

## **Workshop Report: Petascale Computing in the Geosciences (updated 08/11/08)**

In view of The National Science Foundation's recent announcement entitled: Leadership-Class System Acquisition - Creating a Petascale Computing Environment for Science and Engineering, which calls for deployment of a petascale computational facility capable of sustained scientific applications performance approaching a petaflop ( $10^{15}$  floating point operations per second) to solve scientific questions of strategic importance by or around the year 2011, and because the time between now and then is not more than ample for scientists to prepare to use such a computer, a series of workshop around Petascale Computing in the Geosciences were organized to wit GARPA, GARPA2, Petascale Computing and the Geosciences I and II. With the permission of NSF, the workshops GARPA2 and Geosciences II were combined as they have largely overlapping goals. The objectives of these workshops was to examine the opportunities for progress in the geosciences that could be enabled by the petascale computational capability and to determine the steps necessary to ensure that this community is prepared to take advantage of such resources when they come on line. The joint workshop report is below, with a concise bullet list of *recommendations* provided first for ease of reference, and the more detailed *analysis and findings* that led to these recommendations follows. At the completion of the report there were still substantial funds left at SDSC. And two high-profile application tuning efforts were engendered as a result of the workshop between the Performance Modeling and Characterization (PMaC) lab at SDSC and investigators at NCAR, and the seismic wave propagation group led by Jeroen Tromp (two distinct collaborations). Therefore with the permission of the NSF program manager the funds were given a No Cost Extension and retargeted to support these follow-on activities (there was also an SGER award #0637994 to support the first). These two collaborations resulted in world records in performance of a non-hydrostatic weather simulation (WRF) and a seismic wave propagation simulation (SPECFEM3D) but more importantly new science; to wit the appearance of Rosby Waves in a global weather simulation, and the generation of a higher frequency wave through the Earth (approaching frequency of highest such waves that will propagate) than ever before achieved in simulation. These collaborations were finalists for the Gordon Bell Prize in two successive years 2007 and 2008 (result of 2008 competition pending). In many ways these collaborations prototype the kinds of inter-disciplinary collaborations called for in this report, and the conference papers resulting are here appended in Appendices A<sup>1</sup> and B.

### **1. Recommendations**

The potential benefits of petascale computing to advance scientific discovery in the geosciences, and to improve economic competitiveness of the U.S and, and the climate for all world citizens, and to better understand the world we live on, as identified in the analysis and findings below, are manifold. However, to achieve these benefits, several specific steps are required, both of the community of research scientists in the geosciences and computer science that should advance relevant science investigations, and by funding agencies that should provide resources to carry out these science investigations. The recommendations of this workshop, developed from the analysis and findings section are:

---

<sup>1</sup> This version of the WRF paper is updated with new performance results relative to the version turned in with the last Annual Report specifically this one contains the world record results.

- **A portfolio of candidate petascale applications should be established, and development funding should be provided, for collaborative teams of geoscientists and computer scientists to prepare calculations that can both advance scientific discovery and run at petascale.** As a community we need to assemble a portfolio of application classes on the path to petascale. We should *not* strive to make this a large, comprehensive portfolio at first, but rather seek to obtain high success by enabling a few strategic geoscience computations at petascale in the first few years of system availability, particularly to solve science problems of impact. This set of workshop participants identify the following set of broad geoscience application areas as being inherently suitable for development to go to petascale, and have science impact, both in the *near* term (it is not implied this list is exhaustive). Thus candidate applications may be considered from the following broad areas:
  - **Mantle Dynamics** (for example earthquake ground motion prediction using PetaShake at 25 meter, 2 Hz)
  - **Climate** (for example hemispheric “nature” runs to study energy spectrum in atmosphere at the  $k^{-3}$  to  $k^{-5/3}$  kinetic energy spectral transitions).
  - **Coupled Ocean and Weather** (for example a hurricane eyewall calculation with turbulence and mixing at sub 10m resolution)
  - **Space Weather** (for example a full hemisphere coronal model to resolve and improve understanding of loop structure in a constrained coronal heating with loop structures model)
  - **Ecological component of earth system modeling** (for example adding plant cover to climate models)
- Explicitly called out from the above: **projects to scale candidate geoscience calculations to petascale should be undertaken by collaborative teams of geoscientists and computer scientists starting now.** All the calculations identified in the analysis and findings section, even the relatively straight-forward ones, require work by both experts in the domain science and in the associated computer science, working together, to get ready for petascale.
- To ensure the kinds of facility access required to allow collaborative teams to make progress: **interactive access to large numbers of nodes, and a “hardware ladder” should be provided.** Software development and tuning teams today have difficulty obtaining access to 1000 cpus for debugging and interactive code-development. Yet petascale calculations will use 2 orders of magnitude more processors. NSF centers should provide interactive partitions and reservations for up to 10,000 cpus for short periods of time now. NSF should ensure phased deployment of Tier2 and Tier1 systems at 10,000, 100,000 cpus, and more granted to code developers along the way.
- Extending the above beyond the short-term: **petascale community should be cultivated.** To foster a growing interdisciplinary community of collaborating geoscientists and computer scientists we recommend organization of “summer institutes”, focused workshops for carrying these collaborations forward, and key training programs for next-generation interdisciplinary scientists in this field. These should be advertised widely to all agencies with funding/interest in geoscience including (beyond NSF) DOE and DOD (including DARPA and HPCMO). Petascale

application proposals should also be encouraged to propose “High-Performance Computational Geoscience postdocs”.

- Related to the above: **representative benchmarks that represent the computational requirements of geoscience applications should be defined.** A common geoscience benchmark suite would be an important contribution towards furthering petascale computing in the geosciences. It would also simplify procurements for vendors and geoscience application practitioners alike. For practitioners, the benchmark suite could be appropriated en-masse into multiple RFP’s. This would save time and effort in developing RFP benchmark suites. For vendors, by focusing on a single suite, would save money responding to RFPs, and would allow these applications to have greater influence on system design in the future.
- Complimenting the above: **performance models of representative applications should be deployed.** It will be difficult to get benchmark time (as well as development time) on the largest supercomputers. Therefore, we emphasize the importance of having predictive models of system performance based on key architectural parameters both to guide system deployment but (more crucially) code development and tuning for the target platform.
- **Suitable selection criteria should be applied in choosing which specific calculations and teams from the above areas to invest effort and money into for the purpose of enabling viable petascale applications (via several years of effort starting ASAP).**

We propose the following broad evaluation criteria:

- Reward due to science impact
- Strength and interdisciplinary nature of proposed team
- Demonstrated plausibility for petascale (via reports from team initial feasibility studies)
- Needed investment in algorithms and models (via reports from team initial feasibility studies)
- Risk of failing to result in a viable petascale application

We applied the first and last (risk versus reward) in identifying the broad application areas suitable for petascale in the first recommendation. Further effort is required to evaluate teams and carry out feasibility studies.

- Explicit from the above **initial feasibility studies should be carried out ASAP.** Collaborative teams should be formed to further assess the suitability of calculations, including from the broad areas of the first bullet, to achieve petascale. Performance studies should be generated showing where scaling bottlenecks are and strategies for working around these should be explored. To mitigate risk of failure, teams should not receive larger multi-year development funding before justifying further advancement via short (1 year or less) feasibility studies.
- **Innovative uses of a petascale computer should be cultivated.** There is a danger that the more speculative, innovative, uses for a petascale computer will be squeezed out early by the approach embodied in the first two recommendations. To mitigate this danger we propose an additional specialized RFP with modest funding for assessment and development of “shallow end of the pool” research that is “High Reward/High Risk” and “High Reward/High Effort).
- **“Market segmentation” should be done.** We do not believe it is the case all computational problems in geoscience are potentially petascale. At the same time, many geoscience applications may stress memory bandwidth, disk I/O rates, integer functional units, database query rates, in

ways that traditional high performance computing, floating-point intensive (such as physics codes solving systems of PDES (partial differential equations) ) do not. There is a need and opportunity for the community to examine computational requirements of geoscience applications, and, in addition to identifying candidates for petascale, identify attributes that may be used to influence such programs as NSF OCI (Office of Cyberinfrastructure) Tier2 system procurements. The result could be a machine at less than petascale but well suited to the data-intensive applications of geoscience.

- **Software support and maintenance should be supported.** We emphasize the NSF-wide need to develop a model and paradigm, for making software solutions robust, maintainable, and reusable. This likely requires some staff support, as opposed to graduate student, as students can/should invent but should not be expected to harden and maintain software.

The NSF GEO and CISE directorates need to increase investment in people and software applications development commensurate with the outlay in funding for hardware from OCI to enable petascale computing. We recommend a Geoscience Petaflop Computing Initiative, on the model of NMI to foster software development. A focus should be on specialized programs to capture the specific geoscience subfields for development in the context of petascale computing. There should be an emphasis on coupling and coarse-graining techniques to enable scaled-up coupled models. GEO should explore inter or cross agency coordination to address funding for Application Services, libraries, runtime/programming environments needed at petascale.

- **Storage and networking should be supported.** It is recommended that NSF support national infrastructure for data-intensive applications from the Geosciences, including storage and database resources. For example, CISE/DDDAS and GEO could jointly support a solicitation in this area.

The result of following these recommendations will, we believe, both enable a few early successes of petascale geoscience applications solving important science problems running on the petascale facility when it first becomes available, and enable the evolution of a balanced, robust, interdisciplinary, computational geosciences community and infrastructure going forward.

## 2. Analysis and findings

The workshops were structured around the following broad questions:

- I. What are examples of important questions and conceptual challenges in the geosciences that illustrate the potential impact of access to a petascale computational facility?
- II. What strategies will ensure that the geoscience community is in position to take advantage of petascale computational capabilities?
- III. What resources are needed to get teams of geoscientists and computer scientists working together on petascale applications development, with the goal of having operational packages ready by 2011 when petascale resources will come on line?

The remainder of this report describes the findings of around each broad questions.

## 2.A. Petascale Applications

We worked to identify candidate petascale applications in geoscience. The focus included critical issues, outstanding challenges, and potential impact, as well as on scaling up existing applications to petascale, and on turning important questions and conceptual challenges into petascale applications.

In the large view, some candidate petascale calculations in geoscience are already extant as applications, some are even running at terascale ( $10^{12}$ ), while others exist only as abstractions, models, and (in some cases) algorithms for solving them. For these latter conceptual problems, implementation is an issue; the quality of the implementation may be a larger factor than intrinsic suitability, or scientific importance, in determining their success at petascale. Even for the former cases (existing codes), there is no simple proof that either a) terascale applications will naturally scale to petascale, or that b) important science questions would be answered by so doing; rather, the suitability of each for scientific importance, scalability, and computational challenges, must be examined case by case. Still, in any case, validation is an issue. In many computational science problems, determining that a calculation is computing a result of high fidelity to nature and of scientific relevance, will be as or more difficult than implementing the application at petascale to begin with.

With the above larger issues in mind, three guiding principles were used by this group to identify possible candidate petascale applications in the Geosciences: 1) needs of the domain science are more important than simply enabling petaflop calculations 2) people are expensive, machines are cheap (or in other words software is expensive and hardware is relatively cheap), so designing, coding, porting, tuning, and validating applications will be at least as expensive as procuring petascale hardware, and as well are of utmost importance in this drive towards petascale computing. Furthermore, a rather ideological position was taken, that being that 3) “nothing scales to a petaflop, unless otherwise demonstrated or proven.”

With these guiding principles in mind, candidate questions and conceptual challenges were identified and deemed potentially suited for petascale within the available time, given sufficient resources, early start, and ample time for success. These questions are listed below in rough order of deemed readiness/nearness deployment at petascale, with the most mature candidates listed first. In addition, for each category of application, some risk/reward assessment is provided. Risk is loosely defined here as risk of either 1) failing to be deployed at petascale (i.e. due to difficulty of implementation) within the timeframe, or 2) failing to compute a result of significant scientific merit, or both. Reward is of course the opposite i.e. 1) likelihood of running at petascale at “first light” if sufficient resources are devoted to software development, and 2) importance of the underlying problem to geoscientists, or both.

As mentioned above, this is by no means an exhaustive list. It is not an exhaustive list of all the applications discussed at the workshops. The fact many applications are not on the list does not imply they are not important or not ready for petascale. In this report we simply highlight some specific calculation we believe can use the petascale facility at first light to solve important science problems if sufficient funding and effort is put into the development starting now.

### **Mantle Dynamics: Featured Calculation “PetaShake”**

Scientists believe that California is overdue for a large earthquake on the southern San Andreas Fault, and the opportunity is to better understand the basic science of major earthquakes and to apply this knowledge to prepare for them through such measures as improved seismic hazard analysis estimates, better building codes in high-risk areas, and safer structural designs, potentially saving lives and property.

The initial TeraShake simulations were unprecedented data-intensive simulations, producing more than 40 terabytes of data, revealed new insights into large-scale patterns of earthquake ground motion, including where the most intense impacts may occur in Southern California's sediment-filled basins during a magnitude 7.7 southern San Andreas Fault earthquake, and how basins flanked by mountains can form a “waveguide” that could channel unexpectedly large amounts of earthquake wave energy into the Los Angeles basin.

But the TeraShake simulations could reach a frequency of just one half Hertz, modeling only the lowest part of the frequency range of the ground motion. While these simulations provided information engineers can use to explore earthquake impacts on larger multi-story structures, more than 20 floors high, say, the much larger number of smaller structures remain “invisible” in the TeraShake simulations, which failed to capture the higher frequencies that interact with smaller multi-story buildings.

By reducing the computational grid spacing from 200 to 100 meters or even 50 meters, a PetaShake simulations will capture frequencies up one to two Hertz, providing information that can model earthquake impact on the larger number of smaller multi-story structures. In addition to higher frequencies, one can increase physical volume. These improvements in realism are very computationally intensive, however. Each factor of two improvement in frequency resolution increases by a factor of eight the required spatial grid mesh and another factor of two in the timestep, for a total increase of 16, strongly driving the need for the next generation of petascale computing resources.

This calculation is considered low risk high reward as almost perfect scalability has already been demonstrated on the 40 thousand cpu BGW machine.

### **Weather and Climate: Featured Calculation “WRF Nature Run”**

The development of the Weather Research and Forecasting (WRF) modeling system [1] is a multi-agency effort intended to provide a next-generation mesoscale forecast model and data assimilation system that will advance both the understanding and prediction of mesoscale weather and accelerate the transfer of research advances into operations.

It is suitable for use in a broad spectrum of applications across scales ranging from meters to thousands of kilometers. Such applications include research and operational numerical weather prediction (NWP), data assimilation and parameterized-physics research, downscaling climate simulations, driving air quality models, atmosphere-ocean coupling, and idealized simulations (e.g boundary-layer eddies, convection, baroclinic waves). For example, during the past three hurricane seasons, WRF has been run at NCAR in real-time to offer high-resolution (i.e., detailed) forecasts of storms which have threatened landfall.

