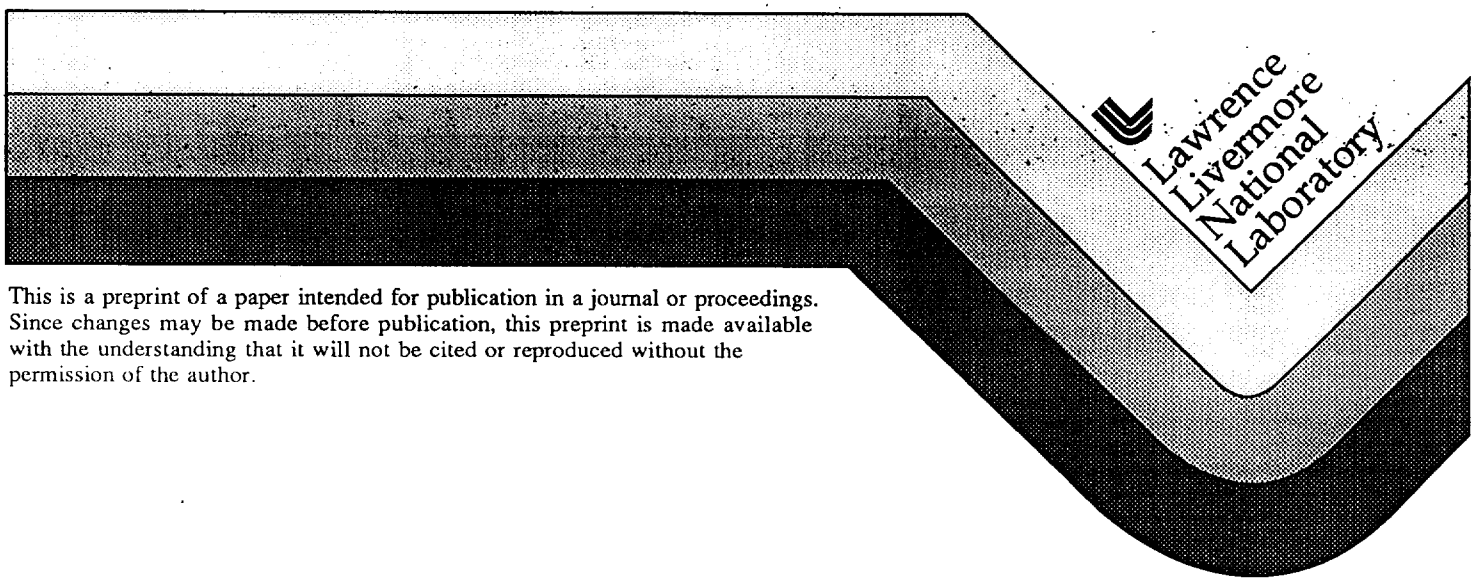
UCRL-JC-125558
PREPRINT

Comprehensive Climate Systems Modeling on Massively Parallel Computers

M.F. Wehner, P.G. Eltgroth, A.A. Mirin, P.B. Duffy, K.G. Caldeira,
J.H. Bolstad, H. Wang, C.M. Matarazzo and U.E. Creach

This paper was prepared for submittal to the
Western Multiconference '97
Society for Computer Simulation
Phoenix, AZ
January 12-15, 1997

October 1996



Comprehensive Climate System Modeling on Massively Parallel Computers

M.F. Wehner, P.G. Eltgroth, A.A. Mirin, P.B. Duffy, K.G. Caldeira,
J. H. Bqlstad, H. Wang, C.M. Matarazzo, U.E. Creach

Climate Systems Modeling Group
Lawrence Livermore National Laboratory
Livermore, CA 94551

A better understanding of both natural and human induced changes to the Earth's climate is necessary for policy makers to make informed decisions regarding energy usage and other greenhouse gas producing activities. To achieve this, substantial increases in the sophistication of climate models are required. Coupling between the climate subsystems of the atmosphere, oceans, cryosphere and biosphere is only now beginning to be explored in global models. The enormous computational expense of such models is one significant factor limiting progress.

