ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, IL 60439

ANL/MCS-TM-186

Computational Load in Model Physics of the Parallel NCAR Community Climate Model

by

John G. Michalakes and Ravi S. Nanjundiah

Mathematics and Computer Science Division

Technical Memorandum No. 186

November 1994

This work was supported by the Atmospheric and Climate Research Division, Office of Energy Research, Office of Health and Environmental Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

Contents

Computational Load in Model Physics of the Parallel NCAR Community Climate Model

John G. Michalakes
Ravi S. Nanjundiah

Abstract

Maintaining a balance of computational load over processors is a crucial issue in parallel computing. For efficient parallel implementation, complex codes such as climate models need to be analyzed for load imbalances. In the present study we focus on the load imbalances in the physics portion of the community climate model's (CCM2) distributed-memory parallel implementation on the Intel Touchstone DELTA computer. We note that the major source of load imbalance is the diurnal variation in the computation of solar radiation. Convective weather patterns also cause some load imbalance. Land-ocean contrast is seen to have little effect on computational load in the present version of the model.

Keywords: CCM2, distributed-memory parallel computing, climate modeling, load imbalance, model physics.

1 Introduction

Climate change studies need numerical models of the earth-atmosphere system to be integrated for extended periods of time (typically, climate models are run for several decades to study global change). Coupled ocean-atmosphere models need to be integrated for much longer periods (100 simulated years). Such simulations, the need for higher resolutions, and the increasing sophistication of physical parameterizations will require extensive computational resources. Scalable parallel computers will provide the increase in computational speed necessary for longer runs at higher model resolutions, but are subject to inefficiency in the form of computational load imbalance.

In this study we discuss the variation of computational load in physics modules of a global climate model and the load imbalances that result when the code is implemented on a massively parallel computer. The study was conducted using PCCM2, a parallel implementation of the NCAR Community Climate model (CCM2) running on the Intel Touchstone DELTA computer.

1.1 Brief Overview of the Model

The CCM2 is primarily a spectral model, meaning that the time integration is done in the spectral domain. The physics and nonlinear advection calculations are done in the

grid point domain. Moisture is handled nonspectrally, using a semi-Lagrangian solver. The version for the present study has a horizontal spectral resolution of T42 and a corresponding grid resolution of approximately 2.8 by 2.8 degrees, giving 64 by 128 horizontal grid points. CCM2 has 18 vertical levels.

Radiation calculations are performed using the delta-Eddington method for the shortwave radiation [1] and and solving the transfer equations for the longwave radiation using absorptivities and emissivities. Moist convection uses the mass flux convective parameterization of [5]. The present version of CCM2 has specified moisture over the land surface. Later versions have incorporated the Biosphere Atmosphere Transfer Scheme (BATS) as an additional option. A detailed description of the CCM2 is given in [6].

The algorithm for the climate model can be summarized as follows:

1. Compute physics and nonlinear interactions in the physical grid-space.
2. Convert the variables to spectral space.
3. Compute the tendencies, and update the variables (excluding moisture) to the latest time step.
4. Perform inverse transform of the variables to physical space.
5. Compute and update moisture in the physical space using the semi-Lagrangian method and repeat steps 1–5.

1.2 Parallel Implementation of the Model

On a sequential computer the solution would be obtained by traversing the entire domain. Parallel computing involves the division of a task into smaller subtasks and the assignment of such subtasks to individual processors. These processors carry out these sub-tasks and communicate with each other when required. One method for dividing work between processors is domain (or data) decomposition. Domain decomposition can be either by latitude or longitude alone (one-dimensional decomposition) or by latitude and longitude (two-dimensional decomposition). The method of parallelizing the dynamics of an atmospheric (spectral) model is discussed in [3]. A similar methodology has been employed for the parallel implementation of the CCM2. The grid-point domain is patch-decomposed over processors in both the latitudinal and longitudinal dimensions, with the added constraint that each processor has both northern and corresponding southern latitudes (Figure 1). Latitudes that are symmetric about the equator are paired on each processor by the spectral transform algorithm. The decomposition of spectral space is not dealt with in this paper, since physics is computed only in grid space. PCCM2 is not decomposed in the vertical dimension.

When decomposing the model domain over processors, it is important that computational load be distributed as evenly as possible. Unevenly distributed load reduces parallel efficiency because processors with lighter load wait for more heavily loaded processors to finish. Therefore, it is necessary to analyze the variation of load during computation to better understand and correct load imbalance.