The growth of computational power is enabling NWP model forecasts within the scale region defined by an observed  $k^{5/3}$  scaling in the kinetic energy spectrum. We have much to learn about how waves and turbulence interact, affecting predictability and optimal sub-grid parameterization, within this region and across the observed transition to larger scales. Without this understanding, we cannot take full advantage of the computational power at our disposal. The version of WRF to be investigated would produce a

suite of “nature runs” that can serve as a basis for current predictability, turbulence, and parameterization study in a multi-scale environment that spans scales above and below the spectral transition. This work would serve as the basis for study at grid scales beyond the capability of current HEC resources.

It is impossible to study predictability in the real atmosphere, making models necessary. The superiority of either increased resolution, or more probabilistic information, can only be established through basic predictability research. A nature run including the transition between the  $k^3$  and  $k^{5/3}$  spectral regimes would facilitate a new generation of predictability studies that are not currently possible. Simple identical-twin experiments on how errors grow within the  $k^{5/3}$  regime and across the transition could be performed. The hypothesis of enhanced mesoscale predictability near topography could also be rigorously addressed. It is additionally extremely difficult to study turbulence in the real atmosphere, and therefore models are attractive. The turbulence community faces several challenges that currently cannot be addressed. Wave-wave interactions within the  $k^{5/3}$  regime and across the transition are poorly understood. Wave-turbulence interaction occurs within the  $k^{5/3}$  regime and across the transition, for example in the jet-stream region of the atmosphere. This nature run will contain instances of both stratified and unstratified turbulence, facilitating their study in a rotating fluid on a sphere and in the presence of many other scales. It would allow the study of gravity waves in a realistic environment, and may include gravity wave breaking. Finally, new closure techniques are necessary for the next generation of NWP model implementations, and proposed stochastic approaches rely on spatially and temporally correlated statistics of mesoscale flows that are also extremely difficult to measure in the atmosphere. These must be quantified to take even the first step toward stochastic parameterization. A nature run that is long enough and with sufficient resolution could prove invaluable in beginning to understand their characteristics, which could then be exploited in next-generation parameterization schemes.

The WRF model is efficient on massively parallel systems, and offers the potential to make use of a great deal of computational power. For example, future goals include supporting a 1km grid resolution, which implies the use of approximately 7,000 petaflops per run, requiring about 4,000 hours on a current system, the “Blue Sky” system at NCAR, which is a 1,600 processor IBM Power 4 system.

The necessarily intensive computational resources required, and the use of a new NWP model, make this a risky endeavor with the potential for significant scientific and educational rewards.

Subsequent to the workshops, NSF GEO ATM funded an SGER (Small Grant for Exploratory Research) to investigate this computation. Substantial progress was made resulting in selection to the finals of the Gordon Bell Prize at SC07. The draft version of the accompanying paper as accepted to the SC07 Gordon Bell track is included as Appendix A.

### **Coupled Ocean to Weather: Featured Calculation “Hurricane Eyewall Intensity Forecasting”**

Over the last several decades, hurricane track forecasting has improved significantly, whereas very little progress has been made in hurricane turbulence, wind intensity, and rainfall along the track. We know where the storm is going but not what it will do when it gets there. Our lack of the skill in intensity and rainfall forecasting may be attributed to two deficiencies (among others) in current computational prediction models: insufficient horizontal resolution and lack of full coupling to the ocean. Extremely high winds around the eye, intense rainfall, large ocean waves, and copious sea spray

push the surface-exchange parameters for temperature, water vapor, and momentum, into regimes unreachable at current levels of model resolution and coupling.

A key to improving hurricane intensity forecasts is to have numerical simulations that are capable of resolving the inner core structures (eye and eyewall) and rainbands in a hurricane and can realistically represent the physical processes governing intensity change, such as the transfer of heat, moisture, and momentum at the air-sea interface, the phase changes of moisture in the atmosphere, as well as flux-carrying turbulent eddies that affect mixing. The next-generation prediction simulation models should then be able to resolve features with very high resolution to capture the gradients across the eyewall boundaries (at  $\sim 1$  km or less), but also capable of representing the turbulent mixing process correctly (at  $\sim 10$  m or less). Rapid increases in available compute power, and recent advance in technology in observations, have made it possible to consider a strategy for deploying the next generation of high-resolution hurricane prediction models. We begin by examining key issues related to grid resolution.

A difficult question to address is the convergence of solutions for hurricane intensity and structure at high grid resolution. An important part of the science is the generation of resolved turbulence in the model. We have conducted simulations on a grid as fine as 185 m. In order to obtain the same intensity as on a grid of 555 m, the viscosity coefficient in the turbulent kinetic energy scheme had to be increased (doubled), upon which the results of coarser resolution were reproduced. The model did not generate turbulence on its own with sufficient intensity to provide the necessary mixing to equilibrate the storm. We would like to explore if, with a grid spacing of 10 m, there will be sufficient turbulence generated to cease intensification. This exploration requires a petaflop calculation.

A related issue: for an idealized, azimuthally invariant initial vortex in a uniform environment, only small-amplitude asymmetries in the eyewall were noted. When the resolution reached 185 m, and with larger mixing (such that the storm intensity was the same as at 555 m), there were pronounced asymmetries that developed in the in the eyewall. An important question, “is what is generating these asymmetries?” and what will happen to them when the resolution is increased by an order of magnitude? To explore these science questions, one could conduct a number of idealized WRF model simulations with 10 meter horizontal grid spacing and 150 vertical levels. Computationally, this entails a 15-billion cell inner-most WRF 10 meter nested domain, with a model time step of 60 milliseconds, and we roughly estimate it will require 18 machine hours per simulated day at a sustained petaflop/second if we can work through issues of scalability, load balancing, and I/O to enable an efficient petascale deployment.

In addition to the model grid resolution exploration, the intensification and decay of a (real or modeled) hurricane largely depends upon two competing processes at the air-sea interface: 1) the heat and moisture fluxes that fuel the storm and 2) the dissipation of kinetic energy associated with wind stress on the ocean surface. Air-sea interaction is especially important in the region between the center of the eye and eyewall where there are extremely large gradients in the wind, temperature, and pressure fields. We wish to further explore the effect of air-sea coupling at very high resolution by using coupled modeling system including a surface wave model and ocean circulation model at grid spacing of  $\sim 1$ -2 km. This addition will increase the size of the computational problem by about 2-fold in data and flops.

This calculation is considered high is high reward as not all the physics governing highly turbulent storms is fully understood.

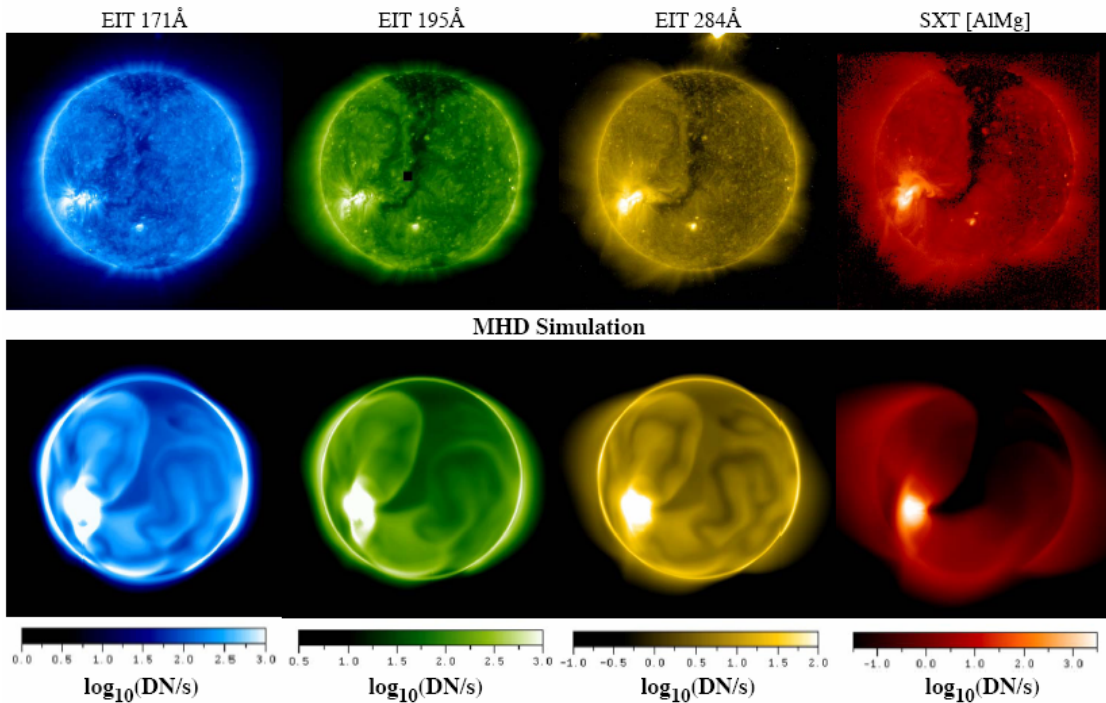
## **Space Weather**

The National Space Weather Program (NSWP) has defined "Space Weather" as the conditions on the Sun and in the solar wind, magnetosphere, ionosphere, and thermosphere that can influence the performance and reliability of space-borne and ground-based technological systems and can endanger human life and health.

As our nation becomes more dependent on advanced technology, it can become increasingly vulnerable to space weather. This is why the National Science Foundation has taken a leadership role in the National Space Weather Program. The solar magnetic field is the ultimate energy source for the most serious space weather effects. Through the medium of the solar wind, coronal mass ejections (CMEs) and solar energetic particles (SEPs) propagate and interact with the Earth's magnetosphere and cause some of the most serious space weather effects. The MAS (Magnetohydrodynamic Algorithm outside a Sphere) code [1], proposed for evaluation, is a state-of-the-art magnetohydrodynamic (MHD) model for realistically predicting the properties of the solar corona, and for investigating the phenomena of solar activity such as CMEs. The MAS code is presently being utilized in the Center for Integrated Space Weather Modeling (an NSF Science and Technology Center based at Boston University) as one of a suite of coupled physics models to describe space weather in the entire Sun-Earth environment.

The solar corona exhibits a wide range of plasma regimes, from strongly magnetized, slowly flowing plasmas low in the corona near sunspots to supersonically flowing plasmas in the solar wind. A successful numerical model must calculate efficiently in these different regimes. The MAS code is relatively mature and has been in use for some time for modeling the solar corona and solar wind. It is built on a rich base of experience in computational physics and the modeling of solar coronal and fusion plasmas. The code has been designed to work effectively in these regimes, but it has not been exhaustively tuned for performance. One of the principal difficulties in the physics and thus in turn for the performance of the code is the wide disparity of time scales present in the coronal plasma. Solar phenomena are often characterized by a slow evolution followed by an impulsive, rapid response as a result of instability, loss of equilibrium, and/or magnetic reconnection. The complex physics makes the computation in turn very challenging; there is evolution of boundary data, the need for staggered, non-uniform structured meshes and implicit and semi-implicit differencing. The comprehensive physics model includes energy transport processes (radiation, parallel thermal conduction, and coronal heating). The MAS model has already run with reasonable (~30%) efficiency on one thousand processors of SDSC's DataStar. The result of a calculation of the solar corona for a specific time period is shown in Figure 1 and compared to observations.

**Quantitative Comparison Between Observed and Computed Coronal Emission**  
SOHO/EIT and Yohkoh/SXT Observations on August 27, 1996



Calculations with  $\sim 10$  million grids take several days on DataStar to compute several days of real time. These calculations just scratch the surface of what is possible. In the present calculations, the photospheric magnetic field measurements (magnetograms) used as boundary conditions are much coarser than are available. Incorporating higher resolution magnetograms is crucial for understanding the details of coronal structure and eruptions of the magnetic field, but will require more than order of magnitude increase in the total number of grid points

This calculation is considered high risk high reward as the underlying physics, particularly of coronal loops, are but poorly understood.

**Ecological Component of Earth System Modeling: Featured Calculation “Groundcover for Climate”**

There is an opportunity to use a petascale facility to enable adding ecological information, such as forest growth, to models of weather and climate. A grand challenge in geoscience is the addition of clouds to ecological modeling. However, clouds and cloud-formation processes interact (in both directions) with the ecosystem.

Ecological models attached to a high-resolution model of climate change (or ocean circulation, etc) that already has clear petascale applicability on its own may be a good path forward. It is debatable whether ecologists will have much say in the design or deployment of these models, unless they begin to collaborate with the climate modelers as soon as possible. Ecologists have several embarrassingly parallel applications that can be “easily” scaled to a petascale system, including:

- stochastic processes that need replication
- parameter sensitivity analysis

- heuristic optimization

Such problems are extremely common in ecological application, as we elaborate here.

1. Stochasticity: Simulation-based ecological models often incorporate demographic stochasticity (random birth/death/movement, etc), environmental stochasticity (random components of climate forcing, resource availability, etc), and/or genetic stochasticity (random mating, mutation, etc). Outcomes are thus stochastic as well, and ecologists wish to ask questions like, “What is the simulated probability that the population size will fall below X within 100 years?” The simulation model must therefore be independently repeated (usually 100s-1000s of times) to generate a distribution of outcomes.
2. Parameter sensitivity (or more generally, model sensitivity): The “true” parameters of ecological models are rarely known, and in fact there are often disagreements about the form of the equations governing those processes. Consequently, ecologists frequently want to characterize the sensitivity of outcomes to input parameter values and model assumptions. This also requires repeated simulation.
3. Optimization: There are (at least) two distinct types of optimization questions that ecologists commonly ask. The first involves fitting parameters to observed data. In all but the most trivial models, it is impossible to use analytical or even simple approximating techniques to identify maximum likelihood estimates of parameters. Increasingly, ecologists are turning to stochastic optimization techniques such as simulated annealing, or the use of various implementations of Markov Chain Monte Carlo to simulate posterior probability distributions in a Bayesian framework. Secondly, applied ecological models often implement heuristic optimization algorithms as decision tools (e.g. identifying the optimal spatial configuration of a land reserve system, given some cost criterion). As with parameter estimation, the simpler algorithms used in the past have been shown to be deficient in complex settings, but more reliable methods require many repeated simulations and long run-times. There is a tremendous need for HPC solutions that can deliver results sufficiently quickly even for models involving many parameters, fine-scale spatial and temporation resolution, and stochastic processes.

Putting this all together, it is clear that the compute time can be overwhelming when coupling one or more the above procedures with even a moderately complex ecological simulation model. Specifically, some model examples include predicting evolution of a collection of interacting species, spatial spread of a disease, or the dynamics of a specific ecosystem. Taking the last example, imagine a regional-scale ecosystem model, the core of which is deployed as a small-scale HPC application (e.g. a single simulation that takes days to complete on a cluster with dozens of nodes). Indeed, the ATLSS group (<http://www.atlss.org>) based at UT/ORNL has spent ~10 years developing and refining a model that integrates a variety of complex and interacting submodels to simulate key Geoscience and environmental components of the Florida Everglades; AFAIK, one submodel has already been parallelized to run on 60+ nodes. Even if a researcher demands just several hundred stochastic replications in such a simulation, performed for each of 100 possible configurations of a proposed reserve system, there would be significant benefit from hierarchically parallelization, to enable a 100k-processor system run (imagine a multi-hundred simultaneous, distributed instantiation of the ecosystem simulation, which itself might be a 64-node data-parallel application). Whether the envisioned petascale system even provides the right architecture for this application could be debated, but the point is that it does not

require significant effort to scale up moderately sized ecological models to result in large computational needs, resulting in the ability to address relevant and interesting problems.