A comprehensive climate system model targeted to distributed memory massively parallel processing (MPP) computers is under development at the Lawrence Livermore National Laboratory. This class of computers promises the necessary computational power to permit the timely execution of climate models of substantially more sophistication than current generation models. Our strategy for achieving high performance on large numbers of processors is to exploit the multiple layers of parallelism naturally contained within highly coupled global climate models. The centerpiece of this strategy is the concurrent execution of multiple independently parallelized components of the climate system model. This methodology allows the assignment of an arbitrary number of processors to each of the major climate subsystems. Hence, a higher total number of processors may be efficiently used. Furthermore, load imbalances arising from the coupling of submodels may be minimized by adjusting the distribution of processors among the submodels.

Comprehensive Climate System Modeling

Reliable prediction of future changes in climate and their impacts on human health and ecosystem functions requires validated coupled climate system models. The major components of the climate system include the atmosphere, oceans, land surface, and sea ice. Initial successes in linking the most comprehensive models of the atmosphere and ocean (general circulation models) are beginning to provide a more sound capability of predicting future climatic change [5]. Before these predictions will be relied on for managing policies regarding environment and energy usage, a more complete assessment of uncertainties associated with the predictions is needed.

The computational expense of running climate models and complex ecological models impedes efforts to evaluate and improve them. Usually a limited number of experiments are affordable. Even on the fastest computers, it can take many months to complete long climate change experiments. Exploitation of massively parallel processing (MPP) technology in stand alone atmospheric and oceanic GCMs has enabled the calculation of heretofore unresolvable phenomena and has significantly advanced the state of the science [11]. Coupled ocean-atmosphere-land models are sure to benefit from similar increases in computational performance. It is clear however, that the attraction of applying massively parallel computers to environmental problems is primarily practical. They can speed validation and improvement of current models, and will make feasible higher spatial resolution studies, which are needed for most environmental assessments. However, the difficulty and expense of porting existing algorithms to this class of machine can be significant.

Parallelism

The issues in porting atmospheric and oceanic GCMs to MPPs are well explored. A principal barrier to efficient use of massively parallel computers in any application is that unless a careful strategy is developed, processors may remain idle much of the time, as they wait for other processors to complete some task. This "load imbalance" is often a result of the many

branches that a sophisticated model may take in the course of a calculation. This problem of load imbalance, although simple to understand, may be difficult to avoid even in a single component model. A further difficulty that limits how many processors can be effectively utilized by a model is that the communication of information from one subdomain to another takes an increasing fraction of the total computational time as the size of the individual domains shrinks. Thus, doubling the number of processors will not necessarily halve the overall execution time. For a very large number of processors, adding additional processors may in fact increase the total execution time because the added communication time will not be entirely offset by the reduction in number of operations required of each processor.

Specific parallelization strategies are best dictated by the nature of the numerical schemes used to solve the model equations. Of chief importance in the design of a parallelization strategy for a climate system model are the approximations of the hydrodynamical equations in the atmospheric and oceanic components. For models based on finite differenced hydrodynamics, two dimensional domain decomposition strategies have proven to possess favorable scaling properties [3,13,14]. Originally proposed by Richardson [9] in 1922, this method covers the globe with subdomains of limited latitudinal and longitudinal extent. Since the atmosphere and ocean are highly stratified, each subdomain contains the entire vertical domain to avoid communication and load imbalances associated with the solution of stiff equations along that direction. The data to be communicated between processors lie near the perimeter of the subdomains. Hence, this decomposition scales well with increased horizontal resolution since the ratio of communication costs to calculational costs is proportional to the perimeter to area ratio. For the same reason, this method scales well with increasing processor count [3].

