      :. :
```

Microarray Data Analysis using Cliques



Source: M. Langston, UTK/ORNL

Extreme Metabolic Pathways Enumeration

Computational Complexity

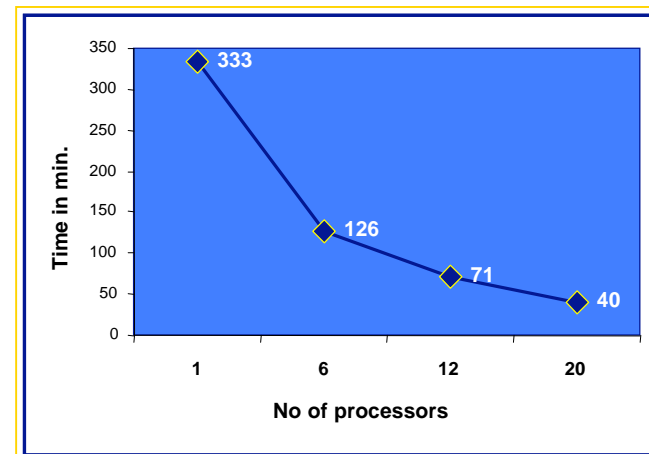
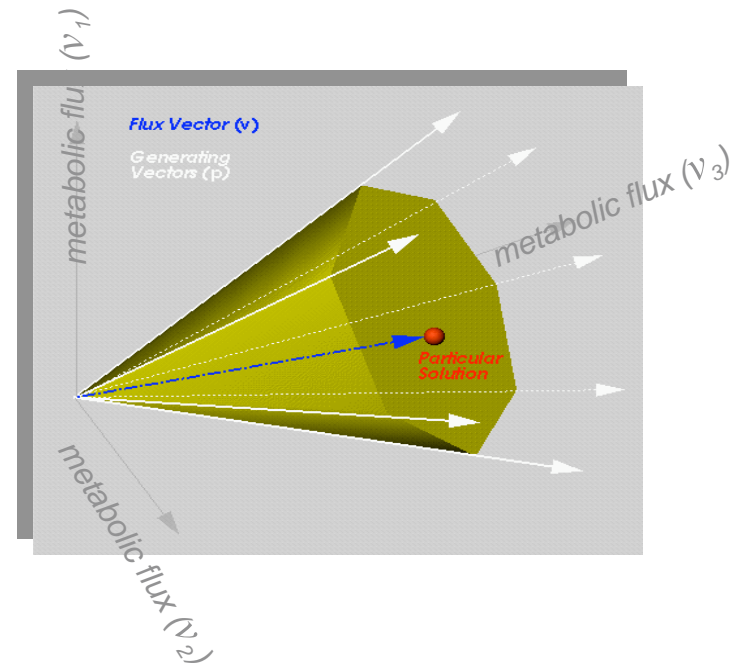
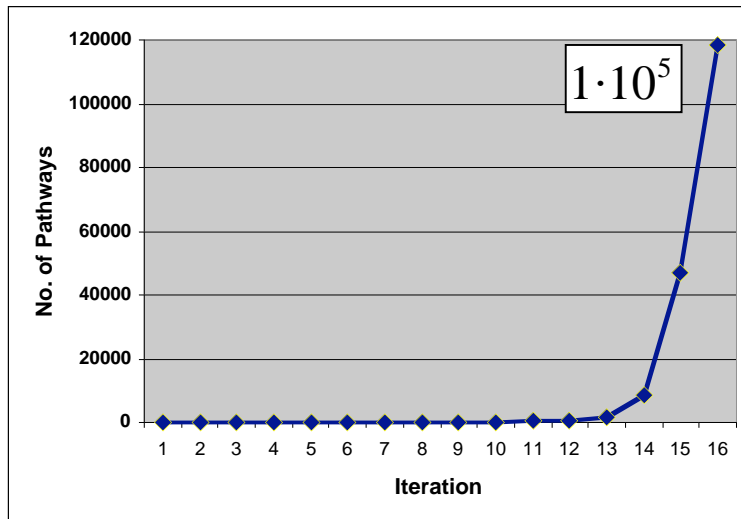
$C = O(M * P^3 * R)$ where

P is number of pathways

R is number of reactions

M is number of metabolites

Grows of pathways:



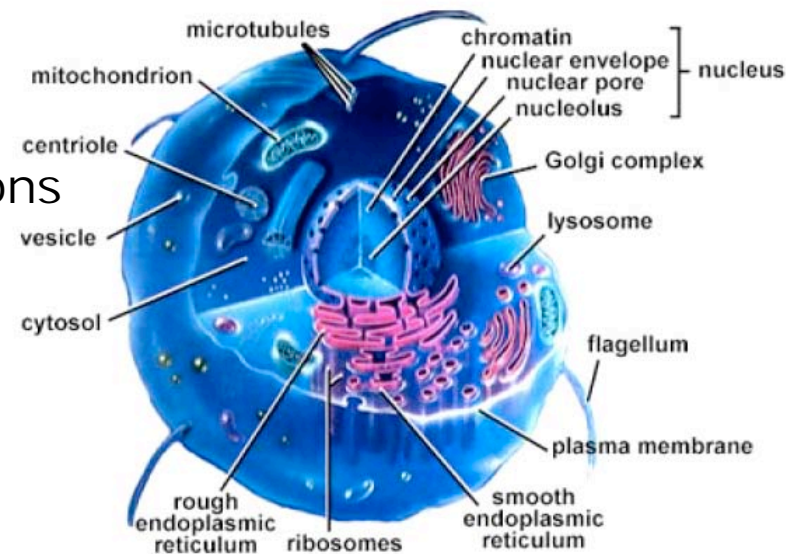
Whole-cell Simulation

Varying timescale

- 10^{-15} s: 3D continuum transport
- 10^{-6} s: signal transduction
- 10^{-3} s: metabolic pathways
- 10^1 - 10^4 s: effects of gene-expressions

Stochastic Simulation

Model 10^{14} biochemical reactions in *E.coli* will take 4^{1024} s = 12 years on a single processor



Petascale Applications —

Chemistry

Impact of Petaflop-scale Computing: Application Area: Chemistry

<p>Programmatic impact to be gained by access to Petaflop-scale computing.</p> <p>In the next few years increases from 100 to 1000 over today's computing power are expected. Roughly speaking, achieving a factor of 100 - 1000 over today's biggest machines would result in Petaflop-scale computing - that is, the ability to obtain from a few tenths of a Petaflop/s peak speed to a few Petaflops/s peak speed in the not-too-distant future. Sustained throughputs would be significantly less, depending on the application.</p> <p>OS wants the assessment to focus on computing capability (the ability to tackle big problems on a single computer run), rather than capacity (the amount of work that can be done in many computer runs by an individual or a community).</p>	<ul style="list-style-type: none"> • Ab initio prediction of interfacial structure of complex ceramic systems where the interface is amorphous on one side and crystalline on the other. A key to tailoring the material properties is to understand the segregation of rare earth and lanthanide dopants to the interface. This process is important for understanding grain growth and the corresponding effects on mechanical properties. • Petaflop-scale computing would be carried out for the complete interfacial system ~ 4-6 nm, which is composed of 1000's of atoms. Currently, with sustained performance of on the order of 750 Gflops, computations are capable on system sizes that are about 10 – 20 times smaller than those in the actual interfacial region. These methods are based on density functional theory and scale $O(N^3)$.
<p>Major scientific challenges to be addressed.</p> <p>These should roughly correspond to the impacts noted above.</p>	<ul style="list-style-type: none"> • Determination of models for the actual interfacial structure. • Determination of the total energy and electronic structure for these systems. • Optimization of the mechanical properties (fracture toughness, etc) for these systems.
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p>	<p>Currently with density functional theory-based codes we can get O(0.75 Tflops) on 256 Itanium 2 processors</p>

<p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p><i>Capability</i> is being emphasized – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)</p>	<p>(SGI, Altix). This is based on sustained performance of about 50% of peak.</p> <p>Turn around time is currently reasonably high, roughly corresponding to the actual computational time which is typically a few days.</p> <p>We need to increase the system size by at least a factor of 10 and more ideally a factor of 20. Single runs performing at the petaflop range would provide the necessary resources to address the critically important problem.</p>
<p>What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?</p>	<p>16 to 256 processors with 256 being the largest used thus far for this particular problem.</p>
<p>What is the Operations Count/Scaling from other computers?</p> <p>To scale performance from today's machines to larger capability machines requires either:</p> <ul style="list-style-type: none"> • An operations count, or • A scaling law based on current performance on current machines <p>If you have used a scaling law to characterize Petaflop-scale performance, please provide the logic used (e.g. compute time scales as n^4, where n is a linear cell dimension), along with the current computer capability. Please also provide the required turn-around time for the longest and typical runs.</p>	<p>The current algorithm scaling goes as $O(N^3)$, independent of the computer. The sustained performance comes from BLAS3 dense linear algebra calculations which dominate the work load.</p> <p>Currently computing on system sizes that need a factor of 10 - 20 increase with a corresponding 1000 – 8000 increase in computation time. Given sustained performance of 0.75 Tflops, a petaflop capability would get us into the correct range.</p>
<p>Projected increase in algorithm efficiency?</p> <p>If you are counting on an increase from better algorithms (historically, algorithm improvements have approximately matched improvements in hardware), please indicate the factor you've used.</p>	<p>Algorithm development is proceeding quite well and is expected to significantly cut the CPU cost in the near future. Currently, methods that offer similar or better accuracies to those currently used are on the horizon and offer scaling of $O(N^3)$ and perhaps near linear is possible.</p> <p>Algorithm developments are based on integrating different levels of theory , so that a part n, where n is nearly the size of the total system N, then the bulk of the computational work is on the levels of theory that treat n and these scale between $O(N) - O(N^3)$ while the small remaining piece $M=N-n$ scales as $O(M^5)$. Recent</p>

	developments using multiresolution techniques offer not only improve scaling but also the accuracies as compared existing methods.
Other	

Impact of Petaflop-scale Computing: Application Area: Chemistry

<p>Programmatic impact to be gained by access to Petaflop-scale computing.</p> <p>In the next few years increases from 100 to 1000 over today's computing power are expected. Roughly speaking, achieving a factor of 100 - 1000 over today's biggest machines would result in Petaflop-scale computing - that is, the ability to obtain from a few tenths of a Petaflop/s peak speed to a few Petaflops/s peak speed in the not-too-distant future. Sustained throughputs would be significantly less, depending on the application.</p> <p>OS wants the assessment to focus on computing capability (the ability to tackle big problems on a single computer run), rather than capacity (the amount of work that can be done in many computer runs by an individual or a community).</p>	<ul style="list-style-type: none">• Molecular-scale devices offer several advantages over conventional technology, including miniaturization that will allow the scaling of component size to the ultimate level of atoms and molecules. Potential benefits include dramatically increased computational speed and lower fabrication costs. A concerted integrated theoretical effort has been aimed at developing the simulation and design tools to quantitatively model electron transport through organic molecules in an open environment where the molecules are placed between semi-infinite (macroscopic) metallic electrodes. The ultimate goal is to compute current-voltage curves in order to reproduce and optimally predict real experiments.• Petaflop-scale computing will be carried out for the complete interfacial system ~ 4-6 nm, which is composed of 1000's of atoms. Currently, with sustained performance of on the order of 750 Gflops, computations are capable on system sizes that are about 10 – 20 times smaller than those in the actual interfacial region. These methods are based on density functional theory and scale $O(N^3)$.
<p>Major scientific challenges to be addressed.</p> <p>These should roughly correspond to the impacts noted above.</p>	<ul style="list-style-type: none">• Determination of the properties of the molecular-scale models taking the environment (the leads), the external applied field (the bias potential) into account.• Treatment of the system as an infinite <i>open</i> (as opposed to periodic) configuration.• Realistic description of the system as a whole, in a quantum mechanical based modeling.
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p>	<p>Currently with density functional theory-based codes we can get O(0.75 Tflops) on 256 Itanium 2 processors</p>

<p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p><i>Capability</i> is being emphasized – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)</p>	<p>(SGI, Altix). This is based on sustained performance of about 50% of peak.</p> <p>Turn around time is currently reasonably high, roughly corresponding to the actual computational time which is typically a few days.</p> <p>There is a need for (1) increasing the system size (the central molecular unit including part of the leads) and for (2) non-equilibrium (i.e. non-zero bias applied to the leads) conditions in which case the geometry and the electronic density of the system may be able to relax</p>
<p>What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?</p>	<p>Typically 64 (but by no means limited to that).</p>
<p>What is the Operations Count/Scaling from other computers?</p> <p>To scale performance from today's machines to larger capability machines requires either:</p> <ul style="list-style-type: none"> • An operations count, or • A scaling law based on current performance on current machines <p>If you have used a scaling law to characterize Petaflop-scale performance, please provide the logic used (e.g. compute time scales as n^4, where n is a linear cell dimension), along with the current computer capability. Please also provide the required turn-around time for the longest and typical runs.</p>	<p>The current algorithm scaling goes as $O(N^3)$, independent of the computer. The sustained performance comes from BLAS3 dense linear algebra calculations which dominate the work load.</p> <p>Currently computing on system sizes that need a factor of 10 - 20 increase with a corresponding 1000 – 8000 increase in computation time. Given sustained performance of 0.75 Tflops, a petaflop capability would get us into the correct range.</p>
<p>Projected increase in algorithm efficiency?</p> <p>If you are counting on an increase from better algorithms (historically, algorithm improvements have approximately matched improvements in hardware), please indicate the factor you've used.</p>	<p>Algorithm development is proceeding quite well and is expected to significantly cut the CPU cost in the near future. Currently, methods that offer similar or better accuracies to those currently used are on the horizon and offer scaling of $O(N^3)$ and perhaps near linear is possible.</p>
<p>Other</p>	

CCSD(T) notes

High-accuracy calculations that are capable of replacing experiment in terms of both reliability and precision are expensive and scale highly-nonlinearly with respect to both system size and precision. In the following, I discuss the cost of calculations using current conventional algorithms, and then how this is expected to change in the near future.

In the following, N refers to the total size of the basis set, O the number of occupied orbitals (essentially half the number of electrons) and V the number of unoccupied orbitals ($N=O+V$). The non-chemically active (core) occupied orbitals will be frozen (i.e., neglected) in the most expensive calculations.

We cannot directly compute most high-accuracy results, and must instead extrapolate to them in a sequence of systematically designed basis sets (Dunning's correlation consistent bases). The ratio N/O is a (crude) measure of the quality of the basis. E.g., nitro-benzene ($C_6H_5NO_2$) has 64 electrons giving 9 frozen core and 23 active occupied orbitals (O).

Table 1. The number of basis functions and number of basis functions per occupied orbital for a sequence of Dunning's correlation-consistent basis sets applied to nitrobenzene.

Basis set	N	N/O
cc-pVDZ	151	6.6
cc-pVTZ	340	14.9
cc-pVQZ	645	28.0
aug-cc-pVDZ	252	11.0
aug-cc-pVTZ	529	23.0
aug-cc-pVQZ	1077	46.8

Predictive calculations require that we compute with at least the TZ and QZ bases, and ideally also the 5Z. Thus we require N/O in the range 20-60.

For closed-shell CCSD(T) the FLOP count is roughly:

$$2N_i (1/4 O^2V^4 + 4O^3V^3) + 2(O^3V^4 + O^4V^3)$$

The no. of iterations will be perhaps 15 for tight convergence. A PFLOP/s day is about 8.6×10^{19} operations. With these FLOP counts (which don't include integral evaluation costs) we can compute in 1 day (assuming no spatial symmetry):

Table 2. Size of CCSD(T) calculation that can be performed in 1 PFLOP day as a function of precision (measured by N/O, the number of basis functions per occupied orbital).

N/O	N	O	Time in days		(C6H6)n
			CCSD	(T)	n
20	2340	117	0.054	0.96	7.8
30	2760	92	0.058	0.95	6.1
40	3120	78	0.06	0.97	5.2
50	3400	68	0.07	0.92	4.5
60	3660	61	0.07	0.90	4.1

This is assuming computation at a sustained PFLOP/s, and the triples part of the CCSD(T) calculation is expected to reach a very high fraction of peak speed on most proposed machines (e.g., 90+% even on most workstation clusters) and algorithms that scale to 10,000 processors have already been designed. Spatial symmetry will reduce the time by a factor of $O(h^{-2})$ (h the order of the point group).

The last column of the table indicates the number of benzene (C6H6) molecules that the number of occupied orbitals corresponds to. It should be clear that fully predictive calculations *using current conventional algorithms* are constrained even on PFLOP architectures to relatively small systems. We need to optimize geometries and examine dynamics, not just compute energies. Also, there are also a vast number of molecules and reactions that are of importance to DOE (it is impossible to enumerate them all), and this level of precision is vital for theory and computation to be relevant to the fields of catalysis, combustion, and atmospheric chemistry, to name just a few.

Indeed, the computational requirement is so vast that an external observer might think that even PFLOP-scale computation is only useful for small systems or at low precision. This is most certainly untrue, and I explain why below.

Considering first high-precision calculations, over the past few years there has been much success in developing *reduced-scaling* methods that reduce the rate at which the computational cost increases with system size. The theoretical best is linear scaling, which has already been demonstrated in low precision calculations on large systems. For reliable and predictive high-precision computation we cannot expect to reach linear scaling in the near future, but cubic scaling is reasonable. The computational cost is determined by both the exponent (for cubic, 3) and the prefactor (i.e., the A in $\text{time} = A * N^3$). This prefactor will depend very strongly upon the required precision, and in most approaches is $A = O(\epsilon^{-4})$. Another way to look at the prefactor is that it is determined by the size of the system that you must exceed before the asymptotic scaling dominates. Much less than this, and the cost is growing at the conventional rate ($O(N^7)$), larger than this, the cost grows at the slower rate.

A critical observation is not only are these new reduced scaling methods becoming available now, but that for the first time computers are large enough to study systems for

which the asymptotic scaling is relevant without sacrificing precision. Thus, we can reexamine the above table, assuming either quadratic or cubic scaling beyond a region of more than two benzene molecules and estimate the prefactor as the conventional cost within this region. This is a rather crude model, but is illustrative of the impact of the methods being developed and tested now by the international chemistry research community.

Table 3. Estimate of the number of benzene molecules that may be simulated using CCSD(T) in 1 PFLOP-day assuming conventional, cubic and quadratic scaling.

N/O	(C ₆ H ₆) _n		
	Conventional	Cubic scaling	Quadratic scaling
20	7.8	49	240
30	6.1	28	104
40	5.2	19	58
50	4.5	14	37
60	4.1	11	25

This table illustrates that the reduced scaling algorithms are opening up a vast new area of chemistry for fully quantitative study. *For instance, we will be able to compute accurate thermodynamics for all of the hydrocarbons and intermediates important to combustion.* It also makes clear that we must continue to develop methods that scale not just better with system size but also with precision.

Looking beyond fully-quantitative computation, a much larger number of calculations are performed using less accurate theories. Although incapable of independently matching experimental precision, these methods are capable of making quantitative predictions when carefully combined with data from either experimental or more precision calculations. Density functional theory and second-order many-body perturbation theory are examples of methods that very successfully interpolate and extrapolate trends and can make quantitative predictions. These methods already have practical linear scaling implementations. When considering chemistry in the real world, which happens in solution, at interfaces, with multiple components and phases, with poor characterization and disorder, these less expensive methods must be used.

For instance, consider modeling a molecular electronic device comprising a gold atomic microscope tip being scanned across a self-assembled monolayer of molecules absorbed onto a gold surface. The end objective is to understand, model, design and control a molecular electronic device comprising multiple such connections. Quantum electron transport methods must be used. Presently, with TFLOP computers, we aspire merely to modeling a single molecule with idealized chemical contacts at the junction of gold and molecule, and without inclusion of the dynamical effects of solvent, vibration, temperature, etc., on the transport properties (these are known to be important, but must

presently be estimated from more idealized models). PFLOP computers will enable us to assemble a fully realistic model and explore the fully structure and dynamics of the combined system.

Subject: RE: help in justifying future DOE "capability" systems for chemistry

Date: Thursday, May 20, 2004 5:36 PM

From: Windus, Theresa L <Theresa.Windus@pnl.gov>

To: Ed Barsis <ebarsis@bmv.com>

Cc: "Peter L. Mattern" <pmattern@bmv.com>

Hi Ed,

Hopefully this is closer to what you want. Again, I should point out that these are samples. Chemists tend to use whatever resource is available to them for as long as they can get it. I also believe that there is going to be a change in the way that we approach the science and that is going to change the "scaling" (I tried to capture this in the second example). If this is not what you are looking for, please give me a call (509-376-4529 for work and 509-528-8722 for cell).

Theresa

To accomplish some of the scientific challenges will require computations at the N^5 scale: A current computation of 3,400 basis functions requires 5 hours of computation to obtain an energy of a large water cluster on the PNNL HP at about 0.6 TF. If we would like to increase the number of basis functions (or the size of the cluster) by a factor of 10, we would need 5 hours * $10^5 / (1000 \text{ TF} / 0.6 \text{ TF}) = 300$ hours = 12.5 days on a PetaFLOP computer. Of course, we are hoping to obtain methods with lower scaling for the same accuracy which would decrease this need by an order of magnitude to 1.25 days (assuming 1 order of magnitude).

In another example, ensemble calculations basically scale linearly with respect to the number of points in the ensemble (where each point can range from N to $N!$ where N is the number of basis functions or atoms in the simulation). These computations will require a factor of approximately 10,000 (as a minimum) more points in the simulation. If a current computation takes 24 hours on a 0.96 TF machine (as mentioned in the throughput section in the Table I sent before), this will require 1 day * $10,000 / (1000 \text{ TF} / 1 \text{ TF}) = 10$ days.

-----Original Message-----

From: Windus, Theresa L [mailto:Theresa.Windus@pnl.gov]

Sent: Wednesday, April 28, 2004 7:56 AM

To: Harrison, Robert J.; David Keyes

Cc: ebarsis@bmv.com; pmattern@bmv.com

Subject: RE: help in justifying future DOE "capability" systems for chemistry

Hi All,

I am also happy to help out. Robert is definitely correct about the open ended nature of chemistry. We also have many theoretical methods and algorithms that are being used, all which scale in different ways and with different hardware requirements. This all makes it difficult to pin down one example and future projections.

Having said that, though, there are certainly trends that will have direct impact on the computations of the future. In the past (and now!), because the computations were so expensive and the hardware was limiting, chemists tended to "restrict" their computations (and sometimes their thinking!) to the minimal amount of information that they can get away with to learn a few important points about the reactions of interest. For example, they look for the stationary points (minima and a few "important" maxima) on a complex potential energy surface and ignore the rest of the surface which is critical to understanding kinetics, dynamics, and dynamical properties. This will certainly still be important and very large computations (either with respect to size or complexity of the computation) will continue to be examined. (We are running some very large protein computations to investigate electron transfer reactions and I know that Robert and Zhengting have recently worked on a Full CI code that scales well and enables very large computations.)

However, with increased computational power and algorithms, chemists are rethinking how they approach the problems and are looking at more "complete" examinations of the surfaces and their dynamics. These types of computations require many electronic structure computations, some which are "independent" of each other and others which are "dependent" on each other. (Explicit examples include nudged elastic band methods, Monte Carlo methods, direct dynamics, and clever algorithms for examining long time scale molecular dynamics.) Each of these computations can easily take several orders of magnitude more computation than is currently being used. They can easily take 100 to 1000 TF.

I hope this is helpful. I am happy to answer questions or clarify anything that I have written above.

