18'th International Conference on Transport Theory (18 ICTT)
Rio de Janeiro, RJ, Brazil, July 20-25, 2003

PERFORMANCE ANALYSIS OF RADIATIVE TRANSFER ALGORITHMS IN A PARALLEL ENVIRONMENT

R. P. Souto[1], H. F. Campos Velho[2], S. Stephany[2], E. S. Chalhoub[2]
[1] Programa de Pós-graduação em Computação Aplicada (CAP/INPE)
[2] Laboratório Associado de Computação e Matemática Aplicada (LAC/INPE)
INPE - Instituto Nacional de Pesquisas Espaciais
Caixa Postal 515, CEP 12245-970 São José dos Campos, SP - Brasil.
E-mail: [roberto,haroldo,stephan,ezzat]@lac.inpe.br

ABSTRACT

Three algorithms for solving the radiative transfer equation (RTE) were studied: Hydrolight, PEESNA and LTSN. These algorithms correspond, respectively, to invariant imbedding, analytical discrete-ordinates and LTS_N methods. As a first step, the performance of each algorithm was evaluated running in the same sequential machine. The related codes were used in a Hydrological Optics typical coastal water test case, in order to calculate the surface-emergent radiation intensities (radiances) given the incident radiances and inherent optical properties such as the absorption and the scattering coefficients. Timing and profiling of the three codes was performed in order to evaluate processing times and to identify performance bottlenecks. Next, each algorithm was studied concerning the feasibility of its parallelization using the MPI message passing communication library and execution in a distributed memory machine, a multicomputer based on IA-32 architecture. The three codes perform spatial discretization of the domain and Fourier decomposition of the radiances obtaining independent azimuthal modes. Therefore, an independent RTE can be written for each azimuthal mode and can be assigned to a different processor, in a parallel implementation. The speed-up that can be achieved increases with the fraction of time spent in the azimuthal mode, but total execution time is also an important issue. Results are discussed and further strategies are proposed.

1. INTRODUCTION

One of the motivations of this work was to improve the performance of a radiative transfer equation (RTE) method in order to solve the inverse problem of estimating inherent optical properties of natural waters. In the employed implicit methodology, the inverse problem is formulated as an optimization problem which minimizes an objective function. For each candidate solution, the value of this function is given by the square difference between experimental values and the data given by the direct model, the RTE. A typical estimation may demand hundreds of iterations and, therefore, RTE method performance is an important issue. As parallel implementations are usually cost effective in order to gain performance, three selected RTE solvers were analized concerning their parallelization potential: the PEESNA [1], based on the analitical discrete ordinate method, the LTSN code, which implements the Laplace tranform on the analitical discrete ordinate equation, and the Hydrolight [2], that uses an invariant imbedding method.

These three methods decompose the radiance in independent azimuthal modes and solve the RTE in each mode. Therefore, parallelization of the code can be accomplished by assigning an azimuthal mode to each processor. Timing and profiling of the three sequential codