PVM Implementation Of Uncertainty Analysis

Jose L. Jimenez and Cheng Wang *
jljimene@mit.edu and chengw@mit.edu
Departments of Mechanical and Chemical Engineering, MIT

In practice, the exact values of numerous quantities entering a model are usually unkown because they can not be precisely measured. Such a problem brings the need to systematically and explicitly address uncertainty in these quantities in order to investigate its effects on the output of the model. To address these uncertainties we are required to perform an uncertainty analysis, whose goals are twofold: first, to obtain the probability density function (PDF) of the model outputs; second to compare the importance of each input uncertainty in terms of its relative contributions to uncertainty in the outputs [Tatang94].

The most common situation is one in which we have a deterministic " black box model," and we know or can estimate the PDFs of the model inputs. The standard method for evaluating the PDF of the output is a Monte Carlo simulation (MCS) [Tatang94], in which random values for each input parameter are generated, and the model is run a large number of times, until the changes on the PDF of the outputs calculated from the new sampling points are small. The main disadvantage of this approach is that the black box model has to be run a large number of times, and the computation time is very large if the model has many uncertain parameters, or if the individual runs of the model take a long time (e.g. climate models).

An alternative method, termed the Deterministic Equivalent Modeling Method (DEMM), has been developed at MIT [Tatang94], which yields the same results of the MCS, but only needs between roughly 1/20 and 1/1000 of the runs of the black box model. This breakthrough makes possible to perform uncertainty analysis on many models for which MCS was too costly.

The main task of both MCS and DEMM is running the black box model for a large set of different input parameters, so its parallelization is easy, due to the lack of interprocessor communication. We have utilized PVM 3.3 to parallelize both methods. Parallel Virtual Machine (PVM) [Geist94] is a software package developed by a research group centered at the Oak Ridge National Laboratory, which allows a heterogeneous network of parallel and serial computers to appear as a single concurrent computational resource.

The specific problem that we computed was the uncertainty analysis of radiative forcing of climate by anthropogenic sulfate aerosols [Charlson92]. Since the climate system is fundamentally driven by radiation, in general any perturbation in the radiation field may cause changes to the system. To be sure, aerosols can affect the earth radiation budget directly through scattering and absorbing solar and terrestrial radiation and indirectly through modifying the cloud field. Although the indirect forcing of aerosols is hard to evaluate at this stage, the direct one can be estimated from radiative transfer equations pro-

*Jose L. Jimenez and Cheng Wang are advised by Prof. Gregory McRae, Department of Chemical Engineering, MIT. Address: Room 16-315, 77 Massachusetts Avenue, Cambridge MA 02139.

vided that some key parameters are given. During recent years, several research groups have addressed this problem and given contradictory predictions for global climate. Given the huge uncertainties for some of the model input parameters, an uncertainty analysis is deemed fundamental to resolve this question.

We have used a model by Charlson et al. [Charlson92], with 9 input parameters, and a single output variable: the direct radiative forcing by the aerosols. We have implemented the PVM system on a network of 6 heterogeneous workstations: two DEC Alpha, three DEC 5000, and a Silicon Graphics Indigo2. We have chosen a master/slave model, in which the master program, running on a DEC 5000, spawns and directs the slave programs on the other five computers. As our computation consists of running a model for a large set of input parameters, the master program divides the input file in small pieces, and sends one to each slave. When a slave has finished the calculations on that piece it sends a signal to the master, which sends the slave another piece of data, until all the input data have been computed on. In this way we achieve dynamic balancing of the processors, so for example if one of the slaves has a widely varying load, we can adapt to it automatically. The optimal size of the pieces of the input data sent to the slaves depends on various parameters in general, including the size of the input file, the time to perform one calculation, the relative speeds of the slaves, and the speed of the communications.

As we can see in the figure, we were able to obtain a speedup factor of 2 when running on the four fastests slaves (two ALPHA, one SGI, one DEC 5000) , compared to the case in which the problem was run in the fastest machine alone (ALPHA). The computing power of the four slaves used is about 2.4 times that of the fastest machine, so taking into account that the slaves had other loads during the execution of our program, PVM uses the resources available very efficiently.

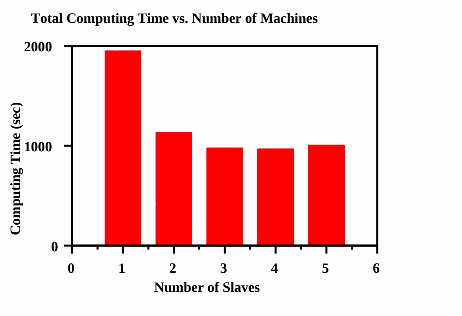

Figure 1: Computing Time

An interesting result is that on a subsequent run in which another slave (DEC 5000, 8 times slower than the fastest machine) was used, the time to complete the task actually increased.

We conclude that the use of PVM for computing the MCS or DEMM method is a very efficient and easy to use way of reducing the computing time, if several workstations of comparable speed are available.

References

[Tatang94] M. A. Tatang. Direct Incorporation of Uncertainty in Chemical and Environmental Engineering Systems. Ph.D. thesis, MIT, October 1994.

[Charlson92] R.J. Charlson and others. Climate Forcing by Antropogenic Aerosols. *Science*, 255:423-430, 1992

[Geist94] A. Geist and others. *PVM 3 User's Guide and Reference Manual.* MIT Press, Cambridge, MA 02139, 1994.