In complicated physics models, computational performance rarely, if ever, follows the behavior predicted by idealized scenarios. In the case of finite difference based GCMs, parallel performance of two dimensional domain decomposition is degraded by the finite number of grid points, by load imbalances, and by nonlocal parameterizations such as the high latitude filtering device commonly used to increase the stable time step [1]. If additional avenues to parallelism can be found, performance may be enhanced. In the case of coupled ocean-atmosphere-land surface models, this additional parallelism is to be found in the natural spatial separation of the model physics. The three dimensional domains of water, air, and soil processes are physically distinct. Coupling among these processes occurs only at surfaces. Hence, concurrent execution of these submodels as a further parallel decomposition strategy is appealing due to scalable communication relative to arithmetic costs.

The LLNL model

At LLNL, we have developed the first step towards a comprehensive climate system model by writing a coupled ocean-atmosphere general circulation model designed explicitly for message passing computers. In this model [17], we have successfully implemented independent domain decompositions for the ocean and atmospheric submodels. Concurrent execution of these models has been achieved by assigning separate sets of processors (within the same machine) to each of these decompositions.

The submodels in the LLNL parallel coupled oceanic-atmospheric general circulation model are based on the UCLA atmospheric general circulation model [1,13,14,15,16] and the GFDL MOM ocean general circulation model [8,16]. Both of these models have been extensively modified in our versions including the addition of a dynamic/thermodynamic sea ice model in the OGCM. We are currently implementing the Simple Biosphere 2 (SiB2) land surface model [10] which provides a further opportunity for concurrency.

Several important performance lessons have been learned. Concurrent execution of independently decomposed submodels offers several advantages over sequential execution of similarly decomposed submodels. By exposing additional parallelism, a significantly larger number of processors may be used effectively. As is discussed below, there are limits as to how many processors an individual submodel can use and still exhibit parallel speedup. In our approach these limits that would apply for sequential execution may be exceeded in the fully coupled model by assigning individual processors to only

a single submodel. This has further benefits in the memory requirements of the coupled model. In our code, all the arrays are dynamically allocated at run time. Only the local memory that is necessary for a processor's assigned submodel is actually allocated by that processor rather than for all submodels, allowing larger problems to be run.

The load imbalance that results from differences in the computational burden of the individual submodels can severely impact performance. Hence, partitioning of processors between submodels must be done in a judicious manner to assure optimal load balance. In our approach, the independence of each submodel's domain decomposition allows the code user to adjust the assignment of processors between the submodels to reduce this idle time. Also, if a subdomain lies completely over water in the land model or completely over land in the ocean model, we do not assign that subdomain a processor, further reducing load imbalance.

Under the assumption that both the computational and communications costs in coupling the submodels are small compared to other costs, we can estimate both the optimal partitioning as well as the potential benefit of concurrent versus sequential execution of the submodels. In Figure 1 we show the actual performance of the atmospheric and oceanic stand alone models as a function of processor count as measured on the Cray T3D. In this example, the resolution of the ocean model is 3 degrees by 3 degrees by 15 levels, and the resolution of the atmospheric model is 4 degrees by 5 degrees by 9 levels. The measure of performance shown is the amount of model time simulated per machine second. In a perfectly parallelizable model, these curves would be linear. That is, the time simulated would increase linearly with the number of processors. In practice, real algorithms require communications and incur load imbalances that degrade performance.

One can use Figure 1 to estimate the optimal machine partitioning for concurrent execution. Supposing that 80 processors were available, for example, one would locate the level of performance for which the total number of oceanic plus atmospheric processors equals 80 (see the horizontal line). The number of processors to be used by a given component model would correspond to the intersection point of the horizontal line with the performance curve of that model. In this case, one would use 10 processors for the ocean and 70 processors for the atmosphere. This corresponds to 0.3 simulated hours per machine second.

