Table Of Contents

Contents

(I) Science, Technology and Management

1 Introduction

This is the second of three linked proposals to develop a 'framework' to support a wide variety of Earth system models. The framework - called ESMF (Earth System Modeling Framework) - will allow models to interoperate and share components, improve their portability and ease of use, enable them to exploit rapidly evolving computer technology, and insulate them from changes in technology. The ESMF development project consists of three parts. Part I implements the core software that will make up the framework itself. Part II, the subject of this proposal, integrates major existing applications into the framework, as the framework itself evolves. Part III implements data assimilation systems under the framework. Parts II (and III) will provide vital feedback to the core framework design and development effort and enable it to be tested and refined on an ongoing basis.

Here, in part II, we propose to aggressively adopt the evolving ESMF infrastructure in several major US Earth system modeling suites. We will adapt components of the Goddard Earth Modeling System[25, 29] (**GEMS**), the GFDL Flexible Modeling System[11] (**FMS**), the MIT Universal General Circulation Model[21, 20, 13] (**MITgcm**), the NCAR Community Climate System Model[17, 3] (**CCSM**) and the NCEP Forecast Suite[16, 18] (**NCEP**) to operate under ESMF. These represent a broad range of models, with differing algorithms and implementation methodologies, ensuring that ESMF can provide support for many, if not all, of the models that are used today in NASA Earth Sciences programs.

The milestones set out here are designed to achieve an orderly transition to ESMF for the above models and demonstrate the benefits of a common infrastructure to the wider community. One important element of our proposal is that we wish to demonstrate that ESMF will not only yield benefits in standard scenarios, but permit complex Earth system models to exploit commodity cluster systems that, because of their low cost, could become ubiquitous and so complement our centralized computer facilities.

We begin, in section 2, by describing the ESMF in broad outline. We go on in section 3 to briefly summarize the reference modeling systems that we propose to implement under ESMF. To set the scene for the plan of work, in section 4 we discuss key concepts in the design of the ESMF (more fully described in Part I). In section 5 the detailed work items are set out.

2 The Earth System Modeling Framework

Richardson [28] proposed that a computation harnessing thousands of human calculators could in principle be used for prediction of the weather. With the advent of modern computers in the 1950's, Charney and collaborators [7] showed that such calculations were actually feasible, laying the groundwork for numerical weather prediction and the use of models in the analysis of data. Subsequently Manabe [19] and others extended the application of models to climatic time scales, and Bryan to the ocean [6]. Today coupled climate models have a vast range of components, and include representations of the atmosphere, ocean, cryosphere, land surface, biosphere, etc. and their interactions. Such models are very complex tools, difficult to build, maintain and use. To further complicate matters there have been, and will continue to be, major changes in computer architecture. We now have available distributed memory architectures, or a mix of distributed and shared memory programming models in the same computer and application. This has placed a huge burden on the already strained software development efforts of our modeling groups.

One of the great strengths of atmospheric, oceanic and climate modeling in the US, is the variety, availability and wide use of models. But this diversity has also led to duplication of effort, a proliferation of models and codes which, due largely to technical reasons, cannot interoperate and that have been unable to keep up with and exploit advances in computing technology. The issues are discussed at length in the NRC report 'Capacity of US Climate Modeling' [8].

Earth system modelers are keenly aware of the need for greater uniformity and interoperability of their codes. Early attempts, by Kalnay and collaborators [15], at a partial solution were limited to 'plug-compatible' rules for single-column physics parameterizations in atmospheric models. It is now recognized that interoperability needs to include dynamical cores and indeed all components of climate models. The old approach, in which each center develops its own solutions, is not just preventing the interchange of scientific codes, but is slowing down progress and becoming prohibitively expensive.

Our goal, then, is to develop a custom software environment - a framework - under which Earth system models can be developed, interact with one another, and execute. The framework provides the substrate on which the 'components' that make up a model, or group of models, are implemented, and through which the components interact. Components that are implemented to fit within the framework can take advantage of the functions that it supports.

2.1 Enhancing Scientific Interchange

The process by which climate system modeling innovations are migrated from their source to other modeling centers is at present slow and tortuous. In general, climate system models have not been designed using good software engineering practices, primarily because the scientists in charge of their development have neither the expertise nor the time to apply such practices. The result has been codes that tend to be tightly tied to local design choices, which may depend heavily both on the underlying hardware and on uninformed choices about software issues. Codes describing scientifically separable pieces of climate system models are often hopelessly entangled with other parts of the model. It has almost always been deemed more efficient for a modeling institution to recode scientifically innovative developments from other centers. In fact, import of new algorithms is most often accomplished by developing code from a text description of the algorithm without even attempting to directly modify the code from the developing institution. This labor intensive process of recoding from scratch has served to stifle the transfer of ideas among modeling centers; the cost to investigate new algorithms is simply too high unless the perceived benefits are enormous.

The ESMF framework proposed here will open the doors to an entirely new method for collaboration between centers and will also facilitate the participation of smaller development groups who are interested in particular components of climate system models. A community using climate models built on the ESMF framework will be able to exchange codes for scientific innovations directly, avoiding the costly recoding step described above. Because details of the underlying hardware are abstracted out of the scientifically interesting code, even modifications needed to enhance performance on different architectures should be minimal compared to the burdensome recoding required at present. As a result, modeling institutions should be able to investigate the impacts of many more scientifically interesting algorithms. This will enhance the exchange of ideas between modeling centers, and should significantly accelerate the pace of improvements in climate system models.

2.2 Prototype Framework Systems

The proposed ESMF will be built from scratch, but exploiting ideas from GEMS, FMS and WRAPPER, the frameworks developed at, respectively, NASA, GFDL and MIT.

GEMS [29] is the modeling framework used at NASA's Seasonal-to-Interannual Prediction Project (NSIPP) [24] to run its forecast and ocean data assimilation systems. Like the proposed ESMF, it uses an object oriented design implemented as a series of functional layers that handle tasks ranging from low-level communication services, to grid definitions and transformations between grids (but in two dimensions), to coordination of high-level coupling between the major computational components. All global models presently in use at GFDL are constructed around a modeling framework that is part of the Flexible Modeling System [11](FMS). FMS is a layered framework that is quite similar to the proposed ESMF, with separate software layers that provide low-level utilities, grid abstraction, and coupling. The MIT WRAPPER software [13] provides a hardware-independent computing environment within which models execute. It is focused on exploiting open-source based commodity cluster platforms and demonstrating their potential for providing abundant low-cost, high-performance resources to the research community [14]. As with the proposed ESMF, support for scalable parallel communication needed by domain decomposition is included within WRAPPER, along with support for component-to-component data transfers and multi-component coordination.

All three systems allow numerical codes to remain in their native implementation language and to continue to use native code for most numerical operations. This approach has proved very successful at ensuring strong performance is preserved for components that are already highly efficient. The proposed ESMF will use a similar architecture. The new framework architecture will also draw on the lessons learned from these proven prototypes to ensure that it is scalable, efficient and portable.

2.3 Support For A Diverse Community

This proposal, if we are successful, will enable the community to harness the talents and ideas of **all** who work in Earth system science, because it provides a flexible mechanism whereby ideas 'out in the field' can be channeled from the base through to the apex, where dedicated modeling groups have access to terascale facilities. Hitherto, this conduit has been blocked both by software problems and the high cost of computing resources which has led them to be centralized. But the technological landscape has changed. Raw computing resources are now ubiquitous, if only we could harness them for Earth science applications. There is no reason why our models cannot be coded so that they can exploit everything from general purpose microprocessor based machines with deep memory hierarchies to highly pipelined special purpose vector systems. The ESMF aims to facilitate this.

New technologies - high performance microprocessors, along with the low overhead, high bandwidth interconnects and software that can exploit them - now make possible a distributed climate modeling activity exploiting 'personal supercomputers'. To give an example of what can be done, the scaling curves in fig. 1 include the performance of MITgcm on a commodity cluster system. This system uses a framework to run the MIT model on a cluster of Intel PC's with high speed interconnect. Sustained performance of 2.1GFlop/s or $35K per sustained gigaflop/s, can be achieved on this thirty-two processor system [14]. This should be compared to the $250K per sustained gigaflop/s typically obtained on more conventional platforms, and illustrated in the other three panels on figure 1.

There is no reason why, for example, a graduate student studying the effect of SST anomalies on the North Atlantic Oscillation using the NCAR community climate model, could not run that model on a small local PC cluster. A version of that same model could be used by NCAR to make climate change projections for IPCC on a teraflop machine dedicated to high end calculations. The component parts that make up the two calculations could be configured differently and target different hardware, but they would be running under the same framework.

Our aim then is to develop a technology which can support terascale simulations but, at the same time, facilitates the complementary role of smaller-scale, affordable and hence more widely accessible computing facilities.

3 The ESMF Reference Modeling Systems

3.1 NASA Modeling Systems:NSIPP

The mission of NASA's Seasonal-to-Interannual Prediction Project (NSIPP) is to develop the use of satellite data for SI prediction. NSIPP is very active in climate model and data assimilation system development. Its coupled atmosphere–ocean general circulation models and land surface model are used to assess the impact of both satellite and conventional data on SI predictions, as well as to produce near-real-time experimental predictions. All of the NSIPP codes run under the GEMS framework on Goddard's 1000-processor Cray T3E-600.

