University of Delaware
Department of Electrical and Computer Engineering
Computer Architecture and Parallel Systems Laboratory

Algorithms, Applications, and Environments for Emerging Petascale Architectures

Guang R. Gao *H. Tufo* *S. Thomas*
R. Loft *R. Govindarajan†* *J. Moreira* *J.Castanos*

CAPSL Technical Memo 44
March 4, 2003

†Supercomputer Education & Research Centre, Dept. of Computer Science & Automation,
Indian Institute of Science

Project Description
ITR/ACI:Algorithms, Applications, and Environments for Emerging Petascale Architectures

a. 1. Introduction

1.1 The Challenge of Realizing Petascale Computing Systems

A sustained calculation rate of one Teraflops was first achieved on Dec 4, 1996 using approximately 7000 processors of the Intel machine ASCI Red at Sandia National Laboratories. Since that time, a variety of applications have approached or exceeded the Teraflops barrier on several supercomputing systems (e.g., 2001, 2002 Gordon Bell awards). Unfortunately, Teraflops computing systems do not provide sufficient computing power to adequately address many important scientific and engineering problems. These "grand challenge" problems, including climate system simulation and numerical weather prediction, nuclear stockpile stewardship, CFD applications in aerospace research, chemistry, drug discovery and bioscience applications, such as protein folding, all seem have a nearly insatiable demand for computation power. It is no surprise therefore that recently the President's Information Technology Advisory Committee (PITAC) report specifically recommended an initiative to achieve Petaflops sustained performance on *real applications* by 2010. Given that a Petaflops is a *million billion* floating−point operations per second and that, if Moore's Law is obeyed, CMOS−based processors are unlikely to sustain much more than 10 Gigaflops by the end of this decade, it follows that systems constructed from CMOS components will require at least 100,000 processors to achieve one Petaflops in this timeframe. In response, the high performance computing community has actively pursued the long−term development of Petascale computing solutions (e.g., the NASA sponsored PETA project under Federal High Performance Computing and Communications (HPCC) program (www.hq.nasa.gov/hpcc/petaflops/definition.html) and the DARPA High Productivity Computing System (HPCS) program (www.darpa.mil/ipto/research/hpcs/index.html)). However, the perceived loss of U.S. supremacy in supercomputing, derived from the remarkable performance achievements of the Earth Simulator, combined with the current geopolitical uncertainty and its associated national security concerns, has served to reemphasize the urgent need for robust, large−scale supercomputing systems.

Several problems are beginning to be recognized in using existing supercomputer architectures to construct Petascale computers. First, the power consumption of Terascale computing systems is becoming a problem. Modern CPU's require lots of power (e.g., Itanium−2 processors require 130 W) because they are complex systems composed of tens or hundreds of millions of transistors and clocked at very high speeds (>1GHz). The more power a CPU draws, the more difficult it becomes to densely package them. By 2010 it is estimated that microprocessors will dissipate 1 kw/cm^2 and have one billion transistors. (Also, as noted by Feng, et al. (Feng 2002), the hotter a CPU gets, the more likely it will fail: these authors have cited unpublished empirical data from two leading vendors that indicates that the failure rate of a compute node doubles with every 10^o C increase in temperature.) Also modern Tera−scale supercomputers are loosely coupled clusters of commercial components that are making less and less efficient use of space. For instance, Feng et al. noted that since the Cray C90, peak performance has increased 2000 times, whereas performance per square foot has risen only 65 times (Feng 2002). Computer room floor space and power both cost money, and a very large clusters occupying football field scale footprints are not cheap to build or maintain. All of this is exemplified by the $350M Earth Simulator in Japan, arguably the fastest machine in the world, which occupies a computer building 55 feet high with a floor area of 55 x 71 yards. The computer alone occupies a floor area of about 3000ft^2, has 3000 km of interconnect cables, consumes 8 Megawatts, and costs approximately $50M/year to operate.

The current parallel software execution models (MPI, threads) have also proven inadequate to the task of efficiently using parallel supercomputers much beyond about 1000 processors. There are many reasons for this. As the machine size increases, the ratio of computational work to communication generally decreases for a fixed problem size. MPI introduces O/S driven latencies, which are not readily hidden at high processor counts under these circumstances. The use of MPI also seems to naturally encourage superfluous copying of buffers and patterns of simultaneous cooperative communication which jam system networks. Simulations of physical systems such as the atmosphere exhibit load imbalances that become more severe as problem size is increased. These are not dealt with easily on distributed memory

systems using MPI, which must explicitly move data over the system network to balance the load. Current thread implementations are more flexible in handling load balancing but exist only within the scope of a single operating system image (typically 10's or 100's of processors) and are too expensive to start, synchronize and stop, and must obey an overly restrictive memory consistency model. Parallel meta−languages, such as HPF, simply inherit the underlying disadvantages of MPI and threads, and merely strive to hide some implementation details from the user.

In December 1999, IBM Research launched a multi−year and multi−disciplinary project called BlueGene. BlueGene is an ambitious project that currently involves more than 50 IBM researchers in laboratories worldwide. One of the stated goals of this project is to investigate biologically important phenomena such as protein folding. An equally important goal is to develop the next generation of Petascale high performance computing architectures. In November 2001 IBM announced a partnership with Lawrence Livermore National Laboratory to build the BlueGene/L (BG/L) supercomputer, a new architecture for high performance parallel computing based on low cost, low power embedded PowerPC technology. The LLNL BG/L system will have 65,536−nodes each capable of 5.6 GigaFlops peak. BG/L has several promising characteristics relative to current Tera−scale systems. First, BG/L's overall cost−performance ratio is expected to be about an order of magnitude less than the Earth Simulator's. Though it will appear three years after the Earth Simulator, its peak floating point rate is expected to be about 9 times higher, representing more than a factor of two improvement over what Moore's Law would predict. BG/L has a very fast combining network that will be useful for broadcast and reduction operations, which are a weak point of all current large−scale clusters. A detailed description of the BlueGene/L system is provided in section 3.

1.2 Our Proposed Response

We propose to investigate and address the technical obstacles to achieving practical Petascale computing in geoscience applications, using the IBM BG/L system as the target compute platform. We believe that achieving sustained Petascale performance will require much more than simply scaling up existing applications. Rather, it will entail addressing a broad range of technical challenges, and will require a close partnership between industry, national laboratories and academia to be successful. We propose to create such a team, composed of computer scientists and architects at IBM's T. J. Watson Research Center, where BG/L is already in development; scientists at Lawrence Livermore National Laboratory (LLNL) where a first large BG/L system will be deployed; computational and atmospheric scientists (application domain experts) at the National Center for Atmospheric Research (NCAR) and McGill University; and computer scientists knowledgeable in high performance computing and fine−grained multi−threading at the University of Colorado and the University of Delaware. The team will proceed by establishing the performance of conventional techniques, evolving new parallel computing paradigms, and proving out radically new algorithms.

We expect the conclusions drawn from our effort to map atmospheric models onto the BG/L architecture to be very general. At their core, the target applications contain many important challenges that frequently arise in parallel computing. These challenges include the scalability of global reductions, dealing with dynamic load imbalances, and coping with non−local operations such as the Fast Fourier Transform (FFT), Legendre Transforms (LT), and global data transpositions. In each case, our methodology will be to compare conventional MPI/OpenMP implementations with fine−grained multithreaded versions implemented under the EARTH fine−grained multithreading environment developed the University of Delaware. The performance of global reduction operations, which are found in many important numerical techniques, such as iterative solvers, is an important limitation of current large−scale computers. The fast integer reduction tree network in BG/L, which may address this issue, will be tested under this proposal in two ways. First, we will work with IBM to test the efficiency and scalability of conventional floating point MPI−based reduction operations on the BG/L tree network. Second, under the EARTH environment, we propose to exploit a novel efficient global reduction scheme based on a dataflow style of computation (Theobald 2000). The models have dynamic load balancing issues associated with the cloud physics parameterizations that are applied, a fundamental problem encountered in other application domains. Under EARTH, we propose to implement a dynamic load−balancing algorithm that uses history information and then employs either a receiver−initiated or sender−

initiated strategy. This will be compared with model performance without load−balancing. Finally, to address non−local operations such as FFT's, LT's and global data transposition, we plan to compare the MPI−based transposition technique with the "percolation model", an extension to the EARTH environment that allows asynchronous remote prefetch operations.

We intend to demonstrate the capabilities of BG/L, which we hope will exceed those of the Earth Simulator, by using it to perform fundamental scientific research. Preliminary performance model results, discussed in section 5.1, confirm the feasibility of using BG/L to simulate grand−challenge atmospheric problems, and suggest that BG/L could dramatically out perform the Earth Simulator, potentially achieving as much as 63 TeraFlops for a 10 km global model with 96 layers. We have identified three important questions in turbulence and atmospheric dynamics, and created an international team of scientists to investigate them. The scientific goals of our team are threefold: first to resolve competing theories concerning the observed kink in the energy spectrum of the earth's atmosphere, representing a separation between large and small−scale dynamics; second, to understand the formation of coherent structures in stratified turbulent flow, arising from quasi−random initial conditions; and third, to understand the origin and dynamics of the tropical Madden Julian Oscillation (MJO), particularly as it relates to the ENSO (El Niño) phenomenon. The details of these three investigations are discussed in more detail in section 2.