From this figure we can also infer possible behavior of a sequentially executed coupled model. As in the above optimally concurrent example, consider using 80 processors in the same domain decomposition arrangement for each submodel. In this mode of execution, all available processors are assigned to each submodel. The AGCM would then be able to execute 0.3 simulated hours in about 0.85 seconds whereas the ocean would require 0.25 seconds. The total execution time in a sequential execution would then be 1.1 seconds as opposed to 1.0 seconds for concurrent execution. A better load balanced sequential scheme could be implemented by constructing a new domain decomposition for the ocean component that uses more subdomains but does not assign processors to those containing only land points. However, this strategy would require the same kind of interprocessor communication to accomplish coupling as in our concurrent scheme.

A ten percent improvement in overall performance is not representative of the potential gains of concurrent execution. The above example is limited by the relatively low number of processors. At higher processor count, the scaling of both the ocean and atmosphere models degrade. This is a result of increasing communication relative to arithmetic cost and of additional load imbalance. On some machines, we have observed that curves such as those in Figure 1 can turn over. Hence, adding more processors actually slows down the calculation. This places a practical limit on the number of processors that can be effectively used. For the T3D, this limit is around 150 processors at the resolutions in Figure 1. Fortunately, this limit increases as resolution is increased. An analysis such as that above using 150 processors instead of 80 would show an improvement much greater than 10 percent for concurrent execution.

One other caveat about Figure 1 should be mentioned. The horizontal resolution of both the ocean and atmosphere models shown here is far from our target resolution. From considerations

of the physical scales in the ocean, we would desire that the ocean model be of significantly higher resolution than the atmosphere model. This adjustment would change the balance of the model by increasing oceanic costs versus atmospheric costs. In this case, the rewards of concurrent execution will be even greater since yet more processors could be efficiently assigned to the ocean.

In fact, the maximum potential benefit of this strategy is seen in the limiting case where the performance of both component submodels is identical. The maximum number of processors that may be effectively used can then be doubled for concurrent execution versus sequential execution. We arrive at this conclusion by noting, as pointed out above, that at some point in the performance curve, increases in number of processors no longer speeds up the execution. Therefore, running each model concurrently at that processor count is twice as effective as sequential execution using all of the processors. This leads us to conclude that this parallelization strategy is most effective when the component submodels are of comparable computational intensity. The rewards to be gained through concurrent execution increase with the number of components. In the case of the ocean-atmosphere-land surface climate model, this concurrency is increased and one would envision scenarios where concurrent operation would provide three-fold improvement over sequential operation.

As just noted, the superiority of concurrent execution depends on the degree of parallelism obtained by the task decomposition. However, the range of tasks on which to base a functional decomposition is not limited to merely those separated by physical domains. In additional to concurrency among model components, one can identify concurrency within model components. The air, water and land domains are surely natural ways to divide a problem since the communication of data between them is usually via two dimensional surfaces rather than three dimensional volumes. Nonetheless, there may be advantages to functional decompositions that require communications of three dimensional variables. This is especially true on high bandwidth machines, such as the Cray T3D. Opportunities that come to mind typically exploit the operator splitting techniques used to solve the numerical approximations to the partial differential equations of the submodels. Examples include solution of the barotropic and baroclinic equations in the ocean model, separation of the sea ice model from the main ocean model, splitting the atmospheric dynamics and column physics into two separate submodels, concurrent execution of the atmospheric GCM and atmospheric chemistry models, and concurrent execution of the oceanic GCM and ocean biochemistry models.

Our approach to the design of our parallel coupled climate model is significantly different from that of other groups [7,12]. In our model, data is communicated between submodels by message passing within the same executable code object. Direct message passing, rather than transfer through intermediary files, offers significant performance advantages. Furthermore, through a sorting algorithm determined at problem initialization, message traffic is routed directly between geographically overlaying subdomains of the atmosphere and ocean models. There are no serial bottlenecks.

Conclusions

The motivations for concurrent execution of independent parallel component submodels within a coupled climate systems model are compelling. Among these are:

(1) Ability to use more processors than would otherwise be prudent.