Typical NSIPP applications range from large ensembles of seasonal forecasts to century-long climate simulations with the full coupled system. The standard coupled model used in most current work consists of an atmospheric GCM run with 2 degree horizontal resolution and 34 vertical levels and an ocean GCM run at 2/3 degree horizontal with 20 layers. The land model represents continental surfaces with some 12000 'tiles', each with its own surface properties (see panel (e) in figure 3). These components are designed to run and couple using very general horizontal decompositions on distributed memory machines.

3.2 GFDL

The GFDL principally focuses on simulations of climate on various time-scales: from short-term seasonal-to-inter-annual simulations to decadal-centennial simulations at the long end. Historically, independent scientific groups working on different research problems maintained their own codes. Over the last few years, the GFDL has united all modeling development efforts around the FMS framework. This has helped to integrate and synthesize model development work across the entire lab: from the development of parametric representations of atmospheric physical processes, to their testing and validation on short time scales, to their validation in the context of coupled climate simulations of climate change. On the ocean side, the GFDL MOM now uses the FMS framework. Models currently available under the FMS umbrella include: a hydrostatic atmospheric grid-point model (BGRID); a hydrostatic atmospheric spectral model (SPECTRAL); a hydrostatic ocean grid-point model (MOM); an ocean surface and sea ice model (SIS); and a land surface processes model (LaD).

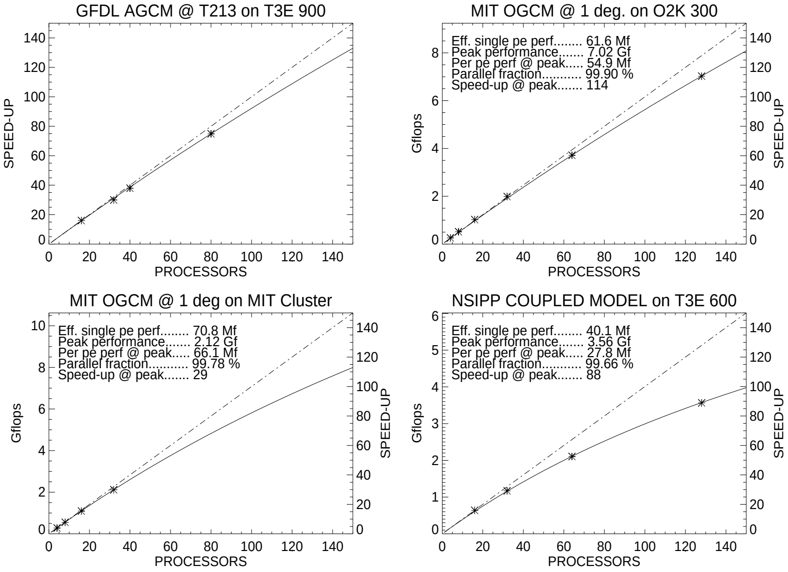

Figure 1: Scaling of some reference models. '*' indicate data points, solid line is an Amdahl curve fit to the data points. Dash-dot line indicates linear scaling based on Amdahl Curve single processor performance. Parallel fraction refers to the percentage of parallel code in the application as defined by Amdahl's Law [2]. Optional milestones (e) and (f) of this project will demonstrate cluster experiments running efficiently beyond 32-processors; the Amdahl curve for the MIT cluster indicates that this is possible, we will deploy other reference models, in addition to MITgcm, using clusters.

Current work on short-term climate evolution is based on coupled-climate models using the atmospheric grid-point model, and the decadal-centennial simulations are envisaged using the spectral model, in conjunction with the other component models listed. The ocean model is used as a component model in climate studies, as well as in solo configurations at various scales down to eddy-permitting, and coupled to simplified atmospheres, like an energy balance model. In addition, other models currently in development, as well as existing independent models, are being brought under the FMS framework: a non-hydrostatic atmospheric grid-point model (ZETANC) and an isopycnal ocean gridpoint model (HIM). The non-hydrostatic model will be used for studies of interactions of flow with topography and as a cloud-resolving model for studies of radiative-convective equilibrium.

3.3 MITgcm

MITgcm is a finite-volume code used for both limited area and planetary scale modeling of the ocean and atmosphere. It supports non-hydrostatic and hydrostatic algorithms with tangent linear and adjoint components. Isomorphisms are used to derive, from one hydrodynamical kernel, atmospheric and oceanic fluid cores. MITgcm is enveloped by a 'wrapper layer' [13], MIT's analogue of the proposed ESMF.

Current projects involve researchers working at universities, NASA labs and national supercomputing centers. They range from numerical studies of laboratory fluids to coupled paleo-climate simulations. Accompanying biogeochemical modules allow the model to be deployed for studies of ocean biological and chemical phenomena. Additionally the suite of codes forms the basis of several ongoing ocean data synthesis and analysis initiatives. In particular MITgcm is at the heart of the MIT/Scripps/JPL collaborative project, ECCO[33] (Estimation of the Circulation and Climate of the Ocean) in which a global model configuration is being constrained by global data sets to yield an estimate of the evolving state of the ocean. These initiatives involve both prognostic simulation and formal data assimilation through sequential estimation and adjoint methods. The latter approach employs an adjoint compiler (TAMC[12]) which automatically differentiates the prognostic model. The data synthesis projects make heavy use of satellite, surface and sub-surface observational data sets. The adjoint compiler is also used in formal sensitivity studies of long-time climatological ocean circulation simulations.

The prognostic model and its automatically generated adjoint counterpart run on a variety of computers. Both large MPP systems (for example the 1100 processor IBM SMP cluster system at SDSC) and vector systems (both Cray and NEC hardware) are used. Additionally, many projects also utilize dedicated low-cost single-processor and SMP clusters, as well as desktop uni-processor and multi-processor workstations.

The MITgcm code is implemented in Fortran. Some optional packages use Fortran 90 but most of the core numerics has been kept Fortran 77 compatible to guarantee portable performance and compatibility with completely open-source based systems.

3.4 NCAR Community Climate System Model (CCSM)

The Community Climate System Model (CCSM) Project is a large focused project to continue improving the CCSM and apply it to important problems in climate research. The investigators are funded or supported by NCAR, NSF, DOE, NASA and NOAA. The project provides community service in many ways, involving a broadly based group of developers, and freely providing both models and simulation results to the climate research community.

From its inception, the CCSM project has been committed to entraining a large and diverse body of researchers into both the continuing development and the application of the CCSM. The project has been remarkably successful at engaging a broad cross section of the climate modeling community in these activities. However, the scientific development of the CCSM has not been matched by progress in the design and implementation of the underlying software. Both development and application of the current CCSM remains challenging.

CSM1 was designed for shared memory architectures. It is extremely efficient on a variety of parallel vector systems and reasonably efficient on the SGI O2000, using up to 128 processors. However, the ocean model and coupler did not use message passing internally, severely limiting the scalability of the coupled model on distributed memory systems. CCSM2 will be a hybrid OpenMP/MPI application able to run on effectively almost any architecture. The scientific development of CCSM2 is expected to be completed in January 2001, but code optimization will continue for several months. The component models all have hybrid (OpenMP) MPI options available. The ocean model has been replaced by POP, LANL's distributed memory model. The sea ice model has been replaced by C-ICE, a hybrid OpenMP/MPI model also from LANL.

The CCSM atmospheric model will be adopted by the NASA Data Assimilation Office (DAO[10]) once the CCSM is running under the framework.

3.5 NCEP Forecast System

The NCEP atmospheric global forecast code is a key component of the NCEP Global Data Assimilation System that provides the backbone of all numerical weather prediction at NCEP. It also is used to make the 4 times per day 120- hour Aviation forecast, the daily 384-hour Medium Range forecast and the 22 per day 384-hour Ensemble forecasts at NCEP. This hydrostatic sigma coordinate model carries surface pressure, temperature, horizontal winds,

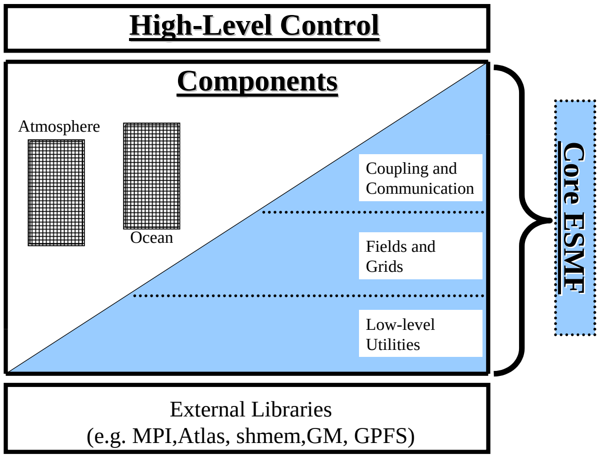

Figure 2: The elements of the proposed Earth system modeling framework. The system comprises 'components' that make up the Earth system application. A driver, in the high-level control layer, assembles the multicomponent application from component layer software. The core ESMF provides a common foundation that allows components to be developed independently of one-another and the hardware architecture. Grey shading denotes software that will be created as part of the proposed part-I ESMF effort. More details on the design of these elements can be found in the part-I proposal. Elements in the component section and example driver programs will be provided by the part-II and part-III efforts.

moisture, ozone and other tracers as its prognostic variables. The code uses the spectral transform method to compute horizontal derivatives, to solve the semi- implicit Helmholtz equation, and to apply subscale horizontal diffusion. The physical grid is used to compute single column physics, including clouds, solar radiation, longwave radiation, gravity wave drag, surface layer exchanges, planetary boundary layer vertical mixing, shallow convection, deep convection, large-scale condensation, and ozone chemistry. The code uses the transpose strategy to distribute data and work across processors. The entire model state is transposed several times every timestep, which for the operational T170 L42 resolution is 450 seconds. The code currently runs on an IBM SP using MPI for communications.