The atmospheric scientists at NCAR and McGill University will work with the computational science experts at NCAR and computer scientists at the University of Colorado and the University of Delaware to implement the target applications on BG/L, using canonical parallelization techniques (e.g. MPI/OpenMP). The team consist of well−qualified experts in mapping such applications to highly parallel systems: it includes one Gordon Bell prize winner, one two−time winner and two members who are recipients of a Gordon Bell Honorable Mention. Simultaneously, members of our team from IBM and the University of Delaware will port and tune the EARTH environment to BG/L. Fine−grained multithread versions of our applications will then be developed to run EARTH under multi−threading environment. The performance of the two approaches will be inter−compared, the process iterated and our conclusions refined.

Intellectual Merit of Proposed Activity:
Our basic premise is that inter−disciplinary research between the computer and atmospheric science communities could be greatly enhanced by a research program such as the one proposed here. Computer architects analyze the performance of computational kernels on new designs, but often in isolation. These kernels lack the full complexity of an atmospheric general circulation model. Conversely, atmospheric modelers tend to propose new algorithms without direct feedback from computer architects. We propose a much tighter coupling between these two groups in order to extract the highest possible performance from the next generation of supercomputers. We will not only discover and publish innovative approaches to tackling the most challenging atmospheric simulations on these machines, but we will also educate and train a new generation of computational and atmospheric scientists.

Potential Broader Impacts:
Ultra−high rates of computation will require radically new hardware and software paradigms. By bringing the designers of Petascale systems and applications into close partnership at an early stage, we hope to accelerate the long−term rate of progress in computational science. This can have a profound impact on the understanding of fundamental scientific questions, such as the physical processes of the atmosphere, which are of great importance to society.

b. ITR Relevance, International Collaborators, Multidisciplinary Nature of Research, Integration of Research and Education:

c. The proposal represents applied computer science research in one of the four critical areas of information technology called out in the PITAC report, namely achieving Petaflops performance levels by 2010. The project brings together an outstanding inter−disciplinary team composed of computer scientists, architects, and atmospheric modelers to realize this goal. Undergraduate and graduate students, and post−docs will participate in all phases of the research program, and will gain hands−on experience working with real Petascale hardware along with state−of−the−art atmospheric general circulation and turbulence models. The team includes two international collaborators, both leading experts in geophysical fluid dynamics and turbulence, located at McGill University in Canada, and Nagoya University in Japan.

The driving applications, geophysical turbulence and climate modeling, along with their core numerical algorithms and implementation requirements, are presented in Section 2. We provide an overview of IBM Blue Gene project and BG/L platform in section 3. In section 4, we present EARTH, a light−weight threading environment designed to address the "memory−wall" problem, discuss the research issues involved in determining how to optimally map EARTH to the BG/L architecture, and then show how to map applications onto EARTH. In section 5, we present research area that need addressing in order to map the driving application algorithms onto BG/L using MPI and then under the EARTH environment. Additionally, we discuss the potential influence of this research on future Peta−scale platform design, in particular BG/C. Finally, in sections 6 and 7, we present project team qualifications, milestones, and management plan.

d. 2 Atmospheric Science

In this section we provide an overview of three important areas of research in geophysical fluid dynamics that require Teraflops and ultimately Petaflops sustained performance. Then we describe the numerical models we will use to investigate these problems along with their basic algorithms and kernels.

e. 2.1 Geophysical Turbulence

Gage and Nastrom (1986) collected observational data indicating that the energy spectrum of the Earth's atmosphere contains a 'spectral kink' separating large and small scales. Their spectra cover scales ranging from 3 km to nearly 10,000 km. The observed spectrum is characterized by a downscale enstrophy cascade at large scales and a inverse energy cascade at small scales. Charney (1971) attributes the −3 slope at scales above 1000 km to quasi−geostrophic turbulence. The mesoscale dynamics follow a Kolmogorov −5/3 spectral slope. Two different mechanisms have been proposed to explain the observed mesoscale spectra. The first is strongly nonlinear and based on quasi 2D turbulence. Lilly (1983) postulates that it is due to stratified turbulence at small scales. The second mechanism is based on a weakly nonlinear wave theory involving the spectrum of internal waves. A 2D inverse cascade was shown not to work in stratified turbulence by Herring and Metais.

At length scales below 1000 km, Lilly (1983) suggests that small scale sources of energy could be provided by thunderstorms and breaking internal waves. Small−scale shear instability may also contribute. He argues that only a small amount of this energy needs to inverse cascade in order to account for the observed mesoscale spectrum. Some of these types of atmospheric flows are nonhydrostatic, and therefore to reproduce the observed energy spectrum of the Earth's atmosphere might require running a global nonhydrostatic model with prescribed heat fluxes. The horizontal resolution of such a simulation would have to be on the order of 1 km in order to resolve the vertical convection leading to mesoscale storms or wave breaking. Given the restrictive time step requirements and enormous number of degrees of freedom involved, it becomes readily apparent that a Petaflops super−computer would be required to carry out such a simulation.

The alternative theory postulates that the observed spectra are due to internal waves. This contribution is from modes not possessing PV, but not necessarily with high linear frequencies. If this is true for the atmosphere, then a 3D hydrostatic primitive equations model may be capable of correctly capturing the dynamics of the Earth's atmosphere. The length scales involved may be accessible at current weather model resolutions. However, the time scales may be more restrictive as the explicit treatment of gravity waves could be important. Nevertheless, it would be extremely important to atmospheric modelers to determine whether or not the primitive equations are adequate for reproducing the observed dynamics of the global circulation. There is tentative evidence suggesting that the spectral kink is visible in results from the GFDL−Princeton SkyHi model, Koshyk, Hamilton and Mahlman (1999). Their approach is to use explicit methods and resolve as much as possible. We propose to investigate further with a spectral transform dynamical core run at higher resolution on the IBM BG/L.

f. 2.2 Coherent Structures

Isolated coherent vortices have been observed to emerge out of random initial data in quite a variety of geophysical fluid flows based on several of the commonly employed model equations. Progress on the temporal scaling laws of the vortex statistics in decaying turbulence has been achieved by appealing to a combination of numerical empiricism and dimensional analysis. For example, Carnevale et al. (1991)

developed an argument predicting vortex radii and mean−square intensities given a numerical simulation's prediction of the fractional area occupied by vortices in decaying two−dimensional turbulence. Central to their argument was the assumption that the vorticity in the core of a vortex remains relatively constant. Following this, Bartello and Warn (1996) measured a self−similar decay of the one−point vorticity probability density. Vorticities fall on the universal curve except for values beyond a characteristic maximum, where probability falls off rapidly. This is due to the fact that vorticity is everywhere bounded by its initial value in two−dimensional flow. Vortex statistics crucially depend on the evolution of this maximum, which is effectively determined by the vortex dynamics. The outstanding issue for this problem is how the decay rate of the characteristic vorticity of the most intense vortices depends on the Reynolds number (i.e. model resolution). Determining it requires massive amounts of resolution and very long model runs. Knowing the answer will provide one of the first clues in a very difficult problem that is central to the future of turbulence theory.

There is preliminary evidence that the same self−similar vorticity decay (although with a different universal function) is present in the continuously stratified 3D quasi−geostrophic equations. It therefore is reasonable to assume that it can be observed in the rapidly rotating stably stratified turbulence of more realistic atmospheric and oceanic models. It will be explored in models employing multiply periodic geometries and the spectral transform method. These models are based on the Navier−Stokes and Bousinessq equations, discretized using high−order pseudo−spectral methods.

g. 2.3 Climate Modeling

Atmospheric moist processes (i.e., processes involving phase changes of water) are a fundamental component of atmospheric dynamics and are the most uncertain aspect of climate change research (IPCC 2001). Consequently, any numerical model that aims to be relevant to weather or climate on Earth must include a realistic representation of moist processes, including the latent heating associated with phase changes (e.g., formation of clouds), and − perhaps more importantly − the development and fallout of precipitation. From the point of view of the large−scale atmospheric energy budget, precipitation reaching the ground is a manifestation of the latent heating of the atmospheric column.

Atmospheric moist processes are arguably the most challenging aspect of any numerical weather or climate model. This is because the presence of moisture leads to a new class of fluid motions, namely moist convection, which is a small−scale phenomenon requiring both very high horizontal and vertical resolution (on the order of a kilometer) to explicitly resolve it numerically. Climate and global weather prediction modelers have been struggling with the representation of moist convection since the early days of atmospheric general circulation modeling. To resolve moist convection, the governing equations must include nonhydrostatic effects (Grabowski and Smolarkiewicz 2002, hereafter GS02). This set of governing equations is considerably more difficult to solve than the hydrostatic primitive equations traditionally used in lower resolution atmospheric models.

