

SIMD Climate Modelling

R. Francis^a, L. Rotstayn^b, D. Abramson^a and M. Dix^b

^aDivision of Information Technology, CSIRO, 723 Swanston St. Melbourne, Australia, 3055.

^bDivision of Atmospheric Research, CSIRO, Private Bag 1 Mordialloc, Australia, 3195.

Abstract

Weather and climate modelling are such major consumers of computer resource that long term climate modeling is considered to be one of the Grand Challenges of high performance computing. The SIMD model is an obvious path to teraflop performance provided applications suit its mode of data parallelism. Consideration should therefore be given to porting climate models to such machines. However, climate and weather modeling in general involves a large number of physics calculations which occur only in subsets of the data elements. It has been conjectured that the flops executed by cells in a given model may vary to such an extent that SIMD execution is ineffective. This paper analyzes a production global circulation model to evaluate its potential for SIMD execution. The analysis shows that the physics routines from an existing production model would operate with a SIMD efficiency of 60% if atmospheric columns are allocated to PEs and 20% if cells are allocated to separate PEs. The results also indicate the factors to be used in a tradeoff between additional PEs and PE processing power for SIMD architectures aimed at climate modeling.

1. Introduction

Numerical weather prediction models have been used for some years to help provide short range predictions of local weather patterns, and are now being used to provide longer range global climate patterns. Whilst early programs only modelled wind dynamics, modern codes attempt to model physical effects such as solar radiation, clouds, rain, snow and ice. The proposed new generation of programs [1] will also include more detailed physical models such as evaporation through vegetation and will be extended to include coupled ocean effects. As a result long term climate modeling constitutes one of the Grand Challenges of high performance computing requiring teraflop performance. The SIMD model is an obvious path to the

required teraflop performance provided that applications suit its mode of data parallelism. However, climate and weather modeling in general involves a large number of physics calculations which occur only in subsets of the data elements. It has been conjectured that the flops executed by cells in a given model may vary by as much as an order of magnitude. Assuming a cell per processor and direct SIMD execution, this order of magnitude variation would cause a significant difference between peak and delivered performance for massively parallel machines executing such codes. Therefore, before launching into the conversion of large climate modeling codes into SIMD form, some analysis of the likely performance characteristics is warranted.

The work reported here covers the analysis of an existing climate model for its workload balance under SIMD execution assumptions. The particular model is a spectral model with 9 vertical layers, a 64 by 56 horizontal grid and comprises approximately 20,000 lines of FORTRAN. This model is an improved form of a four layer model which has been used extensively for the study of climate change and in drought prediction studies [2-5]. The total flop loads for the climate model divide into two categories, physics and dynamics where the physics is calculated in a grid space and the dynamics calculated in spectral and grid spaces. The transform and dynamics activity currently contributes 40% of the total flop requirement of the model, the remainder being the physics calculations. The casting of the transform and dynamics components into SIMD form is a question of mapping the necessary Fourier and Legendre transforms onto the machines [6]. It is expected that the physics proportion will rise as improvements in machine performance and an increasing understanding of weather systems permit cloud micro physics, coupled ocean models and ground and vegetation effects to be included. Therefore physics costs can be expected to dominate model performance and SIMD performance analysis should focus on this area.

To measure the variability of the flop loads presented by the physics calculations, the climate model has been instrumented and executed on a CRAY Y-MP and a Sun workstation. In both cases the instrumentation identifies the basic blocks of the original FORTRAN program, the leaves of the program's control flow graph, and inserts code to capture the flops in each execution of each basic block. The individual components of the flop counts accumulated for each block are identified with the proposed mapping onto the SIMD architecture. If the program were re-coded for a SIMD machine, ideally each basic block would be executed for all cells at the same time. Given such recoding, the total SIMD flops required to execute the program is the sum of the maximum cell flops for each basic block multiplied by the number of processing elements or floating point units. The total SIMD flops can be compared to the actual total flops to obtain a measure of SIMD efficiency. The paper presents results from the analysis of the model showing the distribution of flops across cells in the model for various physics and giving rise to an estimate of the model's performance behaviour under SIMD execution.

2. Climate Model Structure

Atmospheric GCMs such as the 9 layer model described here approximate numerically the equations of fluid dynamics and thermodynamics which

govern the atmosphere using either finite difference methods or spectral methods. In either approach, the approximated equations are numerically integrated forward in time from an initial state in which the global fields of meteorological variables such as temperature, wind, moisture and surface pressure are specified.

2.1 Spectral Transformations

The spectral transform method [7-9] involves a transformation from spectral to gridpoint space at each timestep, followed by a reverse transform from gridpoint to spectral space. When in gridpoint space, the meteorological fields at each of the nine model levels are stored on a rectangular grid of 64 evenly spaced longitudes and 56 latitudes. In spectral space, the fields are represented as truncated two dimensional spectral expansions at each of the nine model levels. Variation in the zonal (east-west) direction is expressed by means of a truncated Fourier series, while variation in the meridional (north-south) direction is expressed by means of a truncated series of associated Legendre polynomials. For example, a field F which varies with longitude (λ) and latitude (ϕ) is expanded as

$$F(\lambda, \phi) = \sum_{m=0}^{21} \sum_{l=m}^{m+21} F_l^m P_l^m(\sin \phi) e^{im\lambda} = \sum_{m=0}^{21} F_m(\phi) e^{im\lambda} \quad \left\{ \begin{array}{l} 0 \leq \lambda \leq 2\pi \\ -\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2} \end{array} \right. \quad (1)$$

Where

P_l^m is an associated Legendre polynomial of order m and degree l

F_l^m are the (complex) spectral amplitudes

$F_m(\phi)$ are the (complex) Fourier coefficients at latitude ϕ

The Fourier coefficients at each latitude are used only as intermediate values when transforming between spectral amplitudes and gridpoint values. The truncation of the series used here is known as rhomboidal truncation at wavenumber 21 (R21) and is commonly used in models of this type.

2.2 Main Program Loop

At each time step, the calculations of the model involve four major activities.

- i Transform spectral amplitudes to gridpoint fields. Calculate the changes ("tendencies") in the (temperature and moisture) fields at each gridpoint due to physical processes (Radiative heating, convection and rainfall, surface effects.) Then transform these modified fields back to spectral space.

- ii Transform spectral amplitudes to gridpoint space. Evaluate the changes in the fields at each gridpoint due to the nonlinear terms in the equations of motion. Then transform these tendencies back to spectral space.
- iii Evaluate the linear terms in the equations of motion (in spectral space.)
- iv Perform the semi-implicit time integration (in spectral space), to update the fields based on the tendencies calculated above.

Step *i* is referred to as the *physics loop* and step *ii* as the *dynamics loop*. Both are loops over latitude and have the following basic structure.

DO FOR LATITUDE = 1 TO 28

DO FOR HEMISPHERE = 1 TO 2

- Convert (selected) spectral amplitudes F_1^m to Fourier coefficients F_m for this latitude by summing terms as indicated in (1)
- Convert Fourier coefficients F_m to 64 gridpoint values F via inverse FFT.
- Do calculations at each of 64 gridpoints. (Either "physics" or "dynamics")
- Convert the modified gridpoint values F to Fourier coefficients F_m for this latitude via FFT.

END DO

Calculate contribution of the two latitudes' Fourier coefficients F_m to the Legendre transform (defined below) i.e. to the new spectral amplitudes F_1^m .

END DO

The Legendre transform

$$F_1^m = \int_{-\pi/2}^{\pi/2} F_m(\varphi) P_1^m(\sin \varphi) \cos \varphi d\varphi \quad m = 0, 1, \dots, 21, \quad l = m, \dots, m+21 \quad (2)$$

is evaluated by Gaussian quadrature over the 56 (Gaussian) latitudes; this method is exact for this choice of latitudes. The iterations are independent of each other as the calculations at one latitude do not affect those at any other latitude, with the exception of the calculation of the Legendre transform.

The reason for combining the northern hemisphere and southern hemisphere calculations as shown rather than simply looping over 56 latitudes, is to exploit the symmetry in the Legendre transform. The symmetry is due to the fact that P is an even function if $l-m$ is even and is an odd function if $l-m$ is odd. Thus for an even value of $l-m$ the contribution to (2) from a northern and corresponding southern latitude can simply be added together. For an odd value of $l-m$ the contribution by the southern latitude is subtracted from the contribution by the northern latitude.