4 Framework Concepts

In order to set out the plan of work in an intelligible way, we briefly describe how the applications discussed in section 3 fit within the framework. Figure 2 shows the envisaged overall system architecture. The figure includes both numerical "components" and core framework software which is used by the components. Core framework software is shaded plain grey while components are shown with hatched shading. Components operate under the direction of a high-level control program, indicated by the box at the top of figure 2. The role of the high-level control program is to orchestrate an overall sequence of computation. The elements of figure 2 are discussed in more detail below.

4.1 Components

The Earth system application is made up of 'components'. For example, in figure 2, one component is simulating the ocean and another the atmosphere, indicated by the hatched boxes. Components can be smaller than a full model; e.g. an atmospheric physics package. All components, whether large or small, have "fields" of physical data on which they operate. For example an atmospheric physics component acts on a field representing atmospheric water vapor content. The physical fields are associated with a grid, which describes the coordinates of the field data, and a decomposition, which specifies how the grid has been subdivided to support parallelism.

We find it useful to introduce the following notation: a component, C, acts on field F on grid G, with decomposition D.

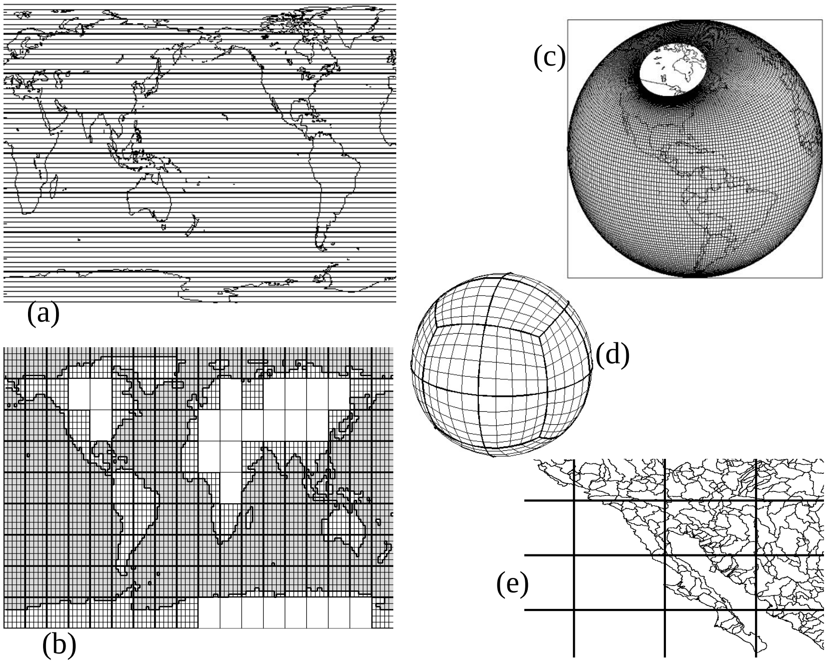

Figure 3: The ESMF will support grids such as those shown here and the parallel decompositions of those grids. The grids shown represent (a) gridding in a spectral model, (b) a regular lat-lon grid with land areas masked, (c) a stretched and rotated lat-lon grid with a displaced pole, (d) a cube projected onto the sphere, (e) a land-surface grid (in this case for south-west North America). Regular cartesian grids and regional variants will also be supported as well as unstructured observational grids. For each grid parallel primitives and coupling primitives supporting appropriate domain decompositions will be included in the framework.

4.2 Fields and Grids

For each component $C_i(i = 1, 2, 3, ...)$ the fields and grids layer of the framework maintains information about the grid(s) $G_1, G_2, G_3, ...$ used by that component and information about grid decompositions $D_1, D_2, D_3, ...$ used for parallel computation. In its most general mode of operation, the framework will provide support for interoperation and scalability by providing high-performance functions that map between a field, or group of fields, F(on grid G using decomposition D in component C) to field F' (on grid G' using decomposition D' in component C'), interpolating data between grids and transforming data between decompositions as necessary.

Figure 3 illustrates some of the grids and decompositions used by the reference models, The "fields and grids" layer of the framework will hold information and support manipulation of fields associated with and between the grids in the figure.

Within the framework two classes of mapping are supported. The first performs **intracomponent communication** (mapping within a single component) whilst the second performs mapping between two different components, **coupling**. The distinction is made because **intracomponent communication** has a profound impact on individual component scalability, while **coupling** concerns both interoperability and scaling.

4.2.1 Intracomponent Communication

Each of the reference modeling systems in section 3 currently contains a set of parallel primitives that are used by individual models within those systems. For grid-point models these primitives typically update the "ghost-zones" surrounding decomposed domain interiors, modifying a field, F, (or a group of several related fields) so that

$$F \mapsto F'$$

with the decomposition D, grid G and component C held fixed. For regular latitude-longitude grids the mapping operator, $\mapsto$, involves simple data copies from the interior section over a decomposed region to the overlap section(s) of neighboring region(s). More complex gridding schemes may require that the $\mapsto$ operator perform small transformations on the data. For example the cubed sphere requires a rotation for points at the cube faces. The framework will include “ghost-zone” mapping schemes, including appropriate transformations, for all the grids in figure 3 that are grid-point based.

For spectral models additional primitives are needed to transpose between the different fourier spaces and between grid-point and spectral space. These operations require a more general mapping operator such that

$$(F, D, G) \mapsto (F', D', G')$$

Some models employ redistributions of data to adaptively load-balance. This operation employs a mapping operator on a decomposition, D, such that

$$D \mapsto D'$$

In all cases, under ESMF, components will integrate forward as a series of compute and communicate phases. The communication phases, in the form of a series of mapping operators, will be supported within the core framework.

4.2.2 Coupling

The coupling transformations that are supported under the ESMF are a superset of the intracomponent communication primitives described in section 4.2.1. All couplings involve mapping between a component, C, and a second component, C'. For the case of data transfer between aligned decomposed grids the coupling method implements mappings of the form

$$(F, C) \mapsto (F', C').$$

Couplings of the form

$$(F, C, D) \mapsto (F', C', D')$$

that transfer data between identical but non-aligned grids are also supported. Transfers between dissimilar grids can be supported through multi-stage mappings such as

$$(F, D, G, C) \mapsto (F', D', G', C) \mapsto (F'', D'', G', C').$$

Some default multi-stage mappings will be included in the core framework. However, it will also be possible to create custom mappings within existing components. It is also possible to create a component that only performs mapping functions and that can serve as an intermediary between two other components.

4.3 High-level control

The high-level control program is responsible for orchestrating overall sequences of component calculation and component-to-component data exchange. This top-most, driver layer, is the highest level element in the ESMF architecture. The specification of a particular driver program is one part of the process of defining a numerical experiment. In other words, the driver layer design defined by the ESMF will specify how a high-level control program is written but not its content.

A simple ESMF compliant driver is that of a single component, for example a stand-alone ocean model. In this case the driver consists of a sequence of code that first requests the creation and initialization of a single realization of the appropriate component. The driver then instructs the component to advance its state by a certain number of steps. Finally the driver requests termination and clean-up.

4.4 Low-level utilities

The final element of the core framework is the low-level utilities layer. This layer contains sets of functions that support interoperability, but that are only indirectly involved in parallel data transfers within and between components. The low-level utilities include calendar and alarm functions, which ensure that different components have a consistent view of time. There are also functions that support disk I/O. These can be used, together with event signaling functions, to, for example, ensure that multi-component calculations can be suspended and subsequently restarted. Functions that support interrogation of parameter settings of one component by another component are also provided, as are functions supporting standard diagnostics and functions supporting performance monitoring.

5 Work Items

The project we are proposing involves much more than adapting existing models to fit within the framework. There will be a close partnership with the Part-I core framework development team in defining and verifying the detailed design and scope of the core framework

Developing and making use of the core ESMF involves transferring logic and algorithms out of the reference models into the framework. Each of the reference models, as they exist at present, employs functions somewhat analogous to the ESMF elements in figure 2. The reference models also contain counterparts of the distinct field, grid and decomposition entities discussed in section 4. Over the course of this project, as the core ESMF evolves, the individual modeling systems will devolve to the framework the responsibility for the ESMF elements identified in figure 2. This will require adapting the reference models to the formalized structure shown in the figure, a structure that is already employed informally by many Earth science models, including the reference models. However, to fit properly within an ESMF container, as an ESMF component, appropriate interfaces between the existing elements of the reference models that correspond to the formal ESMF architecture will need to be defined.

5.1 Defining the ESMF

The early stages of this project will first require creating precise definitions of the functions that are embedded within the reference modeling systems at present. This work contributes primarily to the **Milestone H** target.

5.1.1 Grids

In figure 2 functionality is split between the driver, the components and the core ESMF. As described in section 4 the core ESMF provides support for standardized grids in the form of "grid-classes", G_1, G_2, G_3. The set of grid-classes that the ESMF will initially support will be driven by the reference modeling system requirements. They will include latitude-longitude grids, regional and nested grids, spectral grids and cubic and icosahedral projections onto the spheres, as shown in figure 3. Additionally vector-topology grids are employed in land-surface components of the reference models and unstructured grids are used in handling of raw observational data streams. We will work closely with the Part-I core ESMF team in the detailed definition, development and testing of the grid classes in the core ESMF.