GS02 presented an approach to include moist precipitating thermodynamics into a modeling framework based on a semi−implicit discretization of the anelastic equations. Physical processes considered include the formation of clouds (i.e., condensation of water vapor), development of precipitation, and precipitation fallout from one model vertical level to another. These processes are considered in the spirit of state−of−the−art cloud−resolving models (cf. Grabowski 1998) and include both warm−rain and ice processes. An important feature of the GS02 approach is that it can be applied across a wide range of model temporal and spatial resolutions. The strategy is to treat moist processes using a time step capable of resolving phase changes (on the order of one minute), and then time−averaged tendencies are fed−back to the large−scale dynamics. This differs from the traditional approach employed in climate models based on the primitive equations.

We propose to extend this approach to a primitive equations dynamical core. However, the procedure outlined in GS02 must be modified to account for moist convection. Initially, we will develop a 1D column physics model in which a conventional convective parameterization, due to Emmanuel (Emmanuel 1991, 1999), handles convective processes and the GS02 algorithm treats stratiform moist thermodynamics. A simple surface flux algorithm and representation of radiative processes will also be

included. In this configuration, the model can be run at relatively high horizontal resolution (e.g. a fraction of a degree). The resulting 1D model column physics coupled to a GCM will be validated using an idealized moist Held−Suarez simulation, as in GS02, and moist baroclinic waves based on the test case of Polvani et al (2002). Next the Cloud−Resolving Convection Parameterization (CRCP; a.k.a. super−parameterization; Grabowski and Smolarkiewicz 1999; Grabowski 2001, 2002, 2003) will be interfaced to a GCM. CRCP is a novel technique for representing clouds in atmospheric models. The idea is to imbed a 2D cloud−resolving model in each column of a large−scale model in order to represent small−scale and mesoscale processes. Khairoutdinov and Randall (2001) tested this approach in the community climate system model (CCSM), using a short integration. A stretched vertical coordinate has recently been implemented in the CRCP code, facilitating direct coupling to a pressure vertical coordinate.

Our ultimate scientific goal is to examine the large−scale organization of deep convection in the tropics on time−scales from diurnal to intra−seasonal, such as the Madden−Julian Oscillation (MJO). The formation and propagation of the MJO signature is not well understood and poorly represented in existing weather and climate models (Slingo et al 1994; 1996). In order to investigate the formation of MJO−like coherent structures, we will consider an idealized earth−like aqua−planet test case. The initial test configuration will be the constant sea surface temperature (SST) case explored in Grabowski (2002, 2003). Later we will move to a more relevant test, an aqua−planet with a realistic meridional SST distribution, as suggested by Hayashi and Sumi (1986). Scientifically, CRCP is one way to explicitly represent the impact of moist processes on large−scale dynamics and climate in contemporary GCM's. High spatial resolution simulations using the traditional approach will be compared against coarser resolution CRCP simulations having approximately the same cost.

h. 2.4 Fundamental Numerical Algorithms

Because the spectral transform method is widely employed in current global weather models, the NCAR Built−on−Beowulf (BOB) spectral dynamical core will be employed to simulate the spectral kink. The coherent structures pseudo−spectral models rely on the fast Fourier transform (FFT) and the spectral toolkit (STK) at NCAR will form the basis of this study. We will couple the CRCP physics to the NCAR spectral element atmospheric model (SEAM), including a discontinuous Galerkin scheme for moisture transport. The algorithms contained in these applications consist of the spherical harmonic transform method used by BOB, the pseudo−spectral method based on the FFT, spectral elements employed in SEAM, and the solvers required by both SEAM and the CRCP parameterization scheme.

In the spherical harmonic transform method, physical fields involved in the dynamics are transformed into spectral coefficients by first performing a real FFT in the longitude direction, which is embarrassingly parallel for each latitude and level. The FFT is followed by a Legendre Transform (LT) of the resultant Fourier coefficients in the latitude direction. Computationally, the LT is an embarrassingly parallel, although load imbalanced in wave number operation, due to the triangular wave number structure of the associated Legendre polynomial basis. For a particular wave−number, the LT of a single layer of a field involves the multiplication of a real transform matrix of associated Legendre polynomial coefficients times the complex vector of Fourier coefficients. If multiple layers are included in the operation, the LT operation can be represented as an embarrassingly parallel collection of real matrix−complex matrix multiplies of different sizes. Several challenges arise when one considers mapping the spherical harmonic method onto massively parallel computers. Clearly, the method is highly non−local: global communications of some kind are required to move between the FFT and LT phases. These are generally organized into local computations followed by global transpositions.

In contrast, the spectral element method in SEAM provides the accuracy of the spherical harmonic transform, but confines the "spectral transforms" to many small quadrilateral elements, which tile the sphere by means of a gnomonic projection of the cube (Loft et al 2001). Continuity is maintained between elements by a simple averaging calculation with the neighboring element. Because the number of points per element is typically small (e.g. N=8−16), the calculations of gradients, divergence, interpolations, etc, are naturally cache−blocked matrix−matrix multiplies, and communication is nearest neighbor. These characteristics map well onto microprocessor architectures.

The gravity wave time step restriction in SEAM is overcome using a hybrid parallel, latency tolerant preconditioned conjugate gradient solver. This solver is implemented using reverse communication, and is thus readily adaptable to other applications and preconditioners without modifying the CG core. Further, the solver employs a more scalable variant of the standard CG algorithm in which the two inner products are grouped together (D'Azevedo 1992). The optimization of the inner product (global sum) operation for the CG solver on BG/L, a system noteworthy for having a global reduction network, is a key objective of our collaboration with IBM. The CG solver in SEAM employs an overlapping additive Schwarz preconditioner (Thomas et al 2003). Sub−domain solves are based on a low order finite element approximation of the spectral element Laplacian, resulting in a tensor−product basis. The associated coarse grid solve is potentially a serious bottleneck. We have investigated several coarse grid solver strategies and found the XXT method performs well at processor counts up to 4096 (Tufo and Fischer). The 2D CRCP models are based on a semi−implicit time−discretization of the anelastic equations. An elliptic problem for the pressure is solved using a generalized conjugate residual (GCR) Krylov iterative solver.

Both the traditional and CRCP physics schemes are embarrassingly parallel. CRCP has good parallel load balancing characteristics: the execution−time in empty cells and cells with clouds is found to only differ by 30%, due to faster convergence of the generalized conjugate residual (GCR) Krylov solver. The traditional 1D column model approach suffers from severe load imbalance, since only a fraction of model columns may have active clouds. On the other hand, CRCP is computationally expensive, roughly a factor of 100 times more so than traditional parameterizations. The traditional parameterization may benefit more under the EARTH multi−threading environment. Finally, future improvements to these schemes may introduce local communication between columns. In the case of the 2D CRCP models, the boundary conditions are periodic within each 2D slice. We anticipate that inflow/outflow type conditions may be implemented in the future, implying some form of local communication between neighboring columns

i. 3 BlueGene/L

j. 3.1 BlueGene/L Architecture

BlueGene/L (BG/L) is a new architecture for high performance parallel computers based on low cost embedded PowerPC technology. The main system, to be completed in late 2004 and hosted at LLNL, contains 65,536 compute nodes and has peak performance of 180/360 TF. The basic building block of BG/L is a custom System−On−A−Chip that integrates processing logic, memory and communications logic in the same piece of silicon. Each BG/L chip contains two standard embedded PowerPC 440 cores; each has access to a private, non−coherent, L1 cache. Both processors share a coherent 2 KB L2 cache and a coherent 4 MB L3 cache composed of EDRAM. Each processor drives a 64−bit "Double" FPU that can perform four floating−point operations using extended SIMD instructions. In most scenarios, only one of the 440 cores is dedicated to run user applications while the second processor drives the networks. At a target speed of 700Mhz the peak performance of a node is 2.8 GFlops. If both cores and FPUs in a chip are used, peak performance per node is 5.6 GFlops. Two nodes share a node card that also contains SDRAM−DDR memory. Each node can support up to 2 GB external memory but in the current configuration with 256 Mb DDR chips each node can directly address 256 MB at 5.5 GB/s bandwidth and 75 cycle latency. The low power characteristics of BG/L permit a very dense packaging. Sixteen compute cards can be plugged in a node board. A cabinet with 32 node boards contains 2048 CPUs with a peak performance of 2.9/5.7 TFlops. The complete system has 64 cabinets with a total of 16 TB of memory.

The BG/L ASIC supports five different networks. The main communication network for point−to−point messages for users is the 3D torus. Each node contains six bi−directional links for direct connection with nearest neighbors. The 64K nodes can be organized into a partitionable 64x32x32 3D torus. The network hardware in the ASICs guarantees reliable, unordered, deadlock−free delivery of variable length (up to 256 bytes) packets using a minimal adaptive routing algorithm. It can also provide simple broadcast functionality by depositing packets along a route. At 1.4 Gb/s per direction, the bisection bandwidth of a 64K node system is 360 GB/s. The tree network supports fast configurable point−to−point, broadcast and reductions of packets, with a hardware latency of 1.5 microseconds for a 64K node system. An ALU in the network can combine incoming packets using bitwise or integer operations, forwarding a resulting

packet along the tree. Floating point reductions can be performed in two phases (one for the exponent and another one for the mantissa) or in one phase by converting the floating−point number to a long 2048−bit representation. Following the tree, a separate set of links provides global OR/AND operations (also with a 1.5 microseconds latency) for fast barrier synchronization. Each ASIC contains a Gb/s Ethernet macro for external connectivity and supports a serial JTAG network for booting, control and monitoring of the system through an unarchitected network.

