Jul 1st, 12:00 AM

Functional Parallelization of a Land Surface Model in Regional Climate Modeling

Vimal Sharma

David Swayne

David Lam

Murray MacKay

Wayne Rouse

See next page for additional authors

Presenter/Author Information
Vimal Sharma, David Swayne, David Lam, Murray MacKay, Wayne Rouse, and William Schertzer

Functional Parallelization of a Land Surface Model in Regional Climate Modeling

Vimal Sharma[a], **David Swayne**[a], **David Lam**[b], **Murray MacKay**[c], **Wayne Rouse**[d], **William Schertzer**[b]

[a]University of Guelph, Computing Research Laboratory for Environment, Guelph, ON, Canada.
(vsharma@uoguelph.ca)

[b]Environment Canada, National Water Research Institute, Burlington, ON, Canada.

[c]Meteorological Service of Canada, Downsview, ON, Canada.

[d]McMaster University, Hamilton, ON, Canada.

Abstract: Parallel computing is a very useful tool for computing intensive and time constrained real time problems. Depending on the size of the grid and processors available in the cluster, a group of nodes or processes in the grid can be represented by an individual processor and it can be responsible for their computational needs. This increases the accuracy of the model by allowing finer grid sizes, also leading to savings in time. Our study, utilizes the Canadian Land Surface Scheme (CLASS), a well-tested serial general land/atmosphere interaction model. CLASS is a vertical one-dimensional model and spatially adjacent nodes in the grid do not interact. This model computes heat and moisture fluxes for bare ground (fractional coverage by ground), ground covered with snow (fractional coverage by snow), ground with canopy (fractional coverage by ground), and ground with both snow and canopy. Within each spatial grid cell, these fractions are combined. In this paper, we demonstrate the need of parallelizing the serial CLASS model and discuss the designs to implement it. This will enable finer grid sizes leading to higher accuracy of the model and a corresponding decrease in individual processor computing time, when compared to the serial CLASS model. It was observed that a serial farm kind of design suits our design constraints and has been successfully implemented.

Keywords: CLASS; GCM; Parallel computing.

1. INTRODUCTION

The Laurentian Great Lakes have a significant impact on the weather and climate of the Great Lakes Basin. Alternatively the meteorological conditions can also affect the lake temperature and thermal characteristics. There are many lakes in Canada, both large and small, in which the lake-atmosphere interaction is not well understood. Understanding this relationship is important for addressing the increasing concerns on regional climate change. With regard to climate modeling, there is a critical need to link atmospheric and lake models to achieve better accuracy of prediction of climate impacts on such issues as lake water quantity and quality.

Climate models, like numerical weather prediction models, are computationally intensive. The computational power required increases manifold with finer grid sizes for the models implemented over large lakes and areas. As the grid gets coarser, however, the accuracy of the models tends to suffer.

Parallel computing can be a good tool for this kind of problem. Parallel computing can be used to harness the collective computational power of many

computer clusters available for same problem. Depending on the size of the grid and number of processors available in the cluster, a group of nodes in a grid can be represented by individual processors that can be responsible for its computational needs. Accuracy increases by allowing finer grid sizes, with a potential saving in time.

Our research group is focused on linking a lake component (Swayne et al. [2003]) to the existing serial land model, which needs to be parallelized to harness maximum usage of a parallel system. The first logical step in this scenario is to parallelize the existing land model. We require models to iterate with 15-30 min time step. The time interval does not seem very computationally intensive but when the model is run on a grid comprising of half of Canada with each node running copy of this model, the utility of parallel processing becomes evident.

In this study, the commonly used General Circulation Model (GCM), the Canadian Land Surface Scheme's (CLASS) (Verseghy [1991], Verseghy et al. [1993]) parallel version was designed and implemented.

2. BACKGROUND

The study objective is to parallelize the CLASS model which is coded in FORTRAN. This model has been developed and tested over considerable time (Verseghy [2000], Comer et al. [2000]). The model is a vertical one-dimensional model. It does not have any computational dependencies on its neighboring nodes on the grid. CLASS involves computation of heat and moisture fluxes for bare ground (fractional coverage by ground, FG), ground covered with snow (fractional coverage by snow, FSN), ground with canopy (fractional coverage by ground, FC) and ground with both snow and canopy, every time step for each grid node. The finite-difference one-dimensional heat conservation equation applied to each layer for obtaining the change in average layer temperature $\bar{T}_i$ over a time step Δt is given by (Verseghy [1991]):

$$\bar{T}_i(t+1) = \bar{T}_i(t) + [G(z_{i-1},t) - G(z_i,t)]\frac{\Delta t}{C_i \Delta z_i} \pm S_i \qquad (1)$$

where, $G(z_{i-1},t)$ and $G(z_i,t)$ are the downward heat fluxes at the top and bottom of the layer, respectively, C_i is the volumetric heat capacity of the