5.1.2 Fields

As discussed in section 4, the framework performs a variety of transformations on individual fields, and groups of fields. These same fields are also manipulated within components. We will work with the core ESMF team to define and test mechanisms that allow both the core framework and component models to share common field data structures.

5.1.3 Decomposition

The ESMF provides support for scalable parallelism through domain decomposition. Under ESMF a decomposition, D, comprises two parts. One part, D_g, specifies how the mathematical grid has been decomposed into different subdomains. The other part, D_m, specifies a "machine model" which stores information about how decomposed subdomains are "bound" to underlying system software and/or hardware and how subdomains communicate with one another.

Adapting the reference models to the ESMF decomposition architecture will require implementing the models existing performance critical parallel primitives on top of the ESMF D_g, D_m decomposition scheme. The D_m support that the reference models will require includes ESS testbed support (in which D_m uses MPI[23] and shmem[4]), IBM SP support (D_m uses MPI and IPC shared memory[9]), SGI Origin support (D_m uses MPI, shmem, IPC), commodity clusters support (D_m uses MPI, Myrinet-GM[5]). Here again the developer team of this proposal will work with part-I core ESMF developers to define and test the ESMF decomposition classes.

5.1.4 Communication

Functions for performing many of the mapping operations described in section 4 already exist in the reference models. The mapping operations in the reference models need to be precisely defined and framework methods developed that support these operations. We will again partner with the core ESMF team to define and test the suite of ESMF mapping operators.

5.2 Integration of models into ESMF

Integration of, for example, GFDL's models into ESMF involves swapping FMS (see section 3) software for ESMF in the appropriate GFDL component or coupled models. Low-level utilities will be tested primarily in the context of global spectral and grid-point atmospheric GCMs. Grids and fields level software will be tested in these atmospheric models and also in the MOM ocean model. Finally, coupling level software will be tested in global coupled models comprised of GFDL atmospheric GCMs, MOM, and GFDL land and ice models. Performance and capability of the ESMF software will be compared directly to corresponding performance using FMS and deficiencies of the ESMF prototype versions will be identified to guide further development. The project targets **Milestone E,F and G** provide application level performance feedback to the ESMF core team.

An exactly equivalent process will be carried out for all the reference models set out in section 3. The reference models will be modified so that they can be controlled from an ESMF driver in both single model scenarios and in complex coupled model scenarios. This work will proceed in two stages.

5.2.1 Kernel Tests

In an initial phase a series of code kernels will be selected and extracted from the reference models. This work will span project targets **Milestone H and I**. The Part I and Part II groups will work together to ensure that these tests are manageable pieces of software but are nevertheless representative of the needs of the full modeling systems. For example the custom Helmholtz solver from the NCEP model or the MITgcm might be extracted to form a stand alone kernel. Working with the core ESMF design and implementation team (from the Part I proposal), participants in this proposal (Part II) will then examine the compatibility of each kernel with the framework. This effort will verify the ability of the ESMF support layer to handle the full mix of codes that will be put forward. The kernels will both provide tests of functionality and be tools for understanding scaling and performance. Particular emphasis will be placed on kernel tests that exercise the decomposition support layer parallel operations. Providing a general support layer for these operations, that can work with all the reference models, will present the biggest challenge. Extensive testing will be performed to ensure that adequate provision for these operations is made in the support layer. The parallel operation kernels defined here will become part of a set of standard tests that will be available to validate ESMF installations and to provide examples of ESMF support layer usage.

5.2.2 Driver Tests

In addition to tests aimed at validating support layer functionality, tests will be required to ensure appropriate driver layer support for the reference models. As with the support layer, testing will focus on kernel codes extracted from the reference models. In this case the kernels will be used to define initialization interfaces and a set of state-advancing interfaces that are compatible with the driver layer and the reference models. In addition to testing with a single kernel, the use of the driver layer to compose a set of ensemble experiments using single models will also be exercised. The kernel codes developed here will also be folded into a set standard tests for ESMF validation and demonstration. This work will contribute to project targets **Milestone H and I**.

5.2.3 Full Model Tests

The unit testing described in sections 5.2.1 and 5.2.2 will identify places in the reference models where changes are required in order to transform the models into ESMF components. Under project targets **Milestone I and J** functions will migrate from reference models to the core framework. In particular some functions that exist within the reference models will be implemented and tested in the support layer, for example the halo update parallel primitive. Once it has been demonstrated that these functions are working correctly in the ESMF, the reference models will be changed to use those functions in place on their own native forms. In some reference models this will be achieved initially by creating a “shell” routine that maintains the historical interface, while other reference models will be altered to use the ESMF functions directly.

A technical milestone that we will monitor during this phase is the degree of overlap in support code used by the components derived from the reference models. The reference models currently have elements of the ESMF support layer built-in. It is anticipated that when these functions have been separated out from the models, not only will the resulting gridded-components be reduced in complexity but many of the gridded-components will be able to use the same pieces of ESMF support layer code.

5.3 Interoperability Experiments

The full-blown ESMF components, derived from the reference models, will be employed in a series of interoperability "tests", which will contribute to project targets **Milestone I and J**. The first tests will be simple experiments involving verifying computational interoperability. These tests will verify that reference models from different institutions can indeed exchange data and integrate forward as a coupled application. Once this has been shown we will perform a number of experiments using this new capability.

5.4 Performance, portability and scalability experiments

A major challenge of framework based systems is avoiding heavy performance overheads. Our architecture allows numerical codes to remain in their native implementation language and to continue to use native code for most numerical operations. This should ensure that strong performance will be preserved for codes that are already highly efficient. However, the framework will need to be scalable at both the driver level and at the support layer level. Performance analysis of both parts will be carried out in detail using both kernel test codes and full components. The current performance of a sampling of the reference models is shown if figure 1.

5.4.1 Affordable commodity systems

For project targets **Optional Milestones e and f**, we propose to target a cluster system at M.I.T, using commodity hardware, to examine and demonstrate the capacity of cluster computers to provide rounded support for large scale modeling under ESMF. This will also serve as a valuable stress-test of the ESMF. Getting the framework, including component models, to work efficiently and robustly in a commodity based cluster environment will ensure that the software developed is indeed truly portable and will be adaptable to future technologies.

This effort will include looking at the I/O requirements of the reference models, developing detailed baseline software and hardware configurations that are suitable for reference models. Emphasis here will be placed on producing a complete model of affordable hardware and software that can readily be replicated by other groups who are not necessarily as experienced in computer science as the PI team. We anticipate that this work will have significant impact on the ability of research groups around the US to deploy models locally and so that those groups can become more directly involved in the improvement and enhancement of future generation Earth system models.

6 Management Plan

6.1 Investigator Team

The Principal Investigator of the proposal is **John Marshall**, a professor in the Department of Earth, Atmospheric and Planetary Sciences at the Massachusetts Institute of Technology. Professor Marshall and his group have long experience in developing and using climate models, working in close collaboration with computer scientists - see [14, 13, 30]. He leads the climate modeling team at MIT.

The Co-Investigators are: **Jeffrey Anderson**, Head, Experimental Prediction Group, NOAA Geophysical Fluid Dynamics Laboratory; **Byron Boville**, Senior Scientist and Head, Climate Modeling Section, NCAR; **Cecelia DeLuca**, Software Engineer, Scientific Computing Division, NCAR; **Chris Hill**, Research Engineer, Massachusetts Institute of Technology **Philip Jones**, Staff member, Theoretical Fluid Dynamics Group at Los Alamos National Laboratory **Stephen Lord**, Director, Environmental Modeling Center, National Centers for Environmental Prediction; and **Max Suarez**, NASA Seasonal to Interannual Prediction Project; NASA/GSFC.

6.1.1 Qualifications

Our Investigator Team possesses the combination of skills and backgrounds necessary to help construct and implement key applications under, the ESMF and establish it as a standard throughout the climate and weather communities. The most critical of these are:

Expert understanding of the way in which Earth science models and model components are constructed and combined to support scientific research. The PI and Co-I's have many years of experience, and a proven track record, in the development and use of numerical models in meteorology, oceanography and climate science.
Previous experience creating parallel, high-performance software frameworks and toolkits. Dr. Anderson was a co-developer of the Flexible Modeling System (FMS) framework at GFDL and Dr. Suarez a co-developer of the Goddard Earth Modeling System (GEMS), both frameworks for coupling components of general circulation and climate models. These frameworks have acquired user bases within their institutions. Ms. DeLuca was a co-developer of the STAPL framework for real-time signal processing at MIT Lincoln Laboratory, a software tool currently used in multiple military radar systems. Chris Hill developed MIT's WRAPPER, the framework employed by users of the MITgcm at Scripps, JPL, MIT and Woods Hole.

Experience distributing, maintaining and providing support for software used by a large, open community. Dr. Boville was a founder and is a co-chair of the Community Climate System Model (CCSM) project. The CCSM is distributed to and supports hundreds of researchers internationally. Dr Anderson is a key scientist at GFDL developing models that are used by investigators throughout the world.
Experience in technical management and development in a CMM Level II+ environment. Most of the Investigators in this project include technical management as part of their job. However, the small research-oriented groups common in academic environments and federal laboratories often do not employ systematic software engineering practices. Two co-I's do, however, have significant experience in large-scale, mission critical software development. Ms. DeLuca served as the technical lead for new development on the STAPL framework project at MIT Lincoln Laboratory. This large, DoD-sponsored project employed about 15 FTE software engineers over the course of about 4 years, and was targeted at CMM Level III. Chris Hill has considerable past experience coordinating technical software development practices to comply with the European ISO9001 standard, a standard that is approximately equivalent to CMM Level III practices.
Proven skill in constructing algorithms and data structures to enable very high-performance

codes. All of our investigators have experience constructing high-performance codes on parallel platforms. Groups led by the PI and Co-I's have made numerous contributions to the algorithms used in earth system models.