In addition to the 64K compute nodes, BG/L contains a variable number (1024 in the current design) of I/O and control nodes. Compute nodes and I/O nodes are physically identical only I/O nodes are attached to a Gbit Ethernet network, giving BG/L 1024 Gbit links to external file servers. The I/O nodes execute a version of the Linux kernel for embedded processors and are the primary offload engine for most system services. No user code directly executes on the I/O nodes. Compute nodes execute a single user, single process minimalist custom kernel that provides a familiar POSIX interface, and are dedicated to efficiently run user applications. No real system services execute in the compute nodes; control and I/O operations are shipped to the I/O nodes through the tree. The user's view of a system is of a flat, toroidal, 64K processor system, but the system manager's view is hierarchical: the machine looks like a 1024 node Linux cluster, with each node being a 64−way multiprocessor.

BG/L presents a familiar parallel programming model and a standard set of tools on top of a custom kernel. IBM has ported the GNU tool−chain (binutils, gcc, glibc and gdb) to the BG/L environment. IBM's XL compiler suite is being ported to provide F90 and advanced optimization support. BG/L will also support traditional IBM middleware such as LoadLeveler, and standard libraries such as BLAS and MPI.

BG/L's overall cost−performance ratio is expected to be about an order of magnitude less than the Earth Simulator's (ES), but BG/L also has the potential for technical advantages as well. Though it will appear three years after ES, its peak floating point rate is expected to be about 9 times that of ES, representing more than a factor of two better than Moore's Law would predict. Vector architectures, including ES, generally have excellent memory bandwidth and effective memory latency hiding characteristics, as long as the core algorithms can be expressed in the vector idiom. Moreover, it is easier to predict the performance of vector programs, compared the much more complex behavior of cache hierarchies. But vector nodes in themselves do not address the difficulties of synchronizing, reduction operations, and data movement between nodes. The large node count of BG/L is an alternative way of providing high aggregate memory bandwidth and inter−node communication, and the level one cache are much larger than vector register sets can be. Also, the aggregate non−floating point instruction rates are much higher. Finally, BG/L has a very fast combining network that will be useful for broadcast and reduction operations, which are a weak point of the ES.

IBM researchers have implemented a complete programming and simulation environment for BG/L. An instruction level simulator interprets executables produced by BG/L compilers for each BG/L node. Although not performance accurate (every instruction executes in one cycle), this simulator is architecturally accurate and models most of the features of BG/L: two cores, caches, and networks. The simulator executes approximately 2 million simulated instructions per second on a 1.2 GHz dual processor Linux workstation, with an effective slowdown of about 1000. Complete BG/L systems with multiple nodes (up to 125 in experiments) are simulated on a cluster in parallel. To study performance critical code, gate−level simulators are used but are limited to very small kernels.

Lawrence Livermore National Laboratory, IBM's partner in the BlueGene/L project and the host site for BG/L's planned delivery in early 2005, has a strong interest in aiding the development of a wide variety of applications and programming models for BlueGene/L. While not seeking funding from this proposal, LLNL has resources and expertise that are supportive of it. LLNL plans to run the BGLsim functional simulator on its Linux clusters, which it will make available to participating guests. LLNL will offer periodic workshops and tutorials on the use of the simulator and, eventually, real BG/L hardware, and on tuning codes for better performance on BG/L, as well as limited ongoing support in these areas. LLNL participants can take into account the needs of the climate modeling software and the EARTH runtime

software when deciding on priorities for the optimization of BG/L system software and libraries. When the full BlueGene/L hardware system is delivered to, LLNL will recommend that this project have access to it. Although the support LLNL can offer will be limited by resources and priorities, their participation is a key component of this project.

k. 3.2 Other Planned BlueGene Architectures

BlueGene/C (BG/C) is a new architecture that uses the VLSI technology from IBM's Microelectronics division in a more radical way. The fundamental premise in the architecture of BG/C is that performance is obtained by exploiting massive amounts of parallelism (on the order of eight million threads of execution in a full system), rather than the very fast execution of any particular thread of control. This premise has significant technological and architectural impacts. First, individual processors are kept simple, to facilitate design and large−scale replication in a single silicon chip. Instead of spending transistors and watts to enhance single−thread performance, the real estate and power budgets are used to add more processors. The architecture can also be characterized as memory centric. Enough threads of execution are provided in order to fully consume all the memory bandwidth while tolerating the latency of individual load/store operations.

The building block of BG/C is a single silicon chip and contains memory, processing and interconnection elements. A node can be viewed as a single−chip shared−memory multiprocessor. A node typically contains 8 MB of embedded shared memory and 256 instruction units. Each instruction unit is associated with one thread of execution, giving 256 simultaneous threads of execution in one node. Each group of 8 threads shares one data cache and one floating−point unit. The floating−point units (32 in a node) are pipelined and can complete a multiply−add per cycle. At 500 MHz, this translates into one GFlops peak per floating−point unit, or 32 Gflops per node. Each node has six channels for nearest neighbor communication. With 16−bit channels operating at 500 MHz, a bi−directional bandwidth of 1 GB/s per channel is achieved. The design includes optional off−chip DDR memory: blocks of data are transferred between the external memory and the embedded memory much like disk operations. Larger systems are built by interconnecting multiple nodes in a regular pattern. A system composed of a 32x32x32 3D torus of nodes, would deliver a peak computation rate of approximately one Petaflops.

BG/C systems are not single purpose machines such as MD−Grape but are not truly general−purpose computers either. Combined logic and memory processes have a negative impact: the logic is not as fast as in a pure logic process and the memory is not as dense as in a pure memory process. Due to its single−chip nature, BG/C is a small−memory system. The external DRAM is not directly addressable and the bandwidth to it is much lower. Future generations of BG/C are expected to include larger memory, but the current ratio of 250 bytes of storage per Mflops, compared to approximately 1MB/1MFlops in conventional machines, will likely decrease. BG/C targets problems that exhibit two important characteristics. First, they should be able to exploit massive amounts of parallelism, on the order of a million processors. Second, they should be compute intensive. IBM has previously demonstrated that Molecular Dynamics is one those applications. We would like to investigate if the atmospheric problems described in this proposal will also fit in this category.

IBM has developed a single and multichip simulation environment for a 32−bit version of BG/C and a complete system software stack that includes compilers, kernels, runtime libraries and communications libraries. This environment has been made available to several universities to support exploratory research on SMT architectures. It includes a cost model that permits early performance estimates. It is also highly parametrizable and allows one to study the impact on performance of different design features.

Finally, IBM's BlueGene team is beginning discussions on BlueGene/P. The characteristics of this machine are in very preliminary stage. IBM also plans to present a proposal (called PERCS) to the DARPA's High Productivity Computing Systems (HPCS) initiative later this year. We believe that the lessons learned in this project will have a significant impact on the design of these machines.

l. 4 EARTH: Efficient Architecture for Running THreads

In this section we discuss how to exploit an adaptive fine−grain multi−threaded execution model, such as the Efficient Architecture for Running Threads (EARTH), Hum et al (1995), (1996), on BlueGene architectures for the proposed application studies. We will focus on BG/L, and include a brief discussion on other BlueGene architectures. For a comprehensive background on multithreading, consult Agarwal et al (1990), Alverson et al (1990), Gao et al (1995), Culler et al (1991), Tullsen et al (1995). Survey articles on the evolution of multithreaded architectures are Dennis and Gao (1994), Najjar et al (1999). For a brief overview on related work on fine−grain multithreading, in the context of the work proposed in this proposal, refer to (Theobald 1999).

m. 4.1 EARTH Execution Model

EARTH supports an adaptive eventDriven multi−threaded execution model, containing two thread levels: *threaded procedures and fibers.* A threaded procedure is invoked asynchronously − forking a parallel thread of execution. A threaded procedure is statically divided into fibers −− fine−grain threads communicating through dataflow−like synchronization operations. These generate *events* to signal that control and data dependences are satisfied, triggering a *fiber firing* that schedules a fiber for execution. One effective strategy of fiber formation is to place the source and destination of long−latency operations into different fibers, such as non−local data movement operations (e.g. in caches or near memory). This model permits local synchronization between fibers using only relevant dependences, rather than global barriers. It also enables an effective overlapping of communication and computation, allowing a processor to grab any fiber whose data is ready.

Conceptually, a node in an *EARTH virtual machine* has an *Execution Unit* (EU), which runs the fibers, and a *Synchronization Unit* (SU), which determines when fibers are ready to run, and handles communication between nodes. The EU and SU are communicating through dedicated queues: a *Ready Queue* (RQ) of fibers waiting to run on the EU, and an *Event Queue* (EQ) containing events corresponding to EARTH operations generated by fibers executing in the EU. To address the memory wall problem, threaded function invocation is asynchronous and can be made adaptive − it may first generate a template with little cost and the actual timing/site of its initiation and initialization may be determined by a dynamic load balancing scheme. Asynchronous threaded procedure invocations provide good thread mobility and are effective in balancing dynamically changing workloads.

