Parallelization and Distribution of a Coupled Atmosphere–Ocean General Circulation Model

CARLOS R. MECHOSO, CHUNG-CHUN MA, JOHN D. FARRARA, AND JOSEPH A. SPAHR

Department of Atmospheric Sciences, University of California, Los Angeles, Los Angeles, California

REAGAN W. MOORE

San Diego Supercomputer Center, La Jolla, California

(Manuscript received 15 August 1992, in final form 6 November 1992)

ABSTRACT

The distribution of a climate model across homogeneous and heterogeneous computer environments with nodes that can reside at geographically different locations is investigated. This scientific application consists of an atmospheric general circulation model (AGCM) coupled to an oceanic general circulation model (OGCM).

Three levels of code decomposition are considered to achieve a high degree of parallelism and to mask communication with computation. First, the domains of both the gridpoint AGCM and OGCM are divided into subdomains for which calculations are carried out concurrently (*domain decomposition*). Second, the model is decomposed based on the diversity of tasks performed by its major components (*task decomposition*). Three such components are identified: (a) AGCM/physics, which computes the effects on the grid-scale flow of subgrid-scale processes such as convection and turbulent mixing; (b) AGCM/dynamics, which computes the evolution of the flow governed by the primitive equations; and (c) the OGCM. Task decomposition allows the AGCM/dynamics and OGCM calculations to be carried out concurrently. Last, computation and communication are organized in such a way that the exchange of data between different tasks is carried out in subdomains of the model domain (*I/O decomposition*). In a dedicated computer network environment, the wall-clock time required by the resulting distributed application is reduced to that for the AGCM/physics, with the other two components and interprocess communications running in parallel.

The network bandwidth requirements for the distributed application are analyzed. It is assumed that the wall-clock time required to run the AGCM/physics for the model atmosphere in a dedicated computer environment is fixed at a value corresponding to high network efficiency. The analysis shows that, for computer environments based on nodes equivalent to the Intel Touchstone Delta, a bandwidth approaching that of the Gigabit Network is required for an efficient operation of the distributed application with model resolution double that used in current studies of the climate system if output is visualized in real time.

It is argued that distribution of a climate model based on domain, task, and I/O decomposition has the potential for significant and eventually superlinear speedup in model execution, which will facilitate performance of the long integrations required by climate studies.

1. Introduction

General circulation models (GCMs) of the coupled atmosphere–ocean system are among the most powerful tools for studies on climate and climate variability. GCMs explicitly solve the equations governing fluid motion on a rotating sphere, including parameterizations of physical processes at subgrid scales (e.g., cumulus convection and turbulent mixing). Thus, GCMs can be used to investigate the complex interactions and feedbacks between different components of atmosphere and ocean circulations. Examples of outstanding problems studied with GCMs are El Niño–Southern Oscillation (ENSO) events and the impact on climate of increasing concentrations of greenhouse gases ("the greenhouse effect").

Climate-related studies using GCMs are among the most demanding research activities in terms of computer resources. To investigate ENSO and the greenhouse effect requires decade-long and century-long simulations, respectively, at CPU rates of tens of hours per model year. Further, ensembles of experiments have to be conducted to assess the sensitivity of the climate system. Efforts are constantly being made to develop, optimize, and document the computer codes of GCMs used for climate research or numerical weather prediction. Current operational GCMs have been highly vectorized over the years. The possibility of reducing wall-clock time by parallelizing computations in multiprocessor machines has become a major development activity in recent times (e.g., Chervin and Semtner 1988; Hoffman and Maretis 1990).

Corresponding author address: Dr. Carlos R. Mechoso, Department of Atmospheric Sciences, University of California, Los Angeles, 405 Hilgard Avenue, Los Angeles, CA 90024-1565.

In the near future, networks with bandwidths on the order of a gigabit (Gbit) per second will connect supercomputers with diverse architectures. The question arises as to whether the enhanced computing power implied by this configuration can be used effectively to reduce the wall-clock time demanded by GCMs, as well as other scientific applications. In short, the question is whether a network of supercomputers connected by a high-speed link can be viewed as a "metacomputer" from the point of view of a GCM and other such similar applications.

In this paper, we address that question by discussing procedures for distribution and efficient operation of a coupled atmosphere–ocean GCM in a computer environment consisting of multiple nodes, which can have different architectures and be geographically separated from each other but connected by a high-speed network. We explore the methodology and performance issues for such a distributed scientific application. These issues include (i) code design for a parallel programming environment, (ii) mechanisms for exchanging data among processes, and (iii) hiding network latency with computation. Our research has several objectives:

(a) To explore the possibility of superlinear speedup of computation through the use of the most suitable computer architecture for individual model components.

(b) To enhance capabilities for analysis and display of simulated flow evolution by allowing for remote real-time visualization of model output.

(c) To facilitate closer collaboration among researchers specializing in different model components by providing a system in which modules under development at different research institutions can be easily incorporated into an earth system model.

(d) To increase available resources by allowing for the use of geographically separated computers and mass storage systems.

Several characteristics of the GCM make it both a difficult and ideal application for a distributed computer environment. The model is computation intensive, requires a large amount of core memory, and generates large outputs. The GCM with the lowest resolution used at the University of California Los Angeles (UCLA) performs on the average about 1 billion floating-point operations to simulate 1 hour, requires approximately 10 megawords of core memory, and generates data on the order of hundreds of gigabytes (Gbyte) in a typical simulation aimed to investigate the impact on climate of increased greenhouse gases. The model code has both vector and scalar components. For GCMs used in climate research, one would like to have vector computers, massively parallel processors (MPPs), large-volume high-speed data archiving systems, and visualization systems working seamlessly together.

This study is an integral part of the Corporation for National Research Initiatives (CNRI) Gigabit Testbed Initiative. The major goals of this three-year initiative are to develop architectural alternatives for consideration in determining the possible structure of a wide-area gigabit network serving the research and education communities, and to understand the utility of gigabit networks by the end user. The project is organized around a set of five test-beds consisting of collaborators from universities, national laboratories, supercomputer centers, and major industrial organizations.

The test-beds in the gigabit project are AURORA, BLANCA, CASA, NECTAR, and VISTANET. The leading research organizations in the CASA wide-area test-bed are the Los Alamos National Laboratory (LANL) in Los Alamos, New Mexico; the California Institute of Technology (Caltech) and the Jet Propulsion Laboratory (JPL) in Pasadena, California; and the San Diego Supercomputer Center (SDSC) in association with UCLA. The carriers collaborating in the CASA test-bed are MCI, Pacific Bell, and US West. Supercomputers at participating institutions include CRAY Y-MPs, Intel MPPs, nCUBEs, and Thinking Machines Corporation Connection Machines (see Fig. 1).

The specific goal of the CASA test-bed is to investigate the use of distributed supercomputing over wide-area high-speed networks to provide new levels of computational resources for leading-edge scientific problems. One of the projects under development in the CASA testbed is to run the UCLA coupled atmosphere–ocean GCM distributed between an Intel Touchstone Delta parallel computer and a CRAY Y-MP8/864 vector supercomputer. This experiment requires a network bandwidth substantially higher than any currently available, as discussed in the following.

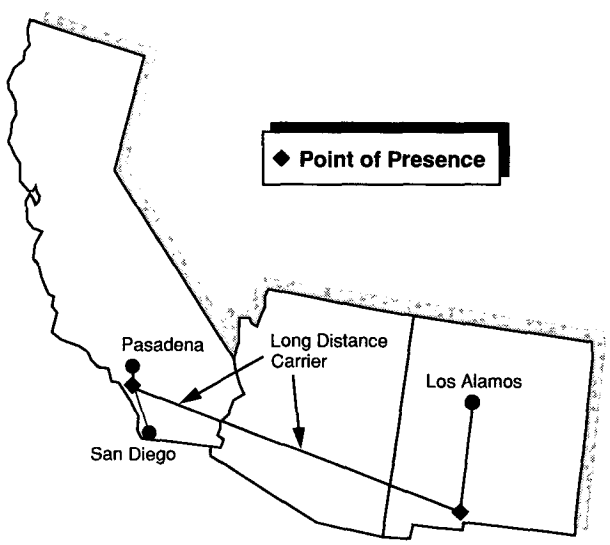

FIG. 1. CASA gigabit wide-area network sites.

We start this paper by describing the coupled atmosphere–ocean GCM in section 2. Several methods of distributing and parallelizing the model code are presented in section 3. The requirements on the network bandwidth for efficient operation of the distributed application are discussed in section 4. Selected results of preliminary experiments on the performance of the distributed GCM are presented in section 5. Our conclusions are described and discussed in section 6.