2.3 Physics Sequence

Within the physics loop the model physics is implemented in a set of routines executed in the order given below. Some of these evaluate only at the surface (S), some at all points simultaneously (A) and some operate over separate columns of points (C).

<i>surfset</i>	S	Set surface type, albedo, roughness length etc. The albedo depends on the season and the presence of snow and/or ice. The seasonal cycle of the albedo is specified over the land and there is no feedback for the effect of climate on vegetation.
<i>qsgam</i>	A	Calculate saturation vapour pressure
<i>negrm</i>	A	Calculate relative humidity
<i>hsflux</i>	S	Calculate surface fluxes of heat, momentum and moisture. These depend on the nature of the underlying surface (temperature, wetness, roughness) and also on the stability and wind speed of the atmosphere.
<i>radfs</i>	C	Calculate clouds from the relative humidity distribution and from the distribution of convection (from the previous time step). Calculate radiative heating rates using the full annual and diurnal cycles of solar radiation and the distributions of ozone, carbon dioxide, water vapour and clouds.
<i>surfupa</i>	S	Update the surface temperature and moisture fields using the surface fluxes.
<i>hvertmx</i>	C	Apply vertical mixing. This depends on the stability of the atmosphere and on the surface fluxes.
<i>gwdrag</i>	C	Calculate gravity wave drag. This is a parameterisation of the momentum flux of waves generated by topography below the scale resolved explicitly by the model.
<i>rainda</i>	C	Calculate large scale precipitation (supersaturation)
<i>conv</i>	C	Calculate convection and rainfall, depending on the temperature and moisture distributions and also the large scale motion field.

surfupb *S* Update surface moisture and snow cover due to precipitation.
 Calculate sea-ice.

The evaluation of the prospective performance of these routines under SIMD execution is the main thrust of the analysis presented in the remainder of this paper.

3. Analysis

The SIMD analysis involves the instrumentation and execution of the model to yield a flop distribution and then the association of those flops with particular notional PEs to give a picture of the workload distribution. This workload can in turn be analyzed to calculate SIMD efficiency.

3.1 Strategy

The first step in the analysis of the potential parallelism in the model is to identify those elements of it which could be executed simultaneously if an arbitrary number of processors were available. The iterates of the loops identified above are obvious candidates. The basic requirement in the analysis is the formation of a mapping from the individual components of the model to a hypothetical set of processing elements (PEs). The re-coding and extension of the model to deliver such code segments to PEs for execution on a particular machine is not required, but the notional identification is essential to the analysis. Given the identification, the model can be modified to accumulate the work as measured by the flops occurring in the model's code segments for each notional PE. Data generated by executing an instrumented model can then be analyzed to provide estimates of the effectiveness of various work load distributions for particular actual processor resources.

This analysis strategy relies on two key assumptions:

- The flops executed in the model are necessary to its result.
- The model's performance is determined by the flops.

In particular the technique does not measure or consider communication overhead, although other tools are under development which can address the data access issue, nor does it measure the effect of any necessary synchronization. However, as the distribution of work is certainly one of the most critical parameters in estimating performance and as floating point operations dominate the calculation, their characterisation is an important ingredient in estimating the potential benefit from parallel execution.

3.2 Program Instrumentation

The tool used to analyze the climate model essentially dismembers a program into its primitive constituent parts and instruments those parts to obtain a detailed picture of the work they perform. The dismembering must be carried out in a way which reveals the parallelism in the workload, and the picture built up by executing the instrumented code can be viewed or analyzed for a variety of purposes including potential performance prediction. The

analysis begins with the original sequential code and a program which processes standard FORTRAN-77 to detect all basic code blocks and to insert a library call at the start of each such block. This library call accepts two parameters; a basic block identifier automatically generated and a notional PE identifier specified as part of the analysis. On each execution it accumulates the flops counted since its previous execution and stores them into a two dimensional array of flop counts. This array can be considered indexed by basic block number and notional PE identifier.