Extensive contacts to promote the ESMF. Our Investigator Team is uniquely qualified to promote the ESMF. Established FMS and GEMS users at GFDL and NASA/GSFC-DAO, respectively, are accustomed to working with a framework and can readily be converted to the ESMF. Hundreds of users of the NCAR Community Climate System Model will be introduced to the ESMF through the yearly CCSM Workshop, which has grown into an international forum for the discussion of issues relating to climate modeling.

6.2 Management Structure

As Principal Investigator, Professor John Marshall will serve as the primary contact and administrator of the proposed work. Professor Marshall will negotiate agreements with NASA and among Investigator Team members, and will arrange for disbursement of funds after payment. He will supervise the overall activities of the Investigator Team and promote the ESMF project to the wider community.

The PI's are members of the core framework oversight team outlined in part I. We will participate in requirement analysis, design reviews, prototyping and testing of our kernel applications over the course of the project. A progress team comprising Marshall, Suarez and Anderson will ensure that the milestones are met, i.e. that progress in the implementation of our applications under framework proceeds according to plan.

Day to day aspects of our project will be coordinated by Chris Hill, a software engineer working at MIT. He will be responsible for establishing and maintaining appropriate software practices, drafting an Implementation Plan for this proposal with input from oversight teams, tracking the progress of the Development Team against the Implementation Plan and revising the Plan as needed, and verifying that unit and system tests are successfully passed.

6.3 Resources sought

We are requesting support for one software engineer to be located in each of the groups whose models are to be implemented under the framework, together with part-support for Chris Hill, the project manager working at MIT. These individuals, working at the PI institutions and in collaboration with the PI and Co-I's, will be responsible for carrying out the work set out in section 5. They will be partnered with developers working on the core framework.

One important aspect of this proposal is to demonstrate the performance of the models running under the framework on a variety of platforms including clusters of commodity hardware. We are therefore requesting support for the purchase of high-performance networking hardware to create a cluster system suitable for use with the framework and component models. A prototype cluster is already in operation at MIT, from which we will build. A high-speed interconnect developed by Arvind at MIT is used at present -see [14], but we anticipate employing Myrinet hardware for this work.

(II) Software Engineering Plan

In this section we present a software engineering plan for the overall ESMF, including software team structure and management, a software process that extends from initial specification through production release and maintenance, and tools and techniques to support development, collaboration, and distribution. Issues primarily relating to specific work items, such as strategies for identifying framework functions in existing codes and migrating code to the framework are described in section 5.

1 Software Teams and Management

For this project a *Development Team* will be established consisting of six software engineers, one located at each modeling institution (GSFC, GFDL, LANL, MIT, NCAR and NCEP). The recruitment or assignment of staff to these positions will be the responsibility of the co-I at the institution. Once established the development team will be coordinated through a development team manager located at MIT as described in section 6.2.

1.1 Progress Planning and Tracking

At the start of each year, development team members will produce a staged, twelve month plan, for reaching the project milestone goals during that year. These plans will be developed in discussion with the core framework *implementation team* of part-I. The plans will take the form of targets, each less than one month apart. Once the targets are agreed, progress through the targets will be tracked and posted so that all participants are aware of the project state. Changes to the plans will also be posted, along with explanations. Initial plans and subsequent changes will be approved by the development team manager.

A brief monthly progress digest will be produced by one member of the development team each month. The responsibility for producing the digest will rotate around team members. The digest will be circulated to all participants.

1.2 Coordination with the core framework effort

Development team members will attend part-I *implementation team* quarterly meetings and will be expected to report on progress during those meetings. In addition development team members will work actively with part-I team members during both initial design and during development and test phase.

An on-line archive will be estalished to support design phase document development and subsequent revision. The part-I core framework defines a set of documents that will be developed. The part-II development team will be involved in the review of those documents and will maintian documents summarising the relationship between the core framework elements and reference models.

The overall software development process will be aligned closely with the process employed by the part-I and part-III teams. Table 1 shows the progression of events in the core ESMF sofware development. The initial set of events, labelled “ESMF Definition” is focused on specifying the ESMF system and procedures as a whole. The second group of events, “Class Implementation” describes the development steps applied to individual software classes. As classes are completed they will be integrated into an evolving prototype of the ESMF. The final development stage, “Integration and Distribution”, involves the integration of classes leading to a software release. The ESMF will have three major software releases, corresponding to project milestones; smaller releases and demonstrations will be scheduled to insure that the project is on track. The annual plans produced by the development team in this project will be coordinated with the timeline of the core ESMF development path.

Core ESMF Software Event Progression

Event	*Product*	*Completion Gate*
ESMF DEFINITION		
Requirements specification Outlines ESMF functional scope and requirements.	Requirements Document *Prepared by:* all collaborators	Document review *Reviewed by:* all collaborators
Architectural description Describes layering strategy, function and interaction of major components.	Architecture Document *Prepared by:* all collaborators	Document review *Reviewed by:* all collaborators
Software process definition Evolving documentation describing software procedures.	Developer's Guide Document *Prepared by:* integrator	Document review *Reviewed by:* software mgr.
Implementation study Assesses existing software, optimal language, threading strategy, more.	Implementation Report *Prepared by:* implementation team	Document review *Reviewed by:* all collaborators
Software implementation and test plan Plan for Implementation and testing based on class dependencies, milestones.	Software Impl. and Test Plan *Prepared by:* software mgr.	Plan review *Reviewed by:* all collaborators
CLASS IMPLEMENTATION		
Class design Includes requirements, function, and interface specification.	Class Design Document *Prepared by:* class developer(s)	Design review *Reviewed by:* Oversight Team, software mgr.
Class implementation A class may be stubbed or partially implemented for a given release.	Prototype code *Prepared by:* class developer(s)	Code review *Reviewed by:* Oversight Team, software mgr.
Class unit test Class is tested stand-alone with a variety of inputs.	Unit test code *Prepared and tested by:* class developer(s)	Unit test *Verified by:* software mgr.
INTEGRATION AND DISTRIBUTION		
Class integration Unit tested class is integrated into an evolving prototype of the ESMF.	ESMF system prototype *Prepared by:* class developer(s), integrator	System test and benchmarking *Verified by:* software mgr.
User documentation updated Class design documentation is updated and converted to user documentation.	User's Guide & Reference *Prepared by:* class developer(s), integrator	Review before software release *Reviewed by:* software mgr.
System release Code and documentation is released. Defects and requests for features are tracked and incorporated into future releases.	System test *Prepared by:* integrator, software mgr.	ESMF system release *Evaluated by:* ESS Project, user community

2 Software Process

The Development Team will follow a software process commensurate with CMM Level II [26, 27]. The process will include many of the procedures recommended by the Software Best Practices Initiative [1]. Documents and reviews will be simplified versions of those described in standard references [22, 31, 32]. We will aim for an effective process free of extraneous overhead.

Documents: We will structure design documents so they can be easily converted to user documentation. All documents will be prepared in a format that easily generates both hardcopy and web-friendly html. We intend to use a documentation generation tool such as doctext to automatically create and update portions of our documentation.

Reviews: Design and code reviews will be used to ensure that the development team efforts are coordinated with the core ESMF effort. Reviews will involve a development team member from this project and a core implementation team member from the part-I project.

Verification and benchmarking: Standard configurations of the reference modeling systems will be developed that verify that the framework fulfils performance, portability and interoperability requirements. New code developments and changes will be continually verified against these benchmarks. Frequent intermediate releases of the evolving codes will be available for testing. The ESS Evaluation Team and vendors will assist with performance evaluation.

Source Availability and Distribution: We plan to develop our code in an open source development environment. Source code and documentation will be distributed via an ESMF website. We plan to hold a series of workshops to introduce the broader community to the ESMF.

Software Maintenance: The individual insititutions, responsible for the reference models, are commited to keeping the models compliant with the ESMF.

NASA ESS Team: We anticipate aligning the plan laid out here with the practices put forward by the NASA in-house ESS team.

3 Software Tools and Techniques

Configuration management: We will likely use CVS for configuration management since it is mature, freely available, and the current community standard. We anticipate maintaining code at the following acceptance levels: **Active** (untested), **Unit tested**, and **Integrated** (code is part of a working ESMF prototype).

Software metrics: we will track a simple set of software metrics throughout development to evaluate progress and predict schedules.

Defect tracking: A tool such as Bugzilla will be used to maintain a database of defects and new feature requests.

Collaborative tools: We plan to employ teleconferences to keep Development Team members in close touch, as well as quarterly face-to-face meetings. We will maintain project mailing lists and discussion forums.

(III) References

References

[1] Airlie Software Council. Principal best practices. Technical report, Software Program Managers Network, 1994. http://www.spmn.com/best_practices.html.

[2] G.M Amdahl. Validity of the single-processor approach to achieving large scale computing capabilities. In *AFIPS Conference Proceedings*, volume 30, pages 483–485, April 1967.

[3] Boville B.A. and P.R.Gent. The NCAR Climate System Model, Version one. *J.Climate*, 11:1115–1130, 1998.

[4] R. Barriuso and A. Knies. *SHMEM Users Guide, Revision 2.0.* Cray Research Inc., May 1994.