2. Description of the coupled GCM

The UCLA atmospheric GCM (AGCM) and the Geophysical Fluid Dynamics Laboratory (GFDL)–Princeton University oceanic GCM (OGCM) are the components of the UCLA coupled atmosphere–ocean GCM.

a. The UCLA AGCM

The UCLA AGCM has been developed over the years by Professor A. Arakawa and collaborators. Constant effort is dedicated to the improvement of the finite-difference schemes and parameterizations of physical processes included in the model, as well as to optimization of the code.

The UCLA AGCM predicts the values of horizontal velocity components, potential temperature, water vapor and ozone mixing ratios, surface pressure, and ground temperature. In an approach unique to this model, the planetary boundary layer (PBL) is treated as well mixed and represented by the model's bottom layer, whose depth is also predicted. Surface fluxes of sensible heat, moisture, and momentum are modeled using a bulk parameterization. The AGCM includes parameterizations of cumulus convection and its interaction with the PBL, stratus clouds, solar and infrared radiative heating, and orographic gravity-wave drag. The geographical distributions of surface albedo and ground wetness are interpolated from prescribed monthly means based on the observed climatology. The reader is referred to Mechoso et al. (1987) and references therein for a more detailed description of these model features.

In the vertical, the model is based on a coordinate system in which the lower boundary, the PBL top, and isobaric surfaces above a prescribed pressure level (100 mb) are coordinate surfaces. The top of the model atmosphere is assumed to be a material surface. The vertical finite differencing used above the PBL guarantees conservations of the global mass integrals of potential temperature and total potential plus kinetic energy under frictionless adiabatic processes.

The UCLA AGCM is a "gridpoint model" and equations are horizontally discretized using a staggered latitude–longitude C grid (Arakawa and Lamb 1977). The scheme for the horizontal advection terms in the momentum equation conserves potential enstrophy and gives fourth-order accuracy for the advection of potential vorticity. The horizontal advection scheme used for the potential temperature is also fourth order and conserves the global mass integral of its square. In the continuity equation, the pressure gradient force, and the definition of absolute vorticity, the differencing is of second-order accuracy.

To maintain linear computational stability near the poles without using an extremely short time interval as required by the decreasing grid size with increasing latitude, a longitudinal filtering is applied to selected terms in the prognostic equations. Integration of the AGCM, therefore, requires both "local" and "nonlocal" calculations. To update model variables at a grid point, the local calculations use values at adjacent grid points both in the horizontal and vertical directions. Nonlocal calculations require information from all other grid points along the same latitude circle.

There are two relatively well-defined components within the AGCM:

1) AGCM/physics, the part of the code that computes diagnostically the effect of subgrid-scale processes on motions resolved by the model. The results obtained by AGCM/physics are supplied to AGCM/dynamics as forcing terms in the hydrodynamic equations.

2) AGCM/dynamics, the part of the code that computes prognostically the evolution of fluid flow governed by the primitive equations.

Typically, a calculation of the AGCM/physics for the model atmosphere requires one order of magnitude more CPU time than that of the AGCM/dynamics. To approximately balance the CPU time required by each model component, the common practice is to advance the model in time by performing one calculation of the AGCM/physics followed by several steps of the AGCM/dynamics with the most recently available forcing fields. The former model component has to be computed frequently enough to resolve the effects of the diurnal variation of insolation. The time step used in the latter component, on the other hand, is limited by the considerations on the aforementioned linear computational stability. For the nine-layer standard horizontal resolution (4° latitude × 5° longitude) AGCM, the AGCM/physics calculation is performed every simulated hour and the time step for the AGCM/dynamics is 7.5 min. Therefore, every calculation of the AGCM/physics is followed by eight time steps of the AGCM/dynamics.

At present, the UCLA AGCM has tropospheric versions with the top at 50 mb and tropospheric–stratospheric versions with the top at 1 mb. Table 1 gives a summary of AGCM timings on a CRAY Y-MP using one processor for the tropospheric versions with the standard horizontal resolution. In a typical run, prognostic variables and selected diagnostic variables such as precipitation and surface fluxes of heat and momentum are stored every 12 simulated hours.

TABLE 1. Timings for AGCM and OGCM with various resolutions.

Horizontal resolution (lat × long)	Vertical resolution no. of levels	No. of grid points		CPU time per simulated day* (s)
		Horizontal	Total	
		AGCM		
4° × 5°	9	3168	28 512	140
4° × 5°	17	3168	53 856	360
		OGCM (tropical Pacific)		
0.3–2° × 1°	27	16 000	432 000	45
		OGCM (global)		
1° × 1°	15	44 888	673 320	70

* Based on CRAY Y-MP, one processor.

The AGCM has been evaluated in a variety of studies, including long-term simulations of monthly mean fields, experimental medium-range predictions, and assessments of the impact of SST anomalies on the atmospheric circulation (e.g., Mechoso et al. 1987).

b. *The OGCM*

The OGCM is based on that developed at the National Oceanic and Atmospheric Administration (NOAA) GFDL/Princeton University by K. Bryan and M. D. Cox (Bryan 1969; Cox 1984). The OGCM predicts the horizontal velocity components, temperature, salinity, and, optionally, other tracers. Density is determined from temperature and salinity using either Knudsen's equation of state (Bryan and Cox 1972) or the formula by UNESCO (1981). The model uses depth as the vertical coordinate. The OGCM is also a gridpoint model, in which the equations are horizontally discretized using a staggered latitude–longitude B grid (Arakawa and Lamb 1977).

The top of the model is assumed to be a rigid lid. The velocities are split into components corresponding to the vertically averaged flow and the deviation from the vertical average. To advance in time the latter component at a grid point involves local calculations, which use values at adjacent grid points both in the horizontal and vertical directions. To advance in time the former component requires nonlocal calculations since an elliptic boundary-value problem has to be solved. These nonlocal calculations are performed after the local calculations are completed.

At present, the OGCM has a tropical Pacific and a global version. The tropical Pacific version, which is especially designed to study the oceanic circulation in the equatorial regions and associated currents characterized by small meridional scales, covers the Pacific Ocean in the latitude belt from 28°S to 50°N. In longitude, the resolution is 1°; in latitude, the mesh size is 1/3° between 10°S and 10°N and increases gradually toward the poles. There are 27 levels in the vertical, with 10 levels equally spaced over the upper 100 m. The ocean depth in this version is assumed to be constant at approximately 4150 m. The northernmost and southernmost parts of the domain are relaxed toward the observed climatology in both salinity and temperature fields. The global version is designed to study the general circulation of the World Ocean as required by climate phenomena with decadal and longer time scales. This version incorporates realistic bottom topography, and can be configured at different horizontal and vertical resolutions. The horizontal resolution of the standard version of the global OGCM is 1° latitude × 1° longitude.

In a typical run, prognostic variables are stored every 10 simulated days. Three-day means of prognostic variables and selected diagnostic variables—such as vertical velocity and heat content of the upper ocean—are stored every three days. Table 1 gives a summary of timings on a CRAY Y-MP using one processor for the tropical Pacific OGCM and the standard resolution version of the global OGCM.

A series of multiyear simulations with the uncoupled tropical Pacific OGCM have been performed to study the the dynamics of equatorial currents and to explore sensitivity of model results to the parameterization of subgrid-scale processes. These simulations generally produce realistic structure for the ocean circulation (Ma et al. 1991).

c. *The coupled GCM*

The UCLA AGCM and the GFDL OGCM are the components of our coupled GCM. The AGCM and OGCM run separately and continually exchange information at the air–sea interface. In the typical situ-

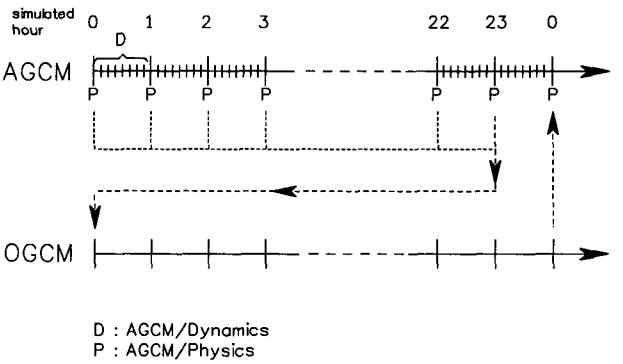

FIG. 2. Diagram showing the procedure of coupling in the UCLA coupled model for the case of a one-day coupling interval. The two horizontal axes show the simulated time of the AGCM and the OGCM, respectively. Arrows between the two axes show the exchange of information between the two models, with dashed lines showing intermediate results that are collected for averaging.

ation when only one processor is available for operation, the AGCM is first integrated for some period of time (coupling interval) and then passes to the OGCM the time-averaged wind stress and heat and water fluxes over that period. The OGCM is integrated for the same period of time and returns sea surface temperature (SST) to the AGCM (see Fig. 2). The interface between the two GCMs is provided by a suite of coupling routines that performs the interpolations required by the difference in grid systems.