Fibers are scheduling quanta generated/optimized by the compiler and contain little architectural state − their invocation/termination only require a few cycles. Fiber scheduling is event−driven and their order of execution is determined at run−time based on the dependence satisfaction and available resource. Event−driven fine−grain multi−threading at the fiber level has been shown to have the unique ability of tolerating latencies, especially those due to irregular and dynamically changing access patterns with poor locality.

n. 4.2 Mapping EARTH onto BlueGene Architectures

Presently, the API for the EARTH virtual machine is programmable through the EARTH Threaded−C language − an extension of C with EARTH primitive operations (Tremblay et al 2000). On the IBM−SP or Beowulf clusters, runtime system (RTS) libraries, running under the native operating system, realize the EARTH virtual machine. Readers are referred to (Theobald 1999) for more detailed information (including EARTH−C, EARTH Threaded−C, EARTH RTS, EARTH−MANNA simulator), and EARTH implementations on other platforms (Kakulavarapu 1999, Morrone 2001)

Based on past experience, the RTS can work with the BG/L custom kernel, or take over custom kernel functions (the latter may not be viable for practical reasons). BG/L architects will suggest tradeoffs and optimize interactions with the BG/L node kernel. If a lower level network API is accessible, several research questions arise. The network infrastructure of BG/L is also of interest. For example, can the reduction/combining power of the dedicated BG/L network hardware be exploited? Without the reduction hardware network, EARTH architects have been using a data−driven style software reduction network based on the capacity of fibers and their event−driven scheduling mechanism There should be a tradeoff to consider with the reduction network and there should be a way exploit both.

To take advantage of the BG/L node architecture, note that the MANNA node consisted of two

processors, one intended for computation and the other for communication. Under EARTH−MANNA, we used one processor as EU and the other as SU. This strategy may also prove effective on BG/L. However, the cache−coherence mechanism between the two processors could cause a "ping−pong" effect on the EQ and RQ queues implemented in the node's shared memory. Under the co−processor model of the BG/L architecture, the two processors communicate through "an uncached region" of memory or through a scratchpad region in the L3 cache. A research question is: when Mapping EARTH to BG/L is this a good choice? If so, how should we best utilize this feature? Are there any other desirable features the architecture can provide in the future? We also note that each processing core has access to a private, non−coherent, L1 cache. Can EARTH take advantage of this feature?

BG/L architects are providing a "symmetric mode" where both CPUs run applications and the user is responsible for cache coherence issues. We plan to examine this mode carefully, and compare it with other modes for implementation tradeoffs. BG/L has a rich set of dedicated I/O nodes with an identical architecture as the compute nodes. This provides interesting and challenging research questions. For example, should we use a similar scheme to divide the compute and I/O operations as provided in BG/L? A strategy is required to overlap I/O operations under fine−grain multithreading at the compute nodes. However, are there any additional advantages in porting the RTS to an I/O node − so some fine−grain multithreading capacity can be exploited?

o. 5 Research Challenges in Mapping Target Applications to BG/L

5.1 A Performance Model of a Conventional MPI Application on BG/L

A performance model of the spherical harmonic transform has been constructed by NCAR in collaboration with IBM scientists. It is similar to models that accurately predict climate model performance on other systems. It uses machine performance values of 5.6 GF peak for the BG/L node, a network link bandwidth of 175 MB/sec, and a communication latency of 5 microseconds.

To understand the performance model, the spherical harmonic transform must be briefly explained. The algorithm is performed on a Gaussian grid, which is equally spaced in longitude and nearly equally spaced in latitude. A vertical terrain following or pressure coordinate is generally treated with a separate finite differencing scheme. The partial differential equations governing the horizontal dynamics are represented in terms of the coefficients of the spherical harmonic basis functions, in terms of which spatial derivatives are then trivially constructed and equations time integrated. Physical fields involved in the dynamics are transformed into spectral coefficients by first performing a Real Fast Fourier Transform (FFT) in the longitude direction followed by a Legendre Transform (LT) of the Fourier coefficients in the latitude direction. Thus, in a message passing implementation, the algorithm may be viewed as a series of distinct phases, each of which is separated by a global transposition.

The performance model is based on a detailed understanding of the decomposition of application across a two dimensional virtual processor domain mapped onto the underlying three dimensional network topology of BG/L. The set of fields to be transformed for the dynamics may be visualized as being initially laid out in a simple two dimensional latitude−longitude decomposition. In terms of the virtual process topology, the global longitude index increases along the x−axis of the virtual processor topology, and the global latitude index increases along the y−axis. The first transposition distributes the layers of the fields across the y−axis and collects all of the longitude indices. A local Real FFT can then be performed. The next transposition is along the y−axis of the virtual process topology distributing all of the Fourier wave numbers and collects all of the latitude indices. A local LT can then be performed. Finally, the Legendre orders of the spectral coefficients must be distributed along the x−axis and layers collected. The equations of motion are then updated in spectral space and the process is reversed back to physical space where non−linear terms are computed and physics parameterizations are applied. Examining the amount of data involved in each transposition clarifies how to map the 2D virtual process topology onto the physical 3D torus of BG/L. Because of truncation, the number of degrees of freedom in the Fourier coefficients are approximately 2/3 the number in the physical fields. Legendre transforms further reduce the number of degrees of freedom to 2/9. Thus the two transposes along the x−axis of the 2D virtual process topology move approximately twice the data as along the y−axis. Therefore, the performance model assumes that the x−axis is localized into small chucks of the 3D torus. The precise

mechanism that achieves this hardware mapping remains unclear, but one avenue could be through the REORDER option to the MPI_CART_CREATE function.

The performance estimate for the FFT is 30% of peak. This number is based on NCAR's experience with the Spectral Toolkit (STK). STK is an ongoing project to produce highly efficient spectral transform software for the geoscience community. STK is written in C++ using a combination of techniques that allow for easy extensibility and flexibility without compromising performance. The resulting code typically achieves 20−40% of the peak Flops rate on a variety current microprocessors, Figure 2. The performance model estimate for the LT is 70%. This is based on NCAR experience with blocking the complex matrix−real matrix multiply inside the LT. Detailed simulations by IBM of a matrix multiply for the 440 core, suggest a maximum achievable efficiency ranging from 70−90% of peak.

The results, shown in figure 1, confirms the feasibility of using BG/L to simulate grand−challenge atmospheric problems, and suggest that BG/L could dramatically out perform the Earth Simulator, achieving possibly as much as 63 TeraFlops for a 10 km global model with 96 layers. If achieved, this would represent an enormous advancement: we estimate BG/L could integrate one simulated year per day.

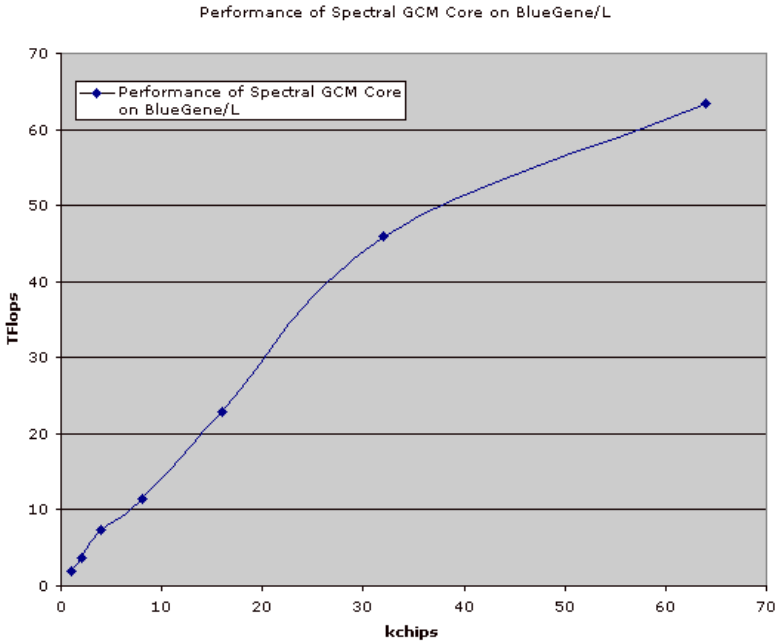

Figure 1: Estimated Performance of the Spherical Harmonic Primitive Equations for BG/L.