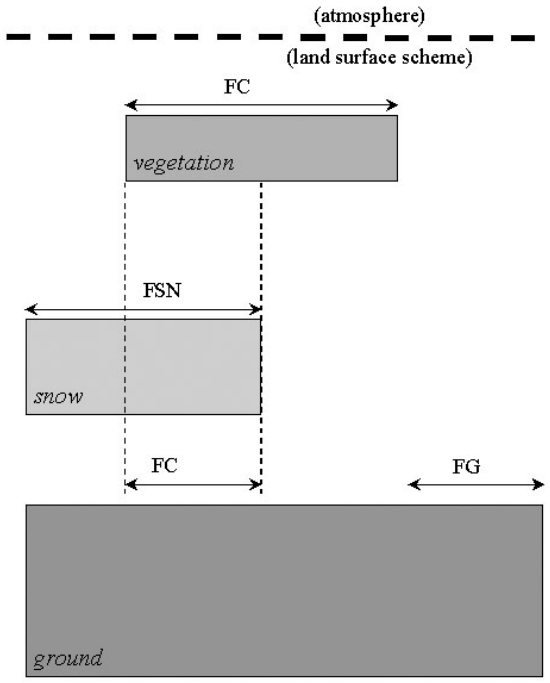

Figure 1. Class layout

soil, Δz_i is the layer depth, and S_i is a correction term applied in case of freezing or thawing, or the percolation of ground water. The change in average volumetric liquid moisture content $\bar{\Theta}_{1,i}$ over a time step is given by (Verseghy [1991]):

$$\bar{\Theta}_{1,i}(t+1) = \bar{\Theta}_{1,i}(t) + [F(z_{i-1},t) - F(z_i,t)]\frac{\Delta t}{\Delta z_i} \qquad (2)$$

where, $F(z_{i-1},t)$ and $F(z_i,t)$ represent the liquid water flow rates at the top and bottom of the layer respectively.

The node or grid square representative value is given by weighted average for each iteration (Fig 1). All the required inputs for the model are provided by the regional climatic model (RCM) and CLASS feeds back its results to the RCM for each node, every iteration. The model iterates at a time interval of 30 minutes for the total time period of one year for the benchmark dataset.

The following are important facts about CLASS:

- Class defines 3 soil layers of fixed thickness 0.10 m, 0.25m and 3.75 m,
- Mean temperature, liquid water content and ice content evolve in time for each layer,
- In soil, heat is transferred by conduction and moisture flux follow Darcy's law, and
- Infiltration of rainwater and phase changes are also accounted.

The following are inputs to CLASS:

- Wind components at top of surface layer,
- temperature at top of surface layer,
- specific humidity at top of surface layer,
- downward global solar radiation,
- downward long wave radiation, and
- precipitation rate.

3. DESIGN CONSTRAINTS

The computation of all the nodes should be completed much faster in wall clock time than 15 min (900 sec) for 15 min simulation time intervals. Actually, the number of grid nodes is far greater than the number of processors available. Linkage to a lake component at a later stage had to be kept in mind for the present design. There should be minimum change to existing tested code developed over the years. The design should not replace any of the routines already implemented in CLASS but call in components (other independent models) to supplement routines that have not been implemented e.g. lake component. Therefore merging CLASS and other models in to one program file are not possible as done by Soulis et al. [2000].

4. DESIGN AND RESULTS

In light of the background in previous section, the following two scenarios arose in design of parallel CLASS:

1. Task parallel implementation
2. Serial Farm

The design of parallel CLASS has been implemented on high performance computing (HPC) Guelph cluster of SHARCNET (The Shared Hierarchical Academic Research Computing Network). The cluster contains 27 Compaq Alpha ES40 nodes with 4 GB memory per node, running Red Hat Linux operating system. Each node has 4, 833 MHz processors.