[5] N. Boden, D. Cohen, R. Felderman, A. Kulawik, C. Seitz, J. Seizovic, and W. Su. Myrinet: A gigabit per second local area network. *IEEE Micro Magazine*, February 1995.

[6] K. Bryan. A numerical method for the study of the circulation of the world ocean. *Journal of Computational Physics*, 4(3):347–376, 1969.

[7] J Charney, R.Fjortoft, and J.von Neumann. Numerical integration of the barotropic vorticity eqaution. *Tellus*, 2:237–254, 1950.

[8] Climate Research Committee. *Capacity of U.S. Climate Modeling to Support Climate Change Assessment Activities.* National Academy Press, 1998.

[9] D. L. Presotto and D. M. Ritchie. Interporcess Communications in the Ninth Edition UNIX System. In *UNIX Research System Papers*, volume II, pages 523–530, 1990.

[10] DAO. Nasa data assimilation office. http://dao.gsfc.nasa.gov.

[11] Geophysical Fluid Dynamics Laboratory. Flexible modeling system. http://www.gfdl.gov/˜fms/.

[12] R. Giering and T. Kaminski. Recipies for adjoint code construction. *ACM Trans. Math. Softwar*, 24:437–474, 1997.

[13] C. Hill, A. Adcroft, D. Jamous, and J. Marshall. A strategy for tera-scale climate modeling. In *Proceedings of Eigth ECMWF Workshop on the Use of Parallel Processors in Meteorology.* World Scientific, 1999.

[14] J. Hoe, C. Hill, and A. Adcroft. A personal supercomputer for climate research. In *Proceedings of Supercomputing 99*, 1999.

[15] Kalnay, E., M. Kanamitsu, J. Pfaendtner, J. Sela, M. Suarez, J. Stackpole, J. Tuccillo, L. Umscheid, and D. Williamson. Rules for intercahnge of physical parameterizations. *Bull. Amer. Met. Soc.*, 70(6):620–622, 1989.

[16] Kalnay, E., M. Kanamitsu, and W.E. Baker. Global numerical weather prediction at the National Meteorological Center. *Bull. Amer. Met. Soc.*, 71:1410–1428, 1990.

[17] Kiehl.J.T., J.J.Hack, G.B.Bonan, B.A.Boville, D.L.Williamson, and P.J.Rasch. The National Center for Atmospheric Research Community Climate Model: CCM3. *J. Climate*, 9:1131–1149, 1998.

[18] M. Kanamitsu. Description of the NMC global data assimilation and forecast system. *Wea. and Forecasting*, 4:335–342, 1989.

[19] S Manabe, J.Smagorinsky, and R.F.Strickler. Simulated climatology of a general circulation model with a hydrologic cycle. *Mon. Wea. Rev.*, 93(12):769–798, 1965.

[20] J. Marshall, A. Adcroft, C. Hill, and L. Perelman. A finite-volume, incompressible Navier-Stokes model for studies of the ocean on parallel computers. *J. Geophys. Res.*, 102(C3):5753–5766, 1997.

[21] J. Marshall, C. Hill, L. Perelman, and A. Adcroft. Hydrostatic, quasi-hydrostatic and non-hydrostatic ocean modeling. *J. Geophys. Res.*, 102(C3):5733–5752, 1997.

[22] McConnell, S. *Code Complete*. Microsoft Press, 1993.

[23] Message Passing Inerface Forum. *MPI: A Message-Passing Interface Standard*, 1994. available from http://www.mpi-forum.org/.

[24] NSIPP. Nasa seasonal-to-interannual prediction project. http://nsipp.gsfc.nasa.gov.

[25] Office Note Series on Global Modeling and Data Assimilation. Geos-3 data assimilation system. Technical Report DAO Office Note 97-06, NASA Goddard Space Flight Center, July 1997.

[26] Paulk, M.C., Curtis, W., Chrissis, M., and Weber, C.V. Capability maturity model for software, version 1.1. Technical Report CMU/SEI-93-TR-24, DTIC Number ADA263403, Software Engineering Institute, February 1993.

[27] Paulk, M.C., Weber, C.V., Chrissis, M. Garcia, S.M., and Bush, M.W. Key practices of the capability maturity model, version 1.1. Technical Report CMU/SEI-93-TR-25, DTIC Number ADA263432, Software Engineering Institute, February 1993.

[28] L.F. Richardson. *Weather Prediction by Numerical Process*. Cambridge at the University Press, 1922.

[29] Schaffer, D.S. and M.J. Suarez. Design and Performance Analysis of a Massively Parallel Atmospheric General Circulation Model. *Scientific Programming*, 8:40–57, 2000.

[30] A. Shaw, K. C. Cho, C. Hill, R. P. Johnson, J. Marshall, and Arvind. A comparison of implicitly parallel multithreaded and data parallel implementations of an ocean model based on the Navier-Stokes equations. *J. Parallel and Distributed Computing*, pages 1–51, 1998.

[31] Software Engineering Research Laboratory Series. Manager's handbook for software development. Technical Report SEL-84-01, NASA Goddard Space Flight Center, November 1990.

[32] Software Engineering Research Laboratory Series. Recommended approach to software development. Technical Report SEL-81-305, NASA Goddard Space Flight Center, June 1992.

[33] The ECCO consortium. Estimating the circulation and climate of the ocean. http://ecco.jpl.nasa.gov, http://ecco.ucsd.edu.

(IV) Biographical Sketches

JOHN MARSHALL

Department of Earth, Atmospheric and Planetary Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139

PROFESSIONAL INTERESTS

My research is directed at understanding key components of the general circulation of the atmosphere and ocean and the development of models to study them. I am interested in a variety of problems in geophysical fluid dynamics and their role in climate, ranging from rotating convection, the global circulation of the ocean and air-sea interaction. I use and develop numerical models of the atmosphere, ocean and climate.

EDUCATION

B.S.	1976	First Class Honors in Physics, Imperial College, London
Ph.D.	1980	Physics Imperial College

EMPLOYMENT

1992–Present	Professor	Massachusetts Institute of Technology
1992	Associate Professor	Massachusetts Institute of Technology
1991–1992	Reader in Physics	Imperial College
1984–1990	Lecturer in Physics	Imperial College
1982–1983	Post-doctoral fellow	University of Oxford

SELECTED PUBLICATIONS

1. Marshall, J., C. Hill, L. Perelman, and A. Adcroft, (1997) Hydrostatic, quasi-hydrostatic, and nonhydrostatic ocean modeling, J. Geophysical Res., 102(C3), 5733-5752.

2. Marshall, J., A. Adcroft, C. Hill, L. Perelman, and C. Heisey, (1997) A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers, J. Geophysical Res., 102(C3), 5753-5766.

3. Adcroft, A.J., Hill, C.N. and J. Marshall, (1997) Representation of topography by shaved cells in a height coordinate ocean model Mon Wea Rev, vol 125, 2293-2315

4. Marshall, J., Jones, H. and C. Hill, (1998) Efficient ocean modeling using non-hydrostatic algorithms Journal of Marine Systems, 18, 115-134

5. Shaw; A. Arvind, Cho, K.-C., Hill, C., Johnson, R.P. and Marshall, J. (1998) A comparison of implicitly parallel multi-threaded and data-parallel implementations of an ocean model based on the Navier-Stokes equations. J. of Parallel and Distributed Computing, vol 48, No 1, 1-51

JEFFREY L. ANDERSON

NOAA/Geophysical Fluid Dynamics Laboratory,
P.O. Box 308, Princeton, NJ 08542

PROFESSIONAL INTERESTS

My research interests include stochastic data assimilation and predcition, seasonal-interannual prediction, and software engineering to support climate system models.

EDUCATION

B.S.	1984	Meteorology and Computer Science, University of Utah
M.S.	1986	Computer Science University of California, Berkeley
Ph.D.	1990	Atmospheric and Oceanic Sciences, Princeton University

EMPLOYMENT

1995–Present	Head	Experimental Prediction Group, Geophysical Fluid Dynamics Laboratory
1992–Present	Lecturer	Atmospheric and Oceanic Sciences, Princeton University
1992-1995	Meteorologist	Geophysical Fluid Dynamics Laboratory
1990-1992	Postdoctoral Scientist	Climate Analysis Center, National Meteorological Center

SELECTED PUBLICATIONS

1. Anderson, J.L. "An ensemble adjustment filter for data assimilation," submitted to *Monthly Weather Review*, 2000.

2. Anderson, J.L. and J.J. Ploshay, "Impact of initial conditions on seasonal simulations with an atmospheric general circulation model," Quarterly Journal of the Royal Meteorological Society, **126**, 2241-2264, 2000.

3. Vitart, F., and J.L. Anderson, "Sensitivity of Atlantic tropical storm frequency and interdecadal variability of SSTs in an ensemble of GCM integrations, to appear in *Monthly Weather Review*, 2000.

4. Anderson, J.L. and S.L. Anderson, "A Monte Carlo implementation of the non-linear filetering problem to produce ensemble assimilations and forecasts," *Monthly Weather Review*, **127**, 2741-2758, 1999.

5. Anderson, J.L., H. van der Dool, A. Barnston, W. Chen, W. Stern, and J. Ploshay, "Capabilities of numerical and statistical methods for extratropical seasonal prediction," *Bulletin of the American Meteorological Society*, **80**, 1349-1361, 1999.