The actual flop counting is achieved in two different ways. If the program is to be run on the CRAY Y-MP, flop counting interacts with that machine's hardware performance registers [10]. If the program is to be run on a UNIX workstation, a more significant operation is required. First, the program is processed into its segments and the result compiled to assembler. Second, the assembler is processed to detect and insert code to count all floating point operations. Third, the modified assembler is finally re-compiled into an object which when executed will generate the required data. Note that the entire maths library and all other libraries linked to the program must be compiled to assembler, processed to count floating point instructions, and then re-compiled into object files if the flops in the maths libraries are to be accumulated.

If each cell were allocated to an actual PE a simple calculation yields an estimate of the number of flops consumed in the SIMD execution of each step. The calculation makes the assumption that on conversion to a SIMD form the program is re-written so that where cells involve the execution of the same basic block in the same step, that the execution of those basic blocks would be arranged to occur simultaneously. A SIMD architecture requires such a program organisation if reasonable efficiency is to be obtained.

Hence the performance estimation relies on two key assumptions:

- Re-coding for SIMD execution is a matter of restructuring the program so that the calculation is spread out over the available PEs in a regular and determined manner.
- The effect of the restructuring is such that basic blocks which must execute on more than one PE will be executed at the same time.

Given the array of flop counts, the sequential execution time is proportional to the sum of all the flops in all basic blocks for all notional PEs. The SIMD execution time is proportional to the sum of the maximum flops in each basic block for any of the PEs assuming each notional PE is allocated to an actual PE. This calculation reflects the fact that SIMD execution represents a single control thread applied across a varying set of enabled PEs. The execution time is determined by the sum of the execution times in each step and not by the number of active PEs. Essentially the basic block index ranges over flops which must be executed at different times and the PE index ranges over flops which can be executed simultaneously.

3.3 SIMD Efficiency

The efficiency of a SIMD implementation does depend on the number of active PEs and can be calculated by comparing the ratio of the sequential time and the SIMD time scaled by the number of PEs assumed for the machine.

That is:

$$\text{Efficiency} = \langle \text{sequential time} \rangle / (\langle \text{number of PEs} \rangle * \langle \text{SIMD time} \rangle) \quad (3)$$

An efficiency of 0.1 means that, on average, only 1 out of every 10 PEs is enabled for every flop or basic block. Alternatively it can be read as saying that the architecture could have executed 10 times as many flops as it did execute in the time required for the program to run. Either way, the efficiency estimate is a key parameter in identifying the match between the problem solution embodied by the program and SIMD architectures.

Additional calculations can be performed by folding the notional PEs in ways appropriate to SIMD machines with fewer PEs (or floating point units) than the problem allows. For instance the allocation of multiple cells to a PE can result in better balanced loads and therefore higher efficiency particularly where the optional nature of a step follows some pattern. In addition, a capability for indexing or for the sharing of floating point units can generate averaging effects which can improve efficiency for optional codes.

4. Analysis Results

The application of the SIMD analysis to the climate model required the identification of notional PEs using program variables. The particular formula and variables used:

$10 * ((ins - 1) * 64 * 28 + (lg - 1) * 64 + (mg - 1) \bmod 64 + (k - 1) \bmod 10)$
 obviously depends on the programming of the model. In this formula the variables *ins*, *lg*, *mg* and *k* identify the hemisphere, latitude, longitude and vertical level respectively. Thus *ins* ranges from 1..2, *lg* from 1..28, *mg* from 1..64 and *k* from 1..10. The effect of the identification is to allocate an individual notional PE to each cell in the modelled atmosphere giving 35840 notional PEs.

4.1 PE Mappings

A direct analysis of these figures produces efficiency and flop times which reflect a mapping of the three dimensional problem onto a SIMD PE array. Knowledge of the problem domain suggests that much of the physics occurs in different layers, for example during surface physics a 3D mapping leaves a significant portion of the machine unused. This property suggests the use of a 2D mapping where columns within the atmosphere are folded to the same PE and using the horizontal dimensions for parallelism. The instrumentation results can be processed by folding the vertical layers considering the total SIMD execution time to be the sum of the maximums across the PEs within each layer. The result is an estimate of performance using 3,584 actual PEs.

The results summarised below also consider the case in which simultaneous execution is permitted for the same program segment applied to different vertical layers. This is the index option available on some SIMD architectures. It is important to bear in mind that the recoding necessary to make use of the feature may be significant.