Regards,
Theresa

Petascale Applications —

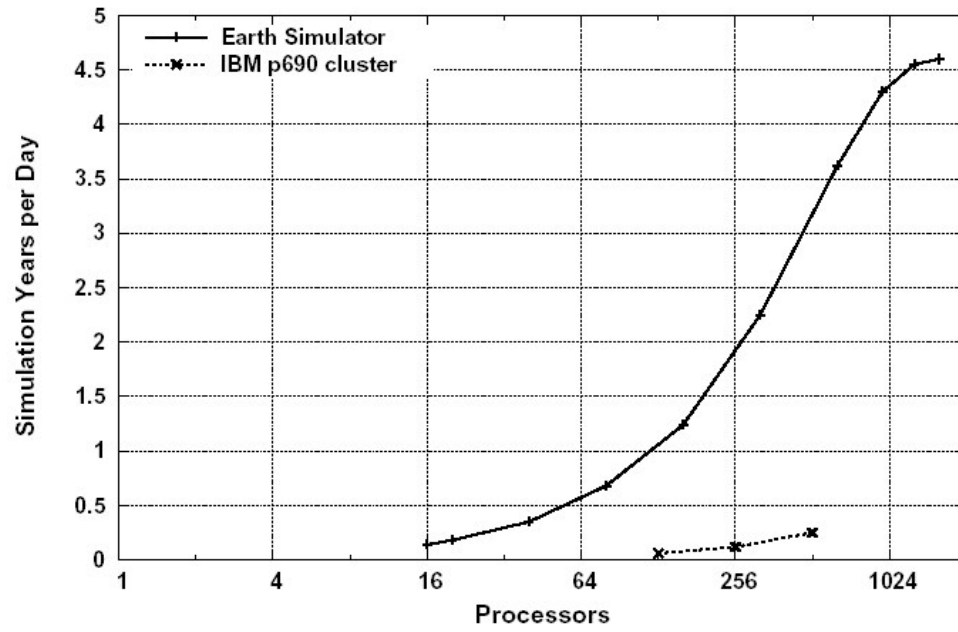
Climate

Impact of Petaflop-scale Computing: Application — Climate Modeling

	Climate Modeling
<p>Programmatic impact to be gained by access to Petaflop-scale computing</p>	<p>With petascale computing we could, in some combination:</p> <ul style="list-style-type: none"> • Improve regional prediction (we probably couldn't get all the way to 10km in the atmosphere) • Realistically simulate mesoscale ocean eddies and improve simulated ocean circulation • Improve understanding of aerosol feedbacks • Move toward source-based greenhouse gas scenarios to improve evaluations of policy changes • Begin to assess the ecological implications of climate change • Increase model fidelity • Examine critical issues such as the collapse of the thermohaline circulation that may occur on time scales ranging from decades to as long as a few centuries.
<p>Major scientific challenges to be addressed</p>	<p>These scientific challenges match the programmatic impacts noted above.</p> <ul style="list-style-type: none"> • Increase resolution of atmospheric models • Increase resolution of ocean models • Add atmospheric chemistry and ocean biogeochemistry • Include dynamic ecosystem models • Replace parameterizations of sub-grid processes with more realistic models • Increase the length of control and climate change runs to reveal tendencies for models to drift and to improve estimates of model variability.
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p> <p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p>[We are emphasizing <i>capability</i> – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)]</p>	<p>For coupled model simulations with 1 degree (100km) ocean/ice resolution and T85 (150km) atmosphere/land resolution, we achieve: 3.5 simulated years/CPU day on 192 processors on an IBM Power 4 system and 16 simulated years/CPU day on 176 processors on the Earth Simulator (with some components not fully optimized).</p> <p>For ocean only simulations at eddy-resolving 0.1-degree (10km) resolution (the largest sims to date), we achieve: 0.12 simulated years/CPU day (20 simulated years) on 500 SP3 processors (7% of peak) 3.6 simulated years/CPU day (several 20-year sims) on 640 EarthSimulator processors (30% of peak)</p> <p>The added expense for doing ocean biogeochemistry is roughly linear in the number of tracers and current simulations use 10-30 additional tracers. Atmospheric chemistry is somewhat more complicated and requires a larger number of tracers.</p> <p>Estimating impacts of new physical parameterizations is difficult because the decision on whether to include</p>

	<p>more physical models often is partly based on performance impact. In other words, a new physical parameterization may greatly improve a model simulation, but if it costs 10x more it won't be included until we have machines that can handle them. Our level of pain is typically 5-10 years/day and if performance falls below that, we start looking at things more carefully.</p>
<p>What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?</p>	<p>Credible architectures for achieving Petaflop-scale capability within 5 years will contain tens of thousands of processors. The ability for codes to run efficiently with large numbers of processors will be critical. Because of the difficulty of load-balancing the components and the necessity for running ensembles, scaling of the full coupled system is not well characterized. It can likely scale up higher than the current 192 processor configuration, but that what is working now from a resource throughput standpoint (e.g. queue structures, machine load, etc.).</p> <p>As shown in figures 1a and 1b below, the ocean at 1-degree (x1) and the atmosphere at T42 (300km, the previous CCSM configuration) each individually scale up to 256 processors, indicating the full model might scale to 512 processors, but not much beyond that. A T85 would likely scale up a little further. Because these grids are not large, scaling to thousands of processors for the current problem is not achievable.</p> <p>For future problem sizes (eddy-resolving ocean at 0.1, atmosphere at T170 or larger), scaling up to a 1000 or 2000 processor might be achievable. Eddy-resolving ocean simulations have been run in production on 480 processors on IBMs and 640 processors of the Earth Simulator and as Fig. 2 shows, can scale up to 1000 processors. Scaling of high-resolution atmosphere models is currently not well known as there are changes to physical parameters that must be made, requiring some substantial runs at those resolutions to determine the best values.</p> <p>The most important scaling issue with climate models is that with increases in grid size, the time step decreases so even with enough processors to handle increased grid sizes, overall throughput will decrease without a corresponding increase in single-processor performance.</p>
<p>What is the Operations Count/Scaling from other computers?</p>	<p>As noted above, we can't simply scale up numbers for climate models assuming a 10000-processor machine since the models likely won't scale up to those processor counts. We can say that on commodity processors, we typically achieve 5-10% of peak single-node performance and on vector machines, we achieve 30-40% of peak.</p>
<p>Projected increase in algorithm efficiency?</p>	<p>While we have historically seen dramatic improvements due to algorithm efficiency, it is unclear that this will continue in the future. The pace and the magnitude of such improvements have decreased in recent years as the architectures and models for the physical components have matured. However, new algorithms for doing chemistry/biogeochimistry are improving the ability to do simulations with large numbers of tracers. For example, a new advection scheme which is geometrically-based that computes the geometric factors once for the first tracer and the remaining tracers are almost free.</p>
<p>Other</p>	

LANL Parallel Ocean Program
POP 1.4.3, 0.1 degree benchmark



CCSM Performance Models and Scaling to Ultrascale Computers For Ed Barsis

The Community Climate System Model consists of four component models, atmosphere, ocean, sea ice and land, and a coupler that regrids data for the different spatial grids and time stepping schemes. The current mode of execution places each component on a distinct set of processors with the atmosphere requiring roughly half the processors and constituting the coupled simulations dominant cost. So to first order, a performance model of the coupled system can be developed by considering the atmospheric model.

The Community Atmosphere Model (CAM) has two major computations, what is called the dynamics, representing the fluid flow calculation of atmospheric winds, and what is called the physics, representing column radiation balances, moist convection and clouds. The dynamics employs a Eulerian spectral transform algorithm for the approximation of all terms in the momentum, mass and energy conservation equations, and a semi-Lagrangian approximation for the transport of moisture and atmospheric trace gases. The physics is embarrassingly parallel, though a significant load imbalance would exist if left in the natural parallel decomposition. So data transposition using message passing as well as shared memory parallelism is employed.

The dynamics calculation is dominated by the spectral transform and a performance model of the spectral transform can be developed to estimate the time for a multi-level calculation. The computational operation counts and communication cost estimates are based on a model in [2] for a one dimensional decomposition and modified by Rich Loft (NCAR) to reflect a simple transpose between FFT and Legendre transform phases including levels. The time for the FFT, the Legendre transform and the communication overhead are estimated using machine dependant rate constants a,b,d, and e.

$$\begin{aligned}\text{Time for FFT} &= a \cdot 5 \cdot (6L+1) \cdot J \cdot I \cdot \log_2(I) \\ \text{Time for LT} &= b \cdot 2 \cdot (6L+1) \cdot J \cdot M^2 \\ \text{Time in COMM} &= d \cdot P + e \cdot 2 \cdot (6L+1) \cdot J \cdot (2M+1)\end{aligned}$$

Nomenclature:

- M wave number resolution, eg. TM
- I number of longitudes ($I \geq 3M+1$)
- J number of latitudes ($J=I/2$)
- L number of vertical levels
- P number of nodes (computational unit doing FFT or LT)
- a computational rate of FFT in flops/node
- b computational rate for LT in flops/node
- d latency factor
- e bandwidth factor

Using this model with estimates of network bandwidth and the speed of a node in computing FFT's and Legendre transforms, we can determine the overall computational

rate of the computer for performing spherical harmonic transforms as well as parallel efficiencies.

Each process in the physics can be modeled based on the relevant process timestep and the number of floating point operations per column. (Here we assume that the number of levels remains fixed and that only the horizontal resolution varies. This was not an unrealistic assumption over the last 5 years.) As more computing power becomes available, modelers will include more “physics” processes driving up the computational cost per column. For example, tropospheric chemistry will be added. The following break out of the physics has been used to estimate overall scaling. (The operation counts are measured from simulations using hardware performance monitors.)

Atmospheric Physics and Chemistry	Ops/col/day	Ops/col/step
LW Radiation	4.50E+06	0.00E+00
SW Radiation	4.90E+06	0.00E+00
Other Physics		1.00E+05
Sulfur Chemistry	0	4.30E+04
Trop Chemistry	0	0.00E+00
Strat. Chemistry	0	0.00E+00
Total Physical Ops	9.40E+06	1.43E+05

The zeros for chemistry operations reflect that the current standard simulation do not include these processes.

The performance of the dynamics is combined with the performance of the physics by calculating the required timestep for a given horizontal resolution (subject to CFL limits) and the frequency of the process updates. The single processor efficiency for the physics is an input parameter to the model as it depends on the level of vectorization and cache utilization achieved. In this way, we avoid trying to predict the actual memory performance of a processor based on hardware specifications.

The number of timesteps required for a century long ‘simulation is given for the standard horizontal resolutions in the table below. It is a fundamental feature of climate simulations that the long time integrations limit the spatial resolution that may be applied. Faster processors, fast memory (eg. vector) and low latency interconnects, are key to fast time stepping.

M	J	I	P	Timestep(min)	Years	Steps/year
	42	64	128	32	20	100 26280
	85	129	258	64	10	100 52560
	170	256	512	128	5	100 105120
	341	513	1026	256	2.5	100 210240
	682	1024	2048	512	1.2	100 438000
	1279	1920	3840	512	8.00E-01	100 657000
	1365	2049	4098	1024	6.00E-01	100 876000

The performance model for scaling estimates the number of Flops in each part of the calculation, physics and dynamics. The dynamics performs at the efficiency computed from the spectral transform performance model and the physics performs at a specified efficiency. This performance is multiplied times the number of time steps that must be taken and a throughput (simulated years per day) is computed. From this we can get the time to solution. Typically, production simulations are only performed if the throughput is greater than 5 years per day.

The following examples show present day performance estimates for the Cray X1 and the IBM p690 (with Federation). The column labeled P signifies the number of processors used on the Cray X1.

M	J	I	P	Total Gflops	Sim.yr/day	Time to solution (days)	Sustained Rate(Gflops)
	42	64	128	1286.01E+04	1.43E+02	7.01E-01	99.31424
	85	128	256	2563.77E+05	4.63E+01	2.16E+00	202.0607
	170	256	512	5122.69E+06	1.34E+01	7.45E+00	418.202
	341	512	1026	10242.18E+07	3.54E+00	2.83E+01	891.5583
	682	1024	2048	20482.06E+08	8.28E-01	1.21E+02	1978.085
	1279	1920	3840	38401.38E+09	2.64E-01	3.79E+02	4214.307
	1365	2049	4098	40962.15E+09	1.84E-01	5.45E+02	4570.644

For the IBM p690 the similar chart of predicted scaling is

M	J	I	P	Total Gflops	Sim.yr/day	Time to solution (days)	Sustained Rate(Gflops)
	42	64	128	1286.01E+04	3.39E+01	2.95E+00	23.60809
	85	128	258	2563.80E+05	1.09E+01	9.15E+00	48.0322
	170	256	512	5122.69E+06	3.18E+00	3.14E+01	99.18668
	341	513	1026	10242.18E+07	8.31E-01	1.20E+02	209.7155
	682	1024	2048	20482.06E+08	1.92E-01	5.22E+02	457.648
	1279	1920	3840	38401.38E+09	5.97E-02	1.68E+03	952.426
	1365	2049	4098	40962.15E+09	4.14E-02	2.42E+03	1029.725

The Cray X1 projections (and performance) indicate that it will be possible to achieve production level throughput at a spectral truncation of T170. This is high resolution for a climate model. In fact, some experiments will be possible at T340. This is a horizontal resolution of about 40km and will be a significant step towards regional climate modeling with the rigor of a fully coupled general circulation model. At this resolution, the events like hurricanes will be simulated realistically in the model, as they are in weather models. The regional rainfall totals that depend on some of these significant storms will be much better simulated.

We will also be able to simulate the global carbon cycle with much greater realism of land use patterns and ecological feedbacks. With the computational power represented by more advanced(planned) hardware such as the Cray X2, we expect to be able to increase resolution again and perform computational experiments with cloud resolving detail in the 10km to 30km scale.



Climate Modeling on a Petaflop Computer

May 24, 2004

Mark Taylor

Mark Boslough

Bill Spatz

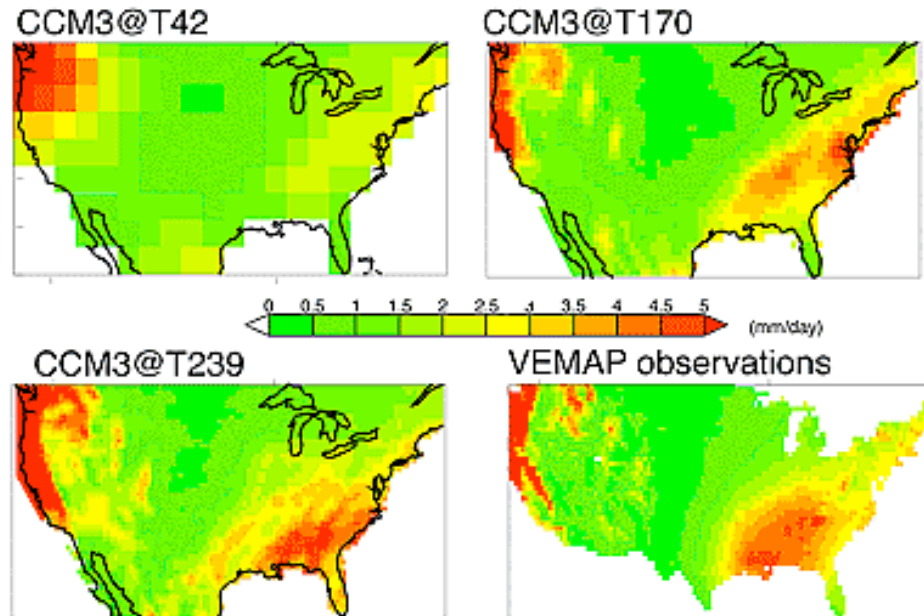


Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy's National Nuclear Security Administration
under contract DE-AC04-94AL85000.





Why 10 km Resolution?



Wintertime precipitation over the United States as simulated by CCM3 at three different horizontal resolutions (300, 75 and 50km), and in the VEMAP observational dataset. Both small- and large-scale (e.g. in southeastern U.S.) features of simulated precipitation appear to converge towards observations as the model resolution becomes finer.

Duffy, Govindasamy, Milovich, and Thompson, LLNL, <http://eed.llnl.gov/cccm/hiresolu.html>

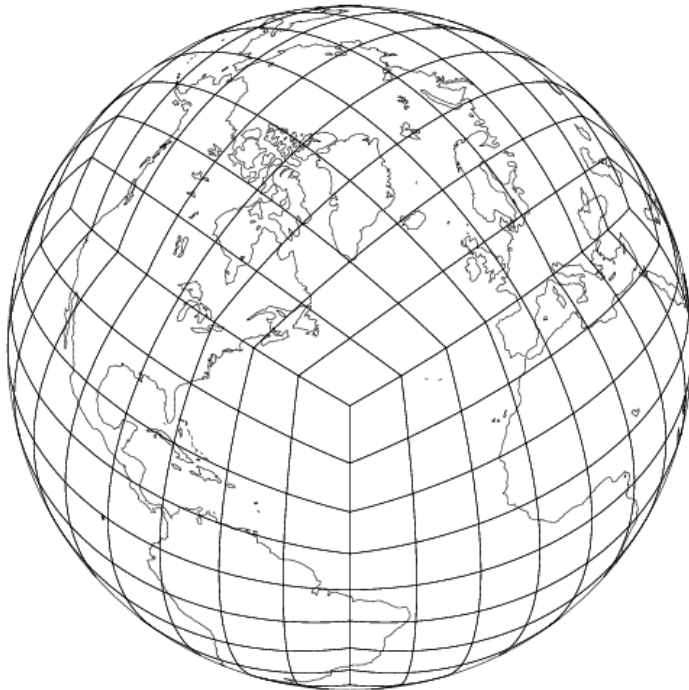


Why 10 km Resolution?

- DOE ScaLeS Report (draft)
 - 10km resolution is an important long term objective for atmosphere and ocean models. At 10km, the atmospheric model will be the dominant component of a coupled model.
- Ocean Model
 - 10km resolution required to be *eddy resolving*. (resolve the eddies which contain most of the kinetic energy in the ocean)
- Atmospheric Model
 - 10km is necessary to resolve important local weather features (storm tracks, topography, monsoons and other precipitation) that impact long term climate. Especially important for understanding the regional impacts of climate change.
 - Regional needed for social impacts of climate change (land use, water resources, agriculture, forest management, conflict due to environmental scarcity)
 - 10km capability would also allow regional forecast models to be replaced with a single global forecast model. (NOAA's National Hurricane Center predicts hurricane landfall using a regional model with a 10km grid.)



SEAM: Spectral Element Atmospheric Model



- ◆ Development funded by DOE, mostly at NCAR:
 - Taylor, Tribbia, Iskandarni, 1997; Taylor, Loft, Tribbia, 1998; Loft, Thomas, Dennis, 2001; Thomas, Loft, 2002; Fournier, Tribbia, Taylor, 2004;
- ◆ Global Atmospheric Circulation Model
- ◆ Spectral elements used in horizontal directions
- ◆ Finite differences used in vertical/radial direction
- ◆ Two dimensional domain decomposition: each processor contains one or more elements and the vertical columns of data associated with those elements.
- ◆ Coupled to the Community Atmospheric Model (CAM) Physics package



SEAM

Performance Requirements

GFLOPS for
Resolution 1x Reality

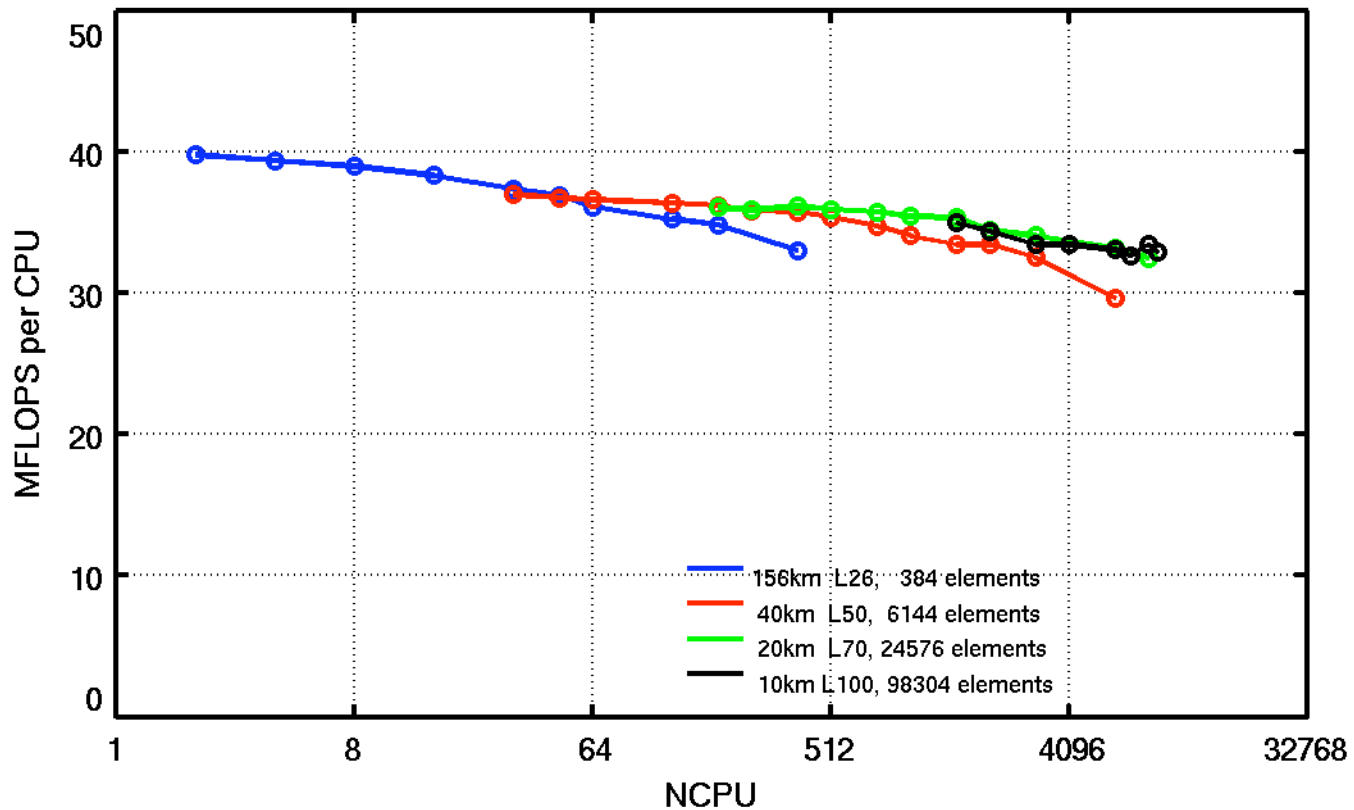
40km/50L	1.6
20km/70L	20
10km/100L	230

Climate application: 1000x Reality

Forecast application: 5x Reality



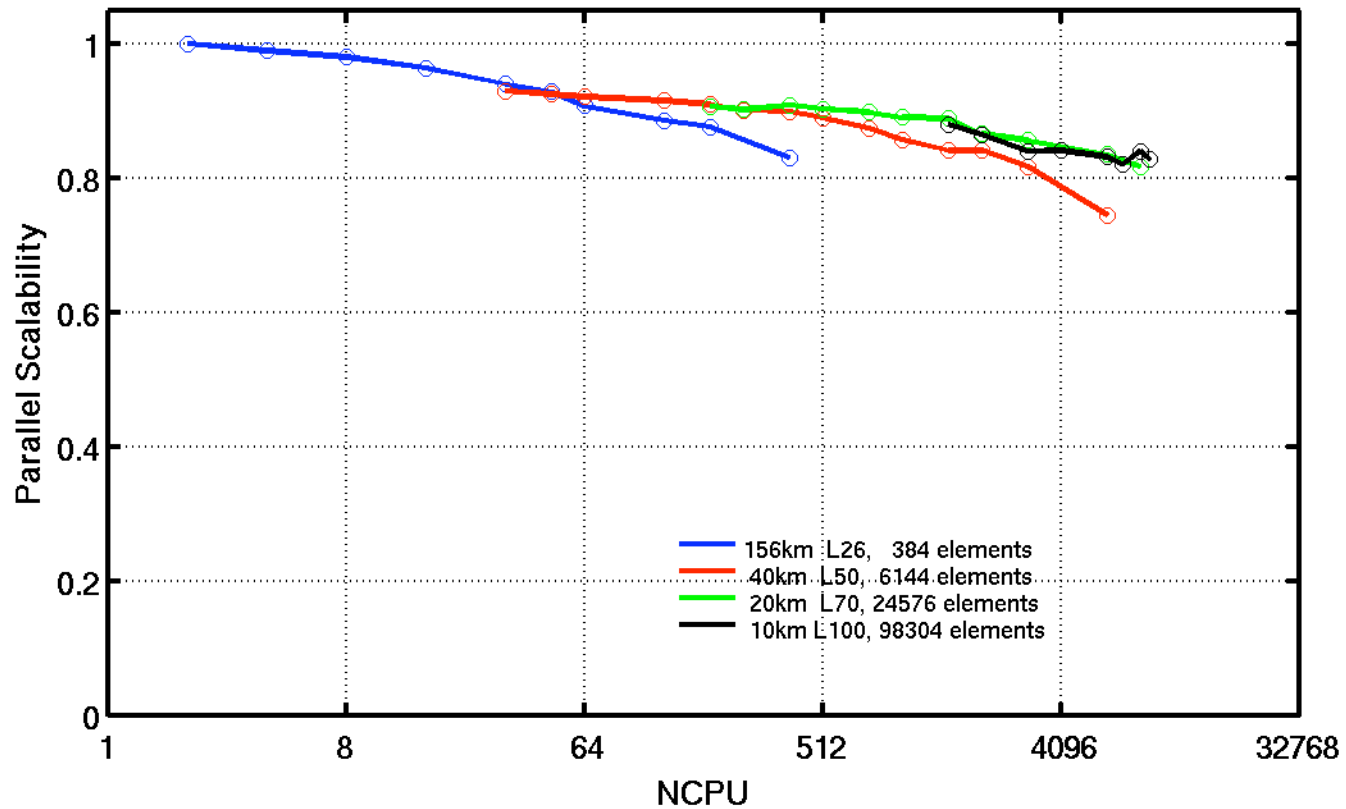
SEAM on ASCI Red



Performance of 4 fixed problem sizes, on up to 8938 CPUs. The annotation gives the mean grid spacing at the equator (in km) and the number of vertical levels used for each problem.