Data integration would be critical to success. A candidate calculation would involve evolutionary correlations of networks and functions (phenotypes). In the worst case, such formulations may lead to NP Hard/Complete problems that would remain intractable even at the petascale level. To the extent that ecologists are able to refine mechanistic mathematical models in a way that is increasingly faithful to reality, one could easily conceive of petascale computing demands for simulating an entire ecosystem from its underlying biological and physical components. However, it is worth pointing out that the tradition in ecology is to simplify and scale back models—indeed, to err on the side of oversimplification; “realistic” models have long been mistrusted in favor of either highly abstract mechanistic (theoretical) models and/or simple phenomenological (statistical) models. In part this is for good reason: ecologists do not yet fully rely on their own more detailed mechanistic models (there being a lack of the ecological equivalents of physical laws, testing approximating models via experimentation and observation is difficult, and each real system seems to have its own unique features). This could in fact partly be historical artifact: few ecologists are even aware of the computational possibilities now afforded by HPC systems. In a sense, one might argue that developments in this area are limited due to apparent belief in computational obstacles that no longer exist—something we believe can be remedied through education and workshop opportunities. Given opportunities for making forays into developing complex ecological simulations despite uncertainties about the models, and having the ability to enhance model output with observed data, has the potential to lead to refinement and progress in this area.

Choosing applications focused on the coupling of ecological with weather and climate models, and supporting them for further work for deployment at the petascale level, appears to be relatively medium risk for success in the given time frame. Much of the risk is dependent on collocating two disciplines that normally do not interact collaboratively, to facilitate synergistic code co-development. Computational possibilities in this area have the potential for relatively high reward towards addressing interesting geoscience problems. The path towards petascale is relatively straight forward, and the science questions of great relevance for better understanding of our environment by the ecological communities.

## **2.B. Petascale Planning**

This group dealt with strategies to prepare for petascale computing, such as: 1) the need for interdisciplinary collaboration, 2) selection criteria to determine which interdisciplinary teams to sustain (given limited funds and resources), 3) software challenges on the way to petascale, and 4) community-wide organizational structures that may foster the long-term needs of petascale computing.

### **1. Need for Interdisciplinary Teams**

The petascale funding announcement has stimulated an intense and broad degree of interaction between computational geoscientists and computer scientists. Our goal is to build a focused R&D effort that simultaneously measurably impacts the quality of life across society, demonstrates leadership and vision in large scale computing for the geoscience sciences, and trains the next generation of engineers, scientists, and mathematicians. To realize our goals we need to formalize the notion of “petascale computing” into an activity worthy of computer scientists’ and geoscientists’ dedicated effort, and that, in turn, enables them to succeed in their research career and in their core mission of training students. To

create and strengthen a community, computer scientists and geoscientists, and their postdocs and students, should have the possibility of receiving travel and living support to visit each other or the petascale site for months at a time.

Long term funding, with a minimum of 5 years, will be needed to ensure continuity, and it should cover all participants: computer scientists, geoscientists, programmers, and so on. This funding should encompass not only the research itself, but the sustained development of application and software libraries as they grow and require maintenance.

Projects should specifically be constructed as interdisciplinary teams working together to deploy geoscience applications at petascale. It will be crucial to have models for performance analysis and scaling to show that an algorithm scales *before* implementing it. There is also need to develop models for quantifying uncertainty (accuracy) and effort put into quantifying performance and accuracy of heuristics (performance modeling); these activities perforce require interdisciplinary collaboration.

## **2. Selection criteria for determining suitable petascale applications**

It is important that selection criteria lead to the formation of outstanding and productive collaborative teams focused on petascale activities. Historically, each 10-fold increase in parallelism required significant time investment, even when the raw compute capability was already available. Finally, since the petascale system will be a scarce resource during the first few years, its use must be managed transparently so it is in line with the selection criteria, and the needs of the funded efforts.

Therefore it is important to understand which applications are “peta ready” today (as was partially addressed by the list of applications above), and which will be ready tomorrow. Moreover, one must also prepare for the emergence of novel classes of calculations and applications that may be as important, or more, and as suitable for petascale as those of today.

One approach would be to distinguish among focused and large-scale development efforts and then apply the appropriate selection criteria and level of funding to each. For applications deemed “peta-scalable” by first-light (2011), the following criteria are important: qualifications of the team, size of the potential user base, suitability of the calculation for petascale resources, and the balance between scientific merit and impact versus risk and feasibility. Specifically, some of the most peta-scalable applications may be relatively low risk but lead to less practical results; while other applications likely to be of great geoscience significance are clearly at the edge of feasibility. It is important that application be evaluated along both dimensions of the risk-reward space and that the resulting portfolio mirror this spread.

The mix of applications may thus include some that are nearly ready for the petascale, others of high importance that require additional software engineering and algorithm improvement, or yet others that require a build-up phase towards substantially more compute-intensive usage. Other types of diversity to consider for support should include both monolithic single applications that spread across the entire machine, and heterogeneous applications where different components of the applications workflow stress the computer architecture in different ways.

## **3. Benchmarks**

Any benchmark suite, to be useful, must measure end-to-end system performance, a key metric for getting science done. In particular, benchmarks that measure the I/O subsystem performance in a variety

of application driven situations are critical. The vendor community in attendance strongly suggested that the application suite contain production benchmarks. This will provide the vendors with a revenue incentive when considering working with these benchmarks. This view was tempered by the idea that benchmark applications should come with unit tests to facilitate the porting and validation process.

A benchmark suite was proposed consisting of the WRF mesoscale meteorological model, the POP 2 global ocean model and. Since the workshop, the CICE-4 sea ice model and the Southern California Earthquake Center (SCEC) TeraShake application have been put forward as additional benchmarks, and the parallel NetCDF I/O benchmark IOR has been suggested as the first I/O code for the GARPA benchmark suite. It is anticipated that other candidate benchmarks will emerge at the second workshop.

#### **4. Software challenges on the way to petascale**

An overwhelming consensus from the group, and indeed the workshop attendees across all groups, is that there is a need to increase support for theory, algorithm and code development, and software engineering to get the best science out of petascale and other high end machines.

A possible format for such support would be a two stage competition for software engineering centers, with the first stage being planning grants and the second stage being full software engineering centers, with each center focusing on a particular application or class of applications. The subject of these centers should include applications services as well as applications themselves, as both levels need considerable effort. There should be provisions for individual investigators who have distinct contributions to make, to contribute to the work of the centers without being an integral part of the centers.

Issues of intellectual property for software rights will be critical to work out in advance, as intellectual property considerations could cripple the dissemination of associated advancements in software, and the development of heterogeneous computing environments.

To ensure that the geoscience community is positioned to take advantage of petascale computing capabilities, it is important to support efforts sustained efforts for the long term, i.e. a “20 years” timeline. A minimum of 5 years is recommended to establish retooling of (large) software, in order to successfully to take advantage of what a petascale system has to offer. Other software will be enabled more quickly, but to focus on only these would be irresponsible to the field of geoscience and the opportunities being presented by establishing a petascale facility.

For support of applications, there is a need to invest in application libraries. With the help of source-to-source transformation (for example, Dan Quinlan's ROSE effort at LLNL) it is possible to implement semantic optimization of class libraries. Efforts of this type can greatly facilitate geoscientists, enabling them to work in terms of abstractions they are comfortable with. Additionally, however geoscientists can greatly benefit from learning key aspects of computer science in focused training sessions key to their application areas (see below.)

Some problems pose challenges for hardware/algorithms (PME/Fast Fourier Transform) and may be suitable for more innovative algorithmic approaches/hardware. Several candidate applications do not fit into traditional numerical frameworks. For example, they may not be floating-point operation intensive or perhaps have poor spatio-temporal locality. This underlines the importance of identifying the computational needs of the candidate applications in advance of deployment to insure petascale

infrastructure is appropriate to their needs (which should be among the first activities of collaborative groups).

Many of the applications identified require multiscale modeling, relevant to many areas of science and engineering including but not limited to computational geoscience (for example, the DoE has such a program for material science). There are various software and algorithmic challenges inherent in such models, in addition to the obvious physics/chemistry challenges of coupling across scale. In terms of the former challenges, for example, managing the components in a petascale system, in particular, “cross-component optimization”, is an important issue. That is, how to optimize multiple components taken together, rather than the classic approach of optimizing individual components separately. Thus, issues of load balancing, tolerating communication delays, memory hierarchy optimizations, etc., all increase complexity of the problem. All of these issues raise interesting research questions, as well as the need for software techniques needed to handle the optimizations. These issues are related to work being performed by the common component architecture (CCA) community, but complimentary in this domain. Frameworks are needed to express such optimizations, and to enable application developers to express, in the form of “application performance meta data”, the information needed to sensibly and collectively optimize the multiple components handling the different parts of a composed multi-scale simulation.

Scalability is a primary requirement for petascale; achieving it involves communications optimization, load balancing, asymptotic complexity analysis, and numerical accuracy assessment. When different models are composed, it is important that their numerical interactions be consistent, stable, etc. Models can not always be composed in a straightforward way, and it is important to understand their collective behavior and associated physical laws. Load balancing across different model; scales and data layouts is particularly hard and ill-understood at present. More research is needed in this domain, and this need is urgent in light of the number of likely processors in a petascale system (conservatively lower-bounded at 100,000).

Many of the issues arising out of multi-scale models are potentially relevant to *any* application running on a multi-scale system. Tolerating communication delays, and handling load balancing are much more difficult at extreme levels of parallelism ( $10^5$  to  $10^6$  processors), because done incorrectly, serious waste in compute resources results. Thus, software techniques that facilitate latency tolerance will play an important role in helping ensuring scalability and efficiency of resources.

## **5. People and Infrastructure organization issues on the path to petascale**

Strategies for long term preparation of potentially new petascale applications include the development of training programs, workshops, and summer schools, with focus on teaching the craft of creating efficient codes. At the same time, sufficient effort needs to be afforded to parallel geoscience application middleware development including appropriate run time systems, frameworks, and libraries. Leveraging of existing and successful programs in these areas is highly desirable.

There is a need to pay specific attention to the workforce pipeline for the applications software development and engineering required. In addition, more focused training programs that solicit proposals from multidisciplinary teams, to jointly train students in high performance computing for geoscience, should be considered. We have a consensus that high performance computing is relatively neglected in computer science departments around the country, and that this is a problem that should be addressed by incentives to train students in high performance computing.

There are also societal issues to be considered. Petascale computing is not simply an extrapolation of prior experience, but raises the stakes for application developers. Raising the scale of parallelism by a factor of 10 will compel some to rethink the algorithm, the implementation, or other software issues, including development costs. The impact of raising the level of parallelism by multiple orders of magnitude opens up considerable opportunity for the geoscience community to take advantage of. Taken from a different perspective, there are quite possibly unknown applications that become possible with the advent of petascale parallelism, which has the potential for opening up entirely new avenues for inquiry in Geoscience, Computer Science, Math, and Physics. These observations bring up two (possibly opposing) viewpoints.

- i. Petascale computing will start out as a small club gradually becoming more widespread over time, at which point we begin again with exaflop ( $10^{18}$ ) computing. It is important that a small number of likely success stories be chosen at the outset, endowed with the human resources needed to thrive, enabling the demonstration of the capabilities of the machine early on.
- ii. Appropriate training should be provided with a long term vision of developing new communities of users: to ensure that a critical mass of experts be available to respond to and disseminate information about new developments, and to develop new software techniques that will be useful to the computational geoscience community at large. Summer camps and summer schools should be held to teach computational geoscientists the latest software techniques, and to teach computer scientists the latest trends in computational geoscience. This should be an ongoing process. Resources should be set aside for innovative ideas to blossom, be they in computer science, computational geoscience, applied mathematics and physics, or other related fields. Small time users should be given the opportunity to use the full scale machine for trying out their ideas.

We believe there is a middle way, embodied in the proposal above to select a limited but diverse portfolio of applications and teams, and that this can foster both viewpoints. The result should be some early success at “first light”, but also more speculative ongoing research and innovation that can lead to new uses of the petascale computational facility to address emerging questions in geoscience.

## **2.C. Petascale Infrastructure**

This group dealt with identifying the computing, software, storage, networking, and people infrastructure needed for geoscience at the petascale level. A high-level take home message is “*invest in people and software at (or preferably) beyond the level of the hardware investment!*”

### **Hardware infrastructure**

Geoscience problems of interest, for example those identified above, are data intensive, compute intensive, communication intensive, in variant combinations, and one size does not fit all. Geo-computing will therefore need multiple types of architectures and resources that map to the diverse hardware portfolio planned by NSF in their “tiers 1, 2 and 3” planning. However a form of “market segmentation” needs to be done to determine which calculations should be done where and to influence some of these architectures to be designed with the special needs of geoscience computing in mind. This is related to the idea that interdisciplinary teams need to start by understanding, modeling, and extrapolating future application requirements before embarking on ambitious code development projects.

Several of the candidate applications described above could benefit from application-specific architectures. Matching heterogeneity in applications and architectures across the NSF portfolio will be very important. It is expected that many candidate petascale applications in geoscience will be bandwidth intensive, with respect to local memory and with respect to inter-processor network bandwidth demands. In fact some key applications may turn out to be solved faster on what NSF terms “tier 2” systems than the petascale system, if those systems are better balanced in terms of memory and communications bandwidth per-processor. Therefore, it will be important to study computational characteristics of applications and associated hardware characteristics in advance to identify memory and communications bandwidth sensitivities. Likewise, it will be important to quantify what portions of candidate calculations are very computationally intensive and could be carried out on coprocessors (such as ASICs, FPGAs, DSPs, GPUs) likely to become available in the same timeframe. Likewise, it will be important to understand which applications are very communications and I/O intensive and will stress machines in these dimensions. In the design of any petascale or “tier 2” I/O infrastructure, it will also be important to address data federation, data availability, and integrity. Also integration of data acquisition systems (sequencers, microarrays, imaging) needs to be addressed more than at present.

Deeper considerations of the specific needs of compute intensive versus data intensive geoscience applications need to be made, as these may not be easily separated. Data intensive applications require stable and scalable file systems, and infrastructure for moving the data in and out of the computer. When such an application is generating hundreds (or thousands) of petabytes, the infrastructure must support storing data, mining and analyzing data, moving data, archiving data, and visualizing data.

By the same token, compute intensive calculations come in different types requiring (1) considerable amounts data, (2) considerable numbers of CPUs, (3) considerable amounts of memory (4) real time/wall clock constraints and (5) combinations of all the previous. Also, even compute intensive applications are not always computing “just one number” as the output. Rather many will generate petabytes or more of output data even if they did not consume a similar amount of input data to start with. Thus even these may require data intensive post processing even if the petaflop calculation is not by itself data intensive.