We have performed multiyear simulations with the nine-layer standard horizontal resolution AGCM coupled to the tropical Pacific OGCM. The model produces a realistic simulation of the seasonal cycle (see Fig. 3). The results do not show evidence of significant climate drift, which is an issue of concern in coupled GCMs without flux corrections even when atmosphere–ocean interactions are allowed only in a single ocean basin (Neelin et al. 1992).

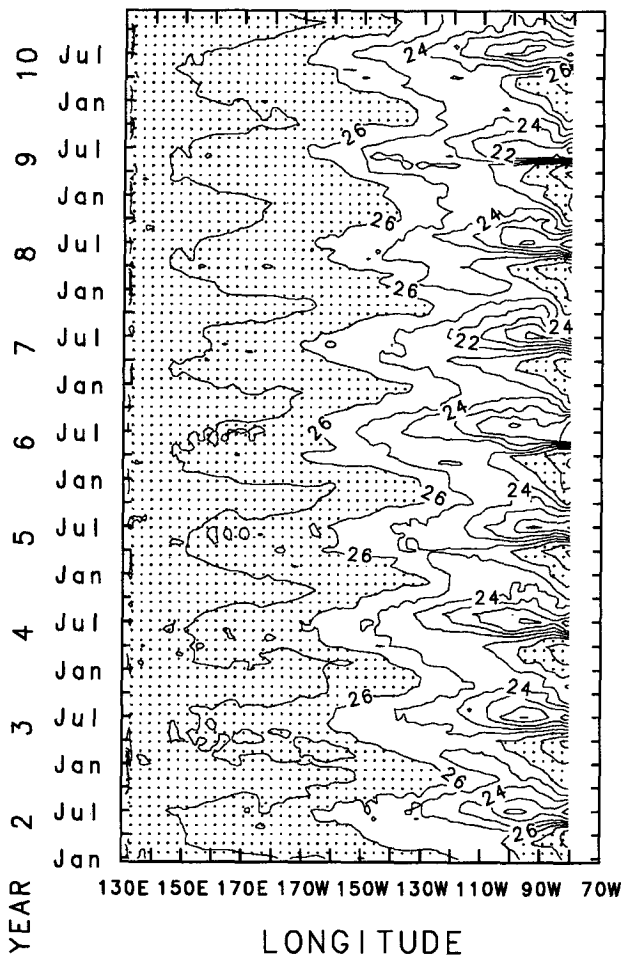

FIG. 3. Time variations of sea surface temperature at equator from a multiyear model simulation. Contour interval is 1°C. Note that the east–west temperature gradient is maintained throughout the year. The SSTs are warmer in the west and colder in the east, and the extent of cold water is largest in July.

3. Parallelization of the coupled model

The coupled GCM can be decomposed in several ways to achieve different levels of parallelism and reduction of wall-clock time. The first level of decomposition is based on *computational considerations,* the second on *functional considerations* specific to a climate code, and the third on *network considerations* specific to distributed applications. In this section we will restrict our attention to the idealized case of dedicated computers and networks.

a. Domain decomposition

Both the AGCM and OGCM are gridpoint models, for which spatial domain decomposition is straightforward. The method is based on dividing the model domain into subdomains, for which calculations are carried out concurrently. Depending on the number of processors available, these subdomains can be as large as a hemisphere or as small as a grid box. Parts of the code that require special consideration are the nonlocal calculations in both the AGCM and OGCM.

For distributed-memory architectures, interprocess communications are conveniently framed in terms of a message-passing arrangement, and it follows that the time spent sending messages in the resulting scheme compared with the time spent in doing calculations scales as the subdomain surface-to-volume ratio. For a three-dimensional fixed problem size, the speedup for the one-dimensional decomposition saturates at large processor count, while the speedup for the two-dimensional decomposition scales as the square root of the number of processors. For this reason, we have focused on a horizontal two-dimensional decomposition for both the AGCM and the OGCM. This approach carries the additional advantage that the AGCM/physics requires no explicit modification for parallelization, since its code does not require communication between model "columns." Preliminary results validate the utility of this paradigm (W. P. Dannevik 1992, personal communication).

b. Task decomposition

The AGCM and OGCM are two well-separated entities. The AGCM has two distinct components as discussed earlier in this paper: AGCM/physics and AGCM/dynamics.

In view of such an inherent modularity of the code, we start by decomposing the coupled GCM into three parts: AGCM/physics, AGCM/dynamics, and OGCM. The decomposed application can be enhanced by additional tasks. A master control program (MCP) can provide a convenient user interface, and monitor communications between different processes. Since data are produced by separate tasks, a dataset manager can be used to collect, organize, and dispose model output to a mass storage system (MSS). Analysis and

display of model results require a real-time visualization system, which can be managed by a separate task. Figure 4 shows the resulting distributed coupled GCM, in which modules can be run as separate tasks on the same computer or on different computers connected by a network.

The decomposition shown in Fig. 4 allows for task parallelization. Specifically, the AGCM/dynamics and OGCM can run concurrently since all forcing fields required by these model components are computed within the AGCM/physics. A substantial reduction in wall-clock time, therefore, can be achieved because two of the principal model components are running in parallel.

The natures of the AGCM/dynamics and AGCM/physics codes are very different. The former code is highly vectorizable, while the latter code is not. In particular, almost all calculations in the AGCM/physics involve grid points in single "model columns," so that minimal communications in the horizontal are needed. Thus, the code of this model component is highly suitable for MPPs. The decomposition in Fig. 4, therefore, suggests a possible superlinear speedup of the code; that is, the speedup achieved can be greater than the number of processors used. The speedup would be achieved by assigning different tasks to computers with architectures that are most efficient for their execution. An example of a heterogeneous environment with potential for superlinear speedup of model execution would have vector capabilities for the AGCM/dynamics and massively parallel capabilities for the AGCM/physics and OGCM.

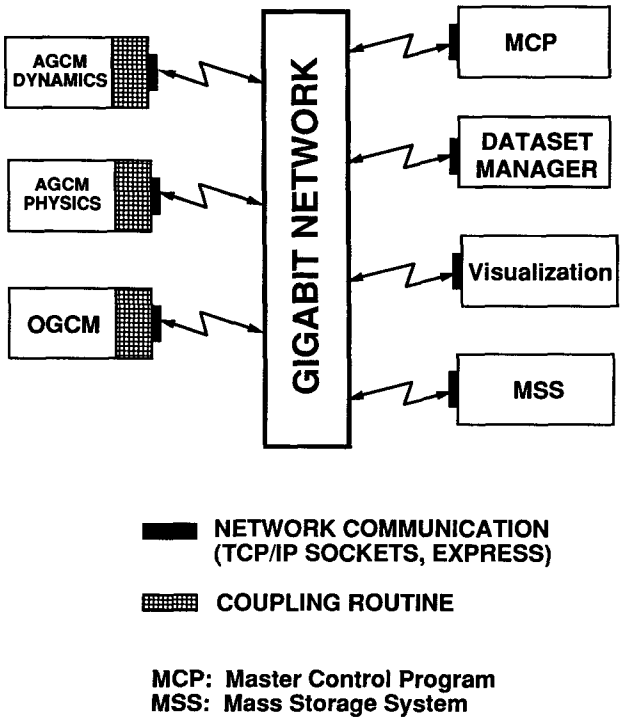

FIG. 4. Schematic diagram of the task-decomposed coupled GCM.

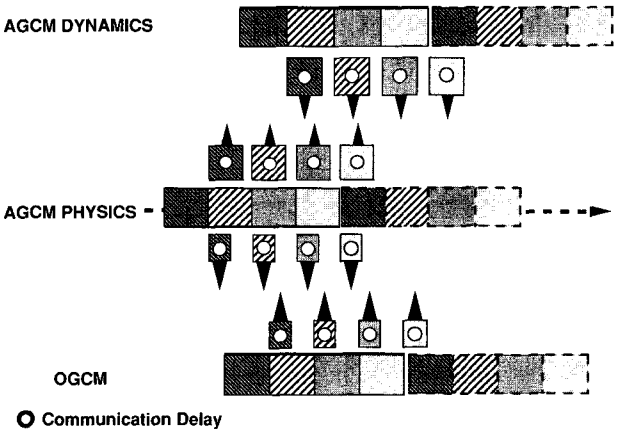

FIG. 5. Timing diagram of the distributed GCM with four I/O subdomains. Computation and communication for each I/O subdomain are depicted by different patterns.

c. *I/O decomposition*

The efficiency of a distributed application is affected by communication delays. We next discuss how to minimize deleterious effects of communication costs on efficiency by overlapping data transmission with model computation. The resulting scheme, which will be called *I/O decomposition,* is based on organizing computation and communication in such a way that exchange of data between different tasks is carried out in subdomains of the model domain. Instead of transfering data after one model component (AGCM/physics, AGCM/dynamics, or OGCM) completes calculation over the entire domain for a coupling interval, the data corresponding to each I/O subdomain is sent as soon as the model component completes calculation over the subdomain. The transmission time for this data, therefore, can be masked by computation in the next I/O subdomain in that component or task.