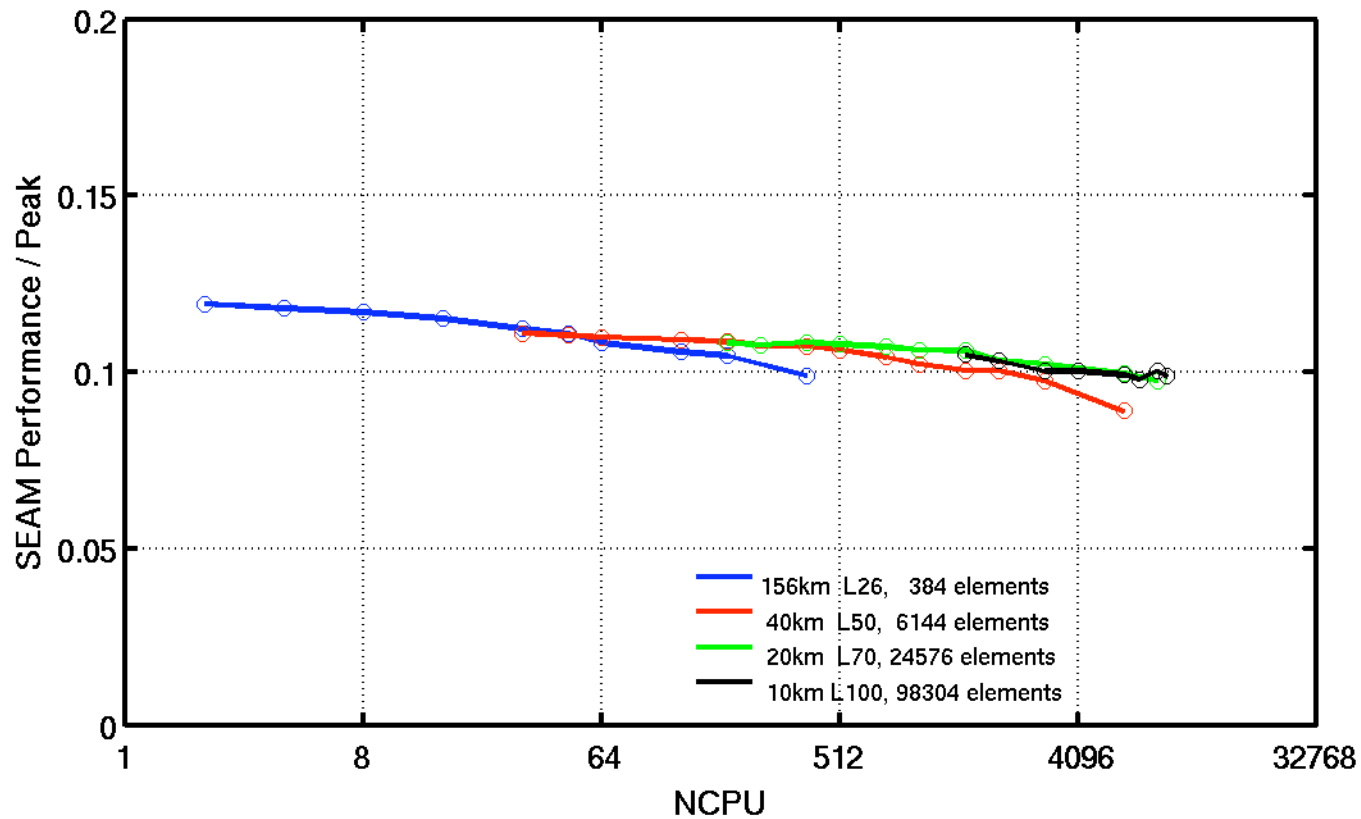
SEAM on ASCI Red



Performance of 4 fixed problem sizes, on up to 8938 CPUs. The annotation gives the mean grid spacing at the equator (in km) and the number of vertical levels used for each problem.



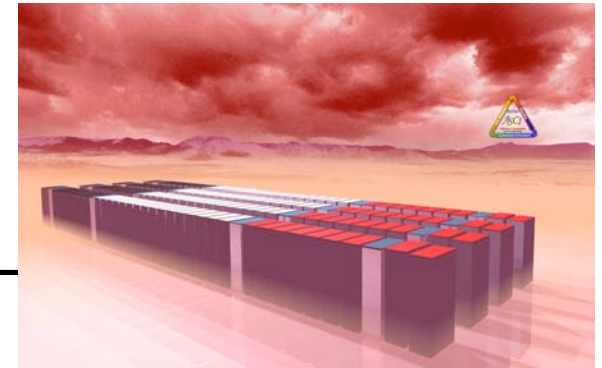
SEAM on ASCI Red



Performance of 4 fixed problem sizes, on up to 8938 CPUs. The annotation gives the mean grid spacing at the equator (in km) and the number of vertical levels used for each problem.



SEAM on Red Storm



- Single processor performance
 - 10-12% Peak: Pentium II, Pentium 4 Xeon
 - 21-25% Peak: IBM Power4, DEC Alpha

- Red Storm Projection
 - AMD Opteron, 64bit, 4GF Peak
 - Assume 25% Peak, Red-like scalability: SEAM ~8TF
 - 10km/100L: Integrate at 34x reality



SEAM on a PetaFlop Computer

- ASCI Red communication/computation balance
- SEAM:
 - Demonstrated scalability to 9000 CPUs
 - Demonstrated scalability to 1 element per CPU
 - Estimate 10km problem should scale to 98,000 CPUs
- 5x more CPUs (50,000)
- 5x faster CPU and memory bandwidth (20GF Peak)
- SEAM: ~200TF
- 10km/100L: Integrate at 870x reality



Earth Simulator



- 640 SMP Nodes, each with 8 vector processors
- AFES Atmospheric Model
 - Global Spectral Model (spherical harmonics)
 - Excellent vector performance (65% of peak on the Earth Simulator) ~24 TF
 - This performance obtainable only up to 640 partitions (at 10km resolution)
 - Resolution: 10km 96L
 - Runs at 57x reality



PetaFlop Earth Simulator

- 5x more SMP nodes
- 5x faster SMP node
 - 5x faster vector processor
 - 5x memory bandwidth
- Retain 8 vectors per node
 - Increasing vectors per node not possible with 5x memory bandwidth increase.
- Global Spectral Model
 - 10km grid limited to 640 SMP nodes: 120TF
 - 10km/96L: integrate at 280x reality

Subject: FW: Info
Date: Friday, May 21, 2004 8:58 AM
From: Ed Barsis <ebarsis@bmv.com>
To: "Peter L. Mattern" <pmattern@bmv.com>

-----Original Message-----
From: John Drake [mailto:drakejb@ornl.gov]
Sent: Friday, May 21, 2004 6:46 AM
To: Ed Barsis
Subject: Re: Info

Ed,

You were working late also.

Nomenclature:

M wave number resolution, eg. TM as in T170
I number of longitudes ($I \geq 3M+1$)
J number of latitudes ($J=I/2$)
L number of vertical levels so the number of points in the
mesh is $I \times J \times L$
P number of nodes or processors (computational unit doing FFT or LT)
For the physics operation counts that is number of floating point
operations (eg exp, **) per column per timestep

The headings for the last performance tables are

M,J,I, P,
Total Gflops,
Simulated years per day,
Time to solution for a century long simulation,
(estimated) Sustained computation rate in Gflop/s using P computational
units.

Since we had a discussion on the meaning of "P" I'll reiterate that.
For the Cray X1, this is an MSP which consists of 4 scalar processors
each with two vector pipes and is peak rated at 12.8 Gflop/s. For the
IBM p690, this is a processor of a 32 processor shared memory node.
Each power4 processor has 4 pipes and is peak rated at 5.2 Gflop/s.

Good luck pulling it all together. I'd be interested in seeing what
you've done. - John

Ed,

We have measured bandwidths of 3-6 Gbytes/s on the Cray X1 for high resolution spectral dynamics. If I enter that number rather than the conservative 1 Gbyte/s used in previous calculations, then the spectral dynamics gets 30% to 40% efficiency. Here are three lines from the dynamics performance model.

M	J	I	P	FFT	LT	Tp	Tcomm	TotTime	S	E	Rate(Gflops)
341	513	256	1024	2.58E-01	2.93E+00	0.00310978	1.22E-02	0.015323238	207.8160629	0.202945374	1276.199984
682	1024	512	2048	1.16E+00	2.34E+01	0.011975273	3.45E-02	0.046439017	528.1196907	0.257870943	3300.203464
1279	1920	1024	3840	4.82E+00	1.54E+02	0.041385225	9.76E-02	0.139016457	1143.168702	0.297700183	7205.258824

bandwidth = 6Gbytes/s

341	513	256	1024	2.58E-01	2.93E+00	0.00310978	9.69E-03	0.012800509	248.7725253	0.242941919	1527.713924
682	1024	512	2048	1.16E+00	2.34E+01	0.011975273	2.44E-02	0.036375145	674.2339935	0.329215817	4213.267182
1279	1920	1024	3840	4.82E+00	1.54E+02	0.041385225	6.23E-02	0.103640841	1533.365239	0.399313864	9664.622025

So with a 3 Gbyte/s bandwidth the T1279 resolution (Japanese ES demo) on 3840 processors would get 30% and with 6 Gbytes/s we would expect 40% efficiency. Of course the numbers we have measured are for much lower processor counts (Pat ran T682 on 32 procs and got 6Gbytes/s). The resolution we are really interested in for climate in the next couple of years is the T341. So we are projecting 20 to 25% efficiency on the dynamics (first lines) and 1024 processors is an attainable number.

The other footnote I'd add is that for higher resolution the dynamics dominates. In fact, the ES demo run at T1279 subcycled the dynamics only calling "physics" intermittently. We have not yet implemented that in our codes because the possibility of running the high resolution cases in a reasonable time is just now presenting itself.

-John

John,

All the entries in the previous table you sent appear to have about the same (low) efficiency. Do you have, and can you send, other data which correspond to "high resolution" and hence 30%-40% resolution?

Thanks

Ed

Subject: RE: [Fwd: Re: Compute requirements for climate]
Date: Friday, May 21, 2004 2:01 PM
From: Boslough, Mark B <mbboslo@sandia.gov>
To: "DeBenedictis, Erik P" <epdeben@sandia.gov>, "Boslough, Mark B" <mbboslo@sandia.gov>, "Spotz, William F" <wfspotz@sandia.gov>, "Taylor, Mark A" <mataylo@sandia.gov>
Cc: "Peter L. Mattern" <pmattern@bmv.com>, Ed Barsis <ebarsis@bmv.com>

Erik,
Thanks for sending Drake's comments. The examples he cites of processes that will require more interprocessor communication (subsurface hydrology and regriding) are local communications only so it doesn't seem to me that it will add that much to the communication. I think he is assuming transform methods when he says the higher resolution increases communication (but not necessarily significant compared to a 10⁴ increase in computation). When he says that addition of more components does not assume the same decomposition as ocean and atmosphere, I think that is more of a load balance issue than a communication one.
Mark

> -----Original Message-----
> From: Erik P. DeBenedictis [mailto:epdeben@sandia.gov]
> Sent: Friday, May 21, 2004 1:47 PM
> To: mbboslo@sandia.gov; wfspotz@sandia.gov; MATAYLO@sandia.gov
> Cc: Peter L. Mattern; Ed Barsis
> Subject: [Fwd: Re: Compute requirements for climate]

>
>
> Gentlemen,
>
> Thank you for discussing compute requirements for climate
> modeling with
> me the other day. You guys noticed that a 10**4 increase in
> local node
> FLOPS would make the application almost embarassingly
> parallel. So today
> I decided to write the authors to see if they had an
> explanation. Their
> response is attached. If you guys are interested, perhaps you
> could read
> it and we could talk about it. It seems to me that John Drake
> is aware
> of the issue and many of the implications.

> Erik

>
>
>

> ----- Original Message -----
> Subject: Re: Compute requirements for climate
> Date: Fri, 21 May 2004 15:10:46 -0400
> From: John Drake <drakejb@ornl.gov>
> Reply-To: drakejb@ornl.gov
> Organization: Oak Ridge National Laboratory
> To: Erik P. DeBenedictis <epdeben@sandia.gov>
> CC: pwjones@lanl.gov
> References: <40967F64.1040908@sandia.gov>
> <40AE380C.2070904@sandia.gov>

> Erik,
> The modeling becoming embarssingly parallel is an interesting
> interpretation, but one I'm not comfortable with. Two current
> experiments would support that point of view, however. By adding
> tropospheric chemistry (with a fairly complete mechanism to
> support air

> quality) one implementation shows a factor of 7 increase. This
> involves no new interprocessor communication other than for load
> balancing and reaction look-up tables. Another experiment is
> performing
> what has been called a super-parameterization to run a cloud resolving
> model within each grid cell of the GCM. Also, embarrassingly parallel
> and could account for much of 100x alone.
> My hesitation is that these are often the kinds of things
> done because
> they will work on machines with poor networks and may not
> represent the
> best (or even correct) way of doing the problem from a scientific
> perspective. For example, the downscaling methods often do
> not provide
> good two way coupling and feedbacks from fine scale to coarse.
> Increasing the resolution of the GCM's clearly doesn't fit the
> embarrassingly parallel, nor does adding more realistic hydrology with
> aquifers and subsurface flows. The addition of more components to the
> model does not always assume the same geographic decomposition as the
> atmosphere or ocean. We are already dealing with independent
> meshes in
> each component and this implies that the interprocessor communication
> could increase due to regridding with the square of the number of
> components.
> In the ScaLeS report we made a point of stating that there are many
> paths toward the development of a comprehensive earth system
> model. But
> like all researchers, climate developers are opportunistic and
> will pick
> the low hanging fruit first. If we say it, it could become a
> self-fulfilling prophecy.
> -John
>

Petascale Applications —

Combustion

Impact of Petaflop-scale Computing: Application — Combustion

	Combustion
<p>Programmatic impact to be gained by access to Petaflop-scale computing -note that we also had to assume continuing algorithmic advances!</p>	<p>We've taken from your chapter the ability to:</p> <ul style="list-style-type: none"> • Predict the level of pollutant emissions, understand the role of larger hydrocarbons, and quantify pressure effects in turbulent flames • Simulate autoignition with realistic fuels in high-pressure turbulent environments • Explore coupling between acoustics and chemistry at the lean flammability limit needed to develop clean and efficient new combustion systems. • Develop predictive models of the growth and oxidation of soot particles LAR • LAR –Yes to the above, and → one of the main points of the chapter is that we are on the verge of simulations that can compute interesting laboratory-scale combustion problems. Petaflop computing would allow us to compute in detail (3D and sufficient chemistry) and directly compare to carefully controlled laboratory turbulent combustion experiments – thus ushering in a new era allowing much faster discovery and validation of understanding needed to impact/invent novel combustion technologies.
<p>Major scientific challenges to be addressed</p>	<ul style="list-style-type: none"> • Develop and validate new chemical mechanisms of increasing complexity and accuracy • Develop models of spray breakup and mixing, radiation properties, and interactions with chemistry, soot, and other particles, and models of reactive interfaces. • Develop and integrate new analysis paradigms for simulation and analysis • Develop detailed simulations that reveal how combustion processes vary across many length scales in turbulent flames. • Extend low Mach number models to include adaptive approaches for closed chambers and techniques for including long wavelength acoustic effects. • Develop scalable algorithms for multi-physics reacting flow. • LAR - Develop reduced representations (lower dimensional models) of dominant modes of turbulent combustion that allow full multiscale simulations with known fidelity (bottom P6). This is a science goal that will benefit from (therefore may not fit in a list of developments required to enable large scale computing) large scale simulations, and will, in turn, enable new approaches in industry (using LES and related tools).
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p> <p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p>	<p>Fig. 3 suggests that the current capability available to combustion modelers is approx. 100 GF/s sustained, or about 1 TF/s peak. A factor of 1000x improvement would be right in the range of interest for our report: approx. 1 PF/s peak. We are looking for a statement that might be something like this: "We did calculations on a 1 TFlops/s (peak) machine, achieving sustained throughput of 100 Gflops/s (or 10% efficiency). The turn-around time is about days."</p> <p>JBB – We have run with 512 processors on Seaborg at NERSC. A single run required 400 hours of wall clock time. At the time the computations were performed we only achieved an aggregate performance of</p>

[We are emphasizing *capability* – the ability to tackle big problems in a single computer run – rather than *capacity* (the amount of work that can be done with many runs.)

13 Gflops/s.

The problem class we are considering is characterized by highly heterogeneous physics and improving scalability is an active area of research within the applied mathematics community.

LAR – Jackie Chen’s simulation on NERSD of 2D H2/air turbulent autoignition at high pressure: NERSC – 9.1 Tflop machine (eff=7-10%, 1000 processors, total sustained Gflops = $1.5 \times 1000 \times 0.8 = 1000$, total run time = 120 hrs for 3 runs, each run was 750,000 time steps at 4 ns (physical scale) simulating 3 ms of autoignition evolution

Would you also indicate how the above example statement would be different for resources normally available and for those simulations shown in Figure 4 -- on page 3 you state that "While these resources exceeded .. normally accessible..."

JBB – Not sure what you are asking here but: The prototype simulation in Figure 4 required 400 hours of wall clock time, which is approximately 3 weeks. If you include the time spent in queues waiting to run, the computation required several months.

If we have interpreted Fig. 3 correctly in light of the values in Figure 2, the programmatic advances that can be anticipated for Petaflop-scale computing can be extracted directly from the text, and have been reproduced above. [Please feel free to improve our paraphrasing or to introduce other concepts.]

On page 8 of the Chapter in SCaLeS II, it is stated that “Turbulent reacting flow computations that resolve the detailed structure of a premixed flame will require approximately 3×10^{16} total FLOPS” First, would you please define "total" and "FLOPS" (is FLOPS the plural of FLOP or is it FLOP/s). Second, please relate this study to the others plotted in Fig. 3, and third, would you please estimate the turnaround time for this computation on a 1 PF/s (peak) platform (we guesstimate the answer to be approx. 300 seconds at 10% efficiency). Please also verify that the number refers to a single run and not an aggregate number of multiple runs.

JBB – FLOPs is the plural of FLOP. The estimate of 300 seconds is correct. Please note that this is a minimalist configuration. Going to a more complex fuel will increase the computational cost by at least an order of magnitude. Going to full multiphase systems will add additional costs (which are hard to estimate)

LAR – Higher fidelity (or more complex) chemistry and multi-phase problems will also benefit from algorithmic improvements

	<p>Also on page 8, you indicate that soot modeling calculations will require approx. 2x over the previous case: would you please estimate the turnaround time for this job, too? (Approx. 600 secs., assuming the same efficiency) JBB – this estimate refers to a micro-scale model of soot particle formation. High-fidelity simulations of sooting flames have much higher computational requirements.</p> <p>If you have any other examples, we would appreciate receiving them. LAR – I have attached a PowerPoint table with some data on currently planned simulations using the Sandia S3D code.....</p>
<p>What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?</p>	<p>These data are not mentioned in the Report. Credible architectures for achieving Petaflop-scale capability within 5 years will contain tens of thousands of processors. The ability for codes to run efficiently with large numbers of processors will be critical. Running today’s codes with large numbers of processors can give useful insights into projected scaling behavior. Please provide us with your experience. LAR – Please see attached PowerPoint indicating linear scalability to over 1000 processors for the S3D code. This code does not make the Low Mach assumption, and so must resolve the acoustic time scales. However, the additional time steps required also accommodate the integration of the detailed chemical kinetics. Because the acoustic coupling is computed, there is no implicit solve over the whole domain that typically suffer more from processor communication and latency. We are currently doing scaling and performance test on other machines. Preliminary results, a 3 GHz Xeon/Infiniband cluster yields 5x speedup over IBM SP2 processors. Scaling is linear to 36 processors with constant load/processor, and is slightly sublinear under constant total (large) load up to 192 processors.</p> <p>JBB – For the low Mach number adaptive codes, we use a broad range of processors, depending on the specific problem. For most 3D applications we use from 128 – 512 processors. We have used as many as 1024. Scaling behavior for the low Mach number adaptive codes is quite good to 128 processors but deteriorates in the 128-1024 range. This deterioration is primarily attributable to poor scaling of linear solvers on large numbers of processors. (The linear systems come from discretizations of partial differential equations on an adaptive grid. They are sparse with considerable specialized structure.) Another factor worth noting here is that limiting factor here is the communications network, not the processor speed. Scaling of the algorithms would improve if the communications were improved relative to the processor speed. As noted above this an active area of research in the applied mathematics community particularly as part of the SciDAC ISIC’s.</p>
<p>What is the Operations Count/Scaling from other computers?</p>	<p>These numbers do not appear in the Report, although an Op-Count is provided on page 8 (see above). To scale performance from today’s machines to larger capability machines requires either:</p>

	<ul style="list-style-type: none"> • An operations count, or • A scaling law based on current performance on current machines <p>If you have used a scaling law to characterize Petaflop-scale performance, please provide the logic used (e.g. compute time scales as n^4, where n is a linear cell dimension), along with the current computer capability used in the scaling. In both cases please provide the required turn-around time for the longest runs.</p> <p>LAR – We do not have a credible scaling law to Petaflop computing yet for S3D.</p> <p>JBB – Operation counts are determined from hardware performance measurements. The inherent scaling law for adaptive methods for flame simulations is between n^3 and n^4 where n is the linear system size, depending on details of the problem. We can make similar estimates for changes in the complexity of the chemical models. The op-count estimates above are based on this type of extrapolation. Again note that the 3×10^{16} is a simplest realistic baseline case.</p>
Projected increase in algorithm efficiency?	<p>If you are counting on an increase from better algorithms, as indicated in your chapter (historically, algorithm improvements have approximately matched improvements in hardware), please indicate the factor you've used.</p> <p>LAR – we did not quantify this in detail, but we are counting on advances at a rate equivalent to the past. Future advances we can foresee are adaptive chemistry, high-order AMR, low mach scaling and load balancing</p> <p>JBB – We have some experience with several aspects of this. Typically, AMR improves things by about a factor of 10. Using a low Mach number results in an improvement of about 100. A reasonable guess for going to a higher-order AMR would be another factor of 20 or so.</p> <p>JBB – Scalable solvers, particularly for the types of sparse systems arising from discretization of partial differential equations is key area where we are relying on improved algorithms. I believe that this will be a critical item across a broad range of application areas.</p>
Other	<p>In response to the question in your email:</p> <p>1)> computer capability, based on Figure 3 that indicates a sustained capability of 0.1 TF/s. However, we do not understand the connection with Figure 2, which appears to put the current sustained capability at 1TF/s. We</p>

would also like you to confirm that the word "effective" means sustained and that the simulations are for the longest single runs and not the aggregate number

Ans: Yes, effective means 'sustained' based on efficiencies determined by benchmarks on NERSC (compressible code efficiency is 10% max)
Fig. 2 - the line represents the effective flops available using the WHOLE MACHINE (i.e. NERSC RS/6000 = ~10TFlops x efficiency of 10%=1000 eff GFlops), combustion runs normally get about 10% of the whole machine. The algorithmic improvements represent increased capabilities derived from algorithms, and plotted as if they were achieved by additional computer power. Thus, this plot is 'make believe' and represents in a very rough way the size of computer required to do the current science if no algorithmic improvements were made.