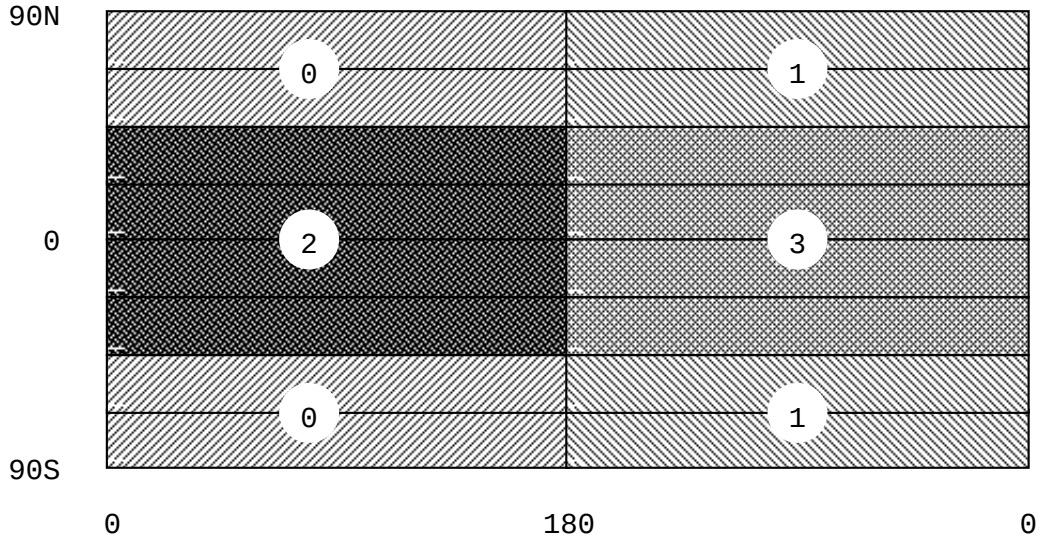

Figure 1: CCM2 domain decomposed over four processors, two decomposing the latitudinal dimension and two decomposing the longitudinal dimension. The latitudinal dimension is decomposed so that latitudes symmetric about the equator are paired on a processor, a property that simplifies implementation of the parallel spectral transform.

A primary source of computational load imbalance in a global climate model is physics. Computational load in physics varies with the state of the model variables. Studies conducted with sequential versions of CCM1 running on CRAY computers showed that the load in physics computations can vary for the following reasons [7]:

1. Spatial variation of the load from
   (a) surface type
   (b) polar night
   (c) weather patterns
   (d) day and night (diurnal cycle)
2. Temporal variation related to the computation of
   (a) radiation variables (once every hour of simulation)
   (b) absorptivities and emissivities (once every 12 hours of simulated time)

The initial study of CCM was performed before the development of the parallel code and had to be conducted using the sequential model with timers placed to capture the time spent at each grid point. Subsequent development of the parallel model, PCCM2, allowed a more direct approach, which is described in the next section. The mesh was decomposed over many processors, and direct measurement of time spent on the processors was used.

2 Instrumenting PCCM2

As with the original study, we are interested in time spent in model physics as a function of location in the model domain (grid point) and as a function of location in the physics code (subroutine or routines). The objective of instrumenting and running the PCCM2 code was to produce a set of timing data that varied over four dimensions. Each datum in this set was the time of an interval between a timer-start and a timer-stop, in microseconds[1]. For a given datum, two dimensions specified its coordinates in the model grid; one dimension specified the point in time (in time steps); and the last dimension specified the section of the code being timed. From this data, it was possible to make inferences about spatial load imbalances (over the first two dimensions) and temporal imbalances (over the third). In addition, the contributing routine or set of routines can be identified (over the last dimension).

Timers were added to the code at appropriate locations to obtain load data during the execution of the code. The sections to instrument were identified with the help of David Williamson and Jim Hack, who have been central to the development of CCM at NCAR and who are members of the working group that produced PCCM2 under the U.S. Department of Energy CHAMMP initiative [2]. The first column of Table 1 shows the sections of the physics subtree that were separately instrumented. The physics routines fall into the following major categories:

- Radiation calculations (RADCTL)
- Cloud modeling (CLDINT)
- Parameterization of moist convection (CONVAD)
- Calculation of surface fluxes (SRFINT)
- Vertical diffusion (VDINTR)
- Gravity wave drag (GWINTR)

To generate timing data in the two horizontal dimensions of the model grid, the model was decomposed as finely as possible over processors so that the timing coming from each processor would serve as a point in the data set. Ideally, and to match the resolution of the original study, one would have a single timing per cell per time step. In other words, each processor would compute and generate timings for a single point in the grid. At T42 resolution (64 latitudes by 128 longitudes) such a decomposition would require 4096 processors and thus is not feasible. However, the loop over latitude is very high in the CCM call-tree, outside the call to physics. Thus, each processor was assigned a number of latitudes, and each latitude was timed separately. In this way, the number of processors needed in the north/south dimension was reduced to only two without affecting timer resolution. The timing runs were conducted on 128 processors of the Intel Touchstone DELTA computer decomposing the grid by 2 processors in latitude and 64 processors in longitude, giving an effective timer resolution of two points per timing per time step for each instrumented section of the physics code.