To illustrate the method, consider the case when there are four I/O subdomains (see Fig. 5). For simplicity, we assume that the three model components require the same amount of wall-clock time for calculation of one I/O subdomain. This can be accomplished by adjusting the number of processors assigned to each task. Note that AGCM/dynamics and OGCM begin their computations for an I/O subdomain as soon as the forcing fields from AGCM/physics are received. One can see that the transmission of data from AGCM/physics to AGCM/dynamics and OGCM for a particular I/O subdomain is concurrent with computation of the former model component for the next I/O subdomain. Similarly, the transmission of data from

AGCM/dynamics and OGCM to AGCM/physics is concurrent with computation for other I/O subdomains. In this way, communication can be masked by computation and all three components of the model can run in parallel. As a result, the wall-clock time required to run the model is reduced to that required by AGCM/physics only.

The organization of computation and communication shown in Fig. 5, however, is valid only if AGCM/physics is computed for every AGCM/dynamics time step. As mentioned already, this is not necessarily the case, and the AGCM/dynamics usually goes through several time steps before another AGCM/physics calculation begins. For example, let us consider that each calculation of the AGCM/physics is followed by n_D time steps of AGCM/dynamics. The finite differencing in the AGCM/dynamics implies that an I/O subdomain cannot be advanced more than one time step without updated information from neighboring subdomains. This information is not available until the forcing fields from AGCM/physics in the next I/O subdomain are transmitted and partial calculation of AGCM/dynamics in that I/O subdomain is completed.

To address this problem, we define I/O subdomains as latitude bands (i.e., longitude–height slabs of the model domain). From the physical point of view, partition of the model domain in latitude bands is natural since both atmospheric and oceanic large-scale motions—particularly the former—have a dominant west–east component. From the computational point of view, latitude bands eliminate consideration of boundary conditions in longitude since model variables are periodic in the west–east direction. The decomposition can become computationally inefficient for narrow bands, however, due to the increased surface-to-volume ratio and associated increases in the ratio between communication and computation time. Nevertheless, I/O subdomains do not have to be narrow enough for this to become an issue of concern.

Next, we subdivide each I/O subdomain into parts for which calculations can be carried out at different times in the simulation. These parts are also latitude bands, taken wide enough to ensure that the additional data required by the finite differencing in any one of them is within the adjacent ones. The fourth-order accuracy used in the finite differencing of the AGCM/dynamics in the current version of the UCLA AGCM implies that the latitude bands cannot be narrower than two grid points (Arakawa and Lamb 1977).

Let us consider a model configuration in which each AGCM/physics calculation is approximately balanced by four AGCM/dynamics calculations, so that we have taken $n_D = 4$. We start by analyzing the idealized case with no communication delays. In this case, it is enough to define three I/O subdomains. Since $n_D = 4$, we further divide each I/O subdomain into four latitude bands. For example, the I/O subdomains can be defined as the sectors 90°–30°S (I), 30°S–30°N (II), and 30°–90°N (III), so that each latitude band is 15° wide.

The resulting organization of the calculation is shown schematically in Fig. 6. In this figure, both I/O subdomains and latitude bands are organized from south (below) to north (above). The horizontal axis in Fig. 6 indicates the sequence of computation from left to right. Symbols along a vertical line, therefore, correspond to latitude bands that are computed concurrently. Roman numerals along the vertical axis denote the I/O subdomains, and symbols along a horizontal line refer to the same latitude band. Crosses denote AGCM/physics calculation, and Arabic numerals denote AGCM/dynamics calculation, with digits indicating time step of execution. We will assume that the AGCM/physics computes only one latitude band at any one time.

The process starts with AGCM/physics computing subdomain I. This is the usual way in which AGCM integrations begin. During this period, which corresponds to the first four crosses on the left-hand side of Fig. 6, the processors assigned to AGCM/dynamics are idle. After AGCM/physics finishes computing subdomain I, the corresponding forcing data are sent to AGCM/dynamics. The transfer of AGCM/physics data to AGCM/dynamics is indicated by the vertical dashed lines. After the transfer of data is completed, AGCM/dynamics can compute time step 1 for all latitude bands in subdomain I. Next, AGCM/dynamics computes time step 2 for all but the northernmost latitude band in this I/O subdomain, since this calculation requires information from subdomain II that is not yet available. AGCM/dynamics can also compute time step 3 for the two southernmost latitude bands, and time step 4 for the southernmost latitude band. Simultaneously, AGCM/physics is computing subdomain II. As the corresponding data become available, AGCM/dynamics starts computing time step 1 for all

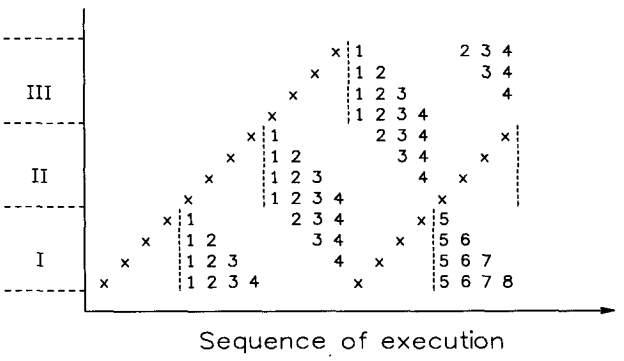

FIG. 6. Diagram showing the sequence of computation when each computation of the AGCM/physics is followed by four time steps of AGCM/dynamics and there are no communication delays. Roman numerals indicate I/O subdomains. Crosses denote execution of AGCM/physics. Arabic numerals denote the AGCM/dynamics time steps being executed.

latitude bands in this subdomain. Notice that, at this and all later times, all processors are active. Time step 2 for the northernmost latitude band in subdomain II cannot be computed, since this calculation requires information from subdomain III. At this time, however, there is enough data to compute the one latitude band in subdomain I for which time step 2 has not been computed. In this way, the number of latitude bands for which the AGCM/dynamics is computed concurrently is equal to the number of latitude bands inside an I/O subdomain except at the very first few time steps, with different latitude bands being computed as the model integration progresses forward in time.

The horizontal axis in Fig. 6 can be interpreted as wall-clock time if the computational load is well balanced; that is, the CPU time required to compute each latitude band is constant for both the AGCM/physics and the AGCM/dynamics, and the former model component is n_D times slower than the latter. Again, these relationships between timings can be accomplished by adjusting the number of processors assigned to each task. If the foregoing requirements are satisfied, Fig. 6 shows that there is no idle time for the processors assigned to AGCM/physics.

To apply the I/O decomposition to the communication between AGCM/physics and OGCM we must consider that for each time step of the OGCM a global operation is needed to advance in time the streamfunction of the vertically averaged flow after the local calculations are completed. The surface heat, momentum, and water fluxes produced by the AGCM/physics for a particular I/O subdomain are required for the local calculations of the OGCM in that subdomain. The SST needed by AGCM/physics for a particular I/O subdomain, on the other hand, is already available after the local calculations of the OGCM in that subdomain are completed. Figure 7 illustrates the sequence of computation in the case when one AGCM/physics calculation is performed every OGCM time step. In this figure we assume that equal times are required for one calculation of the AGCM/physics and the OGCM over their corresponding domains, and that the non-

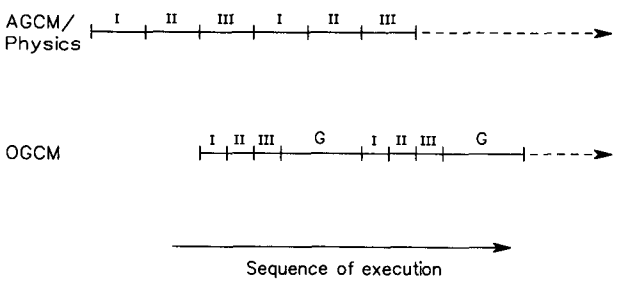

FIG. 7. Timing diagram of the computation of I/O subdomains in AGCM/physics and OGCM. The letter "G" denotes the global calculation of the transport streamfunction in the OGCM.