A related issue, particularly in the cases of applications requiring the movement of vast amounts of data, concerns geosciencenetwork issues. Schemes need to be developed for petabyte data transfer via the internet. This may require upgrades to existing national backbone networks but also, quite seriously, this may involve Fedex ala NetFlix (order data via the Web for next day arrival).

All of these challenges imply rethinking out-of-the-box around new architectures for geoscience computing, not just focus on refitting of existing geoscience problems to fit the petascale (or other high level) facility.

## **Software Infrastructure**

Software costs more than hardware. A strong consensus of the workshop participants is that currently there is an imbalance in NSF support for scalable, robust, easy to use scientific software relative to proposed investments in hardware. Enabling petascale computing in geoscience will require software infrastructure enabling data analysis, mining and visualization. Analyzing massive output data and visualizing will then require more than just high floating-point capability by way of infrastructure; candidate petascale geoscience applications present tremendous issues associated with data handling

(federation of data sources as for example expression, sequence, phenotype, etc.). These applications will stress I/O and file systems, and data federation solutions. Algorithms will need to be developed to deal with uncertainty in data, missing data, and erroneous data (sensitivity analysis). Furthermore, there are two key issues involving interactivity around large-scale data: (1) inordinate amounts of data to move, store, analyze requiring infrastructure supporting interaction (2) human beings often will need to be in the analysis loop. Additional challenges are associated with the connection of sensors and data streaming, as data access rates for I/O become very important.

Viable infrastructure will also need to include scalable codes, scalable algorithms, and scalable memory as well as lots of cpus as was expounded upon by Working Group II above.

In addition to the significant work required in fundamental algorithms and load-balancing, latency tolerance methods, etc., as described, significant efforts need to be focused in the areas of queuing and scheduling. Currently, queuing and scheduling systems do not do a good job of handling different types of needs. There is minimal ability to schedule high-performance computers to accommodate real-time constraints, respond to embedded sensors, be available on demand, and the like uses of interest to geoscientists. Current scheduling policies primarily service throughput jobs. While it may be that this is also deemed to be the best way to manage the petascale system, there is significant doubt. Likely, increased programmer productivity, increased breakthrough science, and better response to the end-user may result from a less heavily loaded resource; one that is reserved for fewer truly petascale calculations, including perhaps some with real-time constraints, rather than the currently heavily loaded, highly utilized, NSF systems.

Additional issues related to software involve fault tolerance issues, which are currently not being adequately addressed in designing software infrastructure for petascale. Given likely state-of-the-art reliability and hardware failure rate trends, it is anticipated that one processor out of one hundred thousand (or a million) will fail every minute on a petascale machine. Who and how does one deal with such failures/minute? By way of example using current semantics, an MPI global operation will block if even one processor fails to respond resulting in code hang-up. Applications need to be re-written to be fault-tolerant, something currently not even possible without updating current semantics of MPI to enable more tolerant of failures. Either vendors need to develop, or the community needs to develop (more likely the latter) fault tolerant APIs and associated semantics for global message-passing systems, in order to enable large parallel codes to be re-written in a fault-tolerant style.

Generally speaking in high-performance computing (not just in geoscience), there is a dearth of scalable, robust, easy to use, interactive, etc. software tools. Many tools that do exist for the purpose are “professor-ware”, so there are lots of tools but not always with the required reliability or associated documentation. NSF could take the lead in finding mechanisms to fund enduring and stable tool efforts and in requiring periodic peer-review of ongoing tools projects.

## **Networking Infrastructure**

### **People Infrastructure**

People cost more than hardware. It is important there be a proportional investment in the people - faculty, staff, and students - who will support the necessary and vital efforts to obtain petascale computing levels. Additionally, true peer collaboration is hard and circumstances must be fostered to overcome discipline silos. Interdisciplinary teams are necessary, involving geoscientists and computer scientists, but also other key disciplines (e.g., math, physics, sociology, economics, etc). Interdisciplinary teams may take the form of 2x2 collaborations as well as “service shop software

models". It is crucial these teams obtain early access to software and hardware at the teraflop level and higher on the path to petascale as it becomes available in order to prototype algorithms and software up to petaflop level computing.

## **WORKSHOP ATTENDEES:**

Richard Moore (SDSC).  
Allan Snavely (SDSC).  
Thomas Jordan (USC)  
Alan Wallcraft (NRL)  
Rich Loft (NCAR)  
Rob Pennington (NCSA)  
Michael Wehner (LBNL)  
Darren De Zeeuw (University of Michigan)  
Shijie Zhong (University of Colorado)  
Laura Carrington (UCSD)  
Kathy Yelick (Berkeley)  
Pat Worley (ORNL)  
Alan Sussman (UMD)  
Shirley Moore (UTK)  
Otto Fringer (Stanford)  
Michael Gurnis (Cal Tech)  
John Lyon (Dartmouth)  
John Dennis (UCAR)  
Bill Putman (NASA)  
Omar Ghattas (UT)  
Stephen Thomas (UCAR)  
Bill Skamarock (UCAR)  
Jeroen Tromp (Caltech)  
Kraig Winters (UCSD)  
Venkatramani Balaji (GFDL)  
John Linker (SAIC)  
Charles Goodrich (BU)  
Kraig Winters (UCSD)  
Yifeng Cui (SDSC)

## Appendix A.

# WRF Nature Run

John Michalakes, Josh Hacker, Richard Loft	Michael O. McCracken, Allan Snively, Nicholas J. Wright	Tom Spelce, Brent Gorda	Robert Walkup
{michalak, hacker, loft} @ucar.edu	{mmcrack, allans, nwright} @sdsc.edu	{spelce, bgorda} @llnl.gov	walkup@us.ibm.com
University Corporation for Atmospheric Research (UCAR), Boulder, CO.	Performance Modeling and Characterization Lab San Diego Supercomputer Center, La Jolla, CA.	Lawrence Livermore National Laboratory, Livermore, CA.	IBM Thomas J. Watson Research Center, Yorktown Heights, NY.

## Abstract

The Weather Research and Forecast (WRF) model is a model of the atmosphere for mesoscale research and operational numerical weather prediction (NWP). A petascale problem for WRF is a nature run that provides very high-resolution "truth" against which more coarse simulations or perturbation runs may be compared for purposes of studying predictability, stochastic parameterization, and fundamental dynamics. We carried out a nature run involving an idealized high resolution rotating fluid on the hemisphere, at a size and resolution never before attempted, and used it to investigate scales that span the  $k^{-3}$  to  $k^{-5/3}$  kinetic energy spectral transition, via simulations. We used up to 15,360 processors of the New York Blue IBM BG/L machine at Stony Brook University and Brookhaven National Laboratory. The grid we employed has 4486 by 4486 horizontal grid points and 101 vertical levels (2 billion cells) at 5km resolution; this is 32 times larger than the previously largest 63 million cell 2.5km resolution WRF CONUS benchmark [10]). To solve a problem of this size, we worked through issues of parallel I/O and scalability and employed more processors than have ever been used in a WRF run. We achieved a sustained 3.4 Tflop/s on the New York Blue system, inputting and then generating an enormous amount of data to produce a scientifically meaningful result. More than 200 GB of data was input to initialize the run, which then generated output datasets of 40 GB each simulated hour. The cost of output was considered a key component of our investigation. Then we ran the same problem on more than 12K processors of the XT4 system at NERSC and achieved 8.8 Tflop/s. Our primary result however is not just scalability and a high Tflop/s number, but capture of atmosphere features never before represented by simulation, and taking an important step towards understanding weather predictability at high resolution.

## Introduction

A fundamental challenge in numerical weather prediction (NWP) is to understand how (or even if) increasingly available computational power can improve weather modeling. An important enabling step towards improving that understanding is to perform a "nature run" to provide a very high-resolution standard against which more coarse simulations and parameter-sweeps may be compared for purposes of studying predictability, stochastic parameterization, and the underlying physical dynamics.

In this work we carry out a nature run at unprecedented computational scale on one of the world's largest supercomputers: we calculate an idealized high resolution rotating fluid on the earth's hemisphere to investigate scales that span the wavenumber ( $k$ )  $k^{-3}$  (largescale) to  $k^{-5/3}$  (smallscale) kinetic energy spectral transition of the observed atmosphere using up to 15K CPUS of the IBM BlueGene/L (New York Blue) at the New York Center for Computational Sciences (NYCCS) a cooperative effort of Stony Brook University and Brookhaven National Laboratory. Then we set a U.S. performance record of a weather code using the XT4 "Franklin" system at NERSC.

This calculation is neither embarrassingly parallel, nor completely floating-point dominated, but memory bandwidth limited in the computational core, and latency-bound with respect to interprocessor communication, and very I/O intensive. In these ways it is representative of many scientific calculations, and therefore achieving a high level of performance is challenging. Our primary result however is not just the high Tflop/s number, or record-

setting scalability of an atmosphere simulation, but an important step towards understanding weather predictability at high resolution<sup>2</sup>.

## Science Motivation

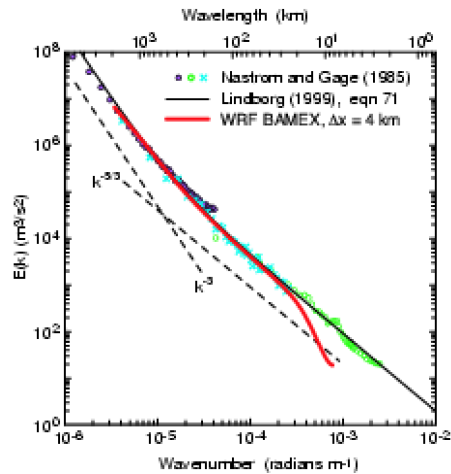
A nature run that includes planetary, synoptic (barotropic and baroclinic), and near-convective scales in the midlatitudes facilitates a new generation of basic research on predictability and turbulence. It is impossible to study predictability in the real atmosphere, making computer models necessary. The superiority of either increased resolution, or more probabilistic information, can only be established through basic predictability research. A nature run including the transition between the  $k^{-3}$  and  $k^{-5/3}$  spectral regimes facilitates a new generation of predictability studies that were not previously possible. For example, simple identical-twin experiments on how errors grow within the  $k^{-5/3}$  regime and across the transition can now be performed. The hypothesis of enhanced mesoscale predictability near topography with increased resolution of the model can now be rigorously addressed.

It is also difficult to study turbulence in the real atmosphere, and therefore models are attractive here as well. The turbulence community faces several challenges; wave-turbulence interactions occur within the  $k^{-5/3}$  regime and across the transition, for example in the jet-stream region of the atmosphere, but wave-wave interactions within the regime and across the transition are but poorly understood.

In the meantime, the growth of computational power is enabling numerical weather prediction model forecasts within the scale region defined by the observed  $k^{-5/3}$  scaling in the mesoscale. Yet we have much to learn about how waves and turbulence interact, better understanding of which will affect predictability and optimal sub-grid parameterization for predictive calculations within this region and across the observed transition to larger scales. Simply increasing the resolution of operational weather forecasts may not result in improved accuracy unless we can improve understanding of the physics and model parameterizations. The long-term goal of our project is therefore to produce a suite of nature runs, including runs at resolutions achievable only with petascale computing, that can serve as a basis for current and future predictability, turbulence, and parameterization study in a multi-scale environment that spans scales above and below the spectral transition. This work describes our achievement of a milestone in that project.

Previous work of Skamarock et al [2] showed that, with dedicated computer time on a large machine and using the Weather Research and Forecasting (WRF) model [1], high-resolution nature runs that can produce the appropriate  $k^{-5/3}$

spectral slope [3] are enabled. The WRF model includes a moist thermodynamic equation making it appropriate for precipitation processes. WRF is fully nonhydrostatic so it is appropriate for deep convection and gravity wave breaking. The numerics are stable enough to make additional damping terms, ubiquitous in typical mesoscale models, less necessary. Figure 1, reproduced from that study, encapsulates some of the evidence that the computational model is stable and of high verisimilitude.



**Figure 1 (courtesy W.C. Skamarock): Spectral energy density in the WRF model compared to observations. The red curve is spectra computed from WRF forecasts at 4 km grid spacing, averaged from 3 May 2003 to 14 July 2003. Both the transition of the spectral slope from  $k^{-3}$  to  $k^{-5/3}$ , and the numerical dissipation range are evident. Observations from Nastrom and Gage (1985) and Lindborg (1999) are shown with points and the solid black curve, respectively.**

Building on that study, the nature run done here contains instances of both stratified and unstratified turbulence, facilitating their study in a rotating fluid on a sphere and in the presence of many other scales. It further allows the study of gravity waves in a realistic environment, including gravity wave breaking. This level of fidelity is unprecedented, and open up new avenues of research to improve NWP.

## The Computational Approach

The WRF model [1] is a limited-area model of the atmosphere for mesoscale research and operational NWP. Developed and maintained as a community model, WRF is in widespread use over a range of applications including real-time NWP, tropical cyclone/hurricane research and prediction, regional climate, atmospheric chemistry and air quality, and basic atmospheric research. The WRF model represents the atmosphere as a number of variables of state discretized over regular Cartesian grids. The model solution

<sup>2</sup> This paper updates results presented in the Gordon Bell Prize track at SC07.

is computed using an explicit high-order Runge-Kutta time-split integration scheme in the two horizontal dimensions with an implicit solver in the vertical. WRF domains are decomposed over processors in the two horizontal dimensions only, and since the solver is explicit in the horizontal, interprocessor communication is logically nearest-neighbor. Each time-step involves 36 halo exchanges and a total of 144 two-way messages between neighboring processes. The decomposition is two-level: first over distributed memory patches and then again within each patch over shared memory tiles. Thus, WRF exploits hybrid parallel (message passing and multi-threaded) architectures such as BG/L, though the runs conducted here were MPI-only. Weather prediction codes are by nature I/O (mostly output) intensive, repeatedly writing out a time series of 3D representations of the atmosphere. WRF used Parallel NetCDF [8] as well as direct calls to MPI-IO for these runs.

## Key aspects of the BlueGene/L architecture for NWP

BG/L presents several opportunities and challenges for efficient implementation of NWP simulations. Details of the tightly integrated large-scale system architecture are covered elsewhere [4]. Overall, Stony Brook's BG/L platform has 18K compute nodes (36K CPUs). Here we briefly cover its general architectural aspects here, focusing on those related to our optimizations to WRF.

Each compute node is built from a single CPU ASIC and a set of memory chips. The compute ASIC features two 32-bit superscalar 700 MHz PowerPC 440 cores, with two copies of the PPC floating point unit associated with each core that function as a SIMD-like double FPU [5]. Each node has 512 MB of physical memory.