4.2 Example Results

Figure 1 present results taken from the detailed analysis of the instrumentation run. The table shows the segment number identifying the basic block of the original FORTRAN. Note that many segments have no flops in them and so for this routine there were only five sections of connected FORTRAN which produced any flop counts. The middle portion of the table presents the average, minimum and maximum flops detected across the PEs subject to the mapping selected during the analysis as well as the total (sequential) flops in the segment and accumulated for the routine and finally the SIMD flops accumulated for the routine. This last figure provides a summation of the maximum flops in any segment across the cells and represents a measure of the *time* the routine would take on a SIMD architecture. The right hand portion of the table performs the efficiency calculation (3) for the segment and for the routine's accumulated flops.

segment	Floating Point Operations (3D mapping)						% Efficiency	
	average	min	max	segment	Sequential	SIMD	segment	total
339	0.90	0	1	32256	32256	1	90	90
340	1.81	0	17	64770	97026	18	10	15
347	0.00	0	2	112	97138	20	0	13
349	5.21	0	12	186624	283762	32	43	24
350	0.06	0	12	2220	285982	44	0	18

segment	Floating Point Operations (2D mapping)						% Efficiency	
	average	min	max	segment	Sequential	SIMD	segment	total
339	9.00	9	9	32256	32256	9	100	100
340	18.07	0	136	64770	97026	145	13	18
347	0.03	0	2	112	97138	147	1	18
349	52.07	48	78	186624	283762	225	66	35
350	0.62	0	30	2220	285982	255	2	31

Figure 1: Report produced by analysis of *rainda* routine.

4.3 SIMD Performance of Physics Routines

The results given in Figure 2 show a large bias towards the radiation calculations (*radfs*) in comparison to other physics. The radiation calculations need only be performed every two to three hours of model time and so the overall efficiency calculation needs to consider the time step of the model.

As the model is currently used, *radfs* is called every four time steps with the other physics calculated every step. As the model resolution is increased, the model timestep will decrease and the number of steps between evaluations of *radfs* will increase. For example, if the model resolution is doubled, the current timestep of 30 minutes will be reduced to 15 minutes, and *radfs* would be called every 8 to 12 steps (ie. every 2 to 3 hours).

Convective Physics	Serial Flops	3 D Mapping		2 D Mapping		2 D + Indexing	
		Flops	% Use	Flops	% Use	Flops	% Use
surfset	49,287	47	2	136	25	47	29
qsgam	2,322,432	72	90	648	100	648	100
negrm	68,179	68	2	84	22	84	22
hsflux	1,621,559	931	4	931	48	931	48
radfs	155,612,159	21,120	20	64,029	67	63,561	68
surfupa	122,181	136	2	136	25	136	25
hvertmx	3,338,959	518	17	3,317	28	3,249	28
gwdrag	1,892,464	110	48	530	99	530	99
cvmix	229,376	16	40	64	100	64	100
rainda	285,982	44	18	255	31	220	36
conv	300,537	149	5	648	12	429	19
surfupb	56,670	120	1	120	13	120	13

Figure 2: Summary of results for the physics routines.

4.4 Performance Factors

Apart from the efficiency calculations, the actual flop counts can be taken to indicate the run time benefit of applying more PEs using the 3D rather than one of the 2D mappings. Essentially a speedup in the physics calculations of about three can be envisaged for an investment of ten times as many PEs. However, as models increase the number of vertical layers the number of PEs required for the 3D mapping will grow linearly while the benefit will not. The performance on the 3D mapping is independent of the number of layers and the performance of the 2D mapping will follow the number of layers only for those routines applied in all layers.

Model Option	3 D Mapping		2 D Mapping		2 D + Indexing	
	Flops	% Use	Flops	% Use	Flops	% Use
Radiation	21,120	20	64,029	67	63,561	68
Other Physics	2,493	13	7,536	45	7,214	47
4 Other + 1 Rad	31,092	18	94,173	61	92,417	62
12 Other + 1 Rad	51,036	17	154,461	55	150,129	57

Figure 3: Calculations showing efficiency for weighted combinations of radiation and other physics.