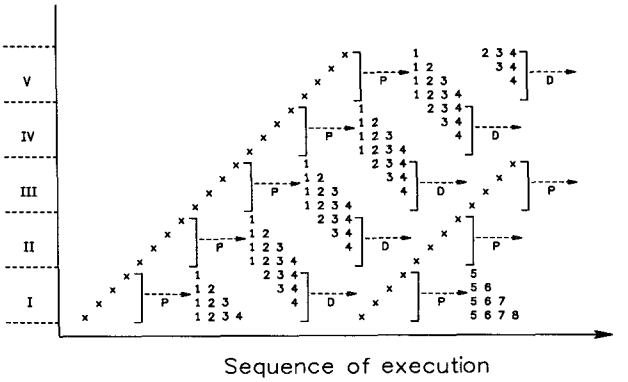

FIG. 8. As in Fig. 6 except in the case with network delays. Dashed arrows denote time when communication takes place, with letter "P" for the transmission of AGCM/physics data to AGCM/dynamics, and "D" for the transmission of AGCM/dynamics data to AGCM/physics.

local (global) calculations in the OGCM require the same amount of time as the local calculations. Once more, this can be accomplished by adjusting the number of processors assigned to each model component. It can be seen that in the case with no communication delays, with three I/O subdomains, the processors assigned to either model component will not become idle waiting for input from the other, and the calculation of the two components can take place concurrently.

The scheme can be extended to a geographically distributed computer environment with communication delays. As shown in Fig. 8, delays can be masked by computations by increasing the number of I/O subdomains. In this figure, the dashed arrows denote the time when transmission of data is taking place, with a letter "P" for the transmission of AGCM/physics data to AGCM/dynamics, and a letter "D" for the transmission of AGCM/dynamics data to AGCM/physics. It is assumed that both transmissions of data complete during one calculation of the AGCM/physics for an I/O subdomain.

Figures 6 and 8 show that when one calculation of AGCM/physics is followed by more than one time step of AGCM/dynamics, the wall-clock time needed for the latter model component for an I/O subdomain is twice that needed by the former model component. This implies that the number of I/O subdomains cannot be less than 3 and that it has to be increased according to the time needed for communication.

4. Network bandwidth requirements

To study the bandwidth requirements for efficient operation of the distributed GCM, we have to estimate the flow of data, the network latencies, and the time within which data transmission has to be carried out. In particular, for a wide-area network, the time required

for communication between remote locations can become a major issue even though in fiber signals travel at near the speed of light. We next obtain such estimates for the coupled atmosphere–ocean GCM.

First, we shall assume that AGCM/dynamics and OGCM are not slower than AGCM/physics. Namely, we expect that the wall-clock time required to complete calculation for each of the first two model components for their corresponding model domains will never be longer than that required by the latter model component for the model atmosphere. Notice that this is clearly the case for the current version of the GCM running in one processor. One can also envision a model configuration in which a very high resolution and CPU-demanding OGCM is coupled to a fast, low-resolution AGCM with a simplified physics module. Our assumption implies that the number of processors assigned to each model component is such that the AGCM/physics is the slowest in terms of wall-clock time.

Second, we shall consider an idealized model for the network. This model assumes that the sending and receiving computers can process the data as fast as the network delivers them. It also assumes that all data transmission can be completed without interruption due to flow control, and that transmissions are performed sequentially regardless of direction. For simplicity, we shall consider that the GCM is distributed between two supercomputer centers connected by a dedicated network, with AGCM/physics running at one of the centers and AGCM/dynamics and OGCM at the other (see Fig. 9). The data transmitted by the network during one calculation of the AGCM/physics for one I/O subdomain, therefore, include the forcing fields produced by AGCM/physics in the computation of the previous I/O subdomain and the fields sent to this model component by the AGCM/dynamics and OGCM for computation of another I/O subdomain (see Fig. 8). Therefore, to mask communication with computation, the following relationship must be satisfied,

$$\frac{T_{\text{phys}}}{n} + D_O + D_s \geqslant T_t + L_0 + E_c + D_c, \quad (1)$$

where T_{phys} is the time required to integrate AGCM/physics one time step for the model atmosphere, and n is the number of I/O subdomains. The other terms on the left-hand side of (1) represent the system overhead D_O and the CPU-resource contention delay D_s. The system overhead D_O accounts for the time required to prepare the AGCM/physics forcing fields to be transmitted to AGCM/dynamics and OGCM, and the time to process the data received from the latter model components. It can be written as

$$D_O = O_P \frac{N}{n}, \quad (2)$$

where O_P depends on the computer used to run AGCM/physics. The CPU-resource contention delay D_s depends on memory access delays (time spent waiting for the memory scheduler to swap the distributed task into memory from disk), I/O access delays (time spent waiting for I/O requests to be processed through the I/O subsystem), and CPU access delays (time spent waiting for the CPU scheduler to connect a process already in memory to a CPU). In this section we will be considering dedicated computers and therefore $D_s = 0$. The first term on the right-hand side of (1) is transmittal time:

$$T_t = \frac{N + N_0}{nB} \approx (1 + r)\frac{N}{nB}, \quad (3)$$

where N is the total size of data exchanged between AGCM/physics, AGCM/dynamics, and OGCM; N_0 is the additional information sent for network protocol functionality; r is the fractional protocol overhead; and B is bandwidth. The second term is latency:

$$L_0 = \frac{2d}{c_f} + D_p, \quad (4)$$

where d is the distance between the two supercomputers between which the application is distributed; c_f is the speed of light in fiber; and D_p represents propagation delays due to hardware devices on the network. The third term is the error correction delay,

$$E_c = \left[2L_0 + \frac{P}{B}\right]\left[\frac{R_p(N/n)}{P}\right], \quad (5)$$

where P is packet size and R_p is packet error rate. The last term D_c represents network contention delay, which depends on the length of network queues that build up at intermediate network routing points. For a dedicated network considered in this section, $D_c = 0$.

The network-related parameters and their estimated values for the projected Gigabit Network are listed in

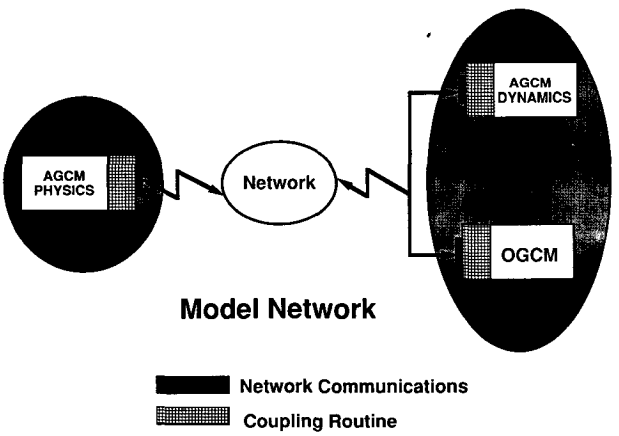

FIG. 9. The configuration of the distributed GCM used to estimate bandwidth requirements.

Table 2 (Moore 1991). In this paper we shall take d = 2000 km, which is the approximate distance between LANL and SDSC. Equation (1) can now be used to determine the minimum bandwidth required for a particular model resolution and network configuration.

Some important implications of (1) can be readily illustrated by considering a simplified case. For situations with dedicated networks and dedicated computers widely separated from each other, the dominant balance is between the computation time, transmittal time, and round-trip latency; that is, (1) can be approximated by

$$\frac{T_{\text{phys}}}{n} \geq \frac{N}{nB} + L_0. \tag{6}$$

According to this equation, the bandwidth has to satisfy the following relationship:

$$B \geq \frac{N/n}{(T_{\text{phys}}/n) - L_0}. \tag{7}$$

The bandwidth efficiency as defined by Moore (1991) is

$$\eta_B = \frac{\text{transmittal time}}{\text{transmittal time} + \text{delay time}}.$$

In the present simplified case, bandwidth efficiency is given by

$$\eta_B = \frac{N/nB}{(N/nB) + L_0}. \tag{8}$$

Substituting the value of the minimum required bandwidth B given by (7) gives

$$\eta_B = 1 - \frac{nL_0}{T_{\text{phys}}}, \tag{9}$$

which implies that larger n results into lower bandwidth efficiency. This is because the cost of latency is incurred for each transmission, while the data exchanged in each transmission is smaller in size. When n is so large that T_{phys}/n is smaller than L_0 there is no possibility of avoiding CPU idle time since latency becomes larger than computation time. These restrictions result in an upper limit on the possible values of n.

There is also a lower limit on n if we require that the data produced by AGCM/dynamics and OGCM for a particular I/O subdomain are received by AGCM/physics before the next computation of AGCM/physics for that subdomain begins. Together with the upper limit mentioned earlier, there is only a limited range of possible choices of n. Here we shall be concerned only with the upper limit, since the lower limit requirement can be easily satisfied.