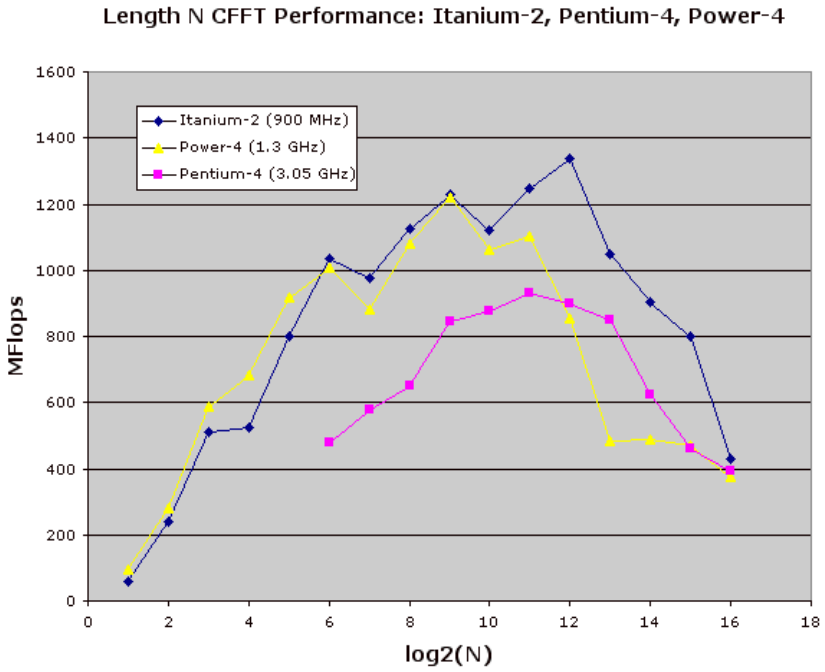

Figure 2: Performance of the NCAR Spectral Toolkit (STK) Complex FFT on different platforms.

p. 5.2 Mapping Our Applications to BlueGene/L with EARTH

In section 2.4, we outlined several computation kernels that are critical to the performance and scalability of the target applications. In this section, we discuss the research questions to be studied and solved in mapping these kernels to BG/L under the EARTH thread environment.

With EARTH's runtime support of fine−grain multithreading, the FFT algorithms can be implemented in a fine−grain event−driven fashion. Under this execution model, parallel computation must be pipelined to avoid the use of barrier synchronization between stages. Furthermore, the communication cost between FFT stages may be long and non−uniform. Fine−grain threading will provide a good way to tolerate such latencies. In an earlier work we have developed two different dataflow−style algorithms for FFT based on coarse−grain and fine−grain parallelism (Thulasiraman et al 2000). In the fine−grain algorithm, the number of threads scales linearly with the product of the input size and the number of processors. The coarse−grain algorithm models FFT as a producer−consumer problem and can be adapted to different architecture parameters for achieving scalable high performance. We propose to apply these FFT algorithms in the proposed application. The new challenges here are to study the tradeoff between fine−grain multithreading overhead and its capacity of latency hiding in the large FFT computation on BG/L.

The Legendre Transform is an embarrassingly parallel collection of matrix multiplies of different sizes. Several challenges remain on large−scale multiprocessors, because the data movement between FFT and LT phases involves highly non−local communication. Mapping each matrix−multiply as an asynchronously scheduled EARTH thread may hide latencies due to non−local communication. We may also overlap the data movement between FFT and LT phases with computation. To this end, we plan to leverage the "percolation model" − an extension to the EARTH model allowing asynchronous long−range pre−fetch operations that may involve data shuffling and reorganization (Jacquet et al 2003).

The conjugate gradient (CG) computation involves a matrix−vector multiplication and an inner product (global reduction). The fine−grained approach exploits a large pool of threads to overlap CG computation and communication, and to efficiently support asynchronous, fine−grained thread synchronization and communication. A novel efficient global reduction scheme, based on dataflow style of computation, has been developed (Theobald et al 2000). It uses multiple event−driven threads to construct nodes in the

dataflow network and employs asynchronous dataflow−like fine−grain synchronization and communication operations. This technique should work well on the fast reduction tree network in BG/L. The cost model in the earlier work will be extended to optimize the overlapped computation vs. communication, fiber−partitioning tradeoffs and take advantage of the rich hardware reduction network/operations.

We propose to investigate the use of dynamic load balancing strategies for the physical parameterizations described in section 2.4. Both the 1D traditional and CRCP packages will experience different levels of dynamic imbalance. The traditional parameterizations experience more dramatic load imbalance than CRCP scheme: nonetheless data migration to restore balance will remain a serious challenge. These load imbalances could become more severe simply due to the sheer scale of the BG/L machine, and are made more difficult to deal with due to the absence of a global shared address space. In earlier work, team members have designed, implemented, and evaluated nine dynamic load−balancing strategies for fine−grain multithreading systems (Cai et al 1999). Significant performance improvements can be achieved by using a dynamic load−balancing algorithm that uses history information and then employs either a receiver−initiated or sender−initiated strategy. Finally, advanced parallel 1D and 2D convection computation may incur fine−grain communication and synchronization between neighboring 1D columns (or 2D slices). A research challenge here is to formulate the new computation pattern with fine−grain communication operations. We propose to tackle this problem with collaboration between computation scientists and computer architects in the team. Intuitively, the power of adaptive event−driven multithreading model should provide interesting dimensions to address such challenges.

q. 5.3 How this project will influence future IBM machines

Blue Gene/L and Blue Gene/C are cellular designs that require efficient mapping of user applications to a three dimensional torus or mesh to achieve full performance. On the other hand, IBM's traditional line of large systems (SP, ASCI Blue, ASCI White or ASCI Purple) are flat machines that do not require explicit mapping of work to specific processors. We believe that this feature has been key in the success of these machines. Unfortunately, a large part of the cost is due to the switches that support this extended functionality. These switches also limit their scalability.

Blue Gene/C also requires a secondary level of parallelism to partition work between the individual threads in a node. Environments like EARTH hide this additional complexity. EARTH can allow us to build simpler and cheaper hardware without significantly affecting the productivity of the application programmers. We would like to know if end users will accept the overhead presented by more complex system software, if they would prefer to tackle the complexity presented by hardware themselves, or if they will still demand from vendors more forgiving and expensive machines.

Both Blue Gene/L and Blue Gene/C (a still unresolved issue in this case) have a secondary network to support global operations in hardware, a feature largely requested by users, that can overcome scalability issues, and that it is not present in the IBM product line. From a hardware point of view, these networks have a cost, in terms of wires and pins in the case of Blue Gene/L, or in terms of performance of the main network in the case of Blue Gene/C. IBM would like to investigate if these networks justify their cost.

6 Project Team Qualifications

r. 6.1 Prior Results for Thomas and Tufo

NSF Grant #CMG−0222282 "An Adaptive Mesh, Spectral Element Formulation of the Well−Posed Primitive Equations for Climate and Weather Modeling", was awarded $501,006 for the period 10/1/03 to 9/31/06. We are currently in the process of staffing the two research positions funded by this project, adding the interpolation−based non−conforming spectral element formulation of Fischer and Kruse to SEAM, testing adaptive mesh refinement technology, and developing a discontinuous Galerkin module for conservative advection. Publications:

Thomas, S. J., J. M. Dennis, H. M. Tufo, and P. F. Fischer, 2003: *A Schwarz Preconditioner for the Cubed−Sphere*, Selected Proceedings of the 2002 Copper Mountain Conference on Iterative Methods, *SIAM J. on Scientific Computing*, to appear.

s. 6.2 Prior Results for Gao

NSF Grant No. 0103723, "Next Generation Software: A framework for Developing Complex Applications on High−End Petaflop−Class Machines", was awarded $728,347 for the period 11/1/01to10/31/03. The first year annual report has been submitted to NSF. Reported progress (University of Delaware part) included refining the base program execution model, developing an executable performance model for fine−grain multithreading, improving the base program model, extensions to the EARTH runtime system, Threaded−C, support for EARTH runtime system on symmetric multiprocessors, and SMP clusters. Publications:

Gao, G. R, K. Theobald, Z. Hu, H. Wu, J. Lu, K. Pingali, P. Stodghill, T. Sterling, and R. Stevens 2002: Next generation system software for future high−end computing systems. NSF Next Generation Systems Program Workshop, held in conjunction with IPDPS−2002.

NSF Grant No. MIPS−9707125, CISE proposal on "Multithreaded Program Execution Model" was awarded $264,975 for the period 7/1/00 to 6/30/04. The project involves research on new models of execution and memory models for parallel architectures, with a focus on multithreaded architectures and high−level language programming paradigms. Publications:

Theobald, K. B., R. Kumar, G. Agrawal, G. Heber, R.K. Thulasiram, and G. R. Gao, 2000: Developing a communication intensive application on the EARTH multithreaded architecture. *Proceedings of the Euro−PAR Conference (Euro−PAR−2000)*, Munich, Germany.

NSF Grant No. CISE−9726388, "Crack Propagation on Teraflop Computers", was awarded $110,000 for the period 10/1/00 to 9/30/03. The project's focus is on the design of algorithms and systems to support the numerical simulation of crack propagation problems on parallel computers. The specific focus is on 3D time−dependent fracture simulations using unstructured, adaptive grids on the IBM SP−2 and on EARTH−MANNA. Publications:

Heber, G, R. Biswas, P. Thulasiram and G. R. Gao 1999: Using multithreading forautomatic load balancing of adaptive finite element meshes. Proceedings of Irregular 99 in conjunction to the International Parallel Processing Symposium (IPPS/SPDP), 969−977.

t. 7 Management Plan and Milestones

This is tightly integrated research project involving 15 researchers from the University of Colorado, the University of Delaware, McGill University, Lawrence Livermore National Laboratory, IBM, and the National Center for Atmospheric Research. The goal of this project is to produce leading edge, world−class research in atmospheric science and high−performance computing. To be successful we'll need to establish at the outset constructive and frequent dialogue between the two groups to foster innovation and generate new ideas. To achieve this we will have an Access Grid meeting once per month where it is expected that all team members participate by providing a brief research update. We will have a yearly retreat in Colorado (since 8 of the 15 are at CU and NCAR). And in year three we will hold a workshop for atmospheric and computational scientists. At the workshop we will present science results (both computer and atmospheric) and provide a series of tutorials on the EARTH environment, programming and operating BG/L, and advanced parallel programming techniques.