I have attached the table you sent with comments highlighted. Bold-> from experience with Sandia compressible DNS code (S3D), RED -> John Bell's experience with low-Mach number code with AMR.

I have also attached a PowerPoint slide with some scaling data for the S3D code and a table of simulations planned in the near future.

1. The data below are estimates for computing a premixed laboratory flame at atmospheric pressure at low turbulent intensity. Our experience to date allows us to make fairly reliable predictions in this type of regime.

The 300 second figure represents a minimalist representation of carbon chemistry for methane

Using a comprehensive mechanism for methane carbon chemistry would increase the time to about 900 seconds.

Including nitrogen chemistry for computation of emissions would increase the time to about 2400 seconds.

Changing the fuel to propane would increase the time to about 7200 seconds.

Changing the fuel to heptane (a surrogate for diesel fuel) would increase

the time to 108000 seconds (30 hours)

2. If I scale the turbulent intensity to more realistic conditions i
increase the overall compute time by about a factor of 1000.

so simple methane becomes 80 hours comprehensive methane carbon chemistyr
become 240 hour

etc.

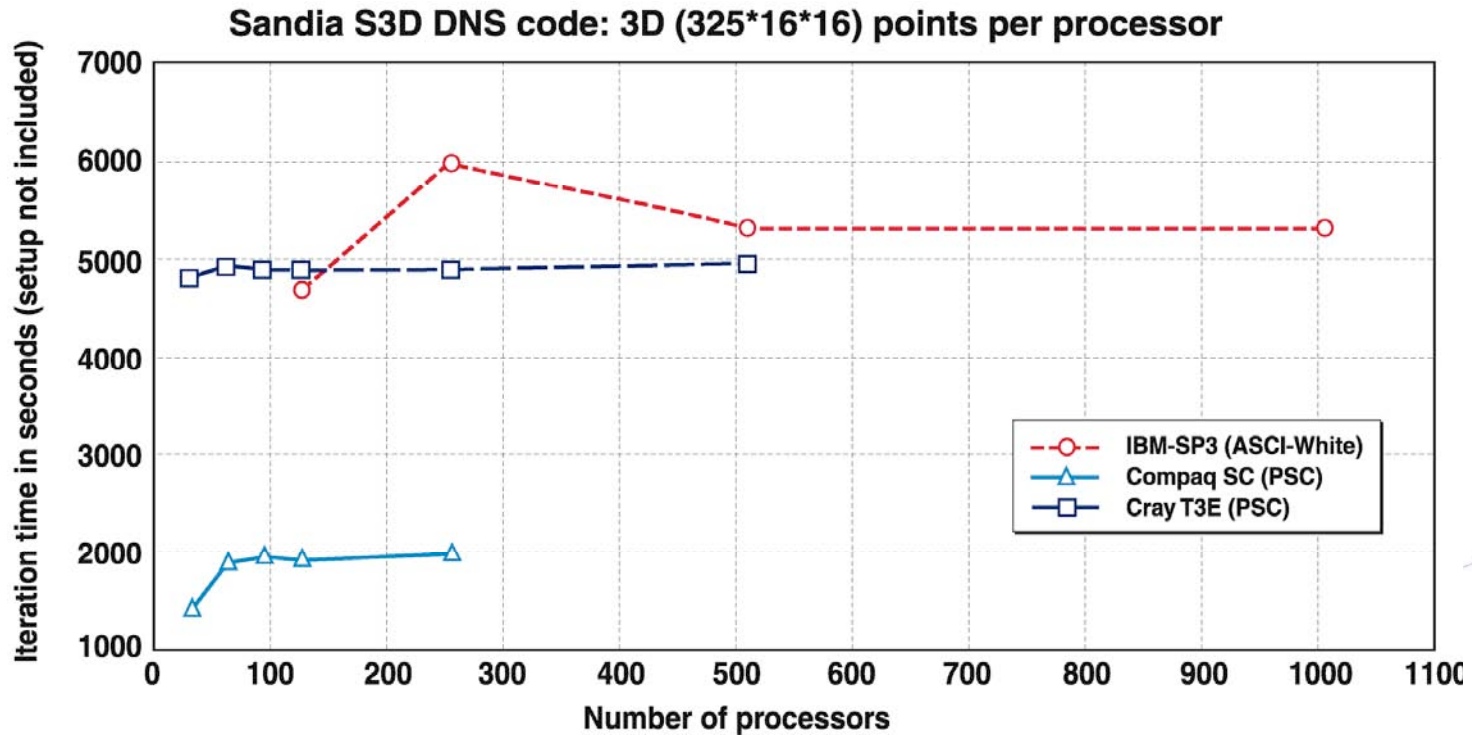
As noted, there are multiphase effects and pressure effects we would also
like to be able to model, each of which would increase computational time
(in some cases dramatically) but I am less certain of how to estimate the
costs.

Hope this helps

Regards

-- John Bell CCSE, MS 50-D, LBNL 1 Cyclotron Rd., Berkeley, CA 94720 Phone:
510-486-5391 Fax: 510-486-6900

3D DNS Code (S3D) scales to over a thousand processors



Scalability benchmark test for S3D on MPP platforms - 3D laminar hydrogen/air flame/vortex problem (8 reactive scalars)

- Ported to IBM-SP3, SP4, Compaq SC, SGI Origin, Cray T3E, Intel Xeon Linux clusters

Ed,

Jackie and I went over her table of science questions, and sized the problems for what an optimal attack of the science question, rather than the

computer. This assumes scaling up from our experience with a cluster of 3 Ghz Xenons (our best performance yet). The S3D code has not been instrumented in detail yet, so the efficiency cannot be specified with any certainty, but it is likely in the 10% range you reference for PetaFlop computing. The extrapolation to Petaflop computing assumes the same linear scaling we have seen to the TeraFlop (192 processors) level, and similar efficiencies for the processors and interconnect, memory access, etc etc. The last problem - 3D autoignition with N-Heptane - comes out to 11,000 hours, but this estimate does not include improvements in the algorithms that

we assumed when we put this as accessible with $1e5$ GigaFlops effective on Fig 3. These computing requirements are PER RUN, with several runs (e.g. with different parameters) deemed necessary usually.

Hope these examples are helpful to you!

larry

New Science	DNS Configuration	Grid points	GFlop-hrs Using S3D 192x 3Ghz Xenon infiniband benchmark	PetaFlop-hrs assuming no algorithmic improvements and similar scaling and efficiencies
What is the flame structure in the thin-reaction zone regime of premixed combustion?	3D stationary turbulent premixed methane-air and hydrogen-air flames, (2X2X2 cm ³)	3.4E09	2 E08	200
What is the flame stabilization mechanism?	3D turbulent methane jet flame (near field) (10x3x3 cm ³),	6.3E09	1.6 E09	1600
How do pollutants depend on mixing in a turbulent jet flame?	3D turbulent CO/H2 jet flame (near field) (36x15x15 cm ³)	9E10	5.4 E08	540
How does turbulent mixing affect multi-stage ignition of hydrocarbon fuels at high pressure?	3D turbulent autoignition of n-heptane with compression heating (1.2 cm ³)	1.36E10	1.1E10	1.1E4

Petascale Applications —

Environment

Impact of Petaflop-scale Computing: Application — Subsurface Transport and Fate

	Subsurface Transport and Fate
Programmatic impact to be gained by access to Petaflop-scale computing	<p>The general Scientific Opportunities have been well laid out in this chapter. However, the opportunities that are within the reach of Petascale capability computing of a few tenths of a Petaflop/s peak to a few Petaflop/s peak might possibly be in excess of what is needed for the advances listed on page 4 for a 1000x increase in capability. The ASCI-Red example (256 processors is about 0.1 Tflop/s peak) implies that scientists in this field do not regularly have access to machines that are 1/1000 of a Petaflop/s peak (or about 1 Tflop/s peak). We also note in this regard that the 4 Petaflop example given in Sidebar 1 would execute in about 40 seconds on a 1 Petaflop/s computer assuming 10% efficiency. (Granted, the 10 zettaflop example would take about 3 years.) In any case, please tell us the current capability assumed (see box 3 below).</p> <p>The allocation of high performance computing resources for subsurface science has not been a priority in the Office of Science. Unfortunately, computational needs in this field are genuine, scaling easily to Tflop size computing. As an example, work in high- resolution 3D geophysical imaging of the subsurface. Typical jobs require about 250 processors, with some high end jobs requiring 1000 processors on the ASCII RED platform, running months to complete the analysis; access to this platform is a special situation and could not be replaced with other machines managed by the Office of Science, given the current allocation of computational resources to subsurface science. This computer crunch comes at a particular bad time in subsurface science. As an example, new opportunities are beginning to emerge for high-resolution 3D subsurface imaging in hydrocarbon and geothermal resource evaluation and environmental site characterization. Solution to these problems is of interest to both industry and government and really does require high-end computational resources. Without adequate resources, work in these problems will be delayed, retarded or even declined. Sufficient computational resources are needed to exploit these new and exciting opportunities.</p> <p>In this box, and the following one, we have taken programmatic and scientific impacts (advances) from your report assuming that an increase of 1000x in capability corresponds to Petascale capability (which may not be the case). If not, please correct the entries in this box and the next box. Also, please feel free to improve, in general.</p> <ul style="list-style-type: none">• Significantly reduce the uncertainty of predictions through "upscaling." Translation: understanding how smaller scale processes and properties should be represented at a larger scale of interest.• Simulate regional ecological impacts. Translation: coupling of comprehensively detailed multiphysics models: groundwater + vadose zone (unsaturated soil zone between land surface and groundwater) + land surface hydrology + river + atmosphere + ecosystem that include multiple fluid phases (gas + liquid [aqueous + nonaqueous] + solid) + biology + chemistry + ecological. Methodology must address a large range of time and space scales that are specific to each process.• Simulate long-term, large-scale, 3-d, high-resolution, 3 phase, multi-fluid flow and multi-component reactive transport.

	<ul style="list-style-type: none"> • Quantify uncertainty in model predictions. While parameter uncertainty can often be quantified by estimation error, the parameterization is based on the use of a specific algorithm representing a selected conceptual process. The largest errors, however, are typically in the selection of conceptual process models and are not typically quantifiable because you don't know the impact of what you haven't modeled. Computationally-efficient, large-scale simulations of complex subsurface phenomena allow a quantitative assessment of alternative conceptual process models in the context of the field-scale system of coupled processes. • We don't quite understand how the computing capability would be used for inversion of real-time multisensor data. Please add this programmatic advance. The analysis of multi-sensor data sets for imaging complex geological systems has been a time consuming and tedious process. Current analysis of data sets has required months to complete using Tflop computational platforms, with hundreds to thousands of processors working on the problem. Imaging the subsurface is a complex task, involving multiple solutions of constrained nonlinear optimization problems. These problems must be solved before one can effectively appraise complex geological systems and determine how noise in our multi-sensor data propagates into our subsurface models. Attacking this problem on a petaflop scale will reduce the analysis of multi-sensor data from months to days, approaching the elusive goal of real time multi-sensor data inversion. This advance will allow for faster and more accurate characterization of complex geological systems, saving significant time and money. • Pore-scale simulation of multiple domains in small volumes (10 cm cube).
Major scientific challenges to be addressed	<p>These scientific challenges match the programmatic impacts noted above, and may also need to be modified as discussed in the box above. Also, please feel free to improve, in general.</p> <ul style="list-style-type: none"> • Link and integrate research at different length scales to allow the development of reliable subgrid parameterizations. • Couple groundwater, vadose zone, watershed, river, meteorological, and ecological processes. • Develop stochastic simulations of conceptual models. One approach is to statistically evaluate hundreds of conceptual models (e.g., geometries of geologic units) with hundreds of realizations (e.g., hydrogeologic parameters) for each conceptual model to understand the range of behavior and uncertainty at a given subsurface site. • Parameter estimation through large-scale inverse modeling. This is typically a large-scale minimization/optimization problem with potentially millions of parameters being estimated simultaneously to best match observations. By increasing the amount and types of observations, more robust parameter selections are possible. • Long-term simulations of 3-D, three-phase (gas, aqueous, non-aqueous) fluid flow in highly-resolved heterogeneous porous media • Long-term simulations of 3-D flow, transport, and multicomponent chemical and microbial reactions in highly-resolved heterogeneous porous media. • Please include scientific advances we may have omitted.

<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p> <p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p>[We are emphasizing <i>capability</i> – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)]</p>	<p>These type of throughput data are not directly indicated in the report. However, the kind of answer we are looking for might be something like this: "We did calculations on a 5 TFlops/s (peak) machine, achieving sustained throughput of 0.5 Tflops/s (or 10% efficiency). The turn-around time is about days." We have run a single subsurface reactive transport simulation on 256 processors with a 15% efficiency on the 6 GFLOPS peak processors. This particular run ran in less than one day. This level of scaling has generally been maintained with increasing problem size from 4 processors up to 256. Our expectation is that we can extrapolate that efficiency across the 1960 processor machine (11+ TFLOPS)</p>
<p>What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?</p>	<p>These data are not mentioned in the report (except for the ASCI-Red example in Sidebar 2). Credible architectures for achieving Petaflop-scale capability within 5 years will contain tens of thousands of processors. The ability for codes to run efficiently with large numbers of processors will be critical. Running today's codes with large numbers of processors can give useful insights into projected scaling behavior. Please provide us with your experience. Testing on a subsurface multiphase flow and reactive transport simulators has gone up to 256 Itanium-2 processors. In the near future, a 1000 processor job will be performed for simulating multifluid compositional flow of a historical mixed waste discharge on the Hanford Site with carbon tetrachloride, lard oil, and aqueous phase co-contaminants. The 3-D simulation resolves multiple length scales of heterogeneous materials and simulates 50 years of operations, including a soil vapor extraction remediation.</p>
<p>What is the Operations Count/Scaling from other computers?</p>	<p>To scale performance from today's machines to larger capability machines requires either:</p> <ul style="list-style-type: none"> • An operations count, or • A scaling law based on current performance on current machines <p>The Subsurface Transport and Fate chapter presents scaling information for the upscaling example on page 4. Would you please provide examples for one or two of the other cases mentioned on page 4. If you have used a scaling law to characterize Petaflop-scale performance, please provide the logic used (e.g. compute time scales as n^4, where n is a linear cell dimension), along with the current computer capability. Please also provide the required turn-around time for the longest and typical runs.</p> <p>There should be no problem upscaling the subsurface imaging applications from Tflop to the Petaflop scale. Currently, the domain is decomposed over a bank of processors. However, we are now considering a data decomposition over another processor bank with subsets of processors in this bank having copies of the decomposed domain. This type of decomposition is nearly perfectly scalable, and will allow for the analysis of large scale multi-sensor data sets and imaging of complex geological systems at an unprecedented level and scale of resolution.</p>
<p>Projected increase in algorithm efficiency?</p>	<p>If you are counting on an increase from better algorithms (historically, algorithm improvements have approximately matched improvements in hardware), please indicate the factor you've used. We didn't include</p>

	this in the estimates.
Other	

A petaflop computational application arises in subsurface 3D geophysical imaging using electromagnetism. Applications arise in environmental site characterization, oil and gas and geothermal resource evaluation. In large-scale applications meshes are on the order 270 million elements, resulting in sparse linear systems of 800 million complex unknowns. These systems represent the discretization of the time harmonic Maxwell equations, and are solved using iterative Krylov subspace methods over a distributed computer system. The solution of a single problem, fixed transmitter at a given frequency, typical requires 2000 matrix-vector multiplies, resulting in 10^{14} floating-point operations. In subsurface imaging applications, the problem (also called the forward problem) must be solved multiple times. Consider first the case of a single transmitter, operating again at a specific frequency. The imaging process using a gradient optimization methodology, requires, 4 solutions of the forward modeling problem per inversion iteration. On average 50 inversion iterations are required to achieve an acceptable data fit resulting in 200 solves of the forward modeling problem, costing on the order 2×10^{16} floating-point operations. This result is for only one transmitter and frequency. In order to realistically image the subsurface at an unprecedented level of resolution and detail requires that we employ hundreds to thousands of transmitters in the imaging process. As an example, consider 50 transmitters at sixteen harmonics, would then require on the order of 1.6×10^{19} floating point operations, Fortunately the imaging problem with multiple transmitters can also be distributed over banks of processors, where within each bank, reside copies of the model discretization problem. This data decomposition is highly parallel, where global communication amongst the various data banks, only needs to be done several times per inversion iteration in order to complete several dot products. The main computational burden occurs with the forward solves, which are independent for each transmitter and frequency.

If we envisage a petaflop platform applied to this problem, we can estimate the time needed for a single solution to the inverse problem (assuming ten percent machine efficiency due to data IO, which cannot be ignored) at

$$t = 1.6 \times 10^{19} / (0.1 * 10^{15}) = 160,000 \text{ sec or } 44.44 \text{ hours.}$$

The imaging software needed to solve such a problem has been under development over the last decade, and it is ready to verify the estimate, mentioned above, given availability petaflop computational resources.

It is critical to note that inversion or imaging can be considered in two phases, solution construction and appraisal. In the construction phase, the imaging problem has been solved once. Solution appraisal on the other hand requires that we understand how data errors propagate into the model. It is also used to test different regularization parameters, using a global line search, to determine the optimal tradeoff between the data fit and model constraints. These model constraints are required to regularize or stabilize the inverse problem, without which superior data fits can be achieved, but at the expense of an image of the subsurface that bears no resemblance to subsurface geological structures or processes. The appraisal process requires hundreds to thousands of solution samples of the inverse problem. Hence the computational cost of solving the inverse

problem at the fine scale envisaged is enormous, and not really practical without computational resources at the petaflop scale.

Subject: Re: help in justifying future DOE "capability" systems for environmental modeli
Date: Thursday, May 20, 2004 11:19 AM
From: M. Wheeler <mfw@ices.utexas.edu>
To: Greg Newman <GANewman@lbl.gov>
Cc: <ebaris@bmv.com>, <yabusaki@pnl.gov>

On Thu, 20 May 2004, M. Wheeler wrote:

> Here is an example of interest to DOE .. Do you need any additional
> information.
>
> Example CO2 sequestration with chemical reactions. The system needs to
> be modeled for 25 years which involves 1000 time steps (coarse
> estimate)
> The model will involve solving coupled compositional three phase flow and
> geomechanics plus 4 chemical species. Here the # of unknowns is 3 for
> displacements, 3 fluid phases + 4 chemical species = 10 unknowns per
> element. With a grid of 500x500x50 = 12.5M elements per time step
>
> Solver iterations 5 Newton x 100 linear iterations = 500 total iterations
> per time step.
>
> Thus we have 12.5M X 12.5M X 10 unknowns X 1000 X 500 = 7.8 10²⁰
> operations.
>
> The above example does not even include optimization and uncertainty
which
> involves solving the above system 1000s of times..
>
> Good Luck.. Mary W

Petascale Applications —

Fusion/Plasma Science

Impact of Petaflop-scale Computing: Application — Plasma Science

	Plasma Science
<p>Programmatic impact to be gained by access to Petaflop-scale computing: (approx. 1 PF/s peak)</p>	<p>Today's largest calculations require about 3×10^{16} operations. This is calculated as: 80 hours x 1024 processors x 100Mflops x 3600 s/hr. Note that this is about 30 seconds on a computer that actually delivered a Petaflop sustained performance. If we could run such jobs for 8 hours, this would be a factor of 1000 increase. These simulations can calculate the turbulence due only to ions (with a simplified electrostatic, adiabatic approximation for the electron response) for a time period of about 1 millisecond, or isolated macroscopic stability events in some of the smallest experiments today using the actual parameters of those experiments.</p> <p>Petaflop-scale capability will enable ...</p> <ul style="list-style-type: none"> • The simulation of macroscopic stability phenomena in some present-day fusion experiments that do not include burning plasma, including the onset conditions, strength, and non-linear saturation mechanisms. • The increase in simulated time of an ITER discharge to 1 sec (approaching the energy confinement time), keeping the simplified electron model. • Resolution of ion, electron, and electromagnetic-scale interactions for small, present-day experiments. • The carrying out of global space weather simulations that couple large solar-terrestrial scales to much smaller scales involving ion dynamics using realistic parameters (Lundquist numbers $S < 10^6$). <p>Modest additional increases (approx. 10x) in capability will be required for the most complete models (two-fluid MHD and electron/ion/electromagnetic turbulence) in the largest present-day plasma fusion experiments, whereas burning plasmas (ITER, FIRE) remain over the foreseeable horizon.</p> <p>The biggest programmatic step will be a comprehensive, integrated simulation bringing together all of the sub-disciplines in fusion simulation and able to predict reliably the behavior of plasma discharges in a toroidal magnetic fusion device on all relevant time and space scales. This will require computation at the petaflop level, as well as gains in algorithms for multi-scale nonlinear problems.</p>
<p>Major scientific challenges to be addressed</p>	<p>To achieve the above impacts, modelers must ...</p> <ul style="list-style-type: none"> • Increase the dimensionless parameter characterizing inverse plasma collisionality in the macroscopic simulations by several orders of magnitude, and include other, extended-MHD effects, in the macroscopic fluid models. • Include full electron/ion physics, including full electromagnetic effects, in the global micro-turbulence models. • Improve the RF models to include time-domain effects, plasma sheaths and other edge effects, and

	<p>direct coupling of the plasma to the antenna.</p> <ul style="list-style-type: none"> • Understand better micro- and macro-instabilities and their implications for plasma stability and transport.
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p> <p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p>[We are emphasizing <i>capability</i> – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)]</p>	<ul style="list-style-type: none"> • Our codes today typically get 100Mflops per processor on the NERSC machines, and scale adequately up to ~1000 processors. • Experience with the Cray X1 with fusion codes is limited, but an all-orders plasma wave code has achieved about 600 Mflops per processor on 128 processors. (= 128 * 12.8GF)
<p>What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?</p>	<p>See above</p>
<p>What is the Operations Count/Scaling from other computers?</p>	<p>See above</p>
<p>Projected increase in algorithm efficiency?</p>	<p>This is very speculative and we cannot predict. If we knew what the better algorithms were, we would be using them already.</p>
<p>Other</p>	

Subject: FW: FW: help in justifying future DOE "capability" systems for Plasma Science
Date: Monday, May 17, 2004 9:16 PM
From: Ed Barsis <ebaris@bmv.com>
To: "Peter L. Mattern" <pmattern@bmv.com>

Peter,
Info
Ed

-----Original Message-----

From: W.M. Nevins [mailto:nevins1@llnl.gov]
Sent: Monday, May 17, 2004 8:00 PM
To: Ed Barsis
Cc: jardin@pppl.gov
Subject: RE: FW: help in justifying future DOE "capability" systems for Plasma Science
Importance: High

Ed,

The most likely use of the computer you describe would be to turn what are today "heroic" runs (using 1000 processors on NERSC for 90 hours) into routine runs -- with no further parallelization we would be using 1% of the computer for 9 hours. This would, in fact, make a big difference because it would enable parameter scans -- which is what you REALLY learn physics from.

We could also get some benefit from using more processors by running problems which required greater spatial resolution -- trapped electron modes, which require up to 10x more grid points (leading to the use of ~ 10,000 processors. We hadn't yet exhausted the benefits of parallelization at 1,000 processors, so there is some reasonable prospect that we could successfully develop code which would benefit from 10,000 processors on a larger problem (our experience is that there is no gain in speed if you don't expand the size of the problem with increases in the number of processors used). The time-scales are not hugely different, so this would be ~10 hour run on 10% of your hypothetical computer (so we could STILL afford to do parameter scans and really learn something).

Sorry if I'm not providing the "right" answer ... but my experience is that you learn far more from sequences of "routine" simulations than from a very few "heroic" simulations. I remain enthusiastic about what can be learned with expanded computer resources, but am reluctant to project our success on problems using up to 1,000 processors by two more orders of magnitude.