[1] On the DELTA, the system timing function used was HWCLOCK.

The collected data was stored in a processor's memory until all the calculations for a time step were completed and subsequently written onto the disk. This procedure was followed to prevent the overhead due to writing of the data from contaminating the load data. The instrumented code was run for one simulated day (72 time steps of 20 minutes each). The data from the first 36 time steps was ignored to avoid the effect on performance of initialization. The initial data corresponded to that of September 1, 1987.

The data for a representative time step in which all routines are active is given in Table 1. The table shows the maximum and minimum time reported by a 2-grid-cell partition in the simulation for each of the instrumented sections of physics. The mean is the average time for all 4096 partitions. The standard deviation, σ, provides one measure of the imbalance between partitions. From the standpoint of how the imbalance affects parallel efficiency, a better measure of imbalance is $Max - Mean$ divided by Max. The mean (not the minimum) is the shortest time for module to execute if load were perfectly balanced. The maximum is the time it would actually take (with the unbalanced load configuration). The next section analyzes the contribution of the physics modules to load imbalance using this measure.

3 Analysis

Not all physics computations are conducted at every time step of integration. PCCM2 in its tested configuration (T42, 20-minute time steps, hourly radiation, and twice daily absorptivity and emissivity calculation) does a representative execution of physics over the course of a 12-hour simulation. It is representative in the sense that the time spent computing physics will contain cost components for all physics modules in proportions that are representative of long runs of the model. We can classify the time steps into the following categories:

a. Radiation time step with calculations of emissivity (ems) and absorptivity (abs): All physics computations are conducted at this step. This step includes the calculation of emissivities and absorptivities (RADABS and RADEMS subroutines) for the longwave radiation. These calculations are conducted once every 12 hours in the model. We shall term this type of time-step as "A".

b. Radiation time step without emissivity and absorptivity: The longwave radiation calculation does not include the computation of emissivities and absorptivities. All calculations for shortwave radiation are conducted. These calculations are done once every hour of integration. This category of time steps is denoted type "B".

c. No-radiation time step: Only convection, diffusion, surface fluxes, and gravity wave drag are calculated during this time step. All time steps other than the radiation time steps are of this category, type "C."

The computational time required for a composite, or average, time step is

$$T_{avg} = \frac{T_A + 11\ T_B + 24\ T_C}{36},$$

Table 1: Computational cost in milliseconds in PCCM2 physics. Statistics are over the simulated 4096 2-cell partitions on the Intel Touchstone DELTA computer for one type-A time step (chosen because all computational modules are engaged). The call tree is indicated by indentation and time shown for a routine includes the times for its subroutines if there are any. A routine marked with • contains in itself or in its subtree conditional code that may or may not execute depending on the state of the model; in the case of others the very small variance is attributable to "noise" — cache effects or other artifacts of the hardware.

Routine		Max	Min	Mean	$\frac{\sigma}{Mean}$	$\frac{Max-Mean}{Max}$
OMCALC		0.518	0.303	0.343	0.01	0.338
CONVAD	•	7.900	3.891	4.610	0.14	0.416
DADADJ	•	0.221	0.100	0.126	0.10	0.430
CMFMCA	•	5.644	1.703	2.347	0.27	0.583
COND	•	1.218	0.996	1.050	0.03	0.138
PHYS	•	466.781	382.242	422.441	0.09	0.096
CLDINT	•	5.955	5.288	5.686	0.02	0.045
CLDFRC	•	3.985	3.377	3.717	0.03	0.067
CLDEMS		0.570	0.418	0.455	0.05	0.202
RADCTL	•	439.063	358.573	397.486	0.09	0.095
RADCSW	•	78.731	0.310	38.039	0.98	0.517
RADALB	•	0.354	0.122	0.168	0.28	0.525
RADDED	•	44.739	0.000	20.895	0.99	0.533
RADCLR	•	72.670	0.000	3.398	0.98	0.538
RADCLW	•	366.300	357.035	358.227	0.00	0.022
RADTPL		1.001	0.824	0.866	0.04	0.135
RADEMS		26.127	25.373	25.562	0.00	0.022
RADABS		336.686	328.222	329.058	0.00	0.023
SRFINT	•	1.176	0.941	1.010	0.04	0.141
SRFFLX	•	0.451	0.272	0.310	0.07	0.313
SRFTSB	•	0.487	0.348	0.370	0.05	0.240
VDINTR	•	3.745	2.691	3.104	0.08	0.171
VDIFF	•	3.282	2.264	2.665	0.10	0.188
MVDIFF		0.807	0.630	0.675	0.04	0.164
GWINTR	•	1.204	0.297	0.563	0.60	0.532