4.1. Task parallel design

The structure of CLASS (Fig 1) at glance showed that there is good potential for computing of snow, vegetation and bare ground components in parallel. A closer look at the code shows that the implementation of CLASS has many COMMON blocks (global variables in FORTRAN) that need to be communicated to all these components. In other words, all the needed global variables have to be broadcast every iteration, to each processor and once computation is complete for each component, all the updated global variables need to be broadcast again. In addition, in case there is any dependency on any common updating variable between the components, critical sections have to be defined and the results communicated among the components. Thus, additional significant communication overhead is added. As a consequence, a different design approach should be explored to implement parallel CLASS.

4.2. Serial farm design and implementation

CLASS, being a vertical model needing no communication with its neighboring nodes, is an ideal candidate for *serial farm* design implementation. In *serial farm* design, a copy of a program, which is self-contained and does not have any dependencies of communication with other grid nodes, is run on as many processors available in the cluster. Thereby, concurrently running as many copies of the program as the number of processors available. As previously mentioned in the design constraints that there are many more grid nodes than processors available for computation. Therefore, there is a need of multithreading with multiprocessing of computation. In this study, a manager program has been coded in C to create and manage the threads for each processor. This manager computes the number of threads needed per processor running CLASS. In the case of an odd number of grid nodes, the last processor computes the extra nodes too. The manager creates all the threads and runs a copy of CLASS for each node in each thread. In this design we only allow (unlock) one thread at a time to complete all its computation for a single iteration. Consequently, there is no time wasted in thread swapping midway and imposing additional overhead. As soon as a thread completes

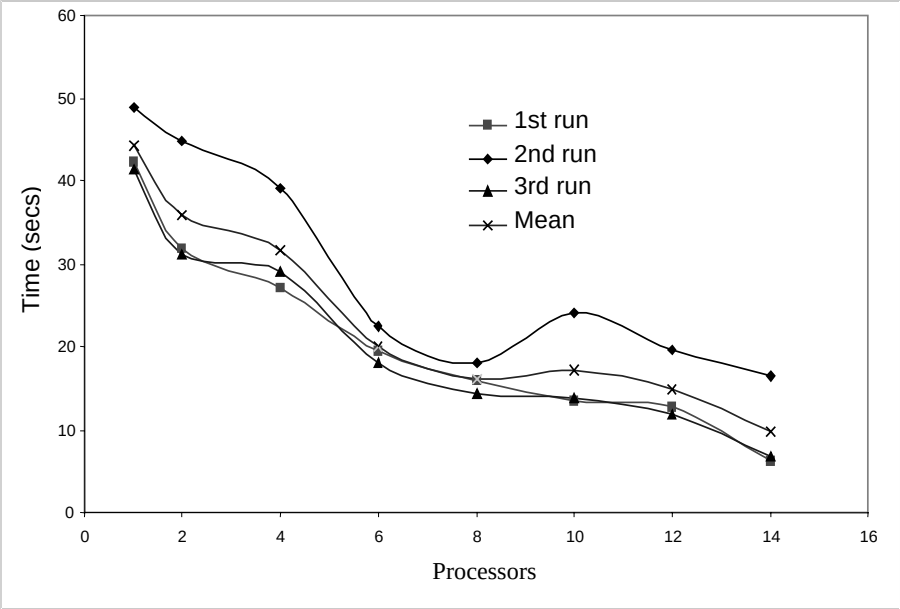

Figure 2. Result of 3 runs for 50 grid nodes with variation of number of processors.

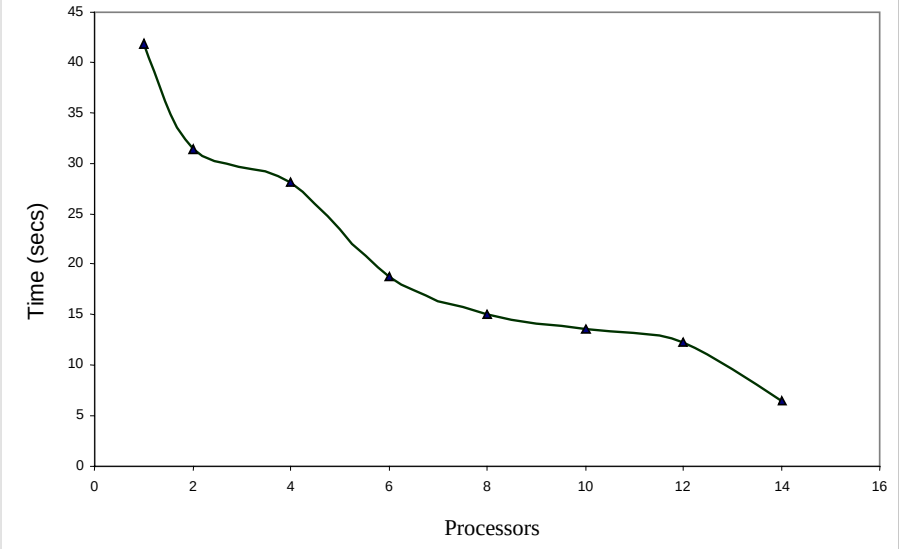

Figure 3. Mean of 2 representative runs for 50 grid nodes with variation of number of processors.