(2) Reduction in load imbalance when null subdomains in the land and ocean components can be removed from the problem.

(3) Improvements in performance by reducing communications costs and load imbalances within the ocean and atmospheric components due to less granularity.

(4) Total flexibility in the choice of number and assignment of processors allowing a more efficient use of production parallel resources.

(5) Reduction in local memory requirements allowing larger problems to be run.

Acknowledgement

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

References

[1] Arakawa, A. and Lamb, V., Computational Design of the Basic Dynamical Processes of the UCLA General Circulation Model, *Methods in Comp. Phys.* **17**(1977) 173-265.

[2] Brown, J. and Mirin, A., MICA, a Facility to Achieve Portability for Message Passing and Dynamic Memory Management in FORTRAN, *NERSC Buffer* **18**, 2(1994) 12.

[3] Dannevik, W., "Geophysical Fluid Dynamics, Large Eddy Simulations and Massively Parallel Computing", in B. Galperin and S. Orszag, eds., *Large Eddy Simulations of Complex Engineering and Geophysical Flows*, (Springer Verlag, New York 1993), 577-595.

[4] Drake, J. and Foster, I., eds., Climate and Weather Modeling, *Parallel Computing*, **21**(10) 1537-1724 (1995)

[5] Intergovernmental Panel on Climate Change (1995) Climate Change, The IPCC Second Scientific Assessment. (in press)

[6] Mechoso, C. R., Ma C.-C., Farrara, J., Spahr, J., Parallelization and Distribution of a Coupled Atmosphere-Ocean General Circulation Model, *Monthly Weather Review*, **121**(7) 2062-2076 (1993)

[7] Mirin, A., et al., Climate System Modeling using a Domain and Task Decomposition Message-Passing Approach, *Comput. Phys. Commun.*, **84** (1994) 278-296.

[8] Pacanowski, R. K. Dixon, and A. Rosati, 1991 The GFDL Modular Ocean Model Users Guide version 1.0, GFDL Ocean Group Technical Report #2.

[9] Richardson, L.F., "Weather Prediction by Numerical Process", Cambridge University Press, Cambridge, England, (1922) p.219

[10] Sellers, P., et al., A revised Land-Surface Parameterization (SiB2) for Atmospheric GCMs. Part 1: Model formulation. Submitted to *J. Climate.*

[11] Semtner, A., Modeling Ocean Circulation, *Science* **269**(5229) 1379-1384 (1995)

[12] Sevault, E., Noyret, P., Terray, L., Thual, O. , Distributed and Coupled Ocean-Atmosphere Modeling, Proceedings of the 6th Workshop on Use of Parallel Processors in Meteorology, ECMWF, Reading, England, (1994) p.512.

[13] Wehner, M.F., et al., Toward a high performance distributed memory climate model. Proceedings of the 2nd International Symposium on High Performance Distributed Computing, Spokane, WA, July 20-23, 1993 IEEE Computer Society (1993) p. 102.

[14] Wehner, M., Mirin, A., Eltgroth, P., Dannevik, W., Mechoso, C.R., Farrara, J., Spahr, J., Performance of a Distributed Memory Finite Difference Atmospheric General Circulation Model, *Parallel Computing* **21**, (1995) 1655-1675

[15] Wehner, M. and Covey, C., Description and validation of the LLNL/UCLA parallel atmospheric GCM, UCRL-ID-123223, Lawrence Livermore National Laboratory, December 1995.

[16] Wehner, M., et al., Climate Systems Modeling on Massively Parallel Computers at Lawrence Livermore National Laboratory, To appear in *Proceedings of the NGEMCOM Workshop*, National Environmental Supercomputing Center, Bay City MI, 1995.

[17] Wehner, M., Bourgeois, A., Eltgroth, P., Duffy, P., Dannevik, W., Parallel Coupled Oceanic-Atmospheric General Circulation Model, Proceedings of the 6th Workshop on Use of Parallel Processors in Meteorology, ECMWF, Reading, England, (1994) p.512.