which represents the cost of 1 time step with radiation and abs/ems calculations, 11 others with radiation but without abs/ems, and the remainder without radiation or abs/ems averaged over the 36-step period between type-A time steps (12 simulation hours). To characterize the effect of physics load imbalance on model performance as a whole over long simulations, we discuss the overall load and imbalance in terms of this average time step. Subsequently, we detail the individual contributions to this overall imbalance from each computational module making up CCM physics during the three different types of time step.

3.1 Composite Load and Imbalance

Over the course of 36 time steps for the hypothetical 2-cell per processor 4096-processor decomposition,[2] time spent in physics is 62 milliseconds per time step. This time is the sum (over time steps) of the maximum time (over the grid of 2-cell partitions) at each time step, divided by the number of steps. The maximum need not occur at the same 2-cell partition at each time step. However, since there is a synchronization imposed by CCM dynamics between calls to physics on successive time steps, and because physics is called for all grid points before the onset of dynamics in a given time step[3], it is reasonable to consider the sum of the individual maxima at each time step as the time spent in physics for the series of time steps. By similar reasoning, one may sum the mean time over 2-cell partitions in the grid from each step, divide by the number of steps, and call this the average time spent per time step. This average or "ideal" time was 45 milliseconds per step and represents the time physics would have taken in a situation of perfect balance. Dividing this ideal time by the maximum time gives an efficiency of 72.6 percent for physics as a whole (or an *in*efficiency of 27.4 percent). The amount of time that would be lost to load imbalance in this decomposition is 17 milliseconds per step, the difference between the maximum and the mean.

The effect of physics inefficiency on total model performance depends on how efficiently the rest of the model, in particular dynamics, is performing. Dynamics in CCM is primarily communication bound, though there is also some computational inefficiency owning to an uneven distribution of Fourier coefficients between processors in spectral dynamics (for wind velocity and temperature) and a disproportionate amount of work at the poles in the semi-Lagrangian dynamics (for moisture). At present, in real runs of the code on the Intel Touchstone DELTA, physics consumes about a third of the total run time when running on the full machine (Table 4). Roughly speaking, for the current implementation of PCCM2 on the full DELTA, the effect of a 33 percent (Section 3.2) computational imbalance in physics will be around 10 percent. As communication efficiency improves with tuning of spectral and semi-Lagrangian dynamics, the effect of physics load imbalance in PCCM2 will become

[2] While the contribution to load from each cell is a fixed quantity, the load imbalance that results depends on the decomposition, how the cells are allocated to processors. The size and shape of partitions affect load imbalance. (Take the trivial case of all cells grouped onto one processor in the shape of the grid itself: the inefficiency due to load imbalance is zero.) Therefore, the timing and efficiency numbers quoted in this discussion are specific to the hypothetical decomposition in force.

[3] This was not true in the vector/shared-memory parallel version of the model, CCM2. The call to physics for each latitude was followed by the call to the FFT for that latitude. To efficiently block FFT communications in the parallel code, PCCM2 separated the calls to physics and the calls to the FFT into separate loops over latitude. Each time step, physics for all latitudes is complete before the synchronization imposed by message passing in the spectral dynamics.

more pronounced.

3.2 Contribution by Module

CCM model physics comprises a number of computational modules (Section 2). How seriously a module affects load imbalance in the parallel model depends on how much imbalance there is in the module and how much time the module contributes to total time spent in physics. Table 2 shows the amount of time processors spent performing useful work in the major modules of CCM2 physics and how much time was lost to load imbalance. The "useful" time is the mean time spent over processors in the hypothetical 2-cell per processor 4096-processor decomposition. The time lost to imbalance is the time spent on the processor that took the longest time (over all modules) minus the mean.