The management structure is relatively flat. Thomas will lead the atmospheric science group. His primary objective is to ensure that the atmospheric and computational scientists at NCAR are making progress towards solving the atmospheric modeling and turbulence problems described in section 3 and meeting the group's milestones (detailed in the following subsections). Gao will lead the computer science group. His primary objective is to ensure that the computer scientists at Delaware, IBM, and LLNL are making progress towards delivery of an efficient implementation of EARTH for BG/L and meeting the group's milestones. Tufo will lead the parallel algorithms group. Primary objective of the group is to investigate novel algorithms and implementations to address the numerical and solver requirements of the atmospheric science group, to inject those ideas into the computer science group, and, thereby, providing linkage between those groups.

u. 7.1 Year 1

Introduce 2D decompositions and STK library into the BOB model. Performance analysis of BOB using MPI on the BG/L. Parallel FFT from STK added to turbulence codes. Preliminary turbulence model runs.

Interface the 1D and 2D CRCP physics to SEAM and conduct idealized experiments. Access to BG/L functional and network simulators. Verification and testing of core kernel code. Access to BG/L prototype 512 node system at IBM for testing at the end of the year. Study how to map EARTH on BG/L architecture. Make necessary extensions to EARTH for efficient implementation on BG/L.

v. 7.2 Year 2

Spectral−kink and turbulence science runs using MPI implementations on simulators and prototype. Full BG/L system delivered to LLNL. High−resolution runs for the Gordon Bell competition. Performance analysis of turbulence codes before scaling up to the full BG/L configuration. High−resolution parallel runs with the SEAM and 1D physics. Build parallel version SEAM−CRCP. Prototype implementation and study of EARTH on BG/L using the functional simulator along with network simulator.

w. 7.3 Year 3

Access to full system at LLNL to pursue science runs. Testing of EARTH application implementations. Implementation of application kernels in the EARTH execution model on BG/L. Start porting science applications codes onto the BG/L architecture.

x. 7.4 Year 4

Porting and testing real applications on the full architecture. Performance evaluation of target applications under the EARTH model. Compare EARTH results with conventional MPI implementations. Refine the EARTH execution model for BG/L.

y. 7.5 International Collaborations

Prof. Peter Bartello is a well−known and highly respected researcher in geophysical fluid dynamics and turbulence. Prof. Bartello holds a joint appointment in the Mathematics and Atmospheric and Ocean Sciences departments at McGill University. He is also an international collaborator with the Geophysical Turbulence Program (GTP) at NCAR. His fundamental contributions include studies of the statistical nature of geophysical flows as a function of rotation and stratification and the impact of numerical time−stepping schemes on dynamics. Prof. Yoshi Kimura is a long−standing collaborator of Dr. Jackson Herring and member of GTP at NCAR. Prof. Kimura is a member of the Mathematics department at Nagoya University in Japan. His research interests include the formation of coherent structures in rotating stratified turbulence.

Project Summary
ITR/ACI:Algorithms, Applications, and Environments for Emerging Petascale Architectures

Recently, great attention in the high performance computing community has been garnered by the revelation that a geoscience application, namely a 10 km resolution atmospheric general circulation model (GCM), ran at 26 Teraflops on the Earth Simulator. Further advances in grand challenge domains such as geo−turbulence, numerical weather prediction and climate modeling will demand access to much higher performance from more flexible and dynamic computational platforms. It is impractical to simply scale up current Teraflops systems to attain Petaflops performance, because of their poorly integrated and inflexible architectural concepts, huge power consumption, poor space utilization and high cost. A new architectural paradigm has emerged in the race to Petascale computing, typified by the IBM BlueGene systems, which are characterized by tightly integrated, low power, densely packaged components, containing O(100,000) or more processors. We propose to create an international team consisting of computer scientists and architects at IBM's T. J. Watson Research Center, computational and atmospheric scientists at the National Center for Atmospheric Research (NCAR) and McGill University, and computer scientists at the University of Delaware, the University of Colorado, and Lawrence Livermore National Laboratory (LLNL) to define, investigate, and address the technical obstacles to achieving practical Petascale computing in geoscience applications.

The team, including four Gordon Bell prizewinners, will investigate how to effectively exploit the capabilities of BlueGene/L to meet the future challenges of atmospheric modeling using new parallel software execution models and algorithms. Specifically, the atmospheric scientists at NCAR and McGill will work with the computational science experts at NCAR and computer scientists the University of Colorado to implement the target applications, which currently run well on O(1000) processors using canonical MPI/OpenMP techniques, on IBM's BlueGene/L (BG/L) computer, initially with simulators and then moving to real hardware. At the same time, computer scientists at University of Delaware will work with IBM and LLNL researchers to implement and extend the EARTH environment on BG/L, focusing on the "*memory wall*" problem facing future Petascale multiprocessor designs. Next we will develop fine−grained multithread versions of our applications, adapting novel latency hiding algorithms for the FFT, matrix multiply, conjugate gradient solver and dynamic load balancing problems at the core of the target applications. The performance of the traditional MPI/OpenMP and fine−grained multithreading approaches on BG/L will be analyzed and inter−compared. Along the way, many questions concerning fine−grained multithreading in future Petaflops systems will be answered. What is the best way to implement EARTH on BG/L's distributed memory architecture? Does fine−grain multithreading scale on BG/L? How can thread percolation be implemented on systems with a multiple level memory hierarchy? Finally, can one characterize the essential hardware features needed to improve the performance of such systems?

Potential Broader impacts: We intend to use BG/L to push the frontiers of atmospheric science by comparing old and new algorithms and physical parameterizations at unprecedented resolutions. Three basic questions of turbulence and atmospheric dynamics have been identified. The first is to resolve competing theories concerning the origin of the observed kink in the energy spectrum of the earth's atmosphere, representing a separation between large and small−scale dynamics. The second question is related to the formation of coherent structures such as vortices, from quasi−random initial conditions in a stratified flow. The third is the origin of the tropical Madden Julian Oscillation (MJO), which dominates the intra−seasonal climate variability in the tropics, but its origin, dynamics and propagation, as well as role in longer−time−scale climate variations, such as the famous El Niño−Southern Oscillation (ENSO), remain unclear.

Intellectual merit of Proposed Activity: This project represents an unprecedented opportunity for advancing the computer science of Petascale systems with O(100,000) processors. Tightly coupled interaction will provide invaluable real world experience and feedback to computer scientists, architects, and atmospheric modelers. Post−docs, graduate and undergraduate students will participate in all phases of the research program. This inter−disciplinary project will educate a new generation of computational and atmospheric scientists and prepare them for careers at the frontiers computational science.

Project References
ITR/ACI:Algorithms, Applications, and Environments for Emerging Petascale Architectures

Agarwal, A. B. −H. Lim, D. Kranz, and J. Kubiatowicz, 1990: APRIL: A processor architecture for multiprocessing. *Proceedings of the 17th Annual International Symposium on Computer Architecture*, pages 104−−114, Seattle, WA, May 28−−31.

Alverson, R., D. Callahan, D. Cummings, B. Koblenz, A. Porterfield, and B. Smith, 1990: The Tera computer system. *Proceedings, 1990 International Conference on Supercomputing*, 1−−6, Amsterdam, June 11−15.

Balaji, V., 2000: Parallel numerical kernels for climate models. *Developments in Teracomputing: Proceedings of the Ninth ECMWF Workshop on the Use of High Performance Computing in Meteorology*, W. Zwiefhofer and N. Kreitz, editors. 277−295.

Bartello, P. and Warn, T., 1996: Self−similarity of decaying two−dimensional turbulence. *J. Fluid Mech.*, **326**, 357−373.

Bitterman, A., 1999: *Superconductors and Cryoelectronics in the Petaflops−Scale Computer Project*, Superconductor and Cryoelectronics, Vol. **12**, No. 1.

Cai, H., O. Maquelin, P. Kakulavarapu, and G. R. Gao, 1999: Design and evaluation of dynamic load balancing schemes under a fine−Grain multithreaded execution model. *Proceedings of the Workshop on Multithreaded Execution, Architecture and Compilation (MTEAC)*, in 1999 IEEE Symposium on High−Performance Computer Architecture (HPCA99), Orlando, Florida.

Carnevale, G.F., McWilliams, J.C., Pomeau, Y., Weiss, J.B. and Young, W. R., 1991, Evolution of vortex statistics in two−dimensional turbulence. *Phys. Rev. Lett.*, **66**, 2735−2737.

Charney, J. G., 1971: Geostrophic turbulence. *J. Atmos. Sci.*, **28**, 1087−1095.