BYRON A. BOVILLE

National Center for Atmospheric Research,
P.O. Box 3000, Boulder CO 80307

PROFESSIONAL INTERESTS

My research has concentrated on developing and applying general circulation models of the lower and middle atmosphere for studies of atmospheric dynamics and climate. I have been one of the central figures in both the scientific and computational development of 4 generations of the NCAR atmospheric general circulation model. More recently, I have concentrated on coupled ocean-atmosphere modeling and was co-chair of the team which developed the NCAR Climate System Model (CSM). I am currently interested in the climate impact of solar variability and the role of the middle atmosphere in climate variability and climate change.

EDUCATION

B.Sc.	1975	1st Class Honors in Meteorology, McGill University, Montreal, Canada
Ph.D.	1979	Atmospheric Sciences, University of Washington, Seattle, Washington

EMPLOYMENT

1999–Present	Head	Climate Modeling Section, Climate and Global Dynamics Division National Center for Atmospheric Research
1992–Present	Senior Scientist	National Center for Atmospheric Research
1981–1992	Scientist I-III	National Center for Atmospheric Research
1979–1981	Postdoc	Advanced Study Program National Center for Atmospheric Research

SELECTED PUBLICATIONS

1. Boville, B. A., J. T. Kiehl, P. J. Rasch, and F. O. Bryan, 2001: Improvements to the NCAR CSM-1 for transient climate simulations. J. Climate, 14, in press.
2. Boville, B. A., 2000: Toward a complete model of the climate system. In "Numerical modeling of the global atmosphere in the climate system", P. Mote and A. O'Neill, eds., Kluwer Academic Publishers, 419–442.
3. Kiehl, J. T., J. J. Hack, G. B. Bonan, B. A. Boville, D. L. Williamson, P. J. Rasch, 1998: The National Center for Atmospheric Research Community Climate Model: CCM3. J. Climate, 9, 1131–1149.
4. Boville, B. A., and P. R. Gent, 1998: The NCAR climate system model, version one. J. Climate, 11, 1115–1130.
5. Hack, J. J., J. M. Rosinski, D. L. Williamson, B. A. Boville, and J. E. Truesdale, 1995: Computational design of the NCAR community climate model. Parallel Computing, 21, 1545–1569.

CECELIA DeLUCA

National Center for Atmospheric Research,
P.O. Box 3000, Boulder CO 80307

PROFESSIONAL INTERESTS

My interests include the design of large, high-performance software systems, particularly those relating to atmospheric science; parallel algorithms; real-time systems, and software engineering tools and processes. I was a design lead on the development of the Space-Time Adaptive Processing Library (STAPL) parallel framework for real-time radar applications. STAPL is an integral part of multiple operational next-generation radar systems and has been ported to several platforms. It extends the serial Vector Signal and Image Processing Library (VSIPL) standard to SMP-cluster architectures. Previous projects have included the development of parallel codes for the simulation of middle atmospheric dynamics, atmospheric chemistry, and remote sensing of atmospheric temperatures.

EDUCATION

A.L.B.	1990	Liberal Arts/Social Sciences, Harvard University, Cambridge, MA
M.S.	1994	General Engineering, Boston University, Boston, MA
M.S.	1996	Meteorology, Massachusetts Institute of Technology, Cambridge, MA

AWARDS

1994	Boston University College of Engineering Outstanding Achievement Award, first in graduating class

EMPLOYMENT

1999–Present	Software Engineer	Scientific Computing Division, National Center for Atmospheric Research
1998–1999	Lead Software Engineer	MIT Lincoln Laboratory
1996–1998	Software Engineer	MIT Lincoln Laboratory
1993–1994	Manager, Technical Support	Omnet Communications

SELECTED PUBLICATIONS

1. Dickinson, R.E., S.E. Zebiak, J.L. Anderson, M. Blackmon, C. DeLuca, T. Hogan, M. Iredell, M. Ji, R. Rood, M. Suarez, K. Taylor, "Need for Infrastructure and Commonality in Climate and Weather Prediction Codes and Data," submitted to *Bulletin of the American Meteorological Society*, 2000.

2. DeLuca, C., C. Heisey, R. Bond and J. Daly, " A Portable, Object-Based Parallel Library and Layered Framework for Real-Time Radar Signal Processing," In *Proceedings of Scientific Computing in Object-Oriented Parallel Environments*, ISCOPE 1997.

3. Heisey, C., C. Adamo, M. Arakawa, P. Baggeroer, J. Daly, C. DeLuca, W. Dale Hall, K. Pickard, and H. A. Spang, "Implementation of the STAP Library and Framework (STAPL) for Real-Time Matrix-Based Signal Processing," In *Abstracts of High Performance Embedded Computing*, HPEC 1998.

PHILIP W. JONES

Theoretical Fluid Dynamics (T-3),
Los Alamos National Laboratory,
Los Alamos, NM 87545

PROFESSIONAL INTERESTS

Current interests involve the use of massively parallel computers to study problems in geophysical and astrophysical fluid dynamics, including atmosphere, ocean and coupled climate modeling, middle atmosphere dynamics and fully-compressible thermal convection.

EDUCATION

B.S.	1985	Physics and Mathematics with distinction, Iowa State University
Ph.D.	1991	Astrophysical, Planetary, and Atmospheric Sciences, University of Colorado

EMPLOYMENT

1993–Present	Staff Member	Theoretical Fluid Dynamics (T-3), Los Alamos National Laboratory
1991–1993	Post-doctoral Research Associate	Geoanalysis Group (EES-5), Los Alamos National Laboratory
1986–1991	Research Assistant	Joint Institute for Laboratory Astrophysics and Center for Applied Parallel Processing, University of Colorado, Boulder

SELECTED PUBLICATIONS

1. Jones, P.W. 1999 "First- and Second-order Conservative Remapping Schemes for Grids in Spherical Coordinates," *Mon. Weath. Rev.*, **127**, 2204-2210.

2. Jones, P.W., Malone, R.C. and Lai, C.A. 1998 "The Los Alamos Coupled Model," *Proceeding of the Second International Workshop on Software Engineering and Code Design in Parallel Meteorological and Oceanographic Applications*, ed. M. O'Keefe and C. Kerr, NASA Publication GSFC/CP-1998-206860.

3. Jones, P.W. 1998 "The Los Alamos Parallel Ocean Program (POP) and Coupled Model on MPP and Clustered SMP Computers," *Making its Mark: Proceedings of the 7th ECMWF Workshop on the Use of Parallel Processors in Meteorology,* ed. G. R. Hoffmann and N. Kreitz (Singapore: World Scientific Publishing).

4. Jones, P.W., Hamilton, K.P. and Wilson, R.J. 1996 "A Very High-Resolution General Circulation Model Simulation of the Global Circulation in Austral Winter," *J. Atm. Sci.*, **54**, 1107-1116.

5. Jones, P.W., Kerr, C.L. and Hemler, R.S. 1995 "Practical Considerations in Development of a Parallel SKYHI General Circulation Model," *Parallel Computing,* **21**, 1677-1694.

STEPHEN J. LORD

Environmental Modeling Center
National Centers for Environmental Prediction
NOAA Science Center, Rm. 207
Washington, DC 20233

PROFESSIONAL INTERESTS

My interests are in managing and participating in all aspects of data assimilation and numerical model development for weather and seasonal climate forecasts. As Director of the Environmental Modeling Center, National Centers for Environmental Prediction, I oversee a staff of 90 who are dedicated to improving operational weather, ocean and climate modeling products to support the NWS mission. I have a strong background in data assimilation and tropical meteorology and have done original research on hurricane numerical modeling and data assimilation.

EDUCATION

B.S.	1969	Physics	Yale University (cum laude)
M.S.	1975	Atmospheric Sciences	University of California at Los Angeles
Ph.D.	1978	Atmospheric Sciences	University of California at Los Angeles

HONORS AND AWARDS

1997	AMS Fellow
1996	NOAA Dept. of Commerce Gold Medal for Implementation of the GFDL Hurricane Model
1993	NOAA Dept. of Commerce Bronze Medal for Applied research on hurricane track prediction

EMPLOYMENT

2000–Present	Director	Environmental Modeling Center, National Centers for Environmental Prediction
1993–2000	Acting Director/Deputy Director	Environmental Modeling Center, National Centers for Environmental Prediction
1989–1993	Meteorologist	National Meteorological Center
1980–1989	Meteorologist	Hurricane Research Division, Atlantic Oceanographic and Meteorological Laborato

SELECTED PUBLICATIONS

1. Pu, Zhao-Xia, S.J. Lord, and E. Kalnay, 1998: Forecast sensitivity with dropsonde data and targeted observations. In press (Tellus)

2. Surgi, N., H.L. Pan, and S.J. Lord, 1998: Improvement of the NCEP global model over the tropics: an evaluation of model performance during the 1995 hurricane season. Mon. Wea. Rev., 126, 1287-1305

3. Tuleya, R. E., and S. J. Lord, 1997: The impact of dropwindsonde data on GFDL hurricane model forecasts using global analyses. Wea. Forecasting, 12, 307-323.

4. Lord, S.J., and J. L. Franklin, 1990: The environment of Hurricane Debby (1982). Part II: Thermydynamic fields. Mon. Wea. Rev., 118, 1444-1459.

5. Lord, S. J., and J. M. Lord, 1998: Vertical velocity structures in an axisymmetric, nonhydrostatic tropical cyclone model. J. Atmos. Sci., 45, 1453-1461.

MAX J. SUAREZ

NASA Seasonal to Interannual Prediction Project
NASA Goddard Space Flight Center
Greenbelt, MD 20771

PROFESSIONAL INTERESTS

Large-scale atmosphere/ocean interactions, climate modeling, numerical methods, parameterization of subgrid-scale processes in atmospheric models, maintenance of the atmospheric general circulation, climate sensitivity.