An alternative way to compute this time would be to take the maximum for a single module of the code and subtract the mean, to determine the inefficiency for that module. However, it is uncertain whether the maximum in each module would occur on the same processor. Therefore, although this method shows the absolute imbalance for a particular module, it would be inappropriate to add together the inefficiencies for different modules. Since we are interested in the *net* effect of imbalances, we used the former method of calculation—considering the time for each module on the processor with the maximum overall physics time. In practice, we discovered that the overall difference between the two ways of calculating the inefficiency is small: adding together times produced by the alternative calculation generates an average physics time step of 1938 milliseconds, which is only 3 percent above the net time of 1881 milliseconds. This suggests there is little canceling out of imbalances in the physics because the imbalance from the diurnal cycle in the radiation module (RADCTL) dominates the rest of the profile.

The times shown are for 1 type-A step (solar radiation with absorptivity and emissivity calculations), 11 type-B steps (radiation), and 24 type-C steps (nonradiation). For the representative period of 36 time steps (one-half of a simulation day) our hypothetical 4096-processor decomposition of model physics consumes 1881 milliseconds, only 1267 milliseconds of which is spent in useful computation. The difference, 614 milliseconds (33 percent), is lost to idle time.

3.2.1 Radiation Calculations (RADCTL)

The most serious source of load imbalance in PCCM2 physics is the radiation package, specifically, shortwave radiation. Radiation comprises 68 percent (864.7/1267.1) of total physics computation over a representative 36-step period. This would be worse except radiation is performed only every third time step (hourly) and the principal component of longwave radiation, RADABS, is so costly that it is performed only every 36th step. The contributions of longwave (RADCLW) and shortwave (RADCSW) to overall radiation (RADCTL) costs is shown in Table 3. Longwave radiation, though costly, is nearly perfectly balanced so its effect on parallel efficiency is negligible. The source of all imbalance in radiation is the shortwave radiation package, RADCSW, because it is computed only in half the grid points (the ones in daylight) at any given time. Figure 2 shows time spent in RADCSW over the grid during a radiation time step. Only some 0.3 milliseconds of work is occurring

Table 2: Time spent in major CCM physics modules in a hypothetical 64 by 64 (4096) processor decomposition at T42 resolution. Time lost to imbalance is calculated in a way that gives the net time lost, allowing for the fact that imbalances in one module may cancel imbalances in another. N is the number of steps that are represented in the timings for a type of step.

type	n	RADCTL	CLDINT	CONVAD	SRFINT	VDINTR	GWINTR	Total
	Time (milliseconds) spent in useful computation							
A	1	397.0	5.6	4.6	1.0	3.1	0.6	411.9
B	11	467.7	62.5	50.8	11.1	33.9	6.2	632.0
C	24	0.0	0.0	110.8	24.8	74.1	13.5	223.2
sub	36	864.7	68.1	166.2	36.9	111.1	20.3	1267.1
	Time (milliseconds) lost to imbalance							
A	1	40.3	0.1	2.6	0.1	0.3	0.5	43.9
B	11	441.6	1.5	23.0	0.3	3.5	4.0	473.9
C	24	0.0	0.0	78.0	1.2	4.8	12.2	96.2
sub	36	481.9	1.6	103.6	1.6	8.6	16.7	614.0
Total	36	1346.6	69.7	269.8	38.5	119.7	37.0	1881.0

Table 3: Time lost to imbalance in the two components of CCM2 radiation: longwave radiation (RADCLW) and shortwave (RADCSW). There is very little imbalance in longwave radiation; nearly all in RADCTL is attributable to shortwave radiation.

type	n	RADCLW	RADCSW
	useful computation		
A	1	358.2	38.0
B	11	37.4	418.2
C	24	0.0	0.0
sub	36	395.6	456.2
	lost to imbalance		
A	1	1.8	38.5
B	11	3.5	437.9
C	24	0.0	0.0
sub	36	5.3	476.4
Total	36	400.9	932.6

Table 4: Total time per average PCCM2 time step and the percentage of time spent in model physics for a series of runs on the Intel Touchstone DELTA

	PCCM2 (milliseconds/time step)				
Mesh	A	B	C	Avg.	
8×8	12944.0	4039.8	1757.0	2747.3	
16×8	6642.5	2192.6	1042.4	1540.4	
16×16	5841.5	1403.0	707.5	1057.9	
32×16	3111.5	854.2	508.6	684.2	
	Physics (milliseconds/time step)				
Mesh	A	B	C	Avg.	percent
8×8	11705.0	2800.6	515.3	1506.5	55
16×8	5867.0	1415.7	264.6	762.8	50
16×16	5277.5	832.9	141.6	490.7	46
32×16	2677.5	419.2	72.4	248.3	36

Table 5: Components of shortwave radiation. RADALB, RADDED, and RADCLR are subroutines called by RADCSW. The RADCSW_{res} entry represents the computation performed in RADCSW itself. It is computed here as a residual; it was not measured directly.