-Bill Nevins-

Subject: RE: FW: help in justifying future DOE "capability" systems for Plasma Science
Date: Monday, May 17, 2004 7:25 PM
From: Ed Barsis <ebarsis@bmv.com>
To: "Stephen C. Jardin" <sjardin@pppl.gov>
Cc: nevins-llnl-gov-offsite <nevins@llnl.gov>, Don Batchelor <batchelordb@ornl.gov>, "Peter L. Mattern" <pmattern@bmv.com>

Steve,
We seemed to have slipped a cog in the communications. I was trying to imply that a Petaflop (peak) computer in about five years would have about 100,000 processors each running at about 10Gflops/s. Would this less pessimistic design (rather than the 1,000,000 processors noted in your response) make a difference in the projected ability of "Plasma Science" to take advantage of such a computer?

Ed

Steve,

Thanks.

For your info, there is a time parallelization method for iterative solvers. Basically, the 2nd time step is started before the first time step is complete, and the third time step is started before the 2nd is complete, etc. So when the 1st time step is complete a smaller amount of work (and time) is taken to complete the 2nd time step, etc. To my knowledge, the work was first done and published by Dave Womble at Sandia.

Thanks again for the response.

Ed

-----Original Message-----

From: Stephen C. Jardin [mailto:sjardin@pppl.gov]
Sent: Sunday, May 16, 2004 3:40 PM
To: Ed Barsis
Cc: nevins-llnl-gov-offsite; Don Batchelor; Peter L. Mattern
Subject: RE: FW: help in justifying future DOE "capability" systems for Plasma Science

There are a lot of clever people out there, so I would not say that anything is impossible, but scaling up to 500,000 or 1,000,000 processors is a huge leap.

This will be especially difficult because you are talking about "strong scaling", ie, leaving the number of grid points and particles fixed, and just increasing the number of processors to get faster running time so you can run for more time steps. At some point, the interprocessor communication begins to dominate and just adding more processors doesn't do any good.

We always say the problem is "you can't parallize over time" because of causality, etc. However, it may be that some clever algorithms will emerge with clever, highly paralleliizable techniques for reaching steady state that somehow solve this seemingly fundamental problem.

Also, I think there may be some other issues in just taking the .1-1 ms code, and running it to 1 sec. They make some assumptions regarding the change in the distribution function be small compared to the original distribution

function, and that would certainly be violated. Again, this could probably be overcome.

When more physics is added to this code, it will only scale worse, since the new physics involves adding elliptic equations each time step, which don't scale as well as the particle advance equations.

I hate to be a nay-sayer, but I would not like to promise that this code could productively use 1,000,000 processors to get long-time simulations.

-steve

From: Ed Barsis [mailto:ebarsis@bmv.com]
Sent: Sat 5/15/2004 1:51 PM
To: Stephen C. Jardin
Cc: nevins-llnl-gov-offsite; Don Batchelor; Peter L. Mattern
Subject: RE: FW: help in justifying future DOE "capability" systems for Plasma Science

Steve,

Thanks much, Don did indeed provide us the filled-in table.

We have a question about the information that was provided. Regarding the ability to extend the current calculations ("keeping the simplified electron model") from 1 millisecc to 1 sec, we would like to know if that could be done with the type of computers likely to be available in the next several years: compared to what you are using now, it is likely that the processor operating frequency will increase by perhaps a factor of 2-4, and that the remaining factor of 1000x more capability will come from more processors (eg 500x-250x more processors). Some of these processors might be called "cores," share memory with other "cores," and be packaged (and sold) as a single processor. Nevertheless, these would be more processors. Is there an inherent reason why simulation of 1 second could not be obtained with this type of computer. It would not be possible to simply run the current code faster because the clocks would only be running 2-4x faster and not 1000x faster. (We say "inherent" reason because we know a great deal of work would be needed to do the additional parallelization, to include other algorithms....

If in fact, a factor of 1000x capability could not be used in this way, then would you be able to use a 100x, ie would 0.1 sec be significant? Or is there some other high impact of 1000x capability?

Thanks again for your help.

Ed, from both of us

Again, thanks much for your help.

Massively Parallel Kinetic Simulation

William Daughton
Plasma Physics Group, X-1

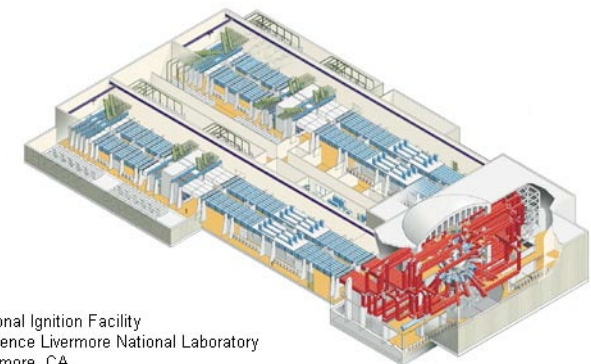
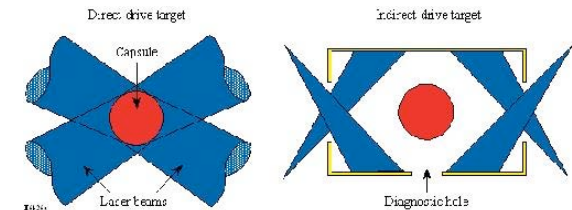
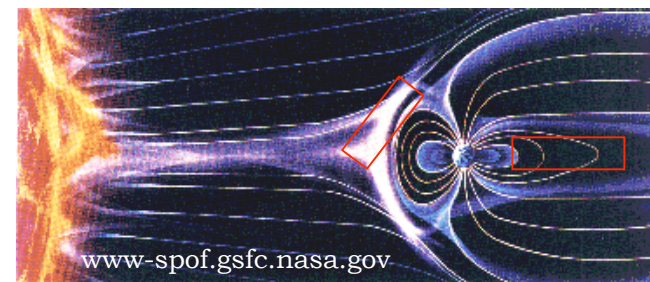
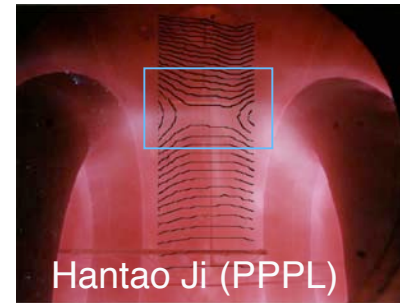
May 19, 2004



Kinetic Simulation of Plasmas

Applications within X-1:

1. **Magnetic Reconnection - Basic process in space, astrophysical, and laboratory plasmas**
2. **Laser Plasma Interactions - Need to understand the importance of parametric instabilities in ICF laser experiments such as NIF**
3. **X-Ray Radiography - Propagation and interaction of electron beam with target (DART)**

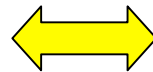


What is a PIC simulation?

- **PIC = particle-in-cell**
- **Statistical approach for solving Vlasov-Maxwell**

Vlasov

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \frac{\partial f_s}{\partial \mathbf{x}} + \frac{q_s}{m_s} \left(\mathbf{E} + \frac{\mathbf{v} \times \mathbf{B}}{c} \right) \cdot \frac{\partial f_s}{\partial \mathbf{v}} = 0$$



Maxwell

$$\begin{aligned} \nabla \cdot \mathbf{B} &= 0 & \nabla \times \mathbf{B} &= \frac{4\pi}{c} \mathbf{J} + \frac{1}{c} \frac{\partial \mathbf{E}}{\partial t} \\ \nabla \cdot \mathbf{E} &= 4\pi\rho & \nabla \times \mathbf{E} &= -\frac{1}{c} \frac{\partial \mathbf{B}}{\partial t} \end{aligned}$$

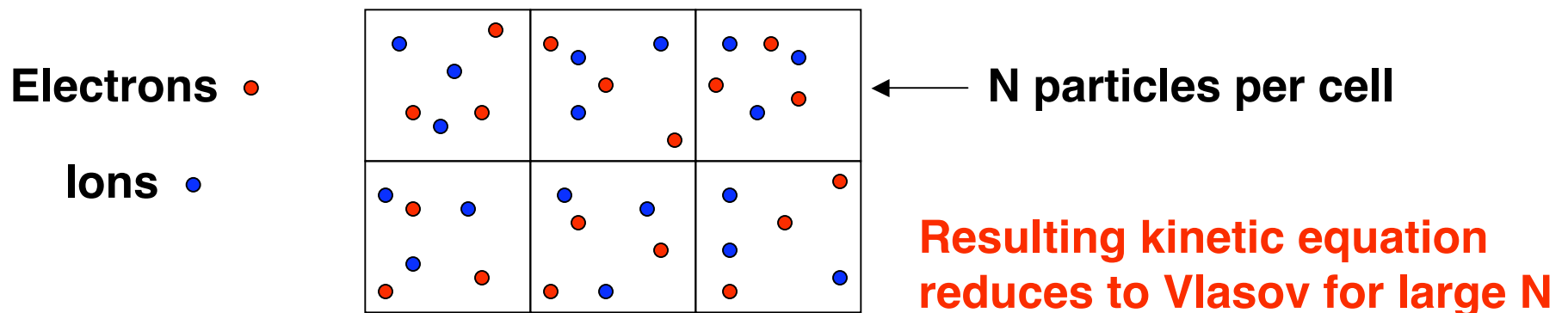
- **Coupled by first 2 moments** $\rho = \sum_s q_s \int f_s d\mathbf{v}$ $\mathbf{J} = \sum_s q_s \int \mathbf{v} f_s d\mathbf{v}$
- **Complete description of collisionless plasmas**
- **Possible to add collisions (difficult to do rigorously)**

How to solve Vlasov equation?

1. **Vlasov Code** - Discretize Vlasov equation directly using finite difference, finite element or spectral approach. Requires discretization of velocity space which is very difficult and can introduce large dissipation.

2. **PIC Method** - Statistical approach

- Introduce “super-particles” - small chunk of phase space
- Create spatial grid (cells)
- Interpolate position and velocity of particles onto grid $\rightarrow \rho$ and \mathbf{J}
- Compute resulting E and B fields
- Push particles using these self-consistent E and B fields.
- Evolution of this system obeys a kinetic equation



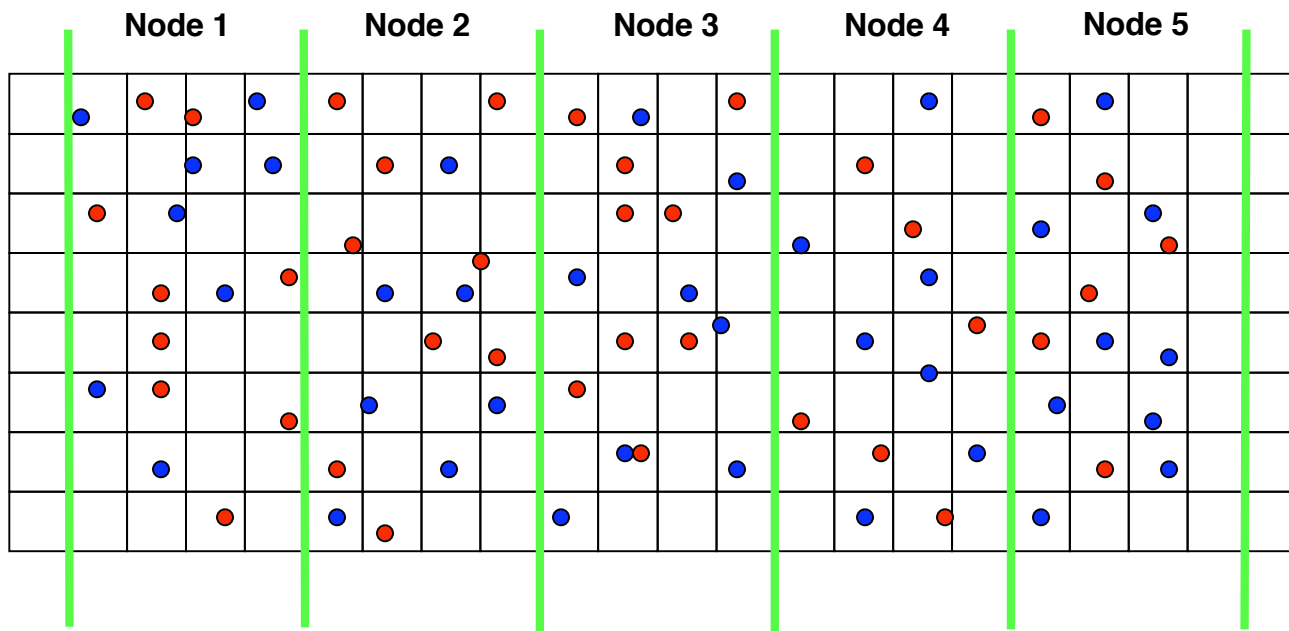
What problems can you solve with PIC?

- Complete theoretical description of a collisionless plasma
- Must resolve all space scales in the Vlasov-Maxwell system
- **Computationally very expensive:** Justified only for problems in which electron kinetic effects are thought to be important
- In space plasma physics, problems simulated with PIC are often on ion scales (current sheets, collisionless shocks, etc)
- Artificial mass ratio to reduce scale separation $m_i/m_e = 100 - 200$
- Cost scales as $(m_i/m_e)^2$ for 2D and $(m_i/m_e)^{2.5}$ for 3D
- Massively parallel computers are required - but are nowhere near fast enough to use real mass ratio in 3D
- **Comparison with theory and/or reduced models is essential to understand and interpret a PIC simulation**

PIC is well suited to parallel computers

Most expensive step is particle push + collecting moments

Use Domain Decomposition



Communication is required to exchange particles between nodes and for the field solve but scaling with processor number is generally very good in these codes.

Need More Computing Power

Two Examples:

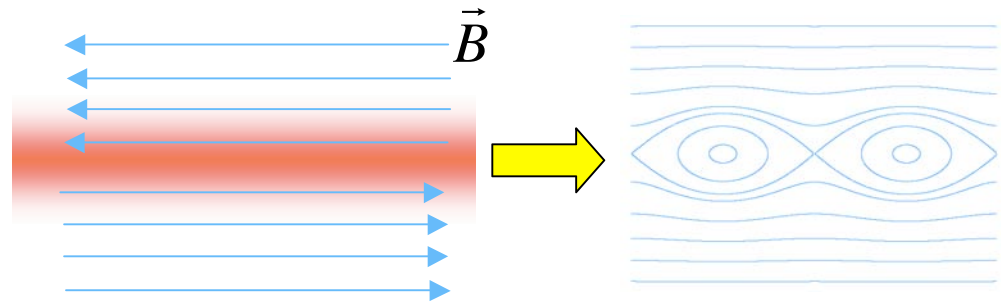
- **Magnetic reconnection: Biggest 2D problems have 5000x5000 grid and 6 billion particles. Really need 3D to answer many of the outstanding questions, but this will require about a factor of 1000 increase in computing power**
- **Laser plasma interaction: Biggest 2D runs are also near the same size as above. Researchers are currently trying to simulation a single speckle, but to develop a first principles, predictive capability there is a need to simulate multiple interacting speckles in 3D. This will also require about a factor of 1000 increase in computing power.**

What is Magnetic Reconnection?

Basic Process in Plasma Physics:

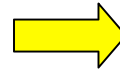
- Magnetic energy \rightarrow kinetic energy
- Topological changes
- Collisional vs **Collisionless**

Collision Frequency $\propto \frac{n}{T^{3/2}}$



Questions

1. How does reconnection proceed so rapidly?



	Resistive Diffusion	Observed Time
Tokamak	1-10 sec	$\sim 10^{-4}$ sec
Solar Flare	10^6 years	~ 20 min
Magnetospheric Substorm		~ 30 min

2. How does it get started in the first place? \rightarrow

Onset problem

Petascale Applications —

Materials

Impact of Petaflop-scale Computing: Application — Materials

	Materials
<p>Programmatic impact to be gained by access to Petaflop-scale computing</p>	<p>Assuming that a factor of 100x- 1000x increase in computing power corresponds to Petaflop/s-scale computing as defined in the cover email, we have taken from your section on Magnetism the ability to:</p> <ul style="list-style-type: none"> • Take key steps towards understanding magnetization reversal (100x increase). • Parameter-free modeling of the dynamics of reversal in small bits (1000x increase). <p>And from “Material Optimization ...”</p> <ul style="list-style-type: none"> • Extend large MD simulations to predict interface mobility • Enable three-dimensional mesoscale simulations to reach large systems sizes relevant for materials processing • Validate multiscale models by means of large-scale atomistic simulations <p>If possible, please add other advances associated with “Materials” If, based on the current level of capability, a 100x -1000x increase is less than Petaflop/s-scale computing then please adjust these advances accordingly.</p>
<p>Major scientific challenges to be addressed</p>	<p>Scientific challenges associated with the programmatic advances noted in the above box are indicated here. Feel free to improve these, and please indicate challenges for the other areas. Also, as noted above these challenges may require adjustment if the assumption about the current level of computing is incorrect.</p> <ul style="list-style-type: none"> • Model the structure of domain walls and their interaction and pinning by defects. • Predict accurately of how phase and grain boundaries move in response to driving forces such as temperature, concentration, or stress. • Develop algorithms of accelerated dynamics to access long time scales.
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made? On the typical calculations that are made?</p> <p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p>[We are emphasizing <i>capability</i> – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)]</p>	<p>These type of throughput data are not directly indicated in the report. However, the kind of answer we are looking for might be something like this: "We did calculations on a 5 TFlops/s (peak) machine, achieving sustained throughput of 0.5 Tflops/s (or 10% efficiency). The turn-around time is about days."</p> <p>Electronic structure codes have been shown to achieve sustained throughput of 512 GFlops/s on a 1.5 TFlops/s machine (i.e. 33% efficiency) (NERSC Paratec code) . First-Principles Molecular Dynamics codes achieve similar performance. Simulations typically are run for several weeks of wall-clock time.</p> <p>In the Materials section, the current and future needs of the community (Table M.2) are indicated in Teraflop-years; however, we need the capability (eg Petaflop/s) for the longest single runs and the typical single run (given a reasonable turn-around time), not the aggregate community requirement which is the capacity.</p> <p>There is a difficulty specifying what would be a desirable target for capability in the PFlops/s range, since no existing software is likely to make efficient use of such a resource when ported as is on such a platform, and what can be done with a PFlops/s resource will depend critically on future progress in software</p>

	implementations.
What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?	Credible architectures for achieving Petaflop-scale capability within 5 years will contain tens of thousands of processors. The ability for codes to run efficiently with large numbers of processors will be critical. Running today's codes with large numbers of processors can give useful insights into projected scaling behavior. The scaling data in Fig M.6 provide some of this information. Please provide us with additional experience if it is available. (Also, would you please explain the meaning of the abscissa in the top graph of Fig M.6.)
What is the Operations Count/Scaling from other computers?	Although scaling is discussed in the report (eg in Algorithm Barriers), numbers are not given explicitly for the required increases of 100x -1000x. To scale performance from today's machines to larger capability machines requires either: <ul style="list-style-type: none"> • An operations count, or • A scaling law based on current performance on current machines If you have used a scaling law to characterize Petaflop-scale performance (100x -1000x increase in computing power), please provide the specific numbers and the logic used (e.g. compute time scales as n^4 , where n is a linear cell dimension), along with the current computer capability used in the scaling. In either cases, please provide the required turn-around time for the longest runs.
Projected increase in algorithm efficiency?	If you are counting on an increase from better algorithms (historically, algorithm improvements have approximately matched improvements in hardware), please indicate the factor you've used. A successful implementation of linear-scaling algorithms for electronic structure is expected to provide an algorithmic speedup of at least 100.
Other	In reference to the discussion in Hardware Barriers, and for your own information, several laboratories have used computers with about 10,000 processors for a number of years. (Personnal comment: This is very impressive indeed. In my own area of expertise, I know of no application running a first-principles simulation on 10,000 CPUs and would be glad to hear about it.)

Impact of Petaflop-scale Computing: Application — Materials

	Materials
<p>Programmatic impact to be gained by access to Petaflop-scale computing</p>	<p>First principles spin dynamics (FP-SD) based on the first principles Landau-Lifshitz-Gilbert (LLG) equation provide a way of performing first principles (parameter free quantum mechanical) simulations of magnetic moments and thereby opens up to study the dynamics of magnetization reversal in bulk and nanostructured materials. In bulk materials this could lead to a first principles understanding of the technical (hysteric) properties of magnets (coercivity, remenance, permeability) and thereby to the development of improved magnets for applications in energy and transportation. In nanoscience understanding of switching of magnetic nano-bits wil have an enormous impact on achievable magnetic storage densities which are rapidly approach critical roadblocks to the current “Moore’s Law” growth - currently the doubling time for areal density (bits/square inch) is ~ 12months.</p> <p>Target calculations would include:</p> <ul style="list-style-type: none"> • Understanding magnetization reversal and the role of magnetic damping in model bulk and nano-magnets. (100x increase). • Parameter-free modeling of the dynamics of reversal in small bits relevant to next generation high density magnetic disc storage (1000x increase).
<p>Major scientific challenges to be addressed</p>	<ul style="list-style-type: none"> • In bulk materials the initial scientific challenges involve the interaction of domain-wall with materials microstructure; i.e. anti-phase boundaries, stacking faults, grain boundaries, etc. Such calculations will require very large simulation cells, even for materials having a large magneto-crystalline anisotropy (MA), (N.B domain wall width is proportional to the size of the MA energy). Thus challenges include: <ul style="list-style-type: none"> ○ Studies of the interaction of domain walls with individual extended defects; antipahase boundaries, stacking faults, grain boundaries ○ Finite temperature studies to see how the interaction of domain walls with microstructural defects depends on temperature. • In switching of magnetic nanoparticles there are a number of important challenges: <ul style="list-style-type: none"> ○ To be able to perform simulations for a sufficient number of atoms to model realistic nano-particles. An appropriate target would be a 5 nm³ Fe nanoparticle containing ~12,000 atoms; (~4,000 of which are either on the surface or subsurface layers). ○ To perform the integration of the LLG equations for sufficient time steps to study reversal – say ~10s of thousands. ○ Currently a first principles theory of the “so called” Gilbert damping is still lacking. Here an intensive research program coupling large scale simulation to experimental studies will be required to gain insights into this problem.

What is the throughput (Tflops/s sustained) today on a *single* run of the longest calculations that are made? On the typical calculations that are made?

Please indicate the code efficiency and/or the computer peak performance.

Please also indicate the turn-around time.

[We are emphasizing *capability* – the ability to tackle big problems in a single computer run – rather than *capacity* (the amount of work that can be done with many runs.)]

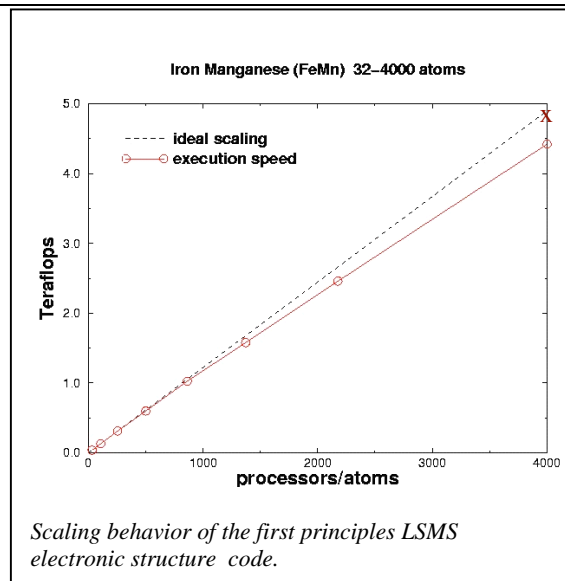
The figure opposite shows scaling studies of the first principles Locally Self-consistent Multiple Scattering (LSMS) electronic structure code. Amongst other uses the LSMS code is used to study the magnetic structure of complex systems. The scaling behaviour of the LSMS code (red circles and solid red line) is shown in the figure opposite. The solid red line is for less than optimal convergence parameters; however, even here the near ideal linear (dashed-line) scaling of the method can be seen. The solid line shows data gathered for a run for a 4000-atom system run on 4000-CPU of the Compaq supercomputer at the Pittsburg supercomputer. The code ran at ~4.8 Teraflop/s on a machine having a peak performance of approximately 6 Teraflop/s. Similar behaviour has been seen on the IBM SP3 at NERSC (Seaborg) -- yes we really do get > 70% efficiency on a regular basis!

More typical runs involve more like a few hundred atoms to a maximum of ~2000 in order to get any reasonable turn around. N.B. at NERSC, which is where, until recently, we had most of our HPC access, there are significant problems associated with running jobs of >3000 CPUs due to operating system limit (hard wired at 3000 CPUs) and MPI implementation which creates large message passing buffers whether you need them or not and results in excessively large memory usage and paging which basically kills the machine for very large jobs (typically the more CPUs you have the more messages are being passed and the more buffers are created.