TABLE 2. Estimated values of parameters for the Gigabit Network.

Parameter	Definition	Value
r	Fractional protocol overhead	0.05
c_f	Speed of light in fiber	2×10^8 m s^{-1}
P	Packet size	5.1×10^5 bit
R_p	Packet error rate	5×10^{-5} error per packet
D_p	Propagation delays	7×10^{-3} s

TABLE 3. Data exchanged between AGCM/physics, AGCM/dynamics, and OGCM per coupling interval in megabits.

AGCM resolution	OGCM horizontal resolution		
	1° × 1°	½° × ½°	⅓° × ⅓°
4° × 5° 9 layers	33	82	165
4° × 5° 15 layers	49	99	181
2° × 2.5° 17 layers	136	186	269
2° × 2.5° 29 layers	263	313	396
1° × 1.25° 33 layers	920	970	1053
1° × 1.25° 57 layers	1933	1983	2066

With all terms in (1) taken into consideration, the minimum bandwidth is given by

$$B_{\min} = \frac{(1 + r + R_p)N/n}{(T_{\text{phys}}/n) + D_O + D_s - L_0 - 2L_0 \dfrac{R_p(N/n)}{P} - D_c}, \tag{10}$$

if

$$\frac{T_{\text{phys}}}{n} + D_O + D_s > L_0 + 2L_0 \frac{R_p(N/n)}{P} + D_c. \tag{11}$$

The bandwidth efficiency in this case is given by

$$\eta_B = \frac{(1 + r)(N/nB)}{(1 + r)(N/nB) + L_0 + \left(2L_0 + \dfrac{P}{B}\right)\dfrac{R_p(N/n)}{P} + D_c}. \tag{12}$$

For the nine-layer standard resolution AGCM using one processor on a CRAY Y-MP, T_{phys} is approximately 3.4 s. The amount of data exchanged for several configurations of the coupled GCM is given in Table 3. If we take n = 5, the estimated minimum bandwidth given by (7) to run the 4° × 5° nine-layer AGCM coupled to the 1° × 1° OGCM distributed over two supercomputers 2000 km apart is 10 Mbit s^{-1}. This value is larger than the maximum bandwidth of a T1 network (1.5 Mbit s^{-1}), but smaller than that for a T3 network (45 Mbit s^{-1}).

Notice that this estimate is based on timings for a GCM with marginal resolution for climate studies and

with the current vectorized version of the code running in single processors of CRAY Y-MPs. As increases in computing power allow for higher model resolutions or as the model code is further parallelized, the minimum required bandwidth will increase. To consider the dependence of minimum bandwidth on model resolution, we have to define how this will be enhanced with increased computing power. To that effect, examination of (9) reveals that the bandwidth efficiency for a particular network configuration (L_0 fixed) is approximately constant for fixed T_{phys}/n. Therefore, we shall assume that model resolution is increased with higher computing power in such a way that T_{phys} remains constant (with n held fixed). Since

$$T_{\text{phys}} = \frac{S_p}{R_c}, \tag{13}$$

where S_p is the number of floating-point operations per AGCM/physics calculation and R_c is the execution rate of the corresponding processors, a constant T_{phys} implies that S_p increases linearly with R_c. The relationship between the size of data exchanged by model components and S_p is not linear, on the other hand, because some operations (matrix inversion, subgrid-scale physics) tend to increase more than linearly with model resolution. Considering the special parameterizations used in the UCLA coupled GCM, a reasonable assumption for the relationship between N and S_p is

$$N = N_0\left(\frac{S_p}{S_{p0}}\right)^{2/3}. \tag{14}$$

Since the CPU overhead is expected to decrease with faster computers, we shall assume that

$$O_P = O_{P0}\left(\frac{R_{c0}}{R_c}\right). \tag{15}$$

The reference values N_0, S_{p0}, O_{P0}, and R_{c0} correspond to the 9-layer standard horizontal resolution version of the AGCM coupled to the 15-layer standard horizontal resolution version of the global OGCM running in one processor of a CRAY Y-MP (N_0 = 33 Mbit, $S_{p0} = 4.42 \times 10^8$ operations, $O_{P0} = 7.2 \times 10^{-10}$ s bit^{-1}, R_{c0} = 130 Mflops).

Using Eqs. (1), (13), (14), and (15), we can obtain B and N/n as functions of T_{phys}/n and R_c. So far, we have not considered the additional data transmissions needed for real-time visualization of GCM output. In principle, such an estimate will depend on the particular climate problem under investigation. For example, one might want to visualize simulated fields in the global domain with reduced spatial and temporal resolutions, or fields in selected geographical regions with the highest available spatial and temporal resolutions. In view of this uncertainty, we have taken the amount of data transmitted for visualization as that corresponding to all the prognostic and standard diagnostic fields that are currently stored for off-line analysis. Although this may seem to be an overestimate, current visualization technology allows for the simultaneous visualization of several variables in different windows of a workstation or overlaid on the same plot.

We calculated the amount of data exchanged and the minimum bandwidth required for two cases depending on whether real-time visualization of model results is included. If real-time visualization is included, we take N_0 = 200 Mbit in (14). Our results are plotted in Fig. 10 for T_{phys} = 3.4 s and n = 5. In this case, we obtain $\eta_B \approx 0.9$ according to (12). The left-hand side of Fig. 10 corresponds to the 9-layer standard horizontal resolution version of the AGCM coupled to the 15-layer standard horizontal version of the global OGCM running in one processor of a CRAY Y-MP. One can see that the bandwidth requirement of the coupled GCM increases rapidly with the increase in computing power. In the figure we also indicate the estimate for running the model with the two most immediate upgrades in resolution. According to Fig. 10, the computing power required by the tropospheric–stratospheric version of the AGCM with double the current resolution in both the horizontal and the vertical (29 layers, 2° latitude × 2.5° longitude) coupled to the OGCM with 1/2° horizontal resolution in such a way that T_{phys} = 3.4 s is approximately 4 Gflops, which can be achieved on the Intel Touchstone Delta. It is clear that a network with a gigabit per second bandwidth is needed to run this configuration with real-time visualization, although the model itself demands a network

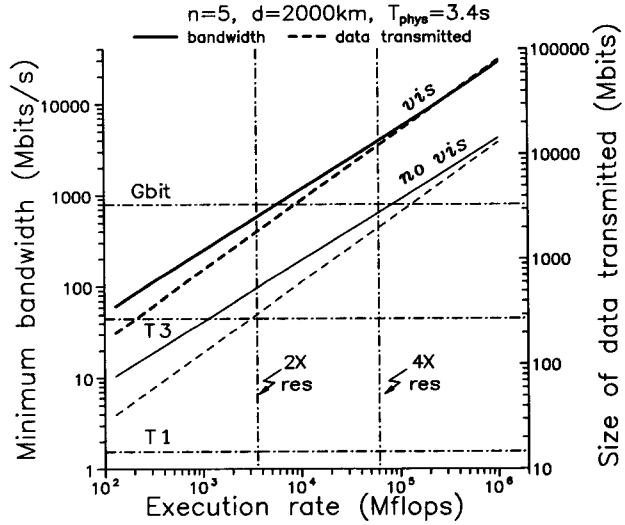

FIG. 10. Dependencies of minimum bandwidth and data exchanged across the network on computing speed for the distributed GCM. A network 2000 km in length and the use of five I/O subdomains were assumed for this calculation. The solid curves are the minimum bandwidths with scales on the left; the dashed curves are the size of data exchanged across the network with scales on the right. The thinner curves are for the case where only data exchanges between AGCM/physics, AGCM/dynamics, and OGCM are considered; the thicker curves are for the case where data required for real-time visualization are also included.

TABLE 4a. The CPU, communication and wall-clock time per simulated day, and the effective bandwidth for the task-decomposed GCM running in a dedicated CRAY Y-MP, with AGCM/physics, AGCM/dynamics, and OGCM exchanging data every simulated hour.

Experiment	CPU time (s)	Communication time (s)	Wall-clock time (s)	Effective bandwidth (Mbyte s^{-1})
Dedicated parallel	218.94	0.82	175.6	40.0
Dedicated sequential	217.29	0.78	235.8	41.6

bandwidth broader than that available in the existing T3 network. On the other hand, if the AGCM resolution is doubled once more (59 layers, 1° latitude $\times$ 1.25° longitude) and the horizontal resolution of the global OGCM is $1/3° \times 1/3°$, then a computing power of 60 Gflops is needed for $T_{phys} = 3.4$ s and the network bandwidth required by the model alone will be near that of the Gigabit Network. In all cases, the application operates at a network bandwidth efficiency of about 90%.