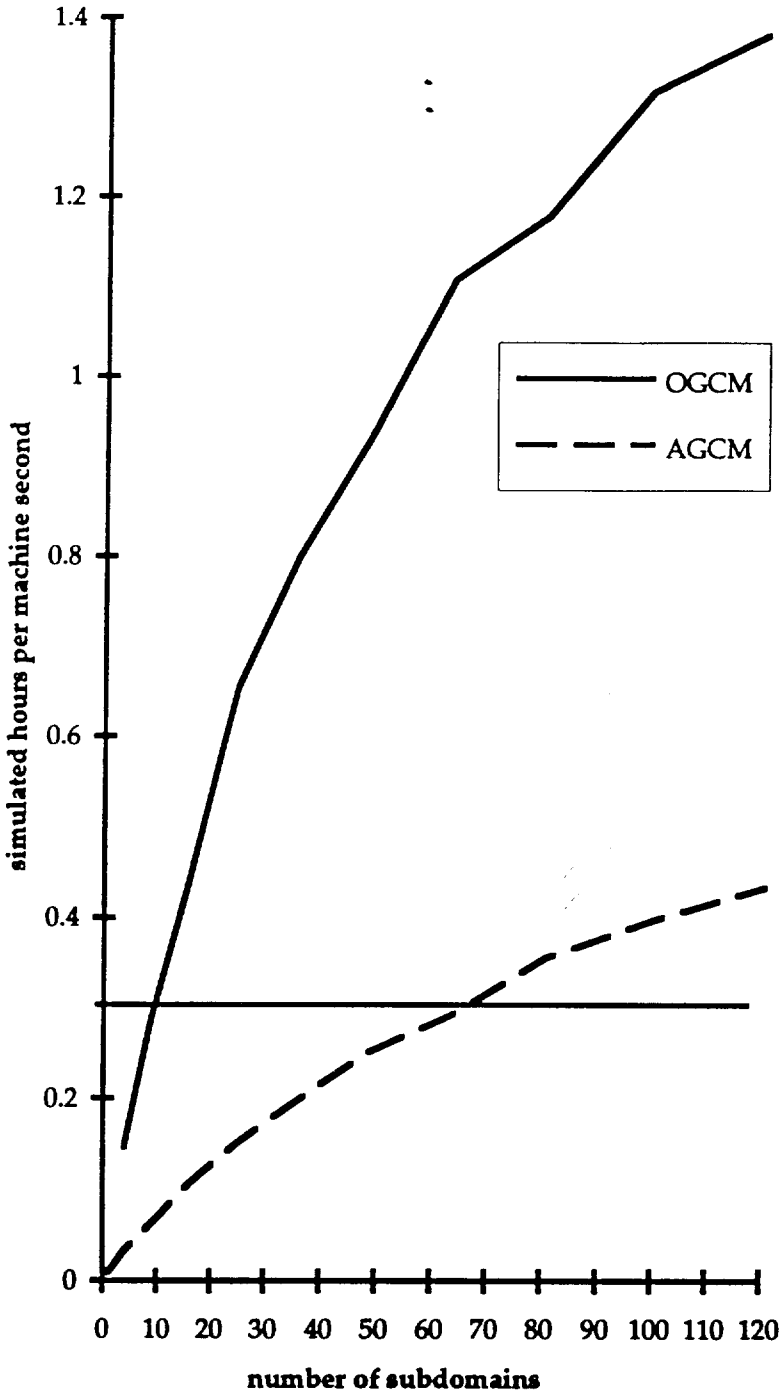


Figure 2. The performance of the atmospheric and oceanic general circulation models as a function of the number of subdomains as measured on the Cray T3D.

Technical Information Department • **Lawrence Livermore National Laboratory**
University of California • Livermore, California 94551