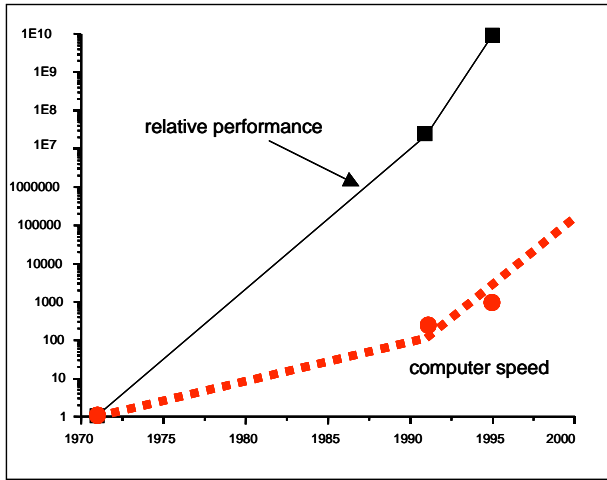
Typical runs are set for ~8 hours in which some 60 iterations are performed (on the SP3), each time step in the integration of the LLG equation typically requires a few ~3-8 self-consistent field (SCF) iterations to determining the forcing fields (N_{SCF_Iters}) For large systems, 100s to ~1000 time steps are required to obtain the ground state - using a form of simulated (spin) annealing. We estimate (based on similar calculations for simple spin (Heisenberg) models that a minimum of several tens of thousands of iterations will be required to study finite temperature dynamics at finite temperature.

SO! For current ground state calculations involving N-atoms using N-nodes:

$$\text{CPU time (wall clock)} = N_{Tsteps} * N_{SCF_Iters} * T_{SCF_Iter}$$



	<p>So for production parameters the time for an SCF iteration, $T_{SCF_Iter} = 8/60$ CPU hours (independent of the number of atoms on Seaborg because of linear scaling), taking $N_{SCF_Iters}=5$ (as an average) and taking $N_{Tsteps}=500$ for optimization of a spin structure CPU time = $(8/60)*500$ hours = 333 Hours (~14 CPU days)</p> <p>Because this is a linear scaling code this number does not change with the number of atoms in the system. If we run N atoms the total CPU time (over all processors) is $333*N$ hours (because with this code we would use N-CPU's). All other timings and estimate can be made from these numbers.</p> <p>To date the largest production runs have been for ~2100 atoms and ~1000 time steps. Thus the estimated flops is the order of $(8/60)*5*1000*2100 = 1,400,000$ CPU hours. Needless to say this was a one-off calculation done during the testing phase of Seaborg.</p> <p>Given that the rating of the NERSC CPUs is ~1.5 GFlops and we run at ~75% efficiency the total operation count is $\sim 1.5*10^6*0.75*1.4*10^6*3600 = 5.67*10^{15}$ Flop</p> <p>Our experience with turn-around is; .if you ask for $\leq 1/4^{th}$ of the processors available on a given MPP, turn around will be typically overnight. This was true on the early Intel Paragons it is still true on the large IBM SP machines (at best). The only time we have been able to run (regularly) on all nodes has been when a machine is new and has not yet been opened for production.</p>
<p>What is the Operations Count/Scaling from other computers?</p>	<p>Given the scaling behaviour in the above figure we can expect that the code will scale well to ~10,000 CPUs. Assuming the CPUs to be ~10X the current IBM SP3 at NERSC (1.5 GigaFlop/s) a 10,000 CPU machine would have a peak speed of $1.5*10^6*10*10000=150$ Teraflop/s</p> <p>This means we could run at $\sim 0.75*150$ Teraflop/s=112.5 Teraflop/s</p> <p>Clearly this would allow simulations of real dynamics which will require $\sim 10-20*10^3$ time steps. But since the CPUs are ~10x faster the runs would in the range of current run times to maybe a factor of two longer.</p>

<p>Projected increase in algorithm efficiency?</p>	<p>In the above we are not relying on algorithm/method improvement. If history is valid we can expect (with time) algorithmic and method improvements to contribute a speed up factor at least equal to that of hardware. See figure opposite.</p>	 <p>Figure M.5 Relative performance increase of Ising model simulations (□) compared the normalized speed of the computers (○) the simulations were executed on. The dashed line is a schematic of the increase in peak performance of the fastest supercomputers since 1972.</p>
<p>Other</p>	<p>In reference to the discussion in Hardware Barriers, and for your own information, several laboratories have used computers with about 10,000 processors for a number of years. I suspect this is only true in a trivial sense since the efficiency is generally very low. When someone runs on 10K CPUs with ~75% efficiency I'll be willing to count it. Using this criterion maybe it may have been done on the Earth Simulator but I doubt elwhere. DO you have chapter and verse to refute this?</p>	

M. Materials [Pre-print of revised SCaLeS-2 chapter] 4034 Words

Science Driven Materials Design: the Road to Technological Innovation

What is the science in materials science?

Materials science is concerned with the discovery of new materials and the understanding, control, and exploitation of their properties. The results of past materials research permeate our everyday lives, from the chips in the computer on which this text was written to the structural and magnetic materials used in generation of the electricity that powers it. At the most basic level materials science asks the simple question “How do we take the ninety or so elements that comprise the periodic table and put them together in combinations that produce materials with useful properties?”

Traditionally the search for new materials and the refinement of existing ones has been accomplished by *Edisonian* trial and error, guided by simple models and the skill and intuition of countless experimenters. Today, however, new materials are increasingly assembled atom by atom or involve previously unimagined complexity, their properties are probed by billion dollar experimental facilities [Advanced Light Source (ALS), Spallation Neutron Source (SNS)] capable of revealing microscopic detail. In addition accurate, robust simulations that are founded in the fundamental equations appropriate to the real material and utilizing the computational power of new generations of high performance computers now have an unprecedented impact on the development of new materials and devices.

A fundamental problem faced by much materials research is that the properties of real materials depend on phenomena that occur at different length and time scales (table

Scale	Quantum	Nanoscopeic	Mesoscopeic	Macroscopic
Length (m)	$10^{-11} — 10^{-8}$	$10^{-9} — 10^{-6}$	$10^{-6} — 10^{-3}$	$> 10^{-3}$
Time (s)	$10^{-16} — 10^{-12}$	$10^{-13} — 10^{-10}$	$10^{-10} — 10^{-6}$	$> 10^{-6}$

Table M.1

M.1). At the smallest scale, properties are determined by the *electron glue* that holds the atoms together (bonding or cohesion). This is the domain of quantum physics. At the macroscopic level, many materials properties – strength, fracture, magnetism – are as much influenced by *microstructure* – crystallites or grains within the material – as the intrinsic bonding of the atoms of the ideal crystal. Between these length scales is the world of nanoscience ($1 – 100 \times 10^{-9}$ m) where materials often display new or unusual properties that hold exciting possibilities for future scientific discovery and technological innovation.

Describing each of these extremes and more importantly bridging the disparate length and time scales associated with them (*multiscale modeling*) poses the *grand challenge* of theoretical and computational materials science. Making progress in

brittle. However recent scientific discoveries involving addition of small amounts of boron, slight modification of the Ni:Al ratio, and control of microstructure, has resulted in a new class of commercial alloys that are ductile, strong at high temperature, and corrosion resistant. These alloys are now resulting in substantial energy and cost savings in the steel, automotive, and chemical industries [fig. M.2]. In 2001, the development of these alloys was listed as one DOE Basic Energy Sciences' 100 most significant scientific advances of the previous 23 years.

In numerous other areas of materials science the basis for future scientific breakthroughs is being laid – understanding the origins of high temperature superconductivity, transition metal oxides with totally new properties and functionality, and the exploration of the fascinating world of nanostructured materials.

M.2 Scientific Opportunities

During the next two decades the opportunity exists to develop a new paradigm for materials research in which modeling and simulation are integrated with synthesis and characterization to accelerate discovery and optimization of materials. Some specific opportunities are summarized in the following sub-sections and in fig. M.3. The examples used are not exhaustive but rather illustrative of the possible impact of this paradigm shift.

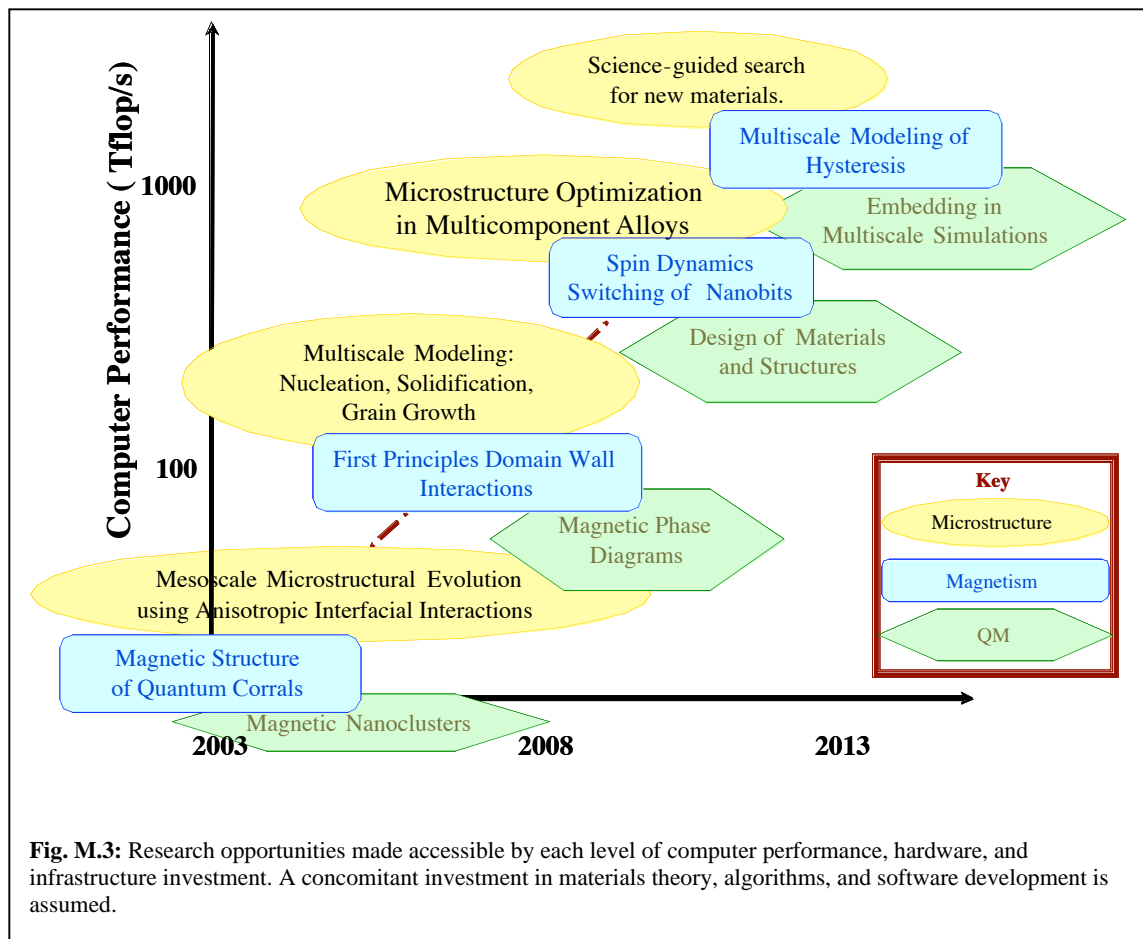
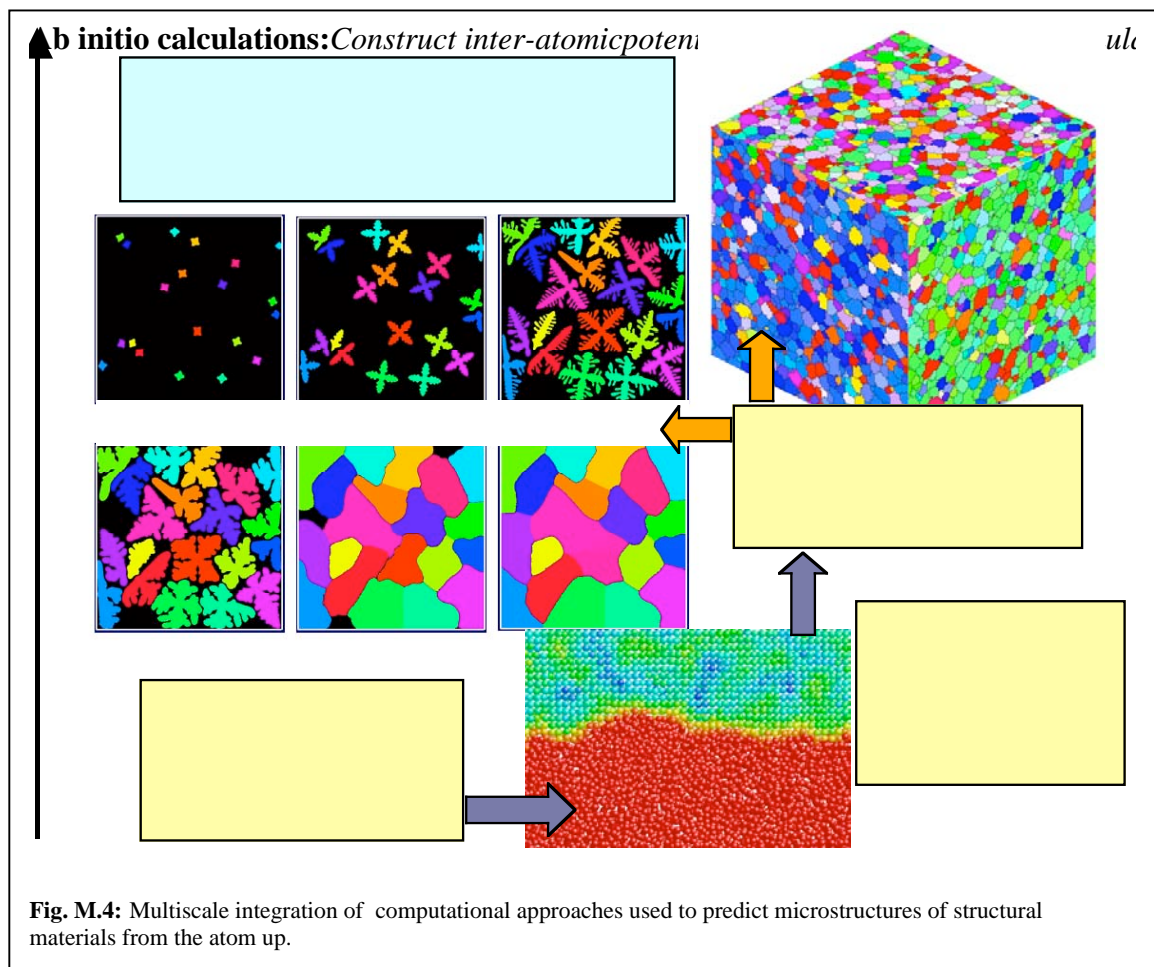


Fig. M.3: Research opportunities made accessible by each level of computer performance, hardware, and infrastructure investment. A concomitant investment in materials theory, algorithms, and software development is assumed.

Material Optimization for Energy and Transportation

Structural materials from nickel-based superalloys used for turbine blades to light-weight aluminum alloys used for automotive parts are pillars of the energy and transportation industry. Without exception, these materials are compartmented on a micron scale by boundaries of complex shapes that divide spatial regions of different composition and/or different crystallographic orientation – broadly called the "microstructure". A materials microstructure controls most of its structural properties – strength, wear, corrosion resistance. Predicting how the microstructure emerges from an initially structureless melt during solidification (casting, welding, etc), and how it evolves during post-solidification processing, is an essential prerequisite for material optimization and is one of today's most important theoretical computational challenges.



The core of this challenge is the accurate prediction of how phase and grain boundaries move in response to driving forces such as temperature, concentration, or stress. This problem is intrinsically multiscale because the two key anisotropic properties that control this motion, the interface *energy* and *mobility*, are determined by details of inter-atomic (quantum physics) forces acting on nanometer/picosecond length/time scales. Whereas the highly non-local fields that determine the local driving force for motion are determined by bulk transport of mass or energy on macroscopic length and

time scales. Furthermore, the vast parameter space that characterizes the interface anisotropy (e.g. five dimensional for grain boundaries in three dimensions!) approaches biological complexity.

Progress in solving this multiscale problem has recently been accomplished through the integration of atomistic scale simulations and mesoscale models (Fig. M.3). Quantum mechanical ab-initio simulations have been used to guide the construction of inter-atomic potentials that can be used in large molecular dynamics (MD) simulations with several million atoms. These simulations, in turn, have made it possible to predict, for the first time, the anisotropy of the interface energy and mobility. Moreover, new mesoscale simulation methods such as phase-field and level set have emerged that incorporate these interfacial properties and thermodynamic data to simulate complex microstructures, which appear nearly indistinguishable from experimental micrographs.

This integration of new techniques holds much promise to guide the optimization of microstructures so as to cut down the current 10-15 years required to commercialize a new material to just a few years, and, even more ambitiously, to guide the search of new materials. However, realizing this promise still requires extension these techniques to multi-component alloys (e.g. 12 components for super alloys), to experimentally relevant length and time scales, and to three dimensions. The 100X to 1000X projected increase in computing power will provide a unique opportunity to achieve these goals by, for example, extending large MD simulations to predict interface mobility over the full range of driving force relevant for microstructural evolution and by enable three-dimensional mesoscale simulations to reach the large system sizes relevant for materials processing. A key target is to model a cubic millimeter of material where the predictions of mesoscale models can be meaningfully interfaced with macroscale industrial codes.

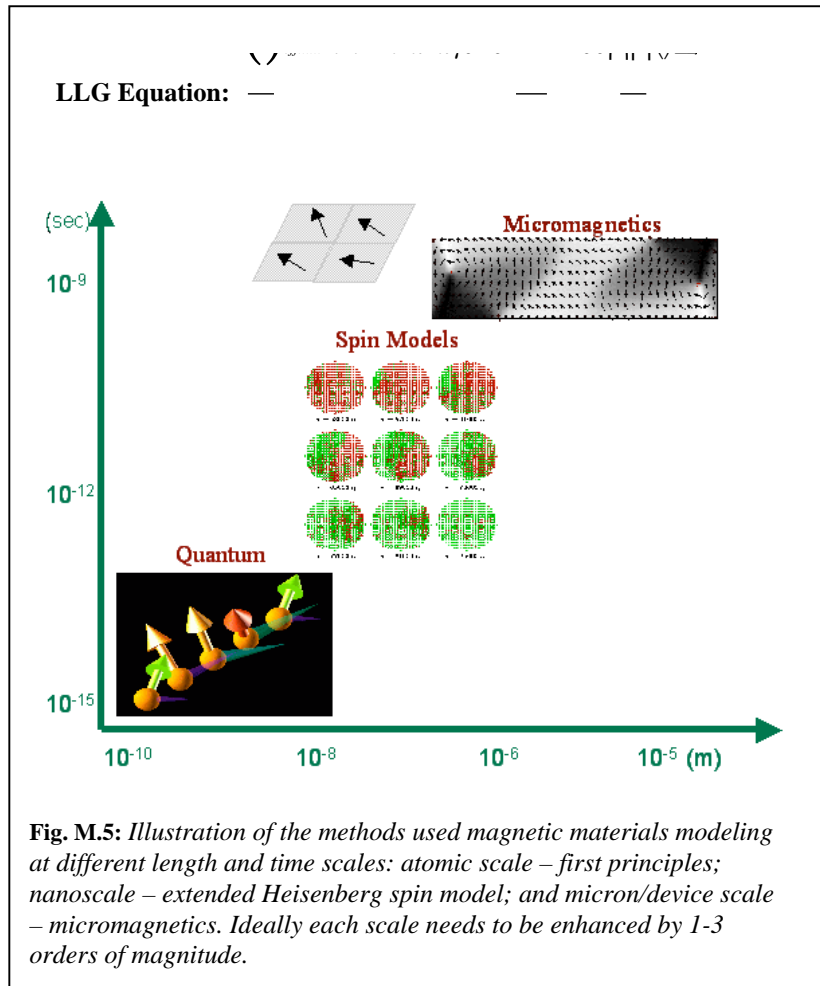
Magnets of the Future: Predictive Modeling of Switching and Hysteresis

Predictive modeling of the technical properties of magnets – energy product, coercivity, remenance – which requires modeling of the dynamics of magnetic moments and how these are reversed or switched – is the central scientific challenge in magnetic materials. It is also one where computational approaches can prove decisive thereby having a profound impact on a wide range of technologies from energy production and utilization (generators, transformers, and motors) to transportation (sensors and motors) and computers (magnetic storage and memory).

While the underlying mechanism for materials magnetism involves electronic interactions at the atomic level, long range, *magnetostatic*, interactions and large-scale features (e.g., domain walls and their interaction with microstructure) are crucial for determining bulk magnetic properties in *real* materials. Consequently, magnetism is an intrinsically multiscale problem. A problem that is, however, greatly simplified by the observation that the basic equation describing the dynamics of magnetic moments at the different length scales is believed to have the same form, the Landau-Lifshitz-Gilbert (LLG) equation (Fig M.3). Albeit that the description of the magnetic moments changes from length scale to length scale – first principles electronic structure methods at the smallest length scales, spin models at intermediate length scales, and continuum *micromagnetics* models with empirical parameters at the device level. Thus, new challenge/opportunity is to develop rigorous approaches to extending and bridging the

models that describe the different length scales and to apply these capabilities to discover and design new magnetic materials.

A 100X increase in computer power would allow the exploration and understanding of the structure of domain walls and their interaction with and pinning by defects – key steps towards understanding magnetization reversal. A 1000X increase would enable detailed parameter free modeling of the dynamics of magnetization reversal in small magnetic bits. Further advances coupled with concomitant advances in microstructure modeling could lead to science based modeling of hysteresis and the design of improved bulk magnets.



Advancing Modeling of the Fundamental Interactions In Complex Systems

The Pathway to New Understanding, New Materials, and New Properties: Increases in computing power have a very large impact on first-principles quantum simulation methods to predict structural and electronic properties of complex materials. First-principles simulations are extremely computationally demanding but are essential to understand the properties of complex materials in detail. Electronic properties are modeled using various first-principles methods depending on the accuracy needed.

The Quantum Monte-Carlo (QMC) method is the most accurate and expensive because it deals directly with many-body effects in a genuine quantum mechanical description. Because, it involves independent statistical sampling QMC is uniquely suited to take advantage of future generations of computers, readily utilizing parallel computation on machines with tens of thousands of nodes. Recently an $O(N)$ algorithm has been developed and applied to the prediction of the optical gap in semiconductor nanostructures consisting of 1000 atoms.

Density Functional Theory (DFT) has been so widely used to model electronic properties in the past decades that it has been called the “Standard Model of Condensed Matter”. One of the most successful methods developed in the past 15 years is First-Principles Molecular Dynamics (FPMD) due to Car and Parrinello, which unifies molecular dynamics and DFT. FPMD is an example of a very powerful simulation tool whose development was accelerated by the large computing power brought about by the first Cray vector computers in the 1980’s.

The importance of first-principles computations of materials properties and their relevance to future industrial applications were recently featured in the Technology Quarterly Review of *The Economist* (June 21st 2003) where Marvin Cohen, father of one of the most successful electronic structure methods, described recent successes of first-principles computations, notably the *prediction* of superconductivity in silicon at high pressure. New computational power coupled with concomitant advances in theory, algorithms, and software engineering, will vastly expand the domain of applicability of first principles methods making such calculations possible for spintronics, super-hard materials, catalytic reactions, and hosts of other applications possible as well as expanding the role of first principles modeling as the foundation upon which multiscale modeling is built.

M.3 Research issues

Research challenges can be broken into three broad classes. Firstly, developing and extending the length and time scales covered by the models used at each scale. Secondly, coupling models across different scales to produce robust and predictive multi-scale modeling capabilities. Thirdly, formal theoretical advances to allow modeling and simulation to address many outstanding problems – formal theory of spin dynamics, origins of pairing in high T_C superconductors, etc.

Extending models can be achieved through improvements in algorithms (e.g., changing from algorithms that scale as N^3 to one that scales linearly in N , where N is the number of atoms in the simulation), better use of computational resources, and parallelization. Larger length scales can generally be achieved through parallelization and domain decomposition. Here, a major goal would be the development of QMC and DFT electronic structure methods that scale linearly with system size to 10,000 to 100,000 processors.