We emphasize that the highest resolution considered for the AGCM is comparable to that used for operational weather prediction at leading centers. The highest resolution considered for the OGCM allows for "eddy-resolving" capabilities, which are achievable in current limited-area ocean models (e.g., Semtner and Chervin 1988). The harnessing of computing power allowed by the network has the potential to achieve execution speed suitable for climate studies with a coupled atmosphere–ocean model.

5. Preliminary experiments

We have performed a series of preliminary experiments to gain insight into the performance of the coupled GCM distributed across a network. These experiments used the task-decomposed GCM without I/O or domain decompositions. Synchronous message passing with TCP/IP sockets was used for interprocess communication. To determine communication time, we measured the period between calls to a communication library routine and when execution control is returned for each message. This period consists of the time required by library routines, protocol, data transfer, and resource contention delays. In general there will be two different values for the length of the period depending on whether it is measured on the sending or the receiving end of the message. We use the smaller of those two values to represent the communication time so that delays due to CPU and memory contentions are excluded. We also measured CPU and wall-clock time for model execution.

The task-decomposed GCM was run on the SDSC CRAY Y-MP during both dedicated and production time. In these runs, the three model components—AGCM/physics, AGCM/dynamics, and OGCM—exchanged data every simulated hour. In some runs, the AGCM/dynamics and OGCM were allowed to run in parallel ("parallel runs"), and in others they were forced to run sequentially ("sequential runs"). The effective bandwidth was calculated as the ratio between the total amount of data exchanged during the run and the total communication time. The results obtained in these experiments are summarized in Table 4. The time used in the initialization of the model components is not included in the values shown in this table.

For the dedicated parallel run, the CPU time for a one-day simulation was 218.94 s, and the wall-clock time was 175.6 s. The wall-clock time for the dedicated sequential run, on the other hand, was 235.8 s. Therefore, a one-fourth reduction of wall-clock time was achieved by having the AGCM/dynamics and OGCM executing in parallel. The effective bandwidth for the interprocess communication on the CRAY Y-MP in these cases is about 40 Mbyte s^{-1}.

In the production parallel run, the wall-clock time was about seven times that in the dedicated parallel run. The wall-clock time for the production sequential run was about triple that in the production sequential run. The increases in wall-clock time are mainly due to memory access delays, which depend strongly on the load of the system and obliterate the benefits of parallelization. The effective bandwidth in these runs decreased to about 28 Mbyte s^{-1}.

Additional tests were made with a simple test program to understand whether the bandwidth obtained

TABLE 4b. The CPU, communication and wall-clock time per simulated day, and the effective bandwidth for the task-decomposed GCM running in a CRAY Y-MP during production time, with AGCM/physics, AGCM/dynamics, and OGCM exchanging data every simulated hour.

Experiment	CPU time (s)	Communication time (s)	Wall-clock time (s)	Effective bandwidth (Mbyte s^{-1})
Production parallel	220.16	1.73	1228.7	27.2
Production sequential	220.45	1.10	795.8	29.2

TABLE 5a. Bandwidths as a function of message size and buffer size on a dedicated system (Mbyte s^{-1}).

Message size (kbyte)	Buffer size 16 kbyte	32 kbyte	64 kbyte	128 kbyte	256 kbyte
64	24.8	42.5	53.8	69.3	69.4
32	24.6	42.4	47.3	46.6	54.8
16	24.5	37.8	43.0		
8	17.9	30.0	31.0		

TABLE 5c. Bandwidths as a function of window size and buffer size for a 64-kbyte message on a dedicated system (Mbyte s^{-1}).

Window scaling factor	Buffer size 64 kbyte	128 kbyte	256 kbyte	512 kbyte
4	48.9	67.4	83.4	76.5
2	48.9	67.1	82.1	80.3
1	48.9	66.9	66.8	66.4
0	53.8	69.3	69.4	

for communication between processors in the CRAY Y-MP was reasonable and to explore the possibility for improvements. We compared the achievable bandwidth as a function of message size, kernel-buffer size for the TCP/IP protocol, and the number of messages in flight allowed between acknowledgments (window scaling factor). Table 5 gives the bandwidth achieved with the test program for both dedicated use of the CRAY Y-MP and for production use on a fully utilized system under the UNICOS operating system version 6.1.5. The values quoted are averages of the read and write rates.

The achievable bandwidth for dedicated use of the CRAY Y-MP depends on both the message size and buffer size. From Table 5a, for a fixed message size, increasing the buffer size increases the bandwidth until the buffer size is about twice the message size. For fixed buffer size, increasing the message size increases the bandwidth until the message size equals the buffer size. The bandwidth achieved in the GCM experiments showed similar dependencies on the message size. For the dedicated run, an average bandwidth of 20.1 Mbyte s^{-1} was obtained for a 7.8-kbyte message, and 37.4 Mbyte s^{-1} for a 25-kbyte message. These are consistent with findings in the simple experiments for the default buffer size of 32 kbyte.

Table 5c indicates that greater bandwidths are generally obtained by increasing the window scaling factor up to 2. In this case, one can obtain a bandwidth of 82 Mbyte s^{-1} for 256-kbyte buffers.

When the same bandwidth performance tests are run on a heavily loaded system, dramatically lower bandwidths are observed. Table 5b shows similar message and buffer size dependencies for the bandwidth, but with maximum values about a factor of 3–4 lower. Allowing multiple messages in flight on the heavily loaded system provided no bandwidth enhancement (Table 5d). At the San Diego Supercomputer Center (SDSC) the I/O load to disk averages 17 Mbyte s^{-1}. This causes contention for I/O resources that completely dominates any time saved by keeping multiple messages in flight between acknowledgments.

We also conducted an experiment in which the GCM was run using two CRAY computers connected by a T1 link (NSFNet), with the OGCM running on one processor of the CRAY Y-MP8/864 at SDSC in La Jolla, California, and the AGCM running on one processor of the CRAY Y-MP8/512 at the National Center for Atmospheric Research (NCAR) in Boulder, Colorado. To our knowledge, this is the first time that two supercomputers at different locations have been used for a GCM integration. In this experiment, the AGCM and the OGCM were run sequentially. Only AGCM/physics and OGCM exchanged data over the network, and the data were exchanged every six simulated hours. For this experiment, the communication time included the delays due to CPU and memory contentions. The result of this experiment is shown in Table 6.

The distributed run between SDSC and NCAR had two remarkable aspects. First, the wall-clock time per simulated day (2593 s) was dominated by the communication time (1930 s), which is much larger than the CPU time (193 s). Second, the standard deviation of the communication time (1220 s, 10 samples) was comparable to the mean value (1930 s). Part of the increase in wall-clock time and variance in communication time comes from the contention for system resources at both locations since the experiment was

TABLE 5b. Bandwidths as a function of message size and buffer size on a heavily loaded system (Mbyte s^{-1}).

Message size (kbyte)	Buffer size 16 kbyte	32 kbyte	64 kbyte	128 kbyte	256 kbyte
64	6.3	12.5	16.0	14.6	17.9
32	6.3	14.5	13.3	12.6	14.6
16	7.1	8.8	8.0		
8	4.4	6.3	6.6		

TABLE 5d. Bandwidths as a function of window size and buffer size for a 64-kbyte message on a heavily loaded system (Mbyte s^{-1}).

Window scaling factor	Buffer size 64 kbyte	128 kbyte	256 kbyte	512 kbyte
4	10.6	15.4	15.8	15.0
2	11.1	15.9	15.5	17.3
1	13.1	16.9	13.4	15.6
0	16.0	14.6	17.9	

TABLE 6. The CPU, communication and wall-clock time per simulated day, and the effective bandwidth for the distributed GCM with AGCM running in a CRAY Y-MP at NCAR and OGCM running in a CRAY Y-MP at SDSC. Data are exchnaged every six simulated hours.

Experiment	CPU time (s)	Communication time (s)	Wall-clock time (s)	Effective bandwidth (Mbyte s^{-1})
Distributed	193	1930	2593	0.02

conducted with both computers in production mode. Simple experiments performed at SDSC (B. Chinoy 1992, personal communication) showed that network delays associated with store and forward of messages also accounted for a large percentage of this delay. As a result, the effective bandwidth available to the distributed GCM was only 0.02 Mbyte s^{-1}.

These preliminary experiments show that to achieve high effective bandwidth use across a gigabit-per-second link with the present CRAY supercomputer technology, the distributed application will have to be run on dedicated systems. The development of sophisticated job-scheduling algorithms will be needed to avoid memory access delays and I/O resource contention delays when such applications are run in competition with production job mixes. The distributed experiment clearly shows the need for a high-speed network with minimum scheduling delays for efficient operation of the distributed climate model.