its computation for the iteration, it signals to the manager program that it has completed its computation. Then the manager locks this thread and allows (unlocks) the next thread to start computation for the next node. In this manner all the threads (each representing one node on the grid) completes computation for a single iteration. The process is repeated again for the next iteration. This design is replicated on all the available processors.

The maximum number of threads that can be created are limited by operating system constraints of 2GB memory per user. Each thread is given 2 MB of memory by the system; 1024 is the maximum number of threads that can be created and used per processor. We cannot terminate any thread because there is a need to maintain the profile by the model, so with this constraint there can be at most 1024 nodes processed on single processor. The design was

successfully implemented. The following two timings were recorded to test the design:

1. multiple runs of 1 thread on 1 processor to estimate the time taken to run CLASS.

2. multiple runs of the design for 50 grid nodes with varying number of processors (1-14)

It was found that after 5 runs of our design that time taken to run 1 thread on 1 processor ranged between 0.86 –3 sec on SHARCNET. The time included the initial setup time taken by CLASS. The difference in timing can be attributed to the queuing of jobs for execution on Guelph SHARNET cluster.

Three runs (with number of processors varying between 1 and 14) for 50 grid nodes were executed to test if our design significantly shortened the amount of time required for computation.

Fig. 2 presents the results from 3 runs done. These runs were done at different time of the day to account for queuing system on the SHARCNET cluster at Guelph. Run 1 and 3 overlap each other and seem to be better representation of timing of our design. The trend observed on Run 2 can be attributed to queuing system of the SHARCNET cluster.

Fig 3 shows the mean of run 1 and 3 timing. It is seen for 50 grid nodes using 14 processors the computation from all the nodes was completed in 85.19% less time as compared to when using 1 processor.

5. CONCLUSIONS

From the above results it can be seen that if we run 900 threads (approximating that it takes 1 sec per processor to run CLASS) with 100 processors we can model about 90,000 grid nodes in 15 minutes. This means if each node represents 25 km^2 area, with our design we can model roughly 2,250,000 km^2 area. It can be concluded that a successful implementation of parallel CLASS using *serial farm* design the above-mentioned constraints has been achieved.

6. ACKNOWLEDGEMENTS

The authors thank David McCaughan, HPC consultant for SHARCNET for his assistance. This research is supported by SHARCNET, Canadian Foundation for Climate and Atmospheric Sciences, NSERC, and Environment Canada. The referees are also gratefully acknowledged.

7. REFERENCES

Comer, N.T., P.M. Lafleur, N.T. Roulet, M.G. Letts, M. Skarupa, and D. Verseghy. A test of the Canadian Land Surface Scheme (CLASS) for a variety of wetland types, *Atmosphere-Ocean*, 38(1): 161-179, 2000.

Soulis, E.D., K.R. Snelgrove, N. Kouwen, F. Seglenieks, and D.L. Verseghy. Towards closing the vertical water balance in Canadian atmospheric models: Coupling of the land surface scheme CLASS with the distributed hydrological model WATFLOOD, *Atmosphere-Ocean*, 38(1): 251-269, 2000.

Swayne, D.A., D.C.L. Lam, M. Mackay, W. Rouse, W.M.S. Schertzer. Preliminary assessment of the interaction between the Canadian Regional Climate Model and lake thermal-hydrodynamic models, *Proc. 5th International Symposium on Environmental Software Systems (ISESS '03), May 27-30, 2003, Semmering, Austria*, 161-177, 2003.

Verseghy, D.L. CLASS – A Canadian Land Surface Scheme for GCMs. I. soil model, *International Journal of Climatology*, 11: 111-133, 1991.

Verseghy, D.L., N.A. Mcfarlane and M. Lazare. CLASS – A Canadian Land Surface Scheme for GCMs. II. vegetative model and coupled runs, *International Journal of Climatology*, 13: 347-370, 1993.

Verseghy, D.L. The Canadian Land Surface Scheme (CLASS): its history and future, *Atmosphere-Ocean*, 38(1): 1-13, 2000.