Research into extending the time scale is an overarching need at all length scales and is one of the most challenging problems in materials science. Here parallel computers have no obvious advantage since time is intrinsically serial. However, advances can have a profound impact on the exploration of new physical phenomena (e.g. growth mechanisms, rare events).

Although development of seamlessly coupled multiscale methods is a *Holy Grail* of materials science, lack of computational resources is generally not the limiting factor, although large simulations are often necessary to validate multiscale models. Addressing this area necessitates researchers with expertise in many different fields, building the teams of materials scientists, mathematicians, and computer scientists will require major changes in the way materials research is traditionally performed. Achieving reliable and robust techniques for coupling/mapping ab initio electronic structure with/onto atomistic molecular dynamics or Monte Carlo simulations and integration of mesoscale models

with existing thermodynamic databases for quantitative modeling of multi-component alloy would be major steps towards the overall goal of multiscale modeling.

M.4 Technology Barriers to research

The diversity of applications in Computational Materials Science makes it difficult to make general statements about the current technological barriers to research. The following are general concepts that are perceived to be barriers by members of the community.

Hardware Barriers

Node count versus node power: While there is a consensus that a large increase in computing power is desirable, the way in which this increased power should be realized is less clear – a large number of moderately powerful processors versus a moderate number of very powerful processors. From an application development standpoint, it is generally preferable to deal with fewer, more powerful processors. However, since it is easier/cheaper to build a supercomputer by assembling a large number of moderately powerful processors, it is important to assess the usability of such a computer in the context of Materials Science simulations. Two realistic examples are 1) a 100 TFlop/s computer built from 100,000 processors of 1GFlop/s each, or 2) a 100 TFlop/s computer built from 10,000 processors of 10 GFlop/s each. It should be noted that for both 1) and 2), the number of processors far exceeds that of currently available computers, so that our conclusions are, to some extent, speculative and are further complicated when (unknown) considerations of bandwidth and latency of the interconnect are taken into account.

For DFT/FPMD — which has $O(N^3)$ complexity — it is reasonable to expect scaling to 10,000 CPUs within 1-2 years given adequate software development investments, while scaling to 100,000 CPUs is a longer term goal. For QMC, classical MD, and continuum models of $O(N)$ complexity, the situation is more favorable since these methods can maintain a reasonable communication/computation ratio by increasing the size of the system studied and therefore the amount of work performed by each processor – so called weak scaling.

In general it is expected that the cost advantage of using a finer granularity (i.e., a large number of small processors) may be offset by the increased cost of application software development.

Memory limitations: Algorithms of $O(N)$ complexity most often use only $O(N)$ variables. As a consequence, their scalability is only a function of the communication cost relative to the computational cost. This is true of QMC and most other models based on the computation of a finite number of solutions of a partial differential equation. On the other hand, algorithms of $O(N^2)$ or $O(N^3)$ complexity often operate on $O(N^2)$ or $O(N^3)$ variables, so that the scalability is ultimately limited by the total memory available and it is unrealistic to expect the memory *per node* of a large supercomputer to grow proportionally to the total number of nodes.

Software Barriers

Massively parallel scalability: Scalability of some applications to a few thousand CPUs has been demonstrated (see Fig. M.6). However it is important to note that this scaling is typically only obtained after considerable investment in software development and that

efforts made to obtain scaling to 2000 CPUs may not be reusable when targeting 10,000 CPUs.

It is generally difficult to *predict* scalability of an algorithm for processor counts beyond currently available since it often involves trial and error and unpleasant surprises (e.g., lack of scalability of MPI collect operations on IBM SP3's beyond 1024 tasks). Consequently, improved performance models are an important priority since they would facilitate the development of high-performance application software *before* a new platform is built.

System reliability: Ideally, system reliability on 100,000-processor platforms should be dealt with by the operating system. Failing this, most (likely all) materials applications will require additional software development to address fault tolerance, given that

long runs (days or weeks of wall-clock time) are the norm.

Support of libraries: The availability of communications (e.g. MPI) and mathematical (e.g. ScaLAPACK) libraries is an important ingredient in the development of scalable application codes.

Software engineering issues: Materials simulation codes must often be rapidly

modified to address ever changing physical models. The cost effectiveness of good software engineering and design practices is slowly being recognized in the community, together with the fact that simulation software typically has a much longer lifetime than most hardware platforms. The cost of maintaining, rewriting or modernizing legacy codes remains an obstacle to research, since this activity is often not recognized as research and thus not funded as such. Several groups have started efforts aiming at improving code reuse within groups, and ultimately throughout the Materials Science community.

Algorithm Barriers

Some simulation methods are naturally suited to parallel computing. QMC is currently only limited by access to sufficient computational resources, and relies on an algorithm that scales extremely well to very large numbers of processors (i.e., is “embarrassingly parallel”). Classical molecular dynamics and most methods based on continuum models are also well positioned to exploit future large platforms using domain decomposition.

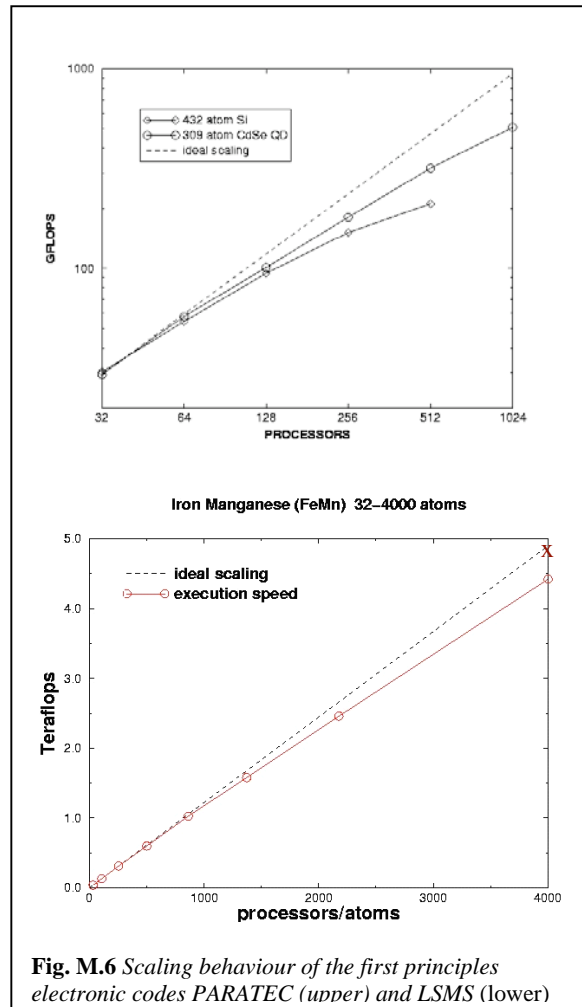


Fig. M.6 Scaling behaviour of the first principles electronic codes PARATEC (upper) and LSMS (lower)

Simulation methods relying on more complex, $O(N^2)$ or $O(N^3)$ algorithms would benefit greatly from larger computing power, although with a less spectacular increase in the length scales that can be described. For instance, DFT simulations, an $O(N^3)$ algorithm, of 256 atoms are routinely carried out on 0.5-1.0 TF computers. An 8-fold increase in the number of atoms, i.e., a 2-fold increase in linear dimension, would require a 512-fold increase in computing power, i.e., a 256-512 TF platform. Furthermore, larger systems usually involve longer simulation and equilibration times, which would further increase the size of computer required. This shows that algorithmic developments that reduce the computational complexity of DFT to $O(N)$ or $O(N\log N)$ are a priority, and must be considered as important as the construction of larger supercomputers.

In addition to the above overarching algorithmic considerations, advances in specific lower level mathematical algorithms would benefit materials applications generally. Particularly important are scalable algorithms for large complex matrices that are either dense or sparse with a known sparsity pattern and portable adaptive meshing and multigrid methods for interface tracking, phase-field and level set.

M.5 Resources required

The resources required fall naturally into two categories. Firstly, state of the art capability and capacity computing. Secondly, people – materials, applied mathematics, and computer science researchers – to support software development and maintenance of methods and software used in cutting edge research.

Computational Resources

Present estimates of annual high performance computing resources used at DOE facilities by materials science is ~2.7 Teraflop-years. Historically, approximately 18% of the computer time available at NERSC is utilized by materials science projects. With current NERSC hardware the annual usage is ~1.8 Teraflop-years. Additional materials projects are serviced by the Center for Computational Sciences (CCS) at ORNL (~ 0.9 Teraflop years annually). Materials scientists are also major users at the NSF supported Pittsburgh Supercomputer Center (PSC) (1.2 Teraflop-years) giving an overall “materials” usage of 3.9 Teraflop-years at the three centers. Using these figures as a base

Resources	Current (2003)	2005	2010	2015
Minimum	2.7	30	300	1000
Target	N/A	80	800	5000
Maximum	N/A	200	10^4	10^6

Table M.2 Aggregate computational requirement for US materials science (Teraflop-years).

Table M.2 gives estimates for requested resources to effect both evolutionary and revolutionary advances. These estimates include requests for both capability (problems requiring all of the computing capacity of the most advanced machines available) and capacity (problems requiring only a fraction of the total resources but for many

independent runs) computing. While the ratio between capacity capability computing is problem dependent a significant increases in both is required in order to address the most challenging materials problems.

Human Resources:

Virtual Research Institutes: A facilities analogy for exploiting high performance computer resources: In order to fully exploit the capability of high performance computing it is necessary to adopt a new approach to accessing and utilizing high-end computational resources. This is necessitated by a number of generic characteristics of computational materials science. Most important is the recognition that, in terms of overall advances in performance – algorithmic efficiency – gains arising from the intellect and ingenuity of the researcher are larger than those from improved hardware; impressive as Moore’s Law is (see Fig M.5). When one adds to this, the multiscale nature of the materials science, the lack a single computer code, or even a small set of codes, which could then be used by the whole community, the need to rapidly respond to the discovery of new phenomena, and continuously tune codes to the most advanced computer architecture, it is clear that a community wide response is needed.

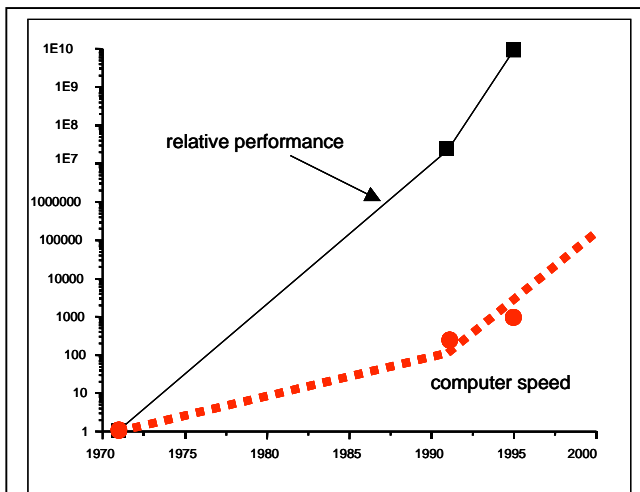
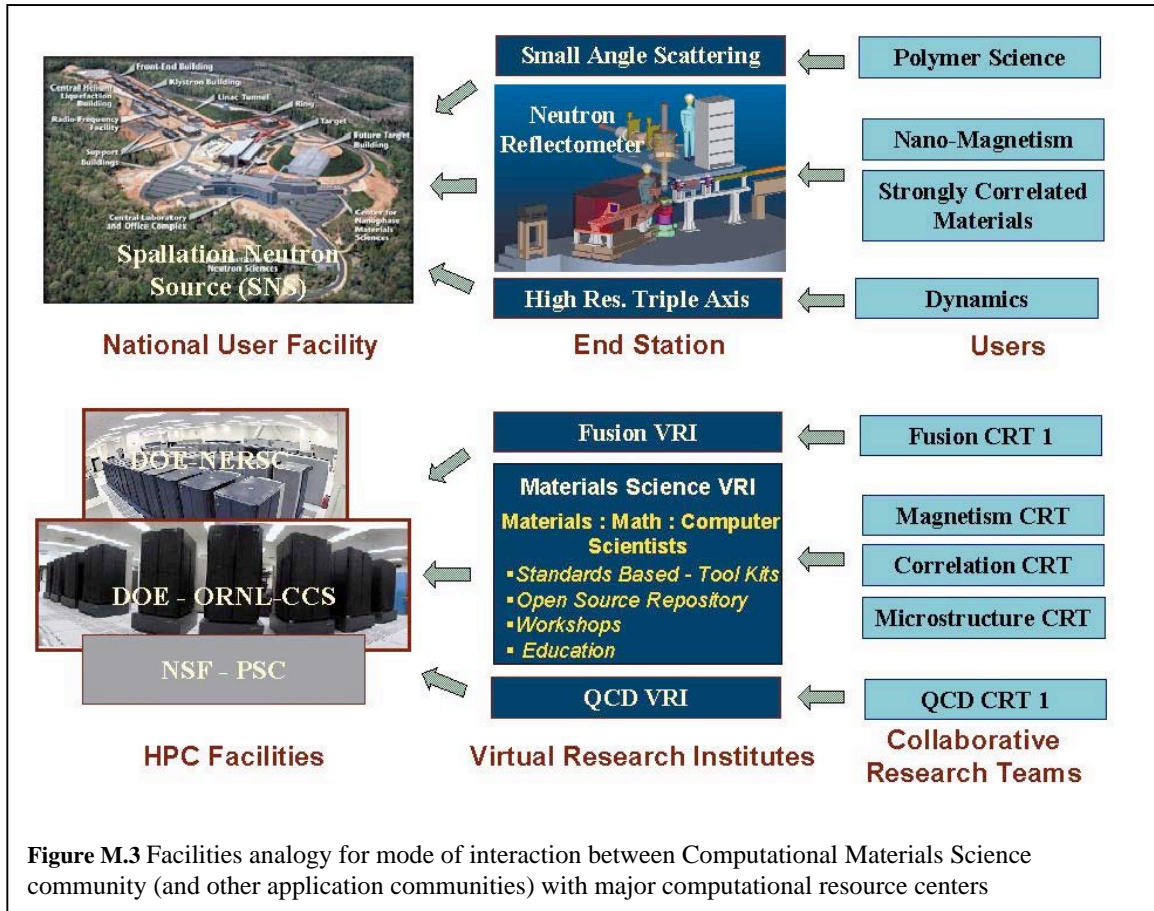


Figure M.5 Relative performance increase of Ising model simulations (□) compared the normalized speed of the computers (○) the simulations were executed on. The dashed line is a schematic of the increase in peak performance of the fastest supercomputers since 1972.

Figure M.3 illustrates the concept of Virtual Research Institute (VRI). The VRI envisioned as the mechanism through which the research community interacts with the high-end computer centers. The purpose of the VRI is to promote the development of a software infrastructure that is flexible, extensible, and tuned to the currently available computer architectures while allowing the field to most rapidly adapt to the ever-evolving computer hardware. In this regard the VRI serves the role of a specialized *end station* through which a focused subsets of the research community (collaborative research teams) interact with a major national facility (the hardware) in much the same way as user end stations at major experimental facilities are the mechanism through which user communities interact with the expensive national experimental facility (e.g. neutron reflectometer end station at the SNS).

Development of the VRI concept will require support for materials scientists, applied mathematicians and computer scientists to develop the science and software base of the VRI. A basis for the development of a Materials VRI is provided the Computational Materials Science Network (CMSN) extending the concept of a Collaborative Research Team (CRT) concept to include applied mathematicians, and computer scientists and by adding specific funding for algorithm and software development. CRTs are collections of

scientists that have on common interest in solving a particular challenging materials problem. Assuming that a VRI can support 8 to 10 CRTs and that each CRT will require funding in the range \$2M to \$2.5M per year a Materials VRI will require funding in the range of \$25M/yr to in addition to direct support to the computer facilities for hardware and infrastructure.



M.6 Metrics of success

A measure of success common to all areas of Materials Science is a reduction in the time spent between the discovery of a new phenomenon and its use in a technological application. Even a 30% reduction in the time to commercialize discoveries appears to be a distinct possibility. Ultimately, the combination of large computational resources, cutting-edge software and numerical methods will then realize the goal of predicting materials properties accurately without recourse to experimental input, and possibly discovering new phenomena and materials by numerical simulation. Finally, direct simulation of experimental quantities can be used to take maximum advantage of experiments performed at the nations advanced characterization facilities thereby greatly reducing the demand on these expensive facilities.

-----Original Message-----

From: Thomas Schulthess [mailto:schulthesstc@ornl.gov]

Sent: Thursday, May 20, 2004 9:48 PM

To: Ed Barsis

Cc: Malcolm Stocks; Jeff Nichols; Thomas C. Schulthess

Subject: ab initio Monte Carlo

The general problem:

Calculating free energies in nanoscale systems.

The challenge:

Computationally expensive ab initio electronic structure methods have to be used to calculate energies of atomic/spin configurations.

An exceedingly large number of atomic/spin configurations have to be visited to get a reliable estimate of the entropy.

The specific problem:

Calculate temperature dependent free energy barrier for switching of the magnetization in nanoparticles.

Application:

Terrabit/square-inch scale data storage requires bits to be stored in nano-meter sized particles. The energy barrier needs to be large compared to kT to prevent data loss due to thermal fluctuations. Materials and nanoparticles with these properties exist. However, it is at present not clear how data can be written, i.e. the magnetization switched deliberately. One way to do this is reduce the energy barrier by locally heating the medium. Generally speaking one needs to understand the energy barrier at the atomistic level to be able to come up with a smart trick to switch the media. Measuring magnetic configurations at the atomic scale experimentally is not possible. Therefore we have to rely on computation for this study.

State of the art in modeling:

Calculating the energy of a general (non-collinear) magnetic configuration is presently possible for 5K atoms with the all-electron locally self-consistent multiple scattering method (LSMS).

Such a run takes of the order of hours on the IBM Power 3 at NERSC.

Demonstrably, the LSMS code scales perfectly up to 10K processors, where the algorithm requires that $N_{\text{atom}} \geq N_{\text{nodes}}$ (distinction between node and processors will be clear in a minute). Note that this is NOT trivially parallel.

Efficiency of the LSMS code is 75-80 percent on IBM Power 3/4 and Compaq-Alpha.

Note, that the efficiency of the code is tied to the efficiency of the implementation of double

complex matrix multiply (ZEGEMM). On the new IBM BG/L architecture, DGEMM has been reported to run at 94% of peak on a dual processor node. The efficiency of ZGEMM is expected to be higher. We will be working with IBM to test the performance of LSMS on BG/L.

What needs to be done:

Sample configuration space by calculating the energy of many configurations, build a density of states (DOS), and from it calculate the entropy configuration to the free energy. One would have to develop some (probably Monte Carlo based) sampling technique. Most likely one could run this in parallel, by running hundreds of LSMS processes in parallel, each running on ~3-5K processors. Since the added communication between LSMS calculations is less than within the LSMS run, one expects that the combined scheme would scale perfectly to however many processors are available.

The general problem:

Simulate phase diagram of high temperature superconductors. Study mechanism of superconductivity and decide which of the many proposed models is correct or propose new model.

The challenge:

Describe a collective quantum effect with a macroscopic number of particles (thermodynamic limit) with strong non-local interactions (strongly correlated).

Algorithmic solution:

Treat strong non-local correlations exactly within a cluster using Auxiliary Field Quantum Monte Carlo (QMC). Account for the macroscopic number of particles by embedding the cluster into an effective medium. The algorithm that does this in a way that the effective medium has the proper symmetries, the resulting Green's function is causal, and the exact solution is recovered with (

XXX

) where N is the number of sites in the cluster, is called Dynamical Cluster Approximation (DCA). We call the combined algorithm DCA/QMC.

The specific problem:

The workhorse within the QMC cluster solver is the BLAS level 2 routine DGER that evaluates a vector outer product to update a determinant. The linear dimension of the determinant is given by $L=n*N$, where N is the number of sites in the cluster and n is the number of time slices in the path integral. The computational complexity goes with L^3 , and the evaluation of DGER is very much limited by the memory bandwidth.

The challenge to a particular computer architecture is, whether N and n and thus the linear dimension L can be made large enough so that the cluster is large enough to capture the physically relevant non-local correlations and the path integral is converged, but still keeping the efficiency of DGER at a reasonable percentage of peak. Obviously this requires a high memory bandwidth.

State of the art in modeling:

On the IBM/Power4 and the Compaq Alpha it is possible to simulate cluster sizes of 4 atoms efficiently. With this it was possible to show that the phase diagram of high temperature superconductors can be qualitatively reproduced with the two dimensional single band Hubbard model. This result, however, is mainly due to finite size effects. The high memory bandwidth on the Cray X1 allows for simulations of cluster sizes of up to 64 sites with efficiency of ~50%. Compared to the IBM, the efficiency of DGER is so good that the matrix multiply (DGEMM), used in the measurements, is taking a significant amount of time. 512 processor Cray X1 -> cluster of size 32 can be simulated with sufficient accuracy that the

minus sign problem is still under control. The cluster size is large enough that 2D Hubbard model does not have finite temperature transition as required by Mermin Wagner theorem

Plans:

Simulate 3D cluster to account for inter-planar coupling. See if this recovers finite temperature transitions in the Hubbard model. If it does, study mechanism of High Tc SC. If not -> expand to multiband model, include phonons, etc. or maybe even abandon the Hubbard model (this of course would put condensed matter theory upside down).

Discuss required computational resources and scaling ...

Petascale Applications —

Nanosciences

Impact of Petaflop-scale Computing: Application — Nanoscience

	Nanoscience
<p>Programmatic impact to be gained by access to Petaflop-scale computing</p>	<ul style="list-style-type: none"> • <i>Ab initio</i> characterization of 1000-atom FePt particle. Perhaps understanding from these calculations will speed the development of higher density storage devices. • Calculation of the electron transport properties of organic molecules. This will allow screening candidates for molecular electronic devices, which have the potential to replace the current silicon-based technology for computing devices that is expected to reach fundamental limits in 10-15 years. • <i>Ab initio</i> molecular dynamics simulation of early key steps in the growth of colloidal quantum dots that is part of wet-chemistry-based self-assembly of nanostructures.
<p>Major scientific challenges to be addressed</p>	<ul style="list-style-type: none"> • Understand and characterize the spin dynamics of 3nm particles • Develop an alternative paradigm for computing to replace limited silicon-based technology. • Understand and control self-assembly in complex environments through multiscale simulation.