6. Conclusions

Computer models of climate epitomize grand challenge applications of computer resources. The models are used to study major environmental problems that can affect human health, economy, and behavior. They demand large amounts of computer resources to run the long simulations required to study climate phenomena, and need mass storage systems for the large outputs produced in the simulations, as well as advanced visualization systems for analyses of the results.

New technologies are developing faster and more powerful computers connected by high-speed networks. Massively parallel processors with an aggregate computational speed on the order of gigaflops will soon be connected by networks with gigabit-per-second bandwidth. For grand challenge applications, one can think of networks of supercomputers with diverse architectures as a "metacomputer," so that corresponding codes demand special procedures for their optimization.

We have analyzed methods for distribution of a climate model consisting of an atmospheric GCM coupled to an oceanic GCM over processors that can be of heterogeneous architectures and located at large distances from each other. To obtain a distributed application running efficiently across high-speed networks, we considered three levels of decomposition.

The first level is based on standard *domain decomposition.* In this method, the model domain is partitioned into subdomains that are assigned to processors in which computations are carried out concurrently. This is relatively straightforward since both the AGCM and OGCM are gridpoint models. There are, nevertheless, "nonlocal" calculations that affect performance. These "nonlocal" calculations arise due to the exacerbated demands on computational stability of the AGCM placed by the convergence of meridians toward the poles and by the solution of the elliptic boundary value problem due to the rigid-lid assumption in the OGCM.

The second level of decomposition is specific to the climate model code. It is based on considering the model's major components (AGCM/physics, AGCM/dynamics, and OGCM) as separate tasks (*task decomposition*). The separation between AGCM and OGCM is natural since the models differ substantially from each other. The split of the AGCM between AGCM/physics and AGCM/dynamics already exists in most codes running in single-processor computers, where these model components are executed sequentially albeit not necessarily with the same frequency. Task decomposition allows for the parallelization of the AGCM/dynamics and OGCM, but introduces delays associated with increased communication between tasks.

To mask communication with computation, we introduce a third level of decomposition (*I/O decomposition*). This novel decomposition is motivated by our interest in an efficient distribution over wide-area high-speed networks, but it can also be used for distribution in a single MPP. At this level of decomposition, computation and communication are organized in such a way that the exchange of data between the different tasks is carried out in subdomains of the model domain. In this organization, calculations for the model tasks are carried out concurrently, although not necessarily for the same I/O subdomain at the same time step. The final result is a coupled GCM consisting of three domain-decomposed tasks, running in parallel and such that the only wall-clock time required is that spent in AGCM/physics.

We also analyzed the network bandwidth required for the efficient operation of the application. For this analysis, we assumed that during computation of the AGCM/physics for one I/O subdomain the network transmits an amount of data equivalent to that produced and received by this model component for each I/O subdomain. Since network bandwidth efficiency is mostly determined by the time required for transmission and latency associated with the finite speed of signal propagation through the network, we fixed the time required for transmission to a value that gives high efficiency for the appropriate value of the latency corresponding to a particular network configuration. Since the CPU time required for the calculation of AGCM/physics can be expressed in terms of model

resolution and power of the computer used for the calculation, the network bandwidth for efficient operation of the distributed application can be determined. We found that a computer environment based on nodes equivalent to the Intel Touchstone Delta will require a bandwidth approaching that of the Gigabit Network for efficient operation of a coupled GCM with a resolution just about double that used in current studies if output is visualized in real time. Otherwise, the amount of data exchanged for computation alone will require a gigabit bandwidth if the resolution of the model is further doubled.

The decomposition of the AGCM into two separate tasks—dynamics and physics—implies the need for communication of three-dimensional data. In principle, increased requirements in communication time are detrimental to the efficiency of the code. We emphasize, however, that the high demands on CPU time of AGCM/physics have resulted in a tendency to compute it less frequently than the AGCM/dynamics in current climate models. Ideally, both physics and dynamics calculations should be performed for each model time step. The task decomposition will allow for improvements in these respects by running AGCM/physics and AGCM/dynamics on computers with architectures selected to best fit the distinct characteristics of their codes. By running AGCM/physics on an MPP, for example, one can achieve a high degree of parallelism without extra communication costs by decomposing the model domain down to single columns. For AGCM/dynamics, on the other hand, one can achieve a high degree of efficiency with fewer but powerful vector processors. In this way, one can approach the optimal configuration of the model, in which the wall-clock time spent in the two model components is approximately balanced.

It is also possible that increased understanding of the subgrid-scale processes might suggest simpler and more efficient algorithms for their parameterization (Moorthi and Suarez 1992). In this scenario, more frequent calculations of the AGCM/physics would be affordable. This will imply the disappearance of the AGCM/physics as a separate model component, and its merging with the AGCM/dynamics and eventually with the OGCM. At that stage, only domain decomposition remains meaningful, and the model calculation will be performed in columns from the top of the atmosphere to the bottom of the ocean.

Preliminary tests for performance of the distributed application were performed. These used a CRAY Y-MP in both shared and dedicated mode, as well as CRAY Y-MPs at two remote supercomputer centers (NCAR and SDSC) linked by the NSFnet (T1 speed). Future tests will involve heterogeneous computer environments connected by high-speed networks, eventually with gigabit per second bandwidth.

Our results demonstrate the potential for major impact on scientific computing of the metacomputer concept made possible by high-speed networks. In a realistic environment, however, the distributed application will have to compete with other processes on individual computers. The initialization and synchronization of processes require sophisticated enough interprocess communication tools, as well as special considerations from system administrators. These issues will have to be addressed if distributed computing across wide-area networks—that is, the metacomputer concept—is to become a viable way of scientific research for grand challenge applications.

Acknowledgments. This work was supported by NSF and DARPA under Cooperative Agreement NCR-8919038 with the Corporation for National Research Initiatives. Development of the coupled atmosphere–ocean model was partially funded by ONR under Grant N00014-89-J-1845. The San Diego Supercomputer Center is acknowledged for providing computing resources and technical support. The authors would like to thank P. Messina and R. Binder for their support to this work and for useful comments. We are also grateful to the editor (M. Navon) and two anonymous reviewers for the perceptive comments on the original version of this manuscript.

REFERENCES

Arakawa, A., and V. R. Lamb, 1977: Computational design of the basic dynamical processes of the UCLA general circulation model. *Methods Comput. Phys.*, **17**, 173–265.

Bryan, K., 1969: A numerical method for the study of the circulation of the world ocean. *J. Comput. Phys.*, **4**, 347–376.

——, and M. D. Cox, 1972: An approximate equation of state for numerical models of ocean circulation. *J. Phys. Oceanogr.*, **2**, 510–514.

Chervin, R. M., and A. J. Semtner, Jr., 1988: An ocean modeling system for supercomputer architectures of the 1990s. *Proc. of the NATO Advanced Research Workshop on Climate-Ocean Interaction*, Oxford, NATO, 87–97.

Cox, M. D., 1984: A primitive equation, 3-dimensional model of the ocean. GFDL Ocean Group Tech. Rep. No. 1, 143 pp.

Hoffmann, G.-R., and D. K. Maretis, 1990: *The Dawn of Massively Parallel Processing in Meteorology.* Springer-Verlag, 200 pp.

Ma, C.-C., Y. Chao, C. R. Mechoso, W. M. Weibel, and D. Halpern, 1991: Comparison of vertical mixing schemes for ocean general circulation models. Preprints, *Fifth Conf. on Climate Variations*, Denver, Amer. Meteor. Soc., 388–391.

Mechoso, C. R., A. Kitoh, S. Moorthi, and A. Arakawa, 1987: Numerical simulations of the atmospheric response to a sea surface temperature anomaly over the equatorial eastern Pacific Ocean. *Mon. Wea. Rev.*, **115**, 2936–2956.

Moore, R. W., 1991: Distributing applications across wide area networks. *Proc. of the Second Gigabit Testbed Workshop*, Washington, D.C., CNRI, 36–43.

Moorthi, S., and M. J. Suarez, 1992: Relaxed Arakawa–Schubert: A parameterization of moist convection for general circulation models. *Mon. Wea. Rev.*, **120**, 978–1002.

Neelin, J. D., and collaborators, 1992: Tropical air–sea interaction in general circulation models. *Climate Dyn.*, **7**, 73–104.

Semtner, A. J., and R. M. Chervin, 1988: A simulation of the global ocean circulation with resolved eddies. *J. Geophys. Res.*, **93**, 15 502–15 522.

UNESCO, 1981: Tenth report of the joint panel on oceanographic tables and standards. UNESCO Technical Papers in Marine Science No. 36, UNESCO, Paris,