Achieving high performance requires that the application be fully domain-decomposable into data structures that can fit this relatively modest memory-per-node. If this can be accomplished, then the network support for scaling is an architectural strength of BG/L which has five networks; we focus on the 3-D torus, the broadcast/reduction tree and the global interrupt for WRF optimizations. Integration of the network registers into the compute ASIC not only provides fast inter-processor communication but also direct access to network-related hardware performance monitor data. Due to limitations on deadlock-free communication, the MPI implementation uses the tree networks only for global (full-partition) collective operations

## 3. Computational Method

As described in Skamarock et al [2] the continuous equations solved in WRF are the Euler equations cast in a flux (conservative) form where the vertical coordinate, denoted as  $\eta$ , is defined by a normalized hydrostatic pressure (or mass) following Laprise [6] as:

$$\eta = (p_h - p_{ht})/\mu \quad (1)$$

where  $\mu = p_{hs} - p_{ht}$  and  $p_h$  is the hydrostatic component of the pressure, and  $p_{hs}$  and  $p_{ht}$  are the values for the dry atmosphere at the surface and top boundaries, respectively. Following common practice we set  $p_{ht} = \text{constant}$ .  $\eta$  decreases monotonically from a value of 1 at the surface to 0 at the upper boundary of the model domain. Using this vertical coordinate, the flux form equations are expressed as

$$U_t + (\nabla \cdot V_u) + P_x(p, \phi) = F_U \quad (2)$$

$$V_t + (\nabla \cdot V_v) + P_y(p, \phi) = F_V \quad (3)$$

$$W_t + (\nabla \cdot V_w) + P_\eta(p, \mu) = F_W \quad (4)$$

$$\Theta_t + (\nabla \cdot V_\theta) = F_\Theta \quad (5)$$

$$\mu_t + (\nabla \cdot V) = 0 \quad (6)$$

$$\phi_t + \mu^{-1} [(\nabla \cdot \nabla \phi) - gW] = 0 \quad (7)$$

$$(Q_m)_t + (\nabla \cdot V Q_m) = F_Q \quad (8)$$

Where  $\mu(x, y)$  represents the mass of the dry air per unit area within the column in the model domain at  $(x, y)$ , hence the flux form variables are defined as  $U = \mu u/m$ ,  $V = \mu v/m$ ,  $W = \mu w/m$ ,  $\Omega = \mu \eta/m$ . And  $m$  is a map-scale factor that allows mapping of the equations to the sphere (see [7]) and is given as  $m = (\Delta x, \Delta y)$  distance on the earth

The velocities  $v = (u, v, w)$  are the physical velocities in the two horizontal and vertical directions, respectively,  $\omega = \eta$  is the transformed 'vertical' velocity, and  $\theta$  is the potential temperature.  $Q_m = \mu q_m$ ;  $Q_m = Q_v, Q_c, Q_i \dots$ , represent the mass of water vapor, cloud, rain, ice, etc., and  $q^*$  are their mixing ratios (mass per mass of dry air).

We also define non-conserved variables  $\phi = gz$  (the geopotential),  $p$  (pressure), and  $\alpha = 1/\rho$  (the specific volume) that appear in the governing equations. The  $P$ 's are pressure gradient terms.

## 4. Data Issues

The nature run is quite data intensive with a large sum memory footprint. During the New York Blue run we used a 2-billion cell, 4486 by 4486 grid with 101 levels. Horizontal resolution (width of an individual grid cell) was 5km; the time step was 6 seconds. Experimentally, the smallest possible run was 2048 processors requiring 287 MB/task for WRF data, not counting buffers, executable size, OS tax etc. Each output interval (1 simulation hour), the model generates 40 GB of data, the size of five 3D fields (three components of wind velocity, potential temperature, and mass) and two 2D fields. Normally, a much larger set of fields is output by WRF; however, this was reduced to the barest minimum to reduce cost. On the input side, the size of the dataset read initially to start the model contained 26 3D fields and was over 200 GB in size.

## 5. Porting and Tuning.

To achieve high performance with WRF on BG/L, the primary hurdles we overcame were 1) the size of main memory on BG/L and 2) the non-scalable I/O scheme in WRF.

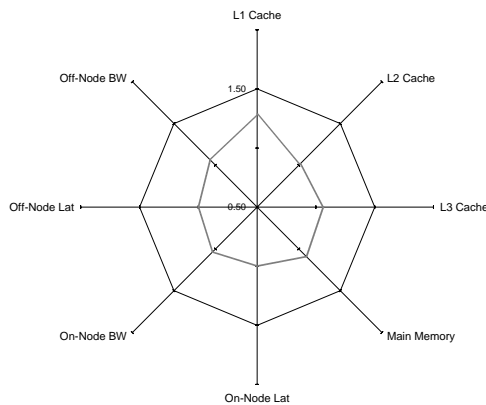


Figure 2 Performance-modeled response of WRF to doubling of processor attributes post-tuning.

Most data structures in WRF scale in memory. The domain decomposition and associated local memory extents used to dynamically allocate state arrays are calculated at run time on each process. However each processor used to keep the global set of boundary conditions. This had been sufficient up to modest numbers (several hundreds) of processors; but with very large grid sizes on thousands of processors, the memory for arrays that store lateral boundary conditions (LBCs) ballooned out using more memory than the rest of model state combined, quickly exceeding the 512 MB physical memory limit. The solution was to fully decompose all dimensions so that each processor only stores the LBCs used in its calculation. This also involved rewriting the code for performing I/O on LBCs. With this optimization, the full state required by each processor fits in memory even on the very large grid 4486 by 4486, 101 levels, on 15K processors.

The other scaling issue we addressed was also I/O related. WRF, like many parallel applications, historically used a single-reader/single-writer scheme for distributed I/O and thus required large, un-decomposed buffers to be stored on at least one process. Again this quickly exceeded the physical memory of one BG/L node. Support for MPI-IO

was added through parallel NetCDF and also direct calls to MPI-IO. Thereby we avoided the need to collect data on a single I/O task. We also encountered the 32-bit addressing limit on Blue Gene/L in the form of inability to define an MPI type large enough write a 3D field (8 GB) in one call to MPI-IO's write routine. Instead, 3D fields were written one level at a time.

Figure 2 shows a performance model developed for WRF using the PMaC methodology [11]. The figure confirms that post-tuning WRF is a well-balanced floating-point intensive code; it is most sensitive to L1 cache bandwidth (i.e. clock speed of processor) for fixed processor counts.

## 6. Performance Measurement and Results

Floating point operations were counted using the APC performance counter library on BG/L. This library accesses the compute node ASIC's hardware performance counters to tracks several events including FPU, some SIMD and load and store operations. Because WRF uses single (32-bit) floating point precision, it was not possible to fully exploit the BG/L "double-hammer" SIMD capability.

Using 15K CPUs of BlueGene/L at Stony Brook we achieved 3.39 Tflop/s and set a parallelism record for number of processors running a weather code. The model was run on 6,144 and 15,360 nodes of NY Blue in co-processor (CO) mode. Only one of the two processors on each node was in use, an advantage for large-memory, high memory-bandwidth applications such as weather models.

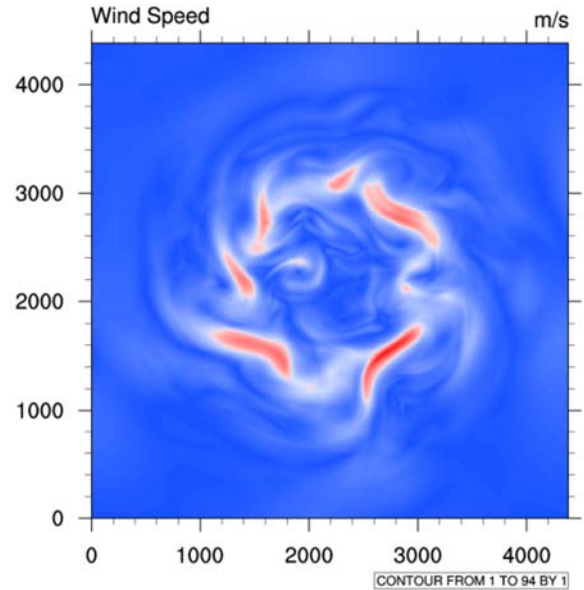
First considering first floating point rate by itself, the BG/L system delivered 1.49 Tflop/s on 6K processors, and 3.39 Tflop/s on 15K. Thus, scaling relative to the 6K rate was better than 90 percent efficient. Meanwhile, the output bandwidth on the BG/L, rather than degrading with increasing numbers of writers, was seen to scale as well: 242 MB/s on 6K processors and 286 MB/s on 15K. The result was good overall performance, even when the cost of writing 40 GB every simulation hour was considered: WRF still achieves 1.44 Tflop/s on 6K processors (3 percent penalty) and 3.17 Tflop/s on 15K processors (6 percent penalty), even with the cost of model output.

Next, using the tuned code, we set a speed performance record for a U.S. weather model running on the Cray XT4 "Franklin" supercomputer at the Department of Energy's National Energy Research Computing Center (NERSC) at Lawrence Berkeley National Laboratory. Running on 12,090 processors of this 100 peak teraflops system, we achieved the important milestone of 8.8 teraflops – the fastest performance of a weather or climate-related application on a U.S. supercomputer.

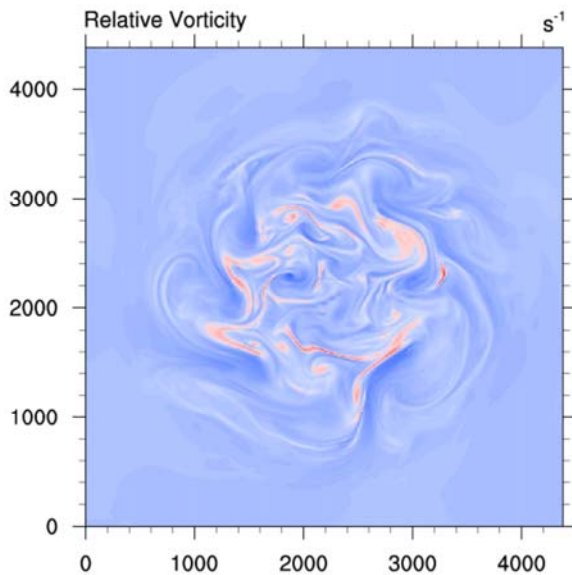
Initial science results are indicated in Figure 3, and demonstrate that the proposed suite of nature runs are feasible. In both panels the mid-latitude wave train is

obvious, encircling the hemisphere. The top panel shows relative vorticity at approximately 5 km above ground level (model level 30) after 3 hours of simulation (model/earth time). In this figure, reds are large positive (cyclonic) vorticity and dark blues are large negative (anticyclonic) vorticity. One thing to note is the multiple scales of flow present, indicative of high resolution, which are not typically seen in simulations of this spatial extent (e.g. global). The 5-km grid spacing ( $\Delta X$ ) in this simulation easily resolves the spectral transition, which will typically reside at scales on the order of  $20\Delta X$ . Gravity waves related to baroclinic development may be present, suggested by the narrow filaments paralleling the streaks of greatest magnitude. More investigation is required to determine whether they are gravity waves, and then to understand whether they are contributing to the spectral transition from  $k^{-3}$  to  $k^{-5/3}$ .

The bottom Figure 3 shows wind speed in m/s at approximately jet-stream level (10 km, model level 60), with red indicating maximum wind speeds in jet streaks. To reach the long-term scientific objectives of this project, this wave train, the associated baroclinic development, and near-convective scales where gravity waves may break, need to be simulated together. This is only possible at high resolution on domains that cover a hemisphere or the globe, so that many instances of waves, turbulence, and their interactions on both sides of the spectral transition are present. We have thus succeeded in opening new lines of investigation, and the path is now open for the next steps.



**Figure 3: Relative vorticity at approximately 5 km above ground (top), and wind speed near the jet stream at approximately 10 km above ground. The entire midlatitude wave train is simulated at high resolution ( $\Delta X=5$  km), and the solutions show the range of scales in the flow.**



## Conclusion

Figure 2 above shows example results from the nature run. We set records for performance and scalability of WRF or any other atmospheric simulation. But our primary achievement is not performance or scalability, rather new science to enable improved numerical weather prediction.

We carried out a WRF nature run that provides very high-resolution "truth" against which more coarse simulations or perturbation runs may be compared for purposes of studying predictability, stochastic parameterization, and fundamental dynamics. We studied idealized high resolution rotating fluid on the hemisphere to investigate scales that span the  $k^{-3}$  to  $k^{-5/3}$  kinetic energy spectral transition of the observed atmosphere using 15K CPUs of BG/L with achieved 3.4 Tflops and thereby opened up new avenues of science investigations via simulation.

## Acknowledgements.

We wish to thank William Skamarock, whose equations and description of WRF's numerical

formulation is reprinted with permission in Section 3. This work was sponsored in part by the National Science Foundation via GEO ATM SGER award #0637994 “Feasibility of Taking the Weather Research and Forecasting (WRF) Model to Petascale”, in part by the OCI award “The Cyberinfrastructure Evaluation Center” and in part by DOE Office of Science through the SciDAC2 award entitled Performance Engineering Research Institute (PERI). We also wish to thank Andrew Vogelmann, Brian Colle, and Efstratios Efstathiadis of the New York Center for Computational Sciences at Stony Brook University and Brookhaven National Laboratory. This work was supported in part by NSF Award #0621611 Workshop: Petascale Computing in the Geosciences.

## References

- [1] W. Skamarock, J. Klemp, J. Dudhia, D. Gill, D. Barker, W. Wang, J. Powers, “A Description of the Advanced Research WRF Version 2”, NCAR Technical Note, 2005.
- [2] W. Skamarock, et al. “A Time-Split Nonhydrostatic Atmospheric Model for Weather Research and Forecasting Applications”, *Journal of Computational Physics*. January 2007.
- [3] W. Skamarock, “Evaluating Mesoscale NWP Models Using Kinetic Energy Spectra”. *Monthly Weather Review*, November, 2004.
- [4] N. R. Adiga et al., “An overview of the BlueGene/L supercomputer” SC2002 – High Performance Networking and Computing, 2002.
- [5] E. L. Bachega, S. Chatterjee, K. Dockser, J. Gunnels, M. Gupta, F. Gustavson, C. Lapkowski, G. Liu, M. Mendell, C. Wait, T.J.C. Ward, “A High-Performance SIMD Floating Point Unit Design for BlueGene/L: Architecture, Compilation, and Algorithm Design” PACT, 2004.
- [6] R. Laprise, “The Euler Equations of Motion with Hydrostatic Pressure as an Independent Variable”. *Mon. Wea. Rev.*, 120, (1992), 197-207.
- [7] G. J. Haltiner, and R. T. Williams, *Numerical Weather Prediction and Dynamics Meteorology*. (2nd edition, John Wiley and Sons, 1980), Inc. 477 pp.
- [8] Parallel NetCDF, see
- [9] <http://www-unix.mcs.anl.gov/parallel-netc>
- [10] <http://www.mmm.ucar.edu/wrf/WG2/bench>
- [11] <http://www.sdsc.edu/pmac>

## **Appendix B**

### **High-Frequency Simulations of Global Seismic Wave Propagation Using SPEC-FEM3D\_GLOBE on 62K Processors**

Laura Carrington <sup>a</sup>, Dimitri Komatitsch <sup>b,c</sup>, Michael Laurenzano <sup>a</sup>, Mustafa Tikir <sup>a</sup>,  
David Michéa <sup>b</sup>, Nicolas Le Goff <sup>b</sup>, Allan Snaveley <sup>a</sup>, Jeroen Tromp <sup>d</sup>

<sup>a</sup> Performance Modeling and Characterization Lab, San Diego Supercomputer Center,  
La Jolla, CA, USA

<sup>b</sup> Université de Pau, CNRS and INRIA Sud-Ouest Magique-3D, Laboratoire de  
Modélisation et d'Imagerie en Géosciences, Pau, France

<sup>c</sup> Institut universitaire de France, Paris, France

<sup>d</sup> Seismological Laboratory, California Institute of Technology, Pasadena, CA, USA

#### **Abstract**

SPEC-FEM3D\_GLOBE is a spectral-element application enabling the simulation of global seismic wave propagation in 3D anelastic, anisotropic, rotating and self-gravitating Earth models at unprecedented resolution. A fundamental challenge in global seismology is to model the propagation of waves with periods between 1 and 2 seconds, the highest frequency signals that can propagate clear across the Earth. These waves help reveal the 3D structure of the Earth's deep interior and can be compared to seismographic recordings. We performed a 3D simulation reaching a shortest period of 3 seconds, setting a new record, using 12K processors of XT4 Franklin at NERSC. The final paper will be seminal; it will easily break the 2 second barrier using 62K processors of Ranger at TACC, and possibly be the first work to meet the ultimate goal in numerical global seismology of a 1 second shortest period. There will be no need to pursue smaller periods, because higher frequency signals do not propagate across the entire globe.