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p> <p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p>[We are emphasizing <i>capability</i> – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)]</p>	<p>Computational nanoscience differs from some of the other science areas (e.g., climate) in that there is a range of computational methods used to address different parts of the problem. We use techniques from computational materials science and quantum chemistry; the application of these methods to nanoscience means the absence of periodicity in three dimensions (the hallmark of bulk systems), the domination of interfacial regions, and the frequent inability to calibrate force fields with experiment. As a result, throughput is strongly influenced by the method used (e.g., quantum chemistry calculation, <i>ab initio</i> molecular dynamics, atomistic molecular and the efficiency of its implementation.</p> <p>For example, reported large-scale calculations using atomistic molecular dynamics (MD) simulations on 10^9 atoms:</p> <ul style="list-style-type: none"> • In excess of 10 Tflops/s sustained on specialized MD hardware (MDGRAPE-2 and WINE-2), exhibiting near 100% parallel efficiency (scaling on multiple processors versus perfect scaling) ~3000 processors and code efficiency (Tflops/s achieved versus theoretical peak) is essentially 100% • In excess of 1Tflops/s sustained has been reported for the public domain biophysical MD program NAMD, exhibiting parallel efficiency of 70% on 2250 processors and code efficiency of 15-20% (calculations performed on Lemieux at Pittsburgh Supercomputing Center) • By contrast, other reports (from NERSC) suggest that the code efficiency in molecular dynamics on IBM SP architectures can be as low as 2-3% <p>Typical atomistic MD simulations are run in blocks of 6 to 24 wall clock hours, with turn-around time of minutes to several hours; a complete run may take 10 to 100 blocks, depending on how much simulated time (ns) is needed for the particular application.</p> <p>[We have assumed that throughput data on computational quantum chemistry and electronic structure calculations will be provided in the computational chemistry and computational materials science sections respectively.]</p>

<p>What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?</p>	<p>The typical number of processors used for molecular dynamics is between 32 and 128. Examples exist of calculations run on as many as 3000 processors.</p> <p>The typical number of processors used for computational quantum chemistry is rather small – only a few processors. This is because of the dominance in the field of Gaussian, a code which does not scale well on shared memory multiprocessor machines. In calculations performed with codes that scale well on parallel computers (e.g., NWChem, GAMESS-UK) the typical number of processors is likely to be ~32-64. Calculations using NWChem have been reported at 90% parallel efficiency on 512 processors.</p> <p>Efficient implementations of methods used in computational nanoscience (atomistic and <i>ab initio</i> molecular dynamics, computational quantum chemistry, density functional theory for electronic structure calculations, mesoscale methods, etc) have all demonstrated good scaling on ~1000 processors. We are not aware of calculations on 10000 processors or more, simply because of the rarity of such machines. In many cases, assuming balanced memory access architectures, there is no reason for many of these codes to successfully scale to 10000 processors and beyond. As a specific example, domain-decomposed molecular dynamics does not scale successfully beyond 1000 processors because the communication between processors is local; there are methods for parallelizing molecular dynamics (data replication) that will not scale to large numbers of processors since they involve global communications.</p>
<p>What is the Operations Count/Scaling from other computers?</p>	<p>The scaling behavior of calculations in computational nanoscience depends on the type of calculation and methods employed. Examples are:</p> <ul style="list-style-type: none"> • Atomistic molecular dynamics for atoms with short-ranged interactions: N^1, where N is the number of atoms • Atomistic molecular dynamics for atoms with long-ranged interactions: $N(\log N)^{3/2}$ (using Particle-Mesh Ewald) • Quantum chemistry calculations with the best available methods and basis set extrapolation: N^3 • Density functional theory, as used in electronic structure calculations and <i>ab initio</i> molecular dynamics: N^3 <p>Implementations of each of the above methods exist that scale inversely with numbers of processors.</p>
<p>Projected increase in algorithm efficiency?</p>	<p>Historically, speed-ups due to improvements in algorithms have been at least as large as speed-ups gained from increase in processor speed. [A concrete example is Monte Carlo algorithms for spin systems, which over a year period experienced ten orders of magnitude increase in speed, only three of which were attributable to increase in processor speed.]</p>
<p>Other</p>	<p>The unit Tflop-years is meant to refer to 1Tflop/s peak performance over a period of 1 year, which we have assumed will translate to approximately 0.2 1Tflop/s sustained achieved performance over a period of 1 year.</p>

Impact of Petaflop-scale Computing: Application — Nanoscience

	Nanoscience
<p>Programmatic impact to be gained by access to Petaflop-scale computing</p>	<ul style="list-style-type: none"> • Ab initio characterization of 1000-atom FePt particle. Perhaps understanding from these calculations will speed the development of higher density storage devices. • Simulate the growth of colloidal quantum dots in wet chemistry. Wet chemical synthesis of colloidal quantum dot is one major way to build nanostructure under the bottom-up approach. Currently, molecular kinetics and patch work are not known, and the experiments are carried out in a trial-and-error basis. Doing ab initio MD or MC simulation for these processes will help to design new synthesis processes. • Calculate the electron transport for different organic molecules. This will help to screen thousands of possible organic molecular candidates for electronic devices to potentially replace the CMOS technology after the Moore's law hits its wall in 15 years.
<p>Major scientific challenges to be addressed</p>	<ul style="list-style-type: none"> • Understand and characterize the spin dynamics of 3nm particles • Understand the role of phonon in electronic quantum transport • Understand the nanocrystal growth or self-assembling process through multiscale simulation • Understand the optical properties and electronic structures for multiple excitons in a nanostructure
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p> <p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p>[We are emphasizing <i>capability</i> – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)]</p>	<p>The long calculations can be a few months (3-4 month) for MD runs on cluster machines. The typical actual large calculations (except the one for benchmarking) are 0.5-1 Tflops/s (256-512 processors of IBM SP3), with 10-20% efficiency (i.e, the actual throughput is 0.1-0.2 Tflops/s). The job can run from hours to a few month.</p> <p>In reality, the majority of the runs are in 32-128 (0.05-0.2 theoretical Tflops peak) processors (based on own experience). For most of those runs, it might take 12 hours to finish.</p> <p>Note, all these depend on the availability of the machine and computer times allocations, and the waiting time to run the job. There is always a tradeoff between different factors. If I have a 50000 processor machine, I certainly can run the whole machine, and I will think about bigger problems to attack. Our physics problems and simulations are dictated by the machine and time we have (capacity). We can easily find problems which cannot be solved by the current day biggest computers and current day algorithms.</p>
<p>What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?</p>	<p>Typical number of processors we used is about 32-128. The largest number of processors we used (just 1 run, since it is difficult to get on, and we don't have that much computer time allocation on NERSC) is 1028. It runs well for very big nanosystems.</p> <p>To run on tens of thousands of processors, we definitely need to change our code, and to some extent to</p>

	<p>our algorithm. The change of computer architecture will always be accompanied by the change of simulation algorithm in all levels. This is especially true in material science. So, it is not simply increasing the hardware power and running the current day code. In other words, it is difficult to predict the future needs based on today's code, algorithms and scaling. It is a rather dynamic process. Due to today's "capacity", in practice people usually run on 16-128 processor range. As a result, most codes only run efficiently on this range of processors. For larger number of processors, new code and new algorithms will be needed. So, in a sense of limited capacity, we really haven't reached the capability of even today's largest computer.</p>
<p>What is the Operations Count/Scaling from other computers?</p>	<p>The computer capability needs of many tasks (e.g. in table 1 of our write-up) depend on the exact algorithm and the approach we will take. Since for many of the tasks, that is not certain (the algorithm itself might depend on the availability of the computer), so it is really difficult to say what is exactly the need.</p> <p>However, we can take an example of ab initio molecular dynamics under car-parrinello algorithm using current code. Under the current algorithm, the calculation scales as N^3, where N is the number of atoms. Current calculation on 100 atom liquid water takes 2-3 month to finish on a 500 processor cluster (peak Flops: ~ 1Tflops/s, actual code runs on 0.1 Tflops/s). That will get results for 20 psecond. To do the same for nanocrystal synthesis, we need at least 1000 atoms. To get the same 20 ps within 2-3 month time, the computer power needs to increase a thousand time, that is one Petaflop. In reality, 20 ps might not be enough. 20 psecond typically only gives you a snapshot for a homogeneous system like water. Growth happens in the scale of a second ! So, new approach and algorithm is needed.</p>
<p>Projected increase in algorithm efficiency?</p>	<p>I guess when the number of atoms approach a few hundreds, the $O(N)$ algorithm for electronic structure calculation can kick in. From there, a linear scaling with the size can probably be reached. But the time to reach that is still a tough problem. Some accelerated MD or Monte Carlo scheme will be needed.</p>
<p>Other</p>	<p>Would you please clarify the use of the unit Tflop-years? We've assumed you meant a computer capable of executing 10^{12} floating point operations per sec continuously for one year. YES. Is this number the peak performance of the computer, or is it the performance sustained on the codes run? I think this is the theoretical peak performance of the computer.</p>

Microthermal Transport in PolySilicon: One Billion-Atom MD Simulation

Aidan Thompson, SNL Dept. 9235

The mechanical response of a polycrystalline silicon surface to intense local heating is of considerable practical importance to Sandia. In particular, a laser radiation source can be used to trigger a mechanical switch in MEMS devices. The material response is strongly affected by the rate at which thermal energy escapes from the heated area into the surrounding material. Lattice excitations or phonons are the dominant mode of microthermal transport in polycrystalline silicon. Though the phonons move through the crystal grains and across grain boundaries at the speed of sound, energy dissipation (i.e. thermalization) occurs due to anharmonicity of the interactions between atoms. In addition, reflection and dissipation of thermal energy at grain boundaries can significantly effect thermal transport rates. Sandia has developed a microthermal transport simulation tool in which the fundamental particles are phonons.[1] Fidelity is limited by the input models that describe how the phonons interact with the material, particularly at interfaces. The GRASP Molecular Dynamics (MD) code is being used to directly simulate the interaction of phonon wave packets with different interfaces. A wave packet is created by introducing a perturbation of atom velocities and displacements in a small region of an otherwise static crystal. The classical equations of motion of all the atoms are then integrated forward in time.[2] Despite the use of periodic boundary conditions, the simulated behavior is expected to be strongly affected by dimensions of the simulation cell, for several reasons. Firstly, the total duration of a simulation of the ballistic behavior of a wave packet is limited by the time it takes for a wave packet to traverse the entire cell, at which point it returns to its starting point. The speed of sound in silicon is 6.4 km/s or 64 Å/ps. This demonstrates that even short simulations require large system sizes. Secondly, dispersion of phonons tend to convert phonons from short wavelengths to longer wavelengths. Phonons with wavelengths longer than the system size are artificially suppressed. Finally, the representation of grain boundaries requires large system sizes, because the super lattice dimensions of an interface where two misoriented crystals meet can be arbitrarily large.

We propose to extend the current program of GRASP simulations to much larger system sizes to meet the technical needs described above. In addition, the combination of relatively short simulation duration (~1 ns) and the requirement of large system size makes the problem a natural fit for a large parallel calculation. Using a standard force field for silicon, the GRASP MD code can achieve 10,000 particle-timesteps/processor-second on a cluster of 466 MHz Compaq alpha processors (communication costs become small for more than 1,000 particles/processor). Assuming a machine peak rate of 500 MFLOPS, and assuming the code runs at 10% of peak, this translates to 5000 floating point operations/particle-timestep. A target periodic system would consist of an elongated box with 100,000 atoms normal to the grain boundary and 100x100 atoms in the plane of the grain boundary. The duration of the simulation would be 1 ns, or 1 million timesteps. This would require 5×10^{12} floating point operations or a peak machine rate of 0.6 PFLOPS, assuming a simulation time of 24 hours and assuming the code runs at 10% of peak. A complete investigation of microthermal transport would require, perhaps, 100 such simulations.

References:

- 1 Microscale Modeling of Energy Transference (ASCI Microsystems Project, Piekos 9113, Webb 1834)
- 2 P. K. Schelling, S. R. Phillpot, and P. Keblinski, "Phonon wave-packet dynamics at semiconductor interfaces by molecular-dynamics simulation", *Appl. Phys. Lett.*, v. 80 p. 2484 (2002)

Petascale Applications —

QCD

Impact of Petaflop-scale Computing: Application — QCD

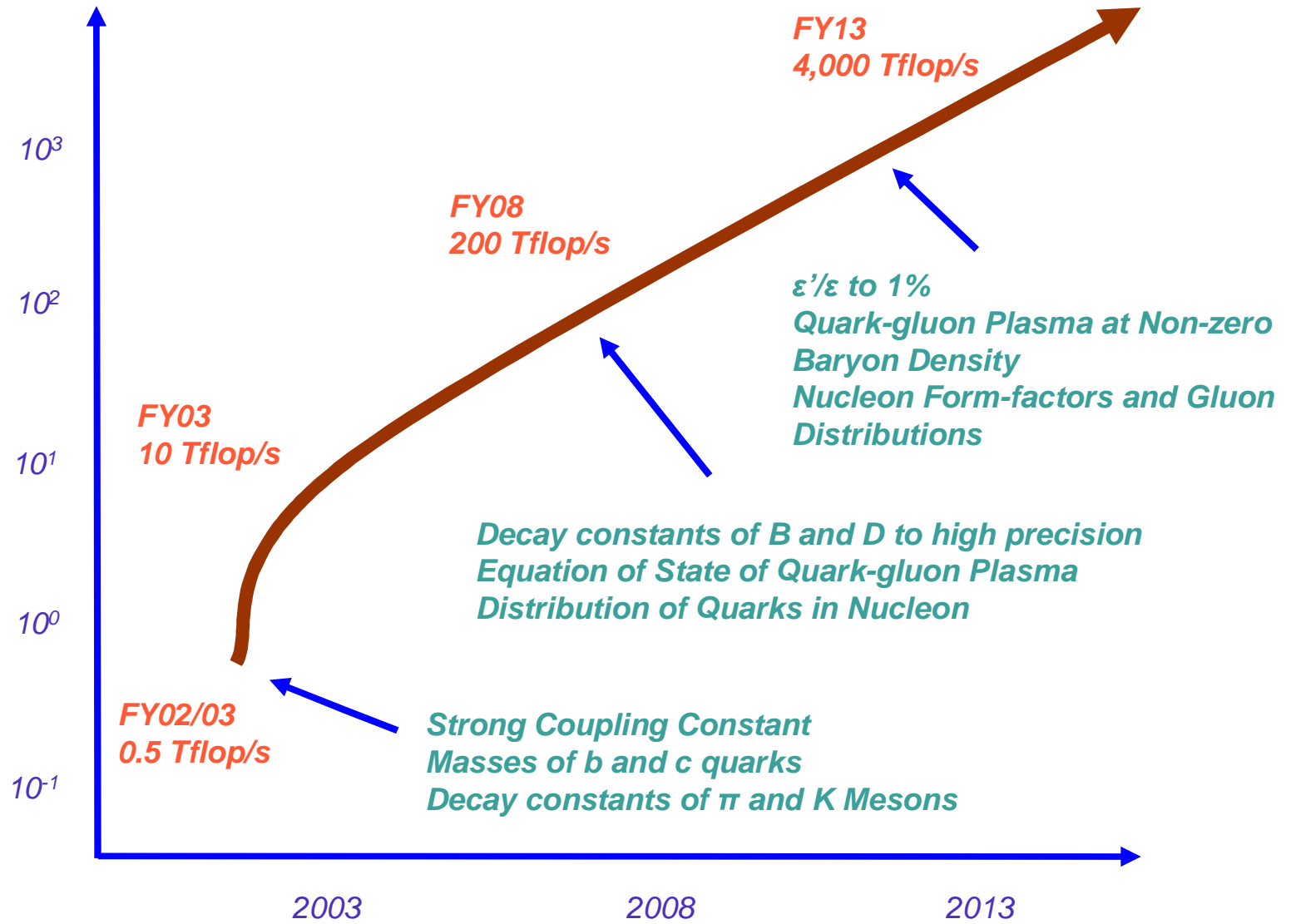
	Lattice QCD
Programmatic impact to be gained by access to Petaflop-scale computing	<ul style="list-style-type: none">o The weak interaction matrix elements of strongly interacting particles will be calculated to an accuracy needed to make precise tests of the Standard Model of High Energy Physics. A significant fraction of the Department of Energy's experimental program in high energy physics is devoted to testing the Standard Model in order to determine whether new physical ideas are needed to understand matter at the shortest distances. In many cases accurate lattice QCD calculations are needed, along with accurate experiments, to make these tests. In a significant number of cases the lattice errors are currently the major impediment to progress.o At sufficiently high temperatures and/or densities, ordinary strongly interacting matter undergoes a transition to a quark-gluon plasma. Petascale QCD calculations will determine the nature of the transition, the temperature and density at which it occurs, and the equation of state of the plasma. This information is needed to understand the development of the universe immediately after the big bang, and to interpret the heavy-ion collision experiments in progress at the BNL Relativistic Heavy Ion Collider.o Petascale QCD calculations will elucidate the quark and gluon structure of nucleons, and make precise determinations of the masses and decay properties of strongly interacting particles, including particles with exotic quantum numbers for which searches are in progress.
Major scientific challenges to be addressed	<ul style="list-style-type: none">o Perform simulations on finer grids in order to improve the accuracy of extrapolations to the continuum limit. QCD is formulated in the four-dimensional space-time continuum. However, in order to perform numerical simulations, a grid with finite grid or lattice spacing must be introduced. It is possible to match the lattice results to the physical, continuum ones through analytic calculations. But to obtain accurate physical results, one must perform calculations for a range of (small) lattice spacings. As the lattice spacing is decreased, the computational work increases as $(1/a)^7$, where a is the lattice spacing. Recent algorithmic advances have increased the lattice spacing from which one can obtain accurate results, and further advances are anticipated in the next several years.o The computational work to perform a calculation increases as the quark mass is decreased, approximately as $(1/m_q)^{2.5}$, where m_q is the quark mass. The

	<p>two lightest quarks, the up and the down quarks, have masses that are much smaller than the other quarks or than the typical QCD energy scale. At present, it is necessary to perform simulations with up and down quark masses larger than their physical values. One must then perform an extrapolation to the physical masses. A theory, known as chiral perturbation theory, guides this extrapolation. Petascale calculations will enable us to significantly improve the accuracy of the extrapolation by working at smaller up and down quark masses, and may eventually enable simulations at the physical up and down quark masses, eliminating the need for extrapolations.</p>
<p>What is the throughput (Tflops/s sustained) today on a <i>single</i> run of the longest calculations that are made?</p> <p>Please indicate the code efficiency and/or the computer peak performance.</p> <p>Please also indicate the turn-around time.</p> <p>[We are emphasizing <i>capability</i> – the ability to tackle big problems in a single computer run – rather than <i>capacity</i> (the amount of work that can be done with many runs.)]</p>	<p>What is perhaps the longest single run in progress is currently being carried out at the Pittsburgh Supercomputer Center. It uses 1024 processors of the Compact AlphaServer, Lemieux. The code has a throughput of 300 megaflop/s per processor on this machine, so the total throughput is 300 gigaflop/s. Lemieux's processors have a peak speed of 2 gigflop/s, so the code is running at 15% of peak. Approximately 6,400 processor-hours are required for each simulation time-unit, and 3,000 time-units will be needed for the full simulation. So, this calculation requires approximately 0.65 teraflop/s-years. Neither Lemieux, nor the NERSC IBM SP, Seaborg, have the capability to allow us to complete the calculation in a reasonable time (one to two years), given the load currently on them. We are counting on additional resources becoming available to do so.</p>
<p>What is the <i>typical</i> number of processors used for your code today? What is the <i>largest</i> number of processors used to-date?</p>	<p>At NSF and DOE centers large jobs typically run on 512 or 1024 processors. The largest jobs I am aware have run on 1500 processors at these centers. Jobs as large as 12,288 processors have been run on the special purpose QCDSF computer at Brookhaven National Laboratory, and jobs using several thousand processors are typically run on this machine, and on a sister 8,192 machine at Columbia University. The total throughput of the two QCDSF machines is said to be 300 gigaflop/s, so the performance per node must be of order 15 megaflop/s. Please note that these machines were built significantly earlier than Lemieux or Seaborg.</p>
<p>What is the Operations Count/Scaling from other computers?</p>	<p>The best way to calculate the computing resources needed for future calculations is to start with the numbers provided in answer to the last two questions, and make use of the facts that 1) computing resources scale as $(1/a)^7$ for fixed quark mass and physical size of the box in which the calculation is done. (The latter means that if we halve the lattice spacing we need to double the number of grid points in each of the three space and one time dimension to keep the box size fixed); the computing resources scale as $(1/m_1)^{2.5}$ as m_1 decreases for fixed lattice spacing and box size (Recall that m_1 is the average mass of</p>

	<p>the up and down quarks); and 3) computing costs scale as L^4 (L is a physical dimension of the box) for fixed lattice spacing and quark mass. The calculation on which I gave details above is for a $40^3 \times 96$ lattice, with a lattice spacing of 0.09 fm, a spatial box dimension of 4.5 fm, and a light quark mass of 1/10 of the strange quark mass, and the strange quark mass fixed at its physical value. With these facts, one can determine the resources needed for future calculations.</p> <p>If you really want to know the operations count, it is as follows. (This is for the formulation of the theory used in the above calculation. Other formulations, which have important uses, have different operations counts, but to include them would be going too far afield):</p> <p> O_{tot} = operations count for the full simulation O_{traj} = operations count for a single simulation time unit O_{site} = operations count for a single step at a single grid site N^l_{cg} = number of conjugate gradient iterations to invert the Dirac matrix for a light (up or down) quark N^s_{cg} = number of conjugate gradient iterations to invert the Dirac matrix for the strange quark. $N^l_{cg} = (m_s/m_l) \cdot N^s_{cg}$. V = lattice volume (in lattice points). So, for the example above, $V=40^3 \times 96$. N_{step} = number of steps per time unit t = the number of time units in the run. </p> <p>Then,</p> $O_{site} = 8,910,000 + 1,187 \cdot (N^l_{cg} + N^s_{cg})$ $O_{traj} = N_{step} \cdot V \cdot O_{site}$ $O_{tot} = t \cdot O_{traj}$
<p>Projected increase in software efficiency?</p>	<p>As in most fields involving large scale computations, algorithm improvements appear to play at least as large a role as increases in computing power in moving the study of QCD forward. In recent years there have been major improvements in formulating QCD on the lattice. These improved formulations require more floating point operations to generate a lattice at a particular lattice spacing and quark mass, but yield far more accurate results for the same input parameters. I have attempted to take this into account in a conservative</p>

	<p>manner. As Yogi Berra would no doubt tell you, predictions about the development of new algorithms are difficult, especially for those that have not been invented.</p>
<p>Other</p>	<p>Could you please provide Fig.2 from Scales 2. (Attached)</p> <p>Dear Ed and Peter,</p> <p>[Above] I have revised the table you sent me regarding the impact of petaflop-scale computing on QCD. A few comments first. In my Scales presentation, and in our field's writeup, all performance figures were given in SUSTAINED teraflop/s. That is, the actual performance of production code that would be used in the scientific studies. I continue to use sustained performance in this note. The percentage of peak speed obtained by good QCD code varies according to the computer being used. For most commercial machines it is of order 10% to 15%. For computers specially designed for QCD it falls in the range of 35% to 50% depending on the specific problem and specific special purpose machine. The unit that is most often used in measuring the size of a QCD problem is teraflop/s-years (TF-YRs). This is the number of floating point operations a computer SUSTAINING one teraflop/s would produce in one year. One TF-YR is approximately 3×10^{19} floating point operations. In order to produce results in time to be useful for the experimental programs in high energy and nuclear physics, the typical turn around time for a major QCD calculation should be of order one year. Finally, it should be noted that QCD calculations are large scale Monte Carlo simulations. A very large fraction of the computer time is spent in generating independent configurations, snapshots of the system being studied. These configurations can be used to obtain a wide variety of physics results. To improve the accuracy of the results, one has to perform simulations on finer grids and with smaller quark masses. Of course, the computational cost increases as one does so. To be specific, the cost of configuration generation increases as $(1/a)^7$, if all other parameters are held fixed. Here "a" is the lattice or grid spacing. The cost of configuration generation increases with decreasing m_l, the average mass of the up and down quarks, the two lightest ones. The cost varies as $(1/m_l)^{2.5}$.</p>

Tflop-year



Distribution List:

- 1 Dr. Everet H. Beckner
Deputy Administrator for Defense Programs
DOE, NA-10
1000 Independence Ave. SW
Washington, DC 20585
- 5 Dr. Raymond L. Orbach
Director, Office of Science
SC-1/Forrestal Building
U.S. Department of Energy
1000 Independence Ave. SW
Washington, DC 20585
- 5 Dr. C. Edward Oliver
Associate Director, Advanced Scientific Computing Res.
SC-30/Germantown Building
U.S. Department of Energy
1000 Independence Ave. SW
Washington, DC 20585-1290
- 1 Dr. Thomas Zacharia
Oak Ridge National Laboratory
P.O. Box 2008, MS6163
Oak Ridge, TN 37831-6163
- 1 Dr. Jeffrey Nichols
Oak Ridge National Laboratory
P.O. Box 2008, MS6164
Oak Ridge, TN 37831-6164
- 1 Dr. B. Ray Stults
Office Director, Office of Science Programs
Los Alamos National Laboratory
P.O. Box 1663, A127
Los Alamos, NM 87545
- 1 Dr. Andrew B. White, Jr.
Los Alamos National Laboratory
P.O. Box 1663, B297
Los Alamos, NM 87545
- 1 Prof. David E. Keyes
Dept. Appl. Phys. & Appl. Math.
Columbia University
200 S. W. Mudd Bldg., MC 4701
500 W. 120th Street

New York, NY 10027

3 Dr. Edwin H. Barsis
BMV Associates, LLC
1538 Catron Avenue SE
Albuquerque, NM 87123

3 Dr. Peter L. Mattern
BMV Associates, LLC
26 Juniper Hill Rd NE
Albuquerque, NM 87122

3	MS-0321	W. J. Camp, 9200
3	MS-0321	R. W. Leland, 9220
1	MS-9018	Central Technical Files, 8045-1
2	MS-0800	Technical Library, 9616