As can be seen from Figure 3, increasing the non-radiation physics by a factor of four has only a minor effect on the overall results. This suggests the

results have a degree of robustness against developments in model physics which may introduce additional layer dependent workloads.

5. Conclusions

The results presented in this paper were taken from a Sun workstation. Similar results were obtained by flop counting using the CRAY Y-MP. The specific numbers differ between these two machines because the flop counting hardware on the CRAY Y-MP counts divisions as four flops, counts some integer operations as flops and the runtime libraries on the machines are independently coded. Despite this, the efficiency figures obtained from runs on the two machines were nearly identical. This represents the fact that the distribution of the workload described by the flop counts in both systems is essentially the same.

One issue often considered in climate modeling concerns workload balance particularly in the physics where large variations exist between cells and where the season and time changes the workload. The data collected in these experiments could be used to reveal patterns in the workload balance but this issue is not relevant to the SIMD execution mode. The efficiency results calculated here are effectively independent of the month or time of day selected for the analysis time step although the specific pattern of the flops changes significantly.

The efficiency results combined with projected improvements in model resolution suggest that SIMD climate modeling should be engineered around a 2D mapping despite the limitation this places on the available parallelism. They argue for a smaller number of faster processors rather than millions of slower processors. While the total SIMD flop times show that a performance gain of a factor of 3 can be obtained by the 3D mapping this comes at a cost in PEs linear in the number of vertical layers. If the horizontal resolution improves by a factor of two or four the resulting growth in horizontal parallelism provides substantial opportunity for larger machines without the drop in efficiency involved in the 3D mapping.

Finally the bit slice mapping for floating point numbers used on the latest CM-2 and the reported performance from that arrangement meshes well with the notion of 2D mapping for climate models. A larger connection machine with the same FPU to PE arrangement would be on the path to a solution capable of providing sufficient SIMD performance for long term climate modeling.

6. Acknowledgement

The work described in this paper was conducted as part of a collaborative project between the CSIRO Divisions of Information Technology and Atmospheric Research within the DIT High Performance Computation Project. The High Performance Computation Project is a joint project between the Commonwealth Scientific and Industrial Research Organisation (CSIRO) Division of Information Technology and the Royal Melbourne Institute of Technology (RMIT).

7. References

- 1 U.S. Department of Energy (DOE), "Building an Advanced Climate Model: Program Plan for the CHAMMP Climate Modelling Program", DOE, Washington, 1990.
- 2 Gordon, H. and B. Hunt, "Interannual variability of the simulated hydrology in a climatic model - implication for drought", *Climate Dynamics* 1, 113-130, 1987.
- 3 Hunt, B. and H. Gordon, "Diurnally varying regional climate simulations", *International Journal of Climatology* 9, 331-356, 1989.
- 4 Hunt, B. and H. Gordon, "Droughts, floods and sea surface temperature anomalies: a modelling approach", *International Journal of Climatology* 11, 347-365, 1991.
- 5 Whetton, P. and A. Pittock, "Australian region intercomparison of the results of some general circulation models used in enhanced greenhouse experiments", CSIRO Division of Atmospheric Research Technical Paper No. 21, CSIRO, Australia, 1991.
- 6 Swarztrauber, P.N. and R.K. Sato, "Solving the Shallow Water Equations on the Cray X-MP/48 and the Connection Machine 2", *Topics in Atmospheric and Oceanic Sciences*, Springer-Verlag, pp. 260-276, 1990.
- 7 Bourke, W., B. McAvaney, K. Puri, and R. Thurling, "Global modeling of atmospheric flow by spectral methods", in: Chang, J., (ed.), *General Circulation Models of the Atmosphere*, Academic Press, New York, 268-324, 1977.
- 8 Bourke, W., "Spectral methods in global climate and weather prediction models", in: Schlesinger, M., (ed.), *Physically Based Modelling and Simulation of Climate and Climate Change - Part I*, Kluwer Academic Publishers, Dordrecht, 169-220, 1988.
- 9 Gordon, H., "A flux formulation of the spectral atmospheric equations suitable for use in long term climate modelling", *Monthly Weather Review* 109, 56-64, 1981.
- 10 Mathieson, I.D., "Vertical Profiling on the CRAY", *JSF Newsletter*, CSIRO, March 1991.