	Max.	Min.	Mean	σ
RADALB	0.4	0.1	0.2	0.5
RADDED	44.7	0.0	20.9	20.7
RADCLR	7.3	0.0	3.4	3.4
RADCSW_{res}	26.3	0.2	13.8	–
RADCSW	78.7	0.3	38.0	37.2

in each 2-cell partition in the nighttime region, compared with 78 milliseconds of work in a daylight 2-cell partition. Within RADCSW, the sources of imbalance are computation within RADCSW itself and in three subroutines to compute surface albedo (RADALB), the delta-Eddington solar scheme (RADDED), and the clear-sky solar computation (RADCLR) (Table 5).

Of the 614 milliseconds lost to load imbalance each 36 time steps, the imbalance in shortwave radiation accounts for 476 milliseconds, or 77.5 percent of the total physics imbalance. For a model run in which physics was 36 percent of the total cost, imbalance in RADCSW would be responsible for 8.5 percent of the total inefficiency attributable to physics load imbalance.

The regularity of this pattern of imbalance suggested a straightforward scheme for correcting a large percentage of the shortwave radiation load imbalance. Before shortwave radiation is invoked in a time step, every other point in a latitude (an east-west row of points) is exchanged between processors, decomposing that row in such a way that, after the exchange, each processor has almost the same number of day and night points. After shortwave radiation, the exchange is reversed. In spite of the cost of performing the exchanges, the load-balancing code resulted in a 6 percent overall improvement in model run times [4].

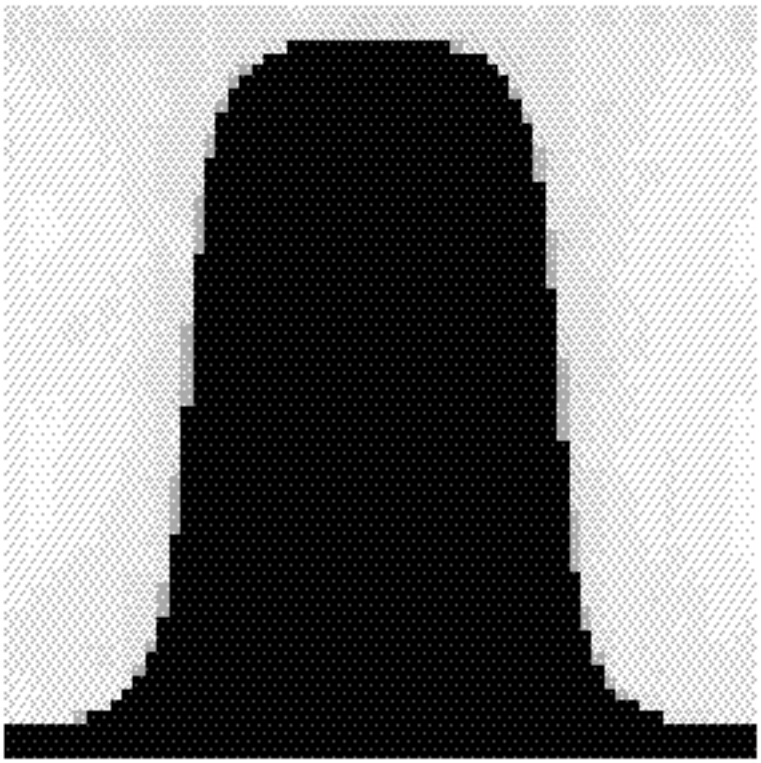

Figure 2: Distribution of load in PCCM2 in the shortwave calculations subtree. Lowest values are dark; highest values are light. The top and bottom of the plot are the north and south poles. Left and right edges correspond to the prime meridian. Each cell represents two grid points in the model domain.

One expects that the effectiveness of the exchange scheme for correcting diurnal cycle imbalance will vary seasonally because the balancing effect is in the east/west dimension only. North/south imbalances associated with seasonal variation in solar declination are not accounted for in the exchange scheme. Thus, the scheme should do well closest to the equinoxes in the simulation when all the latitudes have the same number of daytime and nighttime points. It should do most poorly closest to the solstices, when most latitudes will have different numbers of daytime and nighttime points. However, in the special case of PCCM2, the seasonally induced north/south imbalances in shortwave radiation are not a problem because the model latitudes are decomposed symmetrically about the equator: a processor handling the latitude at 30 N would also be handling 30 S. The domain happens to be decomposed this way to exploit symmetry in the spectral domain. Thus, the lower computations in one hemisphere are offset by higher computations in the corresponding region of the other hemisphere.