We employed performance modeling methods to identify performance bottlenecks and worked through issues of parallel I/O and scalability. Improved mesh design and numbering results in excellent load balancing and few cache misses. The primary achievements are not just the scalability and high teraflops number, but a historic step towards understanding the physics and chemistry of the Earth's interior at unprecedented resolution.

#### **Introduction**

The calculation of accurate synthetic seismograms for 3D global Earth models poses a significant computational challenge, both in terms of the demands on the numerical algorithm and with regards to computer hardware (i.e., memory and CPU requirements). Global seismologists routinely analyze recorded seismic signals with period as short as 1 second. Previous large-scale simulations in 3D Earth models have only been capable of reaching 3.5 seconds [11]. Therefore, our objective is to simulate global seismic wave propagation at periods of 1 to 2 seconds, the highest frequency signals that can propagate clear across the Earth. These waves will help reveal the detailed 3D structure of the Earth's deep interior, in particular near the core-mantle boundary (CMB), the inner core boundary (ICB), and in the enigmatic inner core. The CMB region is highly

heterogeneous with evidence for ultra-low velocity zones, anisotropy, small-scale topography, and a recently discovered post-perovskite phase transition. The Earth's inner core appears to be anisotropic, with dramatic differences between its eastern and western hemispheres, and there are suggestions that it rotates at a slightly different rate than the Earth's mantle. Being able to simulate 3D global seismic wave propagation at these frequencies will help us understand and image these complex structures, an endeavor that will enhance our understanding of the physics and chemistry of the Earth's interior. The SPEC-FEM3D\_GLOBE package has been designed to compute these simulations.

Since the record-breaking 3.5 second frequency run of 2003 which used the Earth Simulator [11], the team has expended a major R&D effort towards breaking the 2 second barrier. Achieving this goal required radical algorithmic changes to SPEC-FEM3D enabling peta-scalability (beyond 10Ks of processors) and incorporation of new algorithms that are both more scientifically accurate and more computationally scalable than ever. Recent algorithm and tuning work is described in Section 4, previous such work involved optimizations to reduce cache misses from 75% to 30%, new mesh designs to nearly eliminate load balancing and improve spatial resolution for the wave, and improvements to the inner Earth core resolution based upon an “inflated” central cube instead of a real cube with flat faces [7]; reduction of the “central cube” bottleneck by cutting the cube in two, reduction of MPI messages by 33% inside each chunk by handling crust mantle and inner core simultaneously, and finally non-iterative coupling between fluid and solid based on the displacement vector [4] instead of velocity as in previous versions of the application. In addition to these enhancements and optimizations, the model has been improved to include more complex Earth models and the capacity to compute sensitivity kernels for inverse problems in addition to forward problems [13]. Thus with the advanced domain science and computer science incorporated in SPEC-FEM3D it amounts to practically a new code and we are ready to break the 2 second barrier using it.

The paper is laid out as follows: in Section 2, a description of the spectral-element method used to solve the seismic wave propagation problem is given. Section 3 briefly describes the current usage for the SPEC-FEM3D application and challenges in moving to shorter seismic periods. In Section 4, we describe the challenges associated with running at large scales (e.g. >10K+ cores), plus the performance analysis, code modifications, and tuning required to address those challenges. Section 5 presents the current results, and section 6 illustrates work to be completed to achieve the ground breaking simulations of global seismic wave propagation at wave periods of 1 to 2 per second for the final submission.

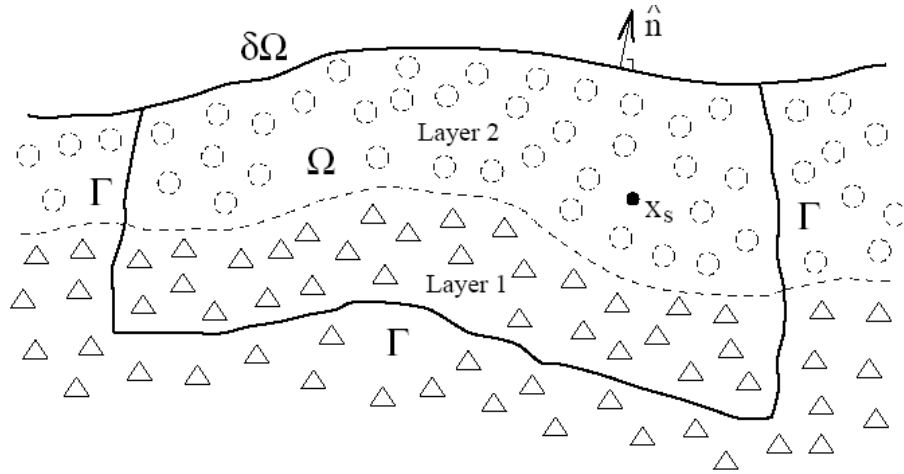
## **Description of the method**

To simulate global seismic wave propagation in 3D anelastic, anisotropic, rotating and self-gravitating Earth models we have developed and implemented a spectral-element method (SEM). The SEM has been introduced more than twenty years ago in computational fluid dynamics [14]. It has gained interest for problems related to 2-D [5, 15] and 3-D [8, 9, 12] seismic wave propagation (for instance following a large earthquake). The method easily incorporates free surface topography and accurately

represents the propagation of surface waves, and lends itself well to parallel computation with distributed memory[6, 11].

## Equations of motion

We seek to determine the displacement field produced by an earthquake in a finite Earth model, as shown in Figure 1. The equations of motion that govern the propagation of seismic waves in the Earth may be solved based upon either a strong or a weak formulation of the problem. In the strong formulation one works directly with the equations of motion and associated boundary conditions written in differential form; this approach is used, for instance, in finite-difference or global-pseudo spectral modeling techniques. In the weak formulation one uses an integral form of the equations of motion, as in finite-element (FEM) and direct solution methods. The SEM is based upon a weak formulation of the equations of motion.



**Figure 1. Finite Earth model with volume  $\Omega$  and free surface  $\delta\Omega$ . An artificial absorbing boundary  $\Gamma$  is introduced if the physical model is not of finite size, and  $\hat{n}$  denotes the unit outward normal to all boundaries. The model can be fully heterogeneous or composed of any number of layers.**

## Strong form

The displacement field  $s$  produced by an earthquake is governed by the momentum equation

$$\rho \partial_t^2 s = \nabla \cdot T + f \quad (1)$$

The distribution of density is denoted by  $\rho$ . The stress tensor  $T$  is linearly related to the displacement gradient  $\nabla s$  by Hooke's law, which in an elastic, anisotropic solid may be written in the form

$$T = c : \nabla s \quad (2)$$

The elastic properties of the Earth model are determined by the fourth-order elastic tensor  $c$ , which has 21 independent components in the case of general anisotropy.

The earthquake source is represented by the point force  $f$ , which may be written in terms of a moment tensor  $M$

$$f = -M \cdot \nabla \delta(x - x_0) S(t) \quad (3)$$

The location of the point source is denoted by  $x_s$ ,  $\delta(x - x_0)$  denotes the Dirac delta distribution located at  $x_s$ , and the source-time function is given by  $S(t)$ .

The momentum equation (1) must be solved subject to a stress-free boundary condition at the Earth's surface  $\partial\Omega$  :

$$T \cdot \hat{n} = 0 \quad (4)$$

### Weak form

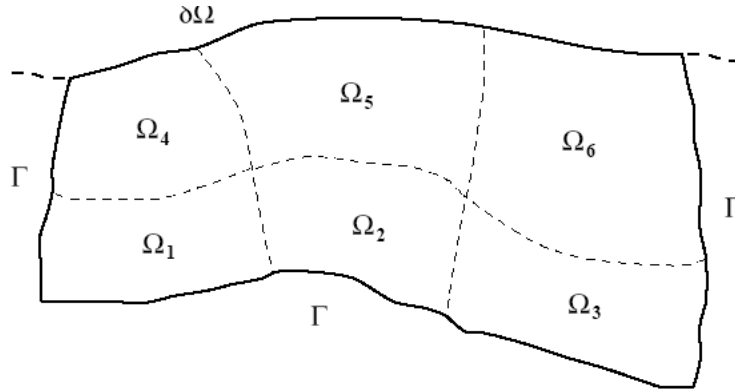
Rather than using the equations of motion and associated boundary conditions directly, one can use an integrated form. This is accomplished by dotting the momentum equation (1) with an arbitrary vector  $w$ , integrating by parts over the model volume  $\Omega$ , and imposing the stress-free boundary condition (4). This gives

$$\int_{\Omega} \rho w \cdot \partial_t^2 s d^3x = - \int_{\Omega} \nabla w : T d^3x + M : \nabla w(x_0) S(t) \quad (5)$$

where the stress tensor  $T$  is determined in terms of the displacement gradient  $\nabla s$  by Hooke's law (2). The source term has  $\int_{\Omega} f \cdot w d^3x$  been explicitly integrated using the properties of the Dirac delta distribution.

### Definition of the mesh

As in a classical FEM, the model volume  $\Omega$  is subdivided into a number of non-overlapping elements  $\Omega_e$ ,  $e = 1, \dots, n_e$ , as shown in Figure 2. Each hexahedral volume element  $\Omega_e$  is mapped to a reference cube. The mapping is defined by the so-called classical Jacobian matrix. Points within this reference cube are denoted by the vector  $\xi = (\xi, \eta, \zeta)$ , where  $-1 \leq \xi \leq 1$ ,  $-1 \leq \eta \leq 1$  and  $-1 \leq \zeta \leq 1$ .



**Figure 2.** For the purpose of computations, the Earth model  $\Omega$  shown in Figure 1 is subdivided into curved hexahedra whose shape is adapted to the edges of the model  $\partial\Omega$  and  $\Gamma$  and to the main geological interfaces.

### ***Representation of functions and numerical integration on the elements***

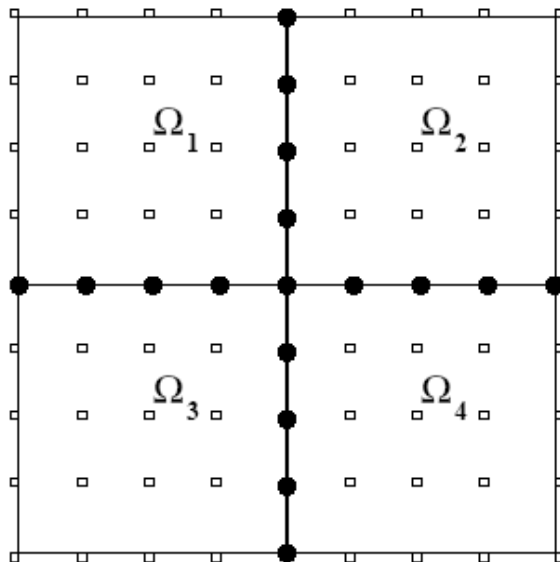
To solve the weak form of the equations of motion (5), integrations over the volume  $\Omega$  are subdivided in terms of smaller integrals over the volume elements  $\Omega_e$ . A high-degree Lagrange interpolant is used to express functions on the elements. The control points needed in the definition of the Lagrange polynomials of degree  $n_\ell$  are chosen to be the classical  $n_\ell + 1$  so-called Gauss-Lobatto-Legendre (GLL) quadrature points. Note that they always include  $+1$  and  $-1$ ; therefore in a SEM some points always lie exactly on the boundaries of the elements.

Any smooth function  $f$  can then be interpolated in a 3D hexahedral element by triple products of Lagrange poly-nomials of degree  $n_\ell$  at these GLL points. In a SEM for wave propagation problems one typically uses a polynomial degree  $n_\ell$  between 4 and 10 to represent a function on the element [8]. The derivative of function  $f$  can then be computed by computing the derivative of the Lagrange polynomials. And numerical integration of this function over volume elements  $\Omega_e$  may be approximated using the Gauss-Lobatto-Legendre integration rule, whose weights can be computed numerically and stored once and for all [3].

### ***Assembling and marching the global system in time***

In the SEM mesh, grid points that lie on the sides, edges, or corners of an element are shared amongst neighboring elements, as illustrated in Figure 3. Therefore, the need arises to distinguish between the grid points that define an element, the *local mesh*, and all the grid points in the model, many of which are shared amongst several spectral elements, the *global mesh*. One needs to determine a mapping between grid points in the local mesh and grid points in the global mesh; efficient routines are available for this purpose from finite-element modeling. Before the system can be marched forward in time, the contributions from all the elements that share a common global grid point need

to be summed. In a traditional FEM this is referred to as the *assembly* of the system. Computationally, this assembly stage is a costly part of the calculation on parallel computers, because information from individual elements needs to be shared with neighboring elements, an operation that involves communication between distinct CPUs (based on message passing with MPI in our case, see for instance Komatitsch et al. [11]).



**Figure 3. Illustration of the local and global meshes for a four-element 2-D spectral-element discretization with polynomial degree  $N = 4$ . Each spectral element contains  $(N + 1)^2 = 25$  Gauss-Lobatto-Legendre points, that constitute the local mesh for each element. These points are non-evenly spaced, but have been drawn evenly spaced here for simplicity. In the global mesh, points lying on edges or corners (as well as on faces in 3-D) are shared between elements. The contributions to the global system of degrees of freedom, computed separately on each element, have to be summed at these common points represented by black dots. Exactly two elements share points inside an edge in 2-D, while corners can be shared by any number of elements depending on the topology of the mesh, which can be non-structured.**

Let  $U$  denote the displacement vector of the global system, i.e.,  $U$  contains the displacement vector at all the grid points in the global mesh, classically referred to as the global degrees of freedom of the system. The ordinary differential equation that governs the time dependence of the global system may be written in the form

$$M\ddot{U} + KU = F, \quad (6)$$

where  $M$  denotes the global mass matrix,  $K$  the global stiffness matrix, and  $F$  the source term. Explicit expressions for the local contributions to the mass and stiffness matrices and further details on the construction of the global mass and stiffness matrices from their elemental expression may be found for instance in [8],[9].