Culler, D. E., A. Sah, K. E. Schauser, T. von Eicken, and J. Wawrzynek, 1991: Fine−grain parallelism with minimal hardware support: A compiler−controlled threaded abstract machine. *Proc. of ASPLOS−IV*, 164−175, Santa Clara, Calif.

D'Azevedo, E. and V. Eijkhout and C. Romine, 1992: Reducing communication costs in the conjugate gradient algorithm on distributed memory multiprocessors. *LAPACK working note 56*, University of Tennessee.

Dennis, J. B, and G. R. Gao, 1994: Multithreaded architectures: Principles, projects, and issues. In Robert A. Iannucci, G. R. Gao, R. H. Halstead, Jr., and B. Smith, editors, Multithreaded Computer Architecture: A Summary of the State of the Art, chapter 1, pages 1−72. Kluwer Academic Publishers, Norwell, Massachusetts.

Emanuel K. A., and M. Zivkovic−Rothman, 1999: Development and evaluation of a convection scheme for use in climate models. *J. Atmos. Sci*., **56**, 1766−1782.

Emanuel, K. A., 1991: A scheme for representing cumulus convection in large−scale models. *J. Atmos. Sci*., **48**, 2313−2335.

Feng, W., M. Warren, E. Weigel, 2002: *The Bladed Beowulf: A Cost Effective Alternative to Traditional Beowulfs*, downloadable at: http://public.lanl.gov/feng/Bladed−Beowulf.pdf (an updated version of *Honey, I Shrunk the Beowulf*, Los Alamos Unclassified Report **02−2582**, April 2002).

Gao, G. R., J. −L. Gaudiot, and L. Bic, 1995: Advanced Topics in Dataflow and Multithreaded Computers.

IEEE Computer Society Press.

Grabowski, W. W., 1998: Toward cloud resolving modeling of large−scale tropical circulations: A simple cloud microphysics parameterization. *J. Atoms. Sci*., **55**, 3283−3298.

Grabowski, W. W., 2001: Coupling cloud processes with the large−scale dynamics using the Cloud−Resolving Convection Parameterization (CRCP). *J. Atmos. Sci*., **58**, 978−−997.

Grabowski, W. W., 2002: Large−scale organization of moist convection in idealized aqua−planet simulations. *Int. J. Numer. Methods in Fluids*, **39**, 843−−853.

Grabowski, W. W., 2003: MJO−like coherent structures: Sensitivt simulations using the Cloud−Resolving Convection Parameterization (CRCP). *J. Atmos. Sci*., **60**, 847−864.

Grabowski, W. W., and P. K. Smolarkiewicz, 1999: CRCP: A Cloud Resolving Convection Parameterization for Modeling the Tropical Convecting Atmosphere. *Physica D*, **133**, 171−−178.

Grabowski, W. W., and P. K. Smolarkiewicz, 2002: A multiscale anelastic model for meteorological research. *Mon. Wea. Rev*., **130**, 939−956.

Hammond, S. W., R. D. Loft, J. M. Dennis, and R. K. Sato: 1995, Implementation and Performance Issues of a Massively Parallel Atmospheric Model. *Parallel Computing***, 21** pp. 1593−1619.

Hayashi, Y.−Y., and A. Sumi, 1986: The 30−40 day oscillations simulated in an ‘‘aqua planet’’ model. *J. Met. Soc. Japan*, **64**, 451−−467.

Hum, H. J.,O. Maquelin, K. B. Theobald, X. Tian, G. R. Gao, and L. J. Hendren, 1996: A study of the EARTH−MANNA multithreaded system. *International Journal of Parallel Programming*, **24,** 319−347.

Hum, H. J., O. Maquelin, K. B. Theobald, X. T., X. Tang, G.R. Gao, P. Cupryk, N. Elmasri, L. J. Hendren, A. Jimenez, S. Krishnan, A. Marquez, S. Merali, S. S. Nemawarkar, P. Panangaden, X. Xue, and Y. Zhu, 1995: A design study of the EARTH multiprocessor. *Proceedings of the IFIP WG 10.3 Working Conference on Parallel Architectures and Compilation Techniques*, (PACT ’95), 59−−68, Limassol, Cyprus.

ICPP Report, Climate Change 2001: Report of the Intergovernmental Panel on Climate Change. Eds. J. T. Houghton, Y. Ding, D. J. Griggs, M. Noguer, P. J. van der Linden, X. Dai, K. Maskell, and C. A. Johnson, Cambridge University Press, United Kingdom and New York, USA, 88, pp.

Jacquet, A., V. Janot, C. Leung, G. R. Gao, R. Govindarajan, and T. L. Sterling, 2003. An executable analytical performance evaluation approach for early performance prediction. *Proceedings of the Workshop on Massively Parallel Processing*, Nice, France, Apr.

Jakob, R., 1993: *Fast and parallel spectral transform algorithms for global shallow water models.* Cooperative Ph.D. thesis 144, University of Colorado and National Center for Atmospheric Research, 127 pp. [Available from University Microfilm, 305 N. Zeeb Rd., Ann Arbor MI 48106.]

Kakulavarapu, P: 1999: *Dynamic load balancing issues in the EARTH runtime system.* Masters thesis, McGill University, Montreal, Quebec, Dec. 1999

Khairoutdinov, M. F., and D. A. Randall, 2001: A cloud resolving model as a cloud parameterization in the NCAR Community Climate System Model: Preliminary results. *Geophys. Res. Let.*, **28**, 3617−3620.

Koshyk, J. N., K. Hamilton, J. D. Mahlman, 1999: *Geophys. Res. Lett*., **26**, 843−846.

Lilly, D. K., 1983: Stratified turbulence and the mesoscale variability of the atmosphere. *J. Atmos. Sci.*,

40, 749−761.

Loft, R. D., S. J. Thomas, J. M. Dennis, 2001: Terascale Spectral Element Dynamical Core for Atmospheric General Circulation Models. *Proceedings of the Supercomputing 2001 Conference*. ACM and IEEE Computer Society, CD−ROM.

Morrone, C. J., 2001: *An EARTH runtime system for multi−processor multi−node Beowulf clusters.* Masters thesis, University of Delaware, Newark, DE, Spring 2001

Najjar, W. A., E. A. Lee and G. R. Gao, 1999: Advances in dataflow computational model *Parallel Computing*.

Rivier, L., R.D. Loft, and L.M. Polvani, 2002: An efficient spectral dynamical core for distributed memory computers. *Mon. Wea. Rev.*, **130**, 1384−1396.

Slingo, J., and Coauthors, 1996: Intraseasonal oscillations in 15 atmospheric general circulation models: results from an AMIP diagnostic subproject. *Climate Dyn.*, **12**, 325−357.

Slingo, J., M. Blackburn, A. Betts, R. Brugge, K. Hodges, B. Hoskins, M. Miller, L. Steenman−Clark, and J. Thuburn, 1994: Mean climate and transience in the tropics of the UGAMP GCM: Sensitivity to convective parameterization. *Quart. J. Roy. Met. Soc.*, **120**, 881−922.

Sterling, T., L. Bergman, 1999: *A Design Analysis* of a *Hybrid Technology Multi−Threaded Architecture for Petaflops Scale Computing,* International Conference on Supercomputing (ICS99), Rhodes, Greece.

Theobald, K. B., 1999: *EARTH: An efficient architecture for running threads*. PhD thesis, McGill University, Montreal, Quebec.

Theobald, K. B., G. Agrawal, R. Kumar, G. Heber, G. R. Gao, P. Stodghill, and K. Pingali. Landing CG on EARTH: A case study of fine−grained multithreading on an evolutionary path. *Proceedings of the Supercomputing Conference (SC−2000)*, Dallas, TX, Nov. 2000.

Thomas, S. J., J. M. Dennis, H. M. Tufo, and P. F. Fischer, 2002: *A Schwarz Preconditioner for the Cubed−Sphere*, Selected Proceedings of the 2002 Copper Mountain Conference on Iterative Methods, SIAM J. on Scientific Computing, to appear.

Thulasiraman, P, K. Theobald, A. Khokhar, G. Gao, 2000: Multithreaded algorithms for the Fast Fourier Transform. *Proceedings of the Symposium on Parallel Algorithms and Architectures (SPAA)*, June 2000, Bar Harbor, Maine.

Tremblay, G., K. B. Theobald, C. J. Morrone, M. D. Butala, J. N. Amaral and G. R. Gao, 2000: Threaded−C Language Reference Manual (Release 2.0), CAPSL Technical Memo 39, Department of Electrical and Computer Engineering, University of Delaware, Newark, Delaware. ftp://ftp.capsl.udel.edu/pub/doc/memos.

Tullsen, D. M., S. J. Eggers and H. M. Levy, 1995: Simultaneous multithreading: Maximizing on−chip parallelism. *Proc. of the 22nd International Symposium on Architecture*, 392−403, Santa Margherita Ligure, Italy, May 1995.

Van Zandt, T. E., 1982: A universal spectrum of buoyancy waves in the atmosphere. *Geophys. Res. Lett.*, **9**, 575−578.

Washington, W. M., J. M. Weatherly, G. A. Meehl, Semtner, A. J., et al.: 2000, Parallel climate model (PCM) control and transient simulations. *Climate Dynamics* 16:755−774.