3.2.2 Other Imbalances

The diurnal cycle in shortwave radiation accounts for 77.5 percent of the load imbalance in PCCM2 physics. The remaining 22.5 percent of imbalance is caused by load imbalances in mass flux convective parameterization (17 percent), gravity wave calculation (2.7 percent),

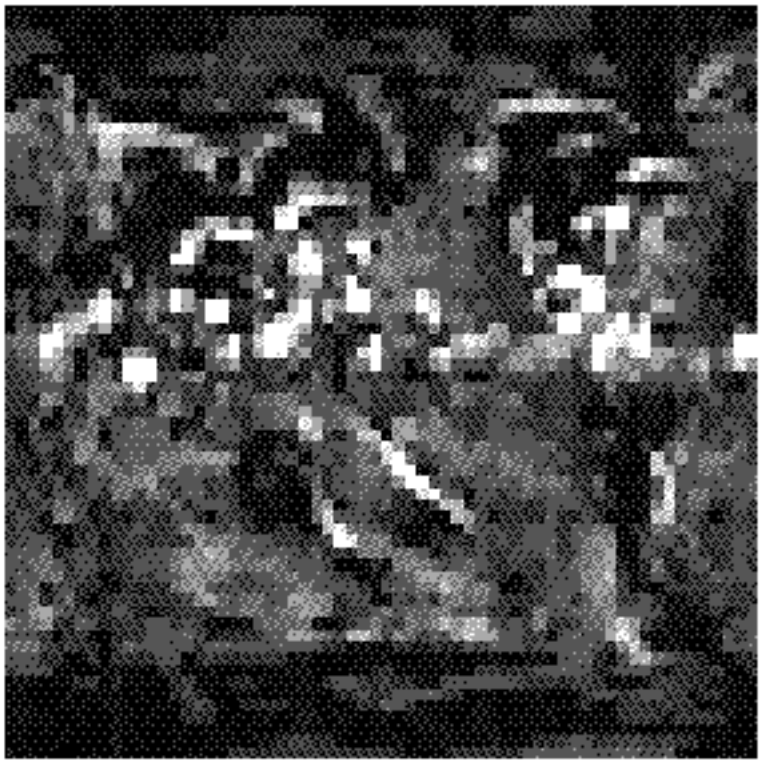

Figure 3: Distribution of load in PCCM2 mass flux convective parameterization subtree

vertical diffusion (1.4 percent), surface temperature calculation (less than 1 percent), and cloud parameterization (less than 1 percent). The load imbalances are from surface type and what is loosely termed weather patterns.

Weather patterns. These appear as irregularly shaped patches of load across the map that can be seen to move in a weather-like fashion as the simulation progresses. The imbalance contributed from within the CONVAD subtree is mostly of this nature (Figure 3). In the CONVAD subtree we notice that the pattern for CONVAD is largely similar to that of CMFMCA (mass flux convective parameterization). The mean computation time in this routine is 2.3 milliseconds, and its contribution to total load imbalance is 2.6 mill-seconds. Thus we note that most of the imbalance in this subtree is caused by mass flux convective parameterization processes. This is understandable because the routine CMFMCA is invoked only when the atmosphere is unstable to moist convection and not all regions have this instability. Load tends to be higher closer to the equator, in the inter-tropical convergence zone, where there is more moist convective activity.

The mass flux convective parameterization is called for every time step in the model, and the characteristics of load do not vary in this routine over the three types of CCM2 time step.

Surface type. Effect of surface type is most noticeable in gravity wave calculations (GWINTR). Figure 4, a plot of processor load in this routine, shows continental outlines