EDUCATION

B.S.	1971	Engineering Science, University of Florida
M.E.	1972	Engineering Science University of Florida
M.A.	1974	Geophysical Fluid Dynamics, Princeton University
Ph.D.	1976	Geophysical Fluid Dynamics, Princeton University

EMPLOYMENT

1983–Present	Meteorologist	NASA/Goddard Space Flight Center
1976–1983	Assistant Professor	UCLA

SELECTED PUBLICATIONS

1. Moorthi, S., and M. J. Suarez, 1992: Relaxed Arakawa-Schubert: A Parameterization of Moist Convection for General Circulation Model. *Mon. Wea. Rev.*, **120**, 978-1002.

2. Koster, R. D., and M. J. Suarez, 1992: A Comparative Analysis of Two Land Surface Heterogeneity Representations. *J.Climate*, **5**, 1379-1390.

3. Suarez, M. J., and D. G. Duffy, 1992: Terrestrial Superrotation: A Bifurcation of the General Circulation. *J. Atmos. Sci..*, **49**, 1541-1554.

4. Held, I. M., and M. J. Suarez, 1994: A Proposal for the Intercomparison of Dynamical Cores of Atmospheric General Circulation Models. *Bull. Amer. Meteor. Soc.*, **75**, 1825-1830.

5. Schaffer, D. S., and M.J. Suarez, 2000: Next Stop: Teraflop; The Parallelization of an Atmospheric General Circualtion Model. *High Performance Computing*, **submitted.**

6. Mehta, V., and M. J. Suarez, 2000: Decadal-multidecadal Variations of ENSO: 1909-1988. *J. Geophys. Res. Letters*, **27**, 121-124.

7. Mehta, V., M. J. Suarez, J. Manganello, and T. L. Delworth, 2000: Oceanic Influence on the North Atlantic Oscillation and Associated Northern Hemishphere Climate Variations: 1959-1993. *J. Geophys. Res. Letters*, **27**, 121-124.

CHRIS HILL

Department of Earth, Atmospheric and Planetary Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139

PROFESSIONAL INTERESTS

My research is directed at applying advanced computing technology to the study of planetary scale circulation. I am interested in how cutting edge technologies become effective research tools and in how algorithms, hardware and software combine to make this possible and profitable. I am currently interested in the role of affordable commodity systems and in very large-scale petaflop/s computing systems.

EDUCATION

B.S.	1987	Honors in Physics, Imperial College, London

EMPLOYMENT

1993–Present	Researcher	Massachusetts Institute of Technology
1987–1993	Engineer	National Grid Company

SELECTED PUBLICATIONS

1. Marshall, J., C. Hill, L. Perelman, and A. Adcroft, (1997) Hydrostatic, quasi-hydrostatic, and nonhydrostatic ocean modeling, J. Geophysical Res., 102(C3), 5733-5752.
2. Marshall, J., A. Adcroft, C. Hill, L. Perelman, and C. Heisey, (1997) A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers, J. Geophysical Res., 102(C3), 5753-5766.
3. Adcroft, A.J., Hill, C.N. and J. Marshall, (1997) Representation of topography by shaved cells in a height coordinate ocean model Mon Wea Rev, vol 125, 2293-2315
4. Hoe, J., C. Hill and A. Adcroft, (1999) A Personal Supercomputer for Climate Research, Proceedings of Supercomputing 99.
5. Shaw; A. Arvind, Cho, K.-C., Hill, C., Johnson, R.P. and Marshall, J. (1998) A comparison of implicitly parallel multi-threaded and data-parallel implementations of an ocean model based on the Navier-Stokes equations. J. of Parallel and Distributed Computing, vol 48, No 1, 1-51

(V) Milestones, Schedule and Costs

The total budget for this proposal is $2,828,052 The dates on these milestones assume that the first payment is received June 2001. The milestones cover implementation of codes on the ESS testbed and on a commodity cluster system consisting of at least sixty-four processors. An initial prototype sixteen processor cluster will be installed at the start of the project. The cluster will then be expanded throughout the project.

No.	**Label**	**Milestone**	**Completion Date**	**Payment Amount**
1	A	Software engineering plan completed *Deliver draft Developer's Guide specifying software procedures and conventions. Basic web distribution and communication mechanisms in place. Online document and source repositories established, along with use policies.*	July. 2001	$207,563
2	E	Code baseline completed *NSIPP, GFDL, MIT, NCAR models running and bench-marked on the ESS testbed. Baseline configurations will be* $\leq 1^o$ *global resolution or equivalent, running on some fraction of the ESS testbed. NCEP prognostic models will be bench-marked and analyzed on NCEP home system. Baseline metrics: lines of shared code between models; performance, scaling and throughput on testbed and, for at least one model, on commodity cluster. Codes available via the Web. Baseline test configurations documented, along with performance analysis identifying code hot-spots. Month by month development targets for year 1 for each reference model established.*	Aug 2001	$312,000
3	H	Design policy for interoperability and community delivery agreed on *Deliver Architecture Document, Implementation Report, Draft Software Implementation and Test Plan. Prototype code used in the Implementation Study will be delivered via the Web. Implementation study will include verifying an API for sharing grids, fields and decompositions between the reference models as ESMF components and the core ESMF. Identify a broader community interested in reviewing/testing the ESMF and mechanism for feedback.*	Dec. 2001	$332,000
4	B	First Annual Report delivered *Submit FY01 Annual Report via the Web. Month by month development targets for year 2 for each reference model established.*	June 2002	$126,700

No.	**Label**	**Milestone**	**Completion Date**	**Payment Amount**
5	F	First code improvement completed *All reference models, including NCEP, running on ESS testbed. All models using some elements of the shared framework code, through calls to the ESMF API. Absolute per processor performance either improved or same as baseline. Demonstrable scaling improvements for all models. Portability between cluster and testbed demonstrated for at least two codes. Price/performance metrics comparing cluster and ESS testbed reported. Benchmark tests performed at global resolution* $\leq \frac{1}{2}^o$ *or equivalent. Throughput comparable to* 1^o *baseline, but using larger fraction of ESS testbed. Metrics measuring number of framework functions used reported. Metrics measured will categorize function use as completely new function, extending an existing function or replacing an existing functions. Codes available via the Web.*	July 2002	\$375,400
6	I	Interoperability prototype from Milestone "H" tested with improved codes *Prototype implementations of all modeling applications using the framework: NSIPP, GFDL, NCEP, MITgcm, CCSM-DAO. May be partial coupling/stubs. Implementation tests will include kernel, driver and full-model tests and prototype tests of models interoperating as framework components. Prototype implementation will include at least 1 of 3 data assimilation prognostic models. Source code delivered via the Web.*	Nov. 2002	\$338,688
7	C	Second Annual report delivered *Submit FY02 Annual Report via the Web. Month by month development targets for year 3 for each reference model established.*	June 2003	\$126,700
8	G	Second code improvement completed *Reference models running on ESS tested and at least two models running on full-size commodity cluster using all main elements (parallel primitives, coupling, I/O, low-level support) of the shared framework code. Absolute per processor performance either improved or same as baseline. Portability between cluster and testbed demonstrated for at least three codes. Price/performance metrics comparing cluster and ESS testbed reported. Benchmark tests performed at global resolution* $\leq \frac{1}{4}^o$ *or equivalent. Demonstrate that scaling and throughput are maintained for this resolution. Metrics measuring number of framework functions used reported. Codes available via the Web.*	July 2003	\$245,800

No.	**Label**	**Milestone**	**Completion Date**	**Payment Amount**
9	J	Full interoperability demonstrated using improved codes *Full implementation of at least 3 of 5 modeling reference modeling systems using the framework: NSIPP, GFDL, MITgcm, NCEP, CCSM-DAO. Full implementation of 2 of 3 prognostic models used in assimilation schemes. Using the framework internally within models/assimilation schemes as well as for coupling components.*	Nov. 2003	$266,200
10	K	Customer delivery accomplished *Full framework system available for download and installation on new systems. Deliver a user-friendly website where the ESMF source code and User's Guide can be obtained, and hold a workshop on framework usage.*	May 2004	$313,000
11	D	Final Report delivered *Submit Final Report to ESS via the Web.*	June 2004	$84,000
opt	e	Installation of prototype high-performance cluster *Between thirty-two and sixty four processor cluster installed for use with framework models at MIT. Configuration will include dedicated network for performance critical parallel processing communication. A separate commodity network will be used for general traffic and for I/O. A scalable I/O subsystem will be included using a set of separate nodes. Exact configuration will be planned in conjunction with Linux clusters team at GSFC. High-performance network will probably be Myrinet, scalable I/O will be through PVFS. Commodity network will be switched 100 Mbps ethernet. Nodes will be PIII or later generation x86 processors with 450MHz or greater speeds. The role of the dedicated network will be carefully analyzed.*	May 2002	$50,000
opt	f	Installation of full-size high-performance cluster *Between sixty-four and 128 processor cluster installed for use with framework models at MIT. Configuration will include dedicated network for performance critical parallel processing communication. A separate commodity network will be used for general traffic and for I/O. A scalable I/O subsystem will be included using a set of separate nodes. Exact configuration will be planned in conjunction with Linux clusters team at GSFC. Information on final configuration, setup and validation tests will be made available in a manual available via the web.*	May 2003	$50,000

(VI) Endorsement Letters

(VII) Education and Public Outreach