A highly desirable property of a SEM, which allows for a very significant reduction in the complexity and cost of the algorithm, is the fact that the mass matrix  $M$  is diagonal by construction. Therefore, no costly linear system resolution algorithm is needed to march the system in time.

Time discretization of the second-order ordinary differential equation (6) is achieved based upon a classical explicit second-order finite-difference scheme, which is conditionally stable (i.e., the time step has an upper limit above which the simulation becomes unstable).

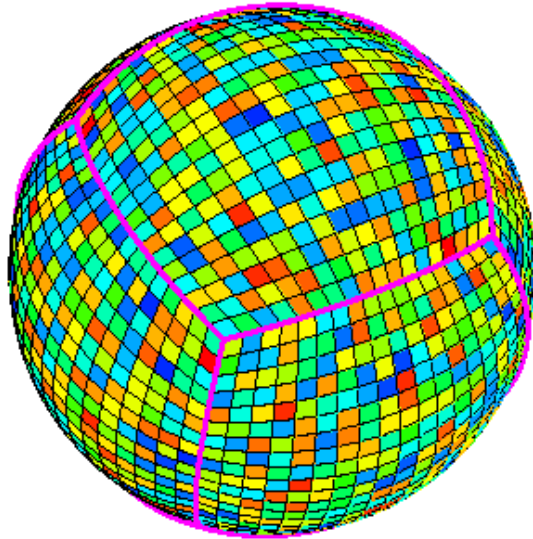
The above numerical techniques are incorporated into our software package called SPECFEM3D\_GLOBE.

## **The SPECFEM3D GLOBE package**

The SPECFEM3D\_GLOBE package was designed to simulate three-dimensional global and regional seismic wave propagation based upon the SEM to solve the equations described in Section 2. The package is maintained on a source code release server at Computational Infrastructure for Geodynamics (CIG) [1], and is being actively developed by a core group of approximately 15 scientists. The package has been extensively benchmarked against semi-analytical normal-mode synthetic seismograms (i.e., curves showing the evolution of displacement with time after the earthquake at a given mesh point) for spherically-symmetric Earth models [9][10]. These benchmarks are very challenging because they involve solid-fluid domain decomposition and coupling, attenuation, anisotropy, self-gravitation, and the effect of the ocean layer located at the surface of the Earth. Our simulations incorporate effects due to topography and bathymetry as well as fluid-solid boundaries, such as the ocean floor, the core-mantle boundary (CMB), and the inner-core boundary (ICB). Thus far, only SPECFEM3D\_GLOBE has been capable of accurately incorporating all of these effects.

The current stable SPECFEM3D\_GLOBE version 4.0 consists of two major programs: MESHFEM3D, the mesher, which generates the spectral-element mesh and SPECFEM3D, the solver, which uses the generated mesh to run the simulation.

The mesher is designed to generate a spectral-element mesh for either regional or entire globe simulations. This work focuses on simulations of the entire globe, which are the most expensive and therefore by far the most challenging. These simulations use a spectral-element mesh which is based upon an analytical mapping from the cube to the sphere called the ‘gnomonic mapping’ or the ‘cubed sphere’ ( see e.g. [17],[16]), which splits the globe into 6 chunks, each of which is further subdivided into  $n^2$  mesh slices for a total of  $6 \times n^2$  slices, as shown in Figure 4. The work for the mesher code is distributed to a parallel system by distributing the slices.



**Figure 4. The cubed-sphere mapping of the globe represents a mesh of  $6 \times 18^2 = 1944$  slices.**

Once the mesh is generated it communicates this information to the solver by writing the mesh to files labeled with the appropriate MPI rank (and usually stored on local disks on each node to avoid an I/O bottleneck). Once the mesher is complete the solver can be launched on the same set of processor cores in the same order and read in the mesh from the mesher files. It is this I/O interaction between the mesher and the solver that creates challenges for some large scale HPC systems. Performance models can help detect these issues and identify solutions to them.

## **Overcoming large-scale challenges**

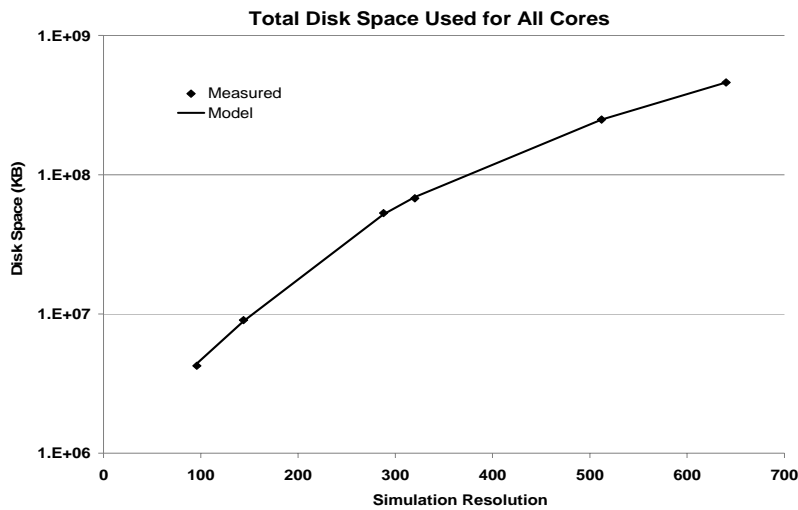
To meet our objective to simulate global seismic wave propagation down to seismic wave period of 1 second (i.e., up to maximum seismic frequencies of 1 Hz) the mesher and solver would each require at least 37 TBs of data. This would require around 62K cores of an HPC system having  $\sim 1.0$ GB of memory per core available to the running application. Running any application at this scale can create bottlenecks and challenges not seen at smaller scale.

There were three separate efforts working on the SPEC3D\_GLOBE package to enable efficient runs at large scale. The first was to remove the I/O bottleneck created between the mesher and solver. The second was to make sure the mesh layout was optimal. The third effort was to do some single processor optimization on the computation routine that dominates the runtime.

### ***Removing the I/O bottleneck***

The original mode of running the SPEC3D code was to first run the MESH3D code which generates the mesh and writes it to local disks. The SPEC3D code then runs immediately after this and reads in those local disk files.

For a system with good local disks this method of running can be quite efficient. But many newly installed larger systems use diskless nodes (to decrease power consumption and to increase node stability by getting rid of mechanical parts). This means that each of these mesher files would then have to be written to a globally mounted file system, creating a large bottleneck for both the mesher and the solver due to I/O contention. The original (current stable) version of the code (version 4.0) writes and reads up to 51 files per core. At around 62K cores, this corresponds to over 3.2 million files that would have to be written and then subsequently read. Furthermore, the amount of data transferred between the two parts of the application will become a factor for a large-scale simulation. Figure 5 shows a simple regression model of the disk space used for a series of resolutions along with the actual disk usage. This model predicts that in order to obtain a 2 second period in the simulation, over 14 TB of data would have to be transferred between the mesher and the solver and to obtain a 1 second period in the simulation, over 108 TB of data transfer is required.



**Figure 5. Total disk space used for communication between MESHFEM3D and SPECFEM3D in the initial stable version of the package. Resolution =  $256 \times 17$  / Wave Period. (Higher resolution is higher frequency).**

Much of this contention can be removed by merging the mesher and solver into a single application and allowing the mesher and solver to communicate via shared memory rather than with I/O. Merging the codes is technically uneasy because it creates challenges in memory management and bookkeeping across the two originally separate applications, and is currently only partially complete. In the current code we have alleviated the need for the write and subsequent read of 39 files per core. For our tests with larger processor counts and resolutions, this has reduced the amount of disk space used by at least two orders of magnitude and has reduced the amount of time spent in I/O by at least a factor of 4. Our intention is to completely remove the use of I/O to communicate between the two parts of the application, eliminating the need to use any disk space for intermediate files along with the associated I/O penalties of using these files.

While the merged code showed promising results by initially removing the I/O bottleneck it created an additional challenge by using more memory (because in the merged version some of the arrays from the mesher and from the solver must be present in memory simultaneously. This is problematic because the more memory per core required, the more cores we will need to meet the goal of simulations accurate down to seismic periods of 1 to 2 seconds.

To reduce this memory usage, optimization efforts are underway to lower the memory high water mark of the merged application. The initial effort for this was to change the arrays in the solver to only be allocated dynamically after the mesher completed rather than being statically allocated at runtime. This decreased the memory usage of the entire application by ~19% for lower resolution runs. Further investigation into memory usage of the application indicates that the memory high water mark should be lower but due to fragmentation caused by allocation and deallocation by the mesher it is higher than initial memory models indicate. Efforts to reduce this fragmentation are currently underway and will be complete for the final submission if accepted. While it is hard to accurately estimate the final effect, it should improve by at least another 10% and possibly much more.

### ***Multilevel Cuthill-McKee sorting***

In the SEM algorithm, one spends a lot of time looping on all the elements (the so-called spectral elements) of the 3D mesh and computing local contributions (local forces and resulting acceleration vectors) at all the internal grid points of each element. Contributions computed at element faces, edges or corners shared between two or more elements are then summed. Therefore in principle (i.e. mathematically) one can loop on the elements in any order and get the same final result because of the associativity and commutativity of the sum operator. (Note that formally this ceases to be true on a computer because of different roundoff depending on the order in which the sub-sums are performed, but in practice only the last one or two decimals are affected and therefore one can still choose any order, and the result is “almost” invariant by permutation down to the last digits). We have checked this experimentally: the same mesh computed with different loop orders on the elements give two sets of synthetic seismograms that are indistinguishable when plotted superimposed.

However, processors have caches and therefore it is important to try to maximize cache reuse and also maximize the effect of prefetching by trying to loop on the neighbors of an element first once the calculations in that element are finished; this way we will increase the probability for common faces, edges or corners to already be in the cache.

To increase spatial and temporal locality for the global access of the points that are common to several elements, the order in which we access the elements can then be optimized. The goal is to find an order that minimizes the memory strides for the global arrays. We use the classical reverse Cuthill-McKee [17] algorithm, which consists of renumbering the vertices of a graph to reduce the bandwidth of its adjacency matrix. Sorting the elements with the Cuthill-McKee algorithm before renumbering the global index table also increases the spatial and temporal locality: spatial locality, because the common points of the connected elements will be stored statistically closer in memory;

temporal locality, because these common points will be re-accessed sooner. We have designed an improved version of that algorithm in which we use multi-level sorting to define groups of typically 50 to 100 elements which all fit together in the L2 cache. Tests performed with SPECFEM3D\_GLOBE on the same mesh with and without sorting show that unfortunately we do not gain much based on sorting: at most 5% in practice. But this is probably in fact good news: it means that previous work we performed to reduce cache misses [7] has worked very well and there are already so few L2 cache misses that it is difficult to further reduce them. An additional explanation is the fact that in the SEM we perform a lot of local operations in each element therefore in percentage the time it takes to move new data in the L2 cache is not crucial compared to the total time it takes to perform the calculations in that element.

### ***Manual use of SSE instructions***

The initial performance model for the SPECFEM3D\_GLOBE application indicates that a large fraction of time (greater than 70%) is spent in two computational routines in which we compute the internal forces and related acceleration vectors in each spectral element of the mesh in two regions of the Earth: the large solid mantle and crust, and the smaller fluid outer core. Inside these two routines, which have a very similar structure, we perform small matrix-matrix products (each matrix has a size of  $5 \times 5$  typically) along cutplanes of 3D arrays (first cut along the  $i$  axis, then cut along the  $j$  axis, and then cut along the  $k$  axis).

It is therefore important to study how we can optimize this crucial section of the two routines. When talking about matrix-matrix products, one immediately thinks about calling a vendor-optimized implementation of the Basic Linear Algebra Subprograms (BLAS-3) subroutine SGEMM, but in our case this turns out to be a poor idea for two reasons. First, the matrices are very small ( $5 \times 5$ ) and therefore the overhead of the BLAS routine is higher than what we can hope to gain. Second, because we have to handle cutplanes along three different directions of a 3D memory block, several of these calls to BLAS would be for blocks not linearly aligned in memory and would therefore first require a memory copy to an aligned 2D block, before the call; this would be more expensive than any potential gain from the BLAS routine.

Tests that we have performed have confirmed that using BLAS calls actually significantly slows down the code compared to our existing regular Fortran loops. We therefore tried another option, which is to use vector instructions provided by a SSE unit (for instance on Intel or AMD processors) or an AltiVec/VMX unit (for instance on IBM PowerPC processors). These units can handle four single-precision floating-point operations in a vector and are very well suited for our small matrix products since we can load a vector unit with 4 floats, perform several "multiply and add" (MADD) operations to compute the matrix-matrix product, and store the results in four consecutive elements of the result matrix (Note that MADD does not exist explicitly in SSE but is rather implemented as a combination of "multiply" and then "add").

These three types of operations (load, MADD and store) are standard in both SSE and AltiVec. Note that, since our matrices are of size  $5 \times 5$  and not  $4 \times 4$ , we use vector instructions for 4 out of each set of 5 values and compute the last one serially in regular

Fortran. Also note that to improve performance we align our 3D blocks of  $5 \times 5 \times 5 = 125$  floats on 128 in memory using padding with three dummy values set to zero. This induces a negligible waste of memory of  $128 / 125 = 2.4\%$ .

Results show that we typically gain between 15% and 20% (with respect to the stable version 4.0 of our code) both with SSE on AMD processors and with AltiVec on another machine equipped with IBM PowerPC970 processors. The relative gain is limited by two factors where first is the limited number of vector registers present in the hardware, which is 16 for SSE and 32 for AltiVec and second is the fact that modern compilers can automatically unroll loops and generate SSE or AltiVec instructions to perform something similar to what we implement manually; therefore the reference time may already include some of the effects of using SSE instructions.

## Performance measurements and models

To meet our objective to simulate global seismic wave propagation down to seismic wave period of 1 second we will need to run on around 62K cores. We have used two different systems to investigate how to reach this goal.

The first is Texas Advanced Computing Center (TACC) Sun Constellation Linux cluster, named Ranger, which has 62,976 processing cores connected with a full-CLOS InfiniBand interconnect. Each compute node in Ranger consists of four 2.0 GHz quad-core AMD Opteron processors with a theoretical peak performance of 32 GFlops and 8 GBytes of memory. The theoretical peak performance of Ranger is thus about 504 TFlops (its Rmax is not known at time of submission).

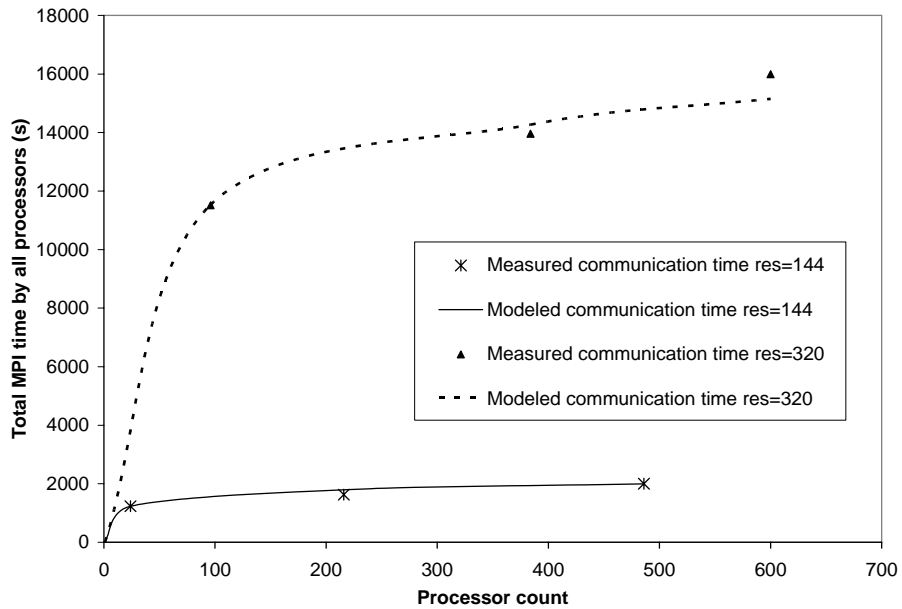
The second is National Energy Research Scientific Computing Center (NERSC) Cray XT4 system, named Franklin. Each of its compute nodes consists of a 2.6 GHz dual-core AMD Opteron processor with a theoretical peak performance of 10.4 GFlops and 4 GBytes of memory. The theoretical peak performance of Franklin is thus about 101.5 TFlops, its measured Rmax is 85 TFlops.

The initial step was to model the communication behavior of SPECFEM3D. To accomplish this we ran several experiments varying the input resolution and the number of processors. In SPECFEM3D, resolution can be changed based on an input parameter called NEX\_XI, which defines the number of elements at the surface along the two sides of each of the six chunks, whereas the number of processor cores can be changed based on an input parameter called NPROC\_XI, which defines the number of MPI processor cores to be used along the two sides of each of the six chunks. For our initial investigation, we varied the processor count from 24 to 1536 and the mesh resolution from 96 to 640 (which corresponds to minimum seismic periods from 45.3 seconds to 6.8 seconds, respectively).

We measured the communication time for each run with IPM (Integrated Performance Monitoring) tool[2], which is a portable profiling tool that provides a performance summary of the computations and communications in a parallel program. IPM has extremely low overhead and is scalable to thousands of processors, which makes it ideal for this purpose.

We measured the total communication time spent in the main loop of the solver component for each run. We ran these experiments on Franklin. Even though we used only Franklin for our modeling runs, we expect similar behavior on other balanced systems for SPECfEM3D. The results showed that the communication time spent in the main loop of the solver component ranges from 1.9% to 4.2% (with an average of 3.2%) of the overall execution time for the runs. More importantly, the lower communication percentages indicate that SPECfEM3D\_GLOBE is dominated by the computation time and is a good candidate to scale up to tens of thousands of processors before the communication time becomes a bottleneck.

The results of modeling runs also showed that the total communication time spent for all processors tends to increase both when the resolution increases and when the number of processors increases. However, it also shows that for a given resolution, the communication time per core decreases as the number of processor increases. Using these observations and measured overall communication time for all processors, we fitted a function to the actual measured communication times for a given resolution. Figure 6 presents the measured and modeled total communication times for all cores for two resolutions. Other resolutions were fitted with similar results. Based on the fitted models for all resolutions used in our modeling runs, we were also able to model the increase in overall communication time for all cores as the resolution increases.

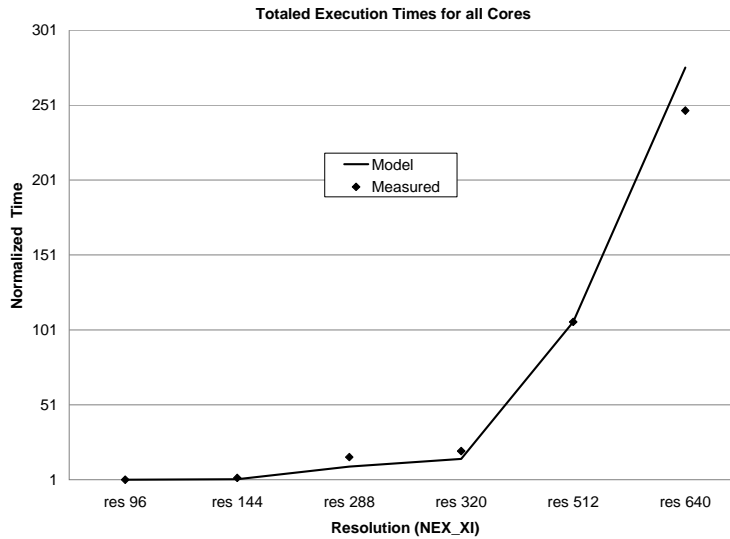


**Figure 6. Fitted curves for total communication time (in seconds) for all cores for different resolutions.**

Using the overall model, we were able to predict the total communication time for all cores of a hypothetical SPECfEM3D run with 12K processors and a resolution of  $NEX\_XI = 1440$  to be around  $7.3E6$  seconds, which corresponds to 599 seconds per core and 3.2% of overall execution time. Similarly, we predict the communication time per core for a SPECfEM3D run with 62K processors and a resolution of  $NEX\_XI = 4848$  to be around 28K seconds, which also corresponds to 4.7% of overall execution time. More

importantly, the results of modeling runs as well as the models we devised using these results indicate that the overall execution time of a SPECFEM3D run is dominated by the computation time and communication is not expected to be the bottleneck for scaling the application to tens of thousands of processors.

Similar to the communication model we also modeled the total runtime for all cores in order to estimate the runtime of our 1 second seismic period run and also confirm that the larger 12K core run did not exhibit any unforeseen bottlenecks. The results of modeling experiments showed that the overall execution time totaled for all computation cores is defined by the resolution used and is independent of the number of cores used. That is, for a given resolution, the execution time per core decreases but the totaled execution time for all cores is almost always the same. Using this fact and the total execution times for all modeling runs, we fitted the measured execution times. Figure 7 shows the actual (e.g. measured) and fitted total execution times for all cores (normalized with respect to the minimum) for different resolutions.



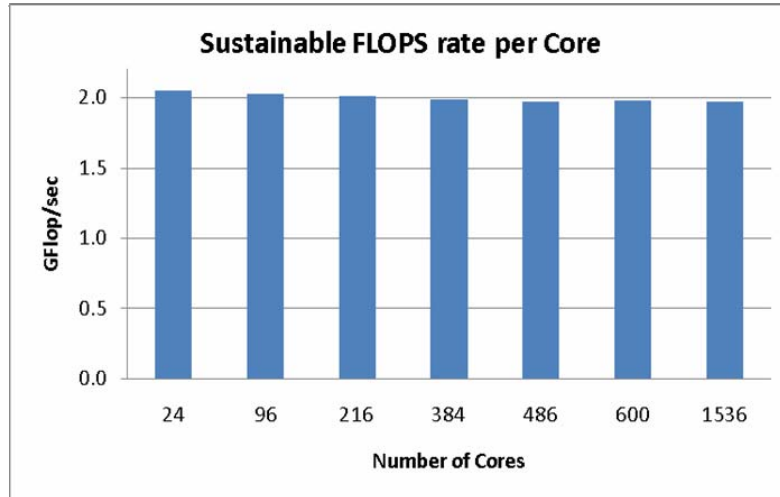
**Figure 7. Predicted and actual total time spent for all cores for different resolutions.**

Figure 7 shows that total execution time of SPECFEM3D for all cores increases significantly (quadratic) as the resolution increases. Using the fitted function, we were able to predict the totaled execution time of all cores of SPECFEM3D run with a 12K processors and a resolution of  $NEX\_XI = 1440$  within 12% error, indicating that no unforeseen bottlenecks emerged as the scaling was increased.

Similar to modeling communication we developed a model for the overall sustained FLOPS rate of the application using the modeling runs. The results show that the sustainable FLOPS rate for SPECFEM3D increases directly proportional to the number of processors it is run on and for the same number of processors slightly increases as the resolution increases. The FLOPS rate ranged between 48 GFlops to 3 TFlops for our modeling runs.

More importantly, the results show that the average sustainable FLOPS rate per core for the majority of the runs is similar and is around 2.0 GFlops on Franklin and the FLOPS rate per core does not vary much as the minimum and maximum rates per core do

not differ much. They also show that SPECfem3D is able to operate sustain 38.5% of the theoretical peak FLOPS rate (~44% of Rmax) of each core (2.0 GFlops vs 5.2 GFlops) on Franklin. Figure 8 presents the overall sustained FLOPS rate in billions (G) for each core for a set of runs for the main loop of the solver component.



**Figure 8. Overall FLOPS rate (G) per core for different numbers of processors.**

Overall, results of our modeling runs show that the sustainable FLOPS rate per core is almost the same independently of the processor count and of the resolution, and that the total sustainable FLOPS rate is directly proportional to the number of processors the application is run on. This indicates that a simple model for predicting the FLOPS rate for any run of SPECfem3D\_GLOBE on Franklin can be devised as  $2 \times CPU$  GFlops. Using this simple model, a run of SPECfem3D\_GLOBE on 12K processors was roughly predicted to achieve a sustainable FLOPS rate of 24 TFlops.

## Results of actual large simulations

The merged SPECfem code run on the NERSC system Franklin was successfully completed on 12,150 cores running for nearly 6 hours achieving around 24 TFlops (44% of Rmax) to model a shortest seismic period of 3 seconds. This confirmed the performance extrapolation model. A full system run on Franklin (19K cores) encountered a node failure during the run and was not able to complete at time of submission but will be redone during the next few weeks. The code also ran successfully on 4,056 cores on Ranger to model a shortest seismic period of 5.2 seconds. Ranger at TACC was also able to successfully start the code on 12,150 cores, but unfortunately there was not enough memory to support the resolution we attempted with the current version of the merged mesher/solver (possibly complicated by MPI buffer configuration tuning ongoing at TACC); in the next few weeks, improved versions of the merged code and more system tuning should be able to overcome this. The success of these runs makes us confident that we will be able to take advantage of a much larger number of cores and a much larger amount of memory, as available on Ranger.

We are obtaining similar fractions of peak on Ranger (the ranger cores are rated 8 Gflops each peak rather than 5.2 Gflops). Thus based on the success and record set

already at these relatively large processor counts, the data from the models presented in Section 5, and the expected completion of the code optimizations discussed in Section 4, we will perform a simulation attempt on a full-system run at TACC on 62,424 cores to achieve a 1 second period and nearly 200 TFlops (around 44% of peak of Ranger) and report the results in the final paper if accepted.

### **Acknowledgements**

Some improvements in the package were implemented while D. Komatitsch and D. Michéa were visitors at the Barcelona Supercomputing Center (BSC, Catalonia, Spain) in the context of the HPC-Europa program. The help of Jesús Labarta, Sergi Girona, José Cela and of the ParaVer analysis tool was invaluable. The authors would also like to thank Rogeli Grima from BSC and Sébastien Deldon from the Portland Group Inc. for fruitful discussion regarding Altivec and SSE instructions. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper. This work was supported in part by Performance Evaluation Research Institute (PERI), (DE-FC02-06ER25760), a DoE Office of Science SciDAC2 Institute, The Cyberinfrastructure Evaluation Center, (NSF-OCI-0516162), and Workshop on Petascale Computing and the Geosciences, (NSF-GEO-0621611). This material is based in part upon research supported by French ANR grant NUMASIS ANR-05-CIGC-002 and European FP6 Marie Curie International Reintegration Grant MIRG-CT-2005-017461. This work was supported in part by NSF Award #0621611 Workshop: Petascale Computing in the Geosciences.

### **References**

1. Computational Infrastructure for Geodynamics (CIG).
2. IPM: Integrated Performance Monitoring.
3. Canuto, C., Hussaini, M.Y., Quarteroni, A. and Zang, T.A. *Spectral methods in fluid dynamics*. Springer-Verlag, New York, 1998.
4. Chaljub, E. and Valette, B. Spectral element modelling of three-dimensional wave propagation in a self-gravitating Earth with an arbitrarily stratified outer core. *Geophys. J. Int.*, 158 (131-141).
5. Cohen, G., Joly, P. and Tordjman, N. Construction and analysis of higher-order finite elements with mass lumping for the wave equation. *Proceedings of the second international conference on mathematical and numerical aspects of wave propagation, SIAM*. 152-160.
6. Fischer, P.F. and Rønquist, E.M. Spectral-element methods for large scale parallel Navier-Stokes calculations. *Comput. Methods Appl. Mech. Engrg.*, 116. 69-76.
7. Komatitsch, D., Labarta, J. and Michéa, D. A 21 billion degrees of freedom, 2.5 terabytes simulation of seismic wave propagation in the inner core of the Earth on MareNostrum. *Proceedings of the 8th World Congress on Computational Mechanics (WCCM8) and the 5th*

*European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008).*

8. Komatitsch, D. and Tromp, J. Introduction to the spectral-element method for 3-D seismic wave propagation. *Geophys. J. Int.*, 139 (3). 806-822.
9. Komatitsch, D. and Tromp, J. Spectral-element simulations of global seismic wave propagation-I. Validation. *Geophys. J. Int.*, 149 (2). 390-412.
10. Komatitsch, D. and Tromp, J. Spectral-element Simulations of Global Seismic Wave Propagation-II. 3-D Models, Oceans, Rotation, and Self-Gravitation. *Geophys. J. Int.*, 150. 303-318.
11. Komatitsch, D., Tsuboi, S., Ji, C. and Tromp, J., A 14.6 billion degrees of freedom, 5 teraflops, 2.5 terabyte earthquake simulation on the Earth Simulator. in *Proceedings of the ACM/IEEE Supercomputing SC'2003 conference*, (Phoenix, Arizona, USA, 2003).
12. Komatitsch, D. and Vilotte, J.P. The Spectral-element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures. *Bull. Seismol. Soc. Am.*, 88 (2). 368-392.
13. Liu, Q. and Tromp, J. Finite-Frequency Kernel Based on Adjoint Methods. *Bulletin of the Seismological Society of America*, 96 (6). 2383-2397.
14. Patera, A.T. A Spectral element method for fluid dynamics: laminar flow in a channel expansion. *J. Comput. Phys.*, 54. 468-488.
15. Priolo, E., Carcione, J.M. and Seriani, G. Numerical simulation of interface waves by high-order spectral modeling techniques. *J. Acoust. Soc. Am.*, 95 (2). 681-693.
16. Ronchi, C., Iacono, R. and Paolucci, P.S. The "Cubed Sphere": a new method for the solution of partial differential equations in spherical geometry. *J. Comput. Phys.*, 124. 94-114.
17. Sadourny, R. Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. *Mon. Wea. Rev.*, 100. 136-144.