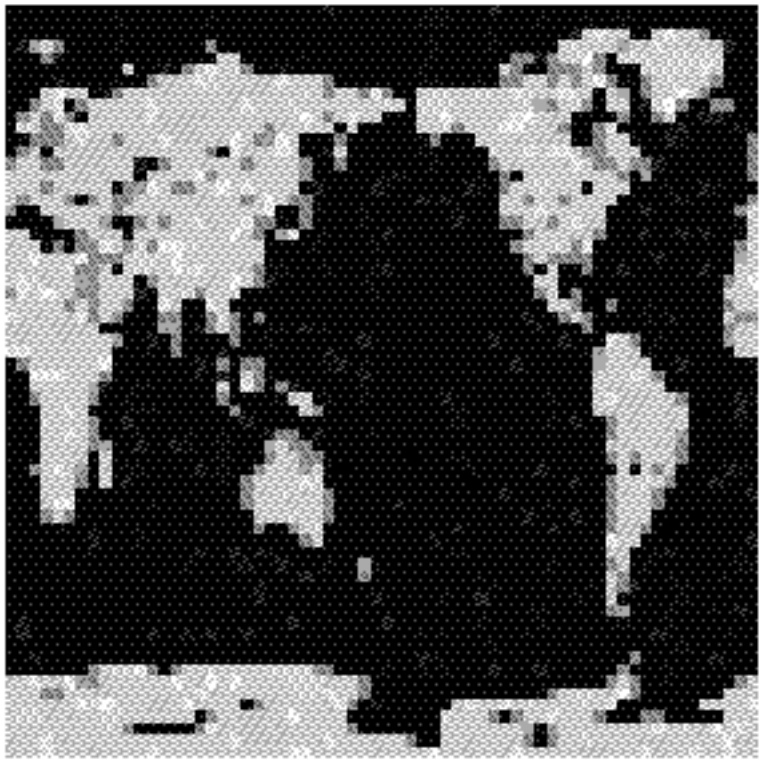

Figure 4: Surface type causes an imbalance in the gravity wave computations of CCM2. However, there is little effect on parallel efficiency because the amount of computation in this routine is small.

clearly; however, its contribution to both mean computation (0.5 milliseconds out of a total of 422 milliseconds or about 0.1 percent) and load imbalance (0.5 milliseconds out of a total of 44 milliseconds or about 1 percent) is small. Other routines showing some influence from surface type imbalance are RADALB (in the radiation subtree), CMFMCA (convection), and VDIFF (in vertical diffusion). Density plots for CLDFRC and SRFFLX show some suggestion of continents as well, though much less distinctly. For purposes of improving parallel efficiency, imbalance stemming from different surface types does not appear to be large enough to be worth attempting to fix in PCCM2.

4 Conclusion

The physics computations of the parallel version of CCM2 has been analyzed for load imbalances. We note that both the mean load and imbalance vary with the the type of time step being computed (no-radiation time step, time step with radiation and emissivity and absorptivity calculations, or time step with radiation but without the calculation of emissivity and absorptivity). The diurnal variation of shortwave radiation is the major cause of load imbalance (about 75 percent of the total imbalance during an average time step). This imbalance is due to the additional computation required over the grid points in the day region (receiving solar radiation). Attempts are being made to reduce this imbalance

in the parallel model by moving computations from more heavily loaded daylight regions to the less-loaded nighttime processors [4]. Weather patterns (resulting in moist convective instability) are also a major cause of imbalance (about 17 percent of the total imbalance during an average time step). Their occurrence in space and time is not predictable *a priori*, and although remediation would also involve redistribution of work between processors, the strategy would need to be dynamically adaptive.

The present method of parallelization, which exploits the symmetry about the equator and allocates similar latitudinal ranges of the opposite hemisphere to the same processor, effectively negates the polar day/night asymmetry.

Surface type did not cause major load imbalances in this version of the model, though it was noticeable in the calculations of gravity-wave drag and vertical diffusion. Load imbalances might be more severe, however, if the BATS surface hydrological model or some other coupled model is used.

References

[1] B. P. Briegleb, *Delta-Eddington approximation for solar radiation in the NCAR community climate model*, J. Geophys. Res., 97 (1992), pp. 7603–7612.

[2] Department of Energy, *Building an Advanced Climate Model: Progress Plan for the CHAMMP climate modeling program*, DOE Tech. Report DOE/ER–0479T, U.S. Department of Energy, Washington, D.C., December 1990.

[3] I. Foster, W. Gropp, and R. Stevens, *The parallel scalability of the spectral transform method*, Mon. Wea. Rev., 120 (1992), pp. 836–850.

[4] I. Foster and B. Toonen, *Load Balancing Algorithms for the NCAR Community Climate Model*, Tech. Rep. ANL/MCS-TM-190, Argonne National Laboratory, Argonne, Illinois, April 1994.

[5] J. J. Hack, *Parameterization of moist convection in the NCAR Community Climate Model (CCM2)*, J. Geophys. Res., (1993), p. submitted.

[6] J. J. Hack, B. A. Boville, J. T. Kiehl, P. J. Rasch, and D. L. Williamson, *Description of the NCAR community climate model (CCM2)*, Tech. Rep. NCAR/TN-382+STR, National Center for Atmospheric Research, Boulder, Colorado, June 1993.

[7] J. G. Michalakes, *Analysis of Workload and Load Balancing Issues in the NCAR Community Climate Model*, Tech. Rep. ANL/MCS-TM-144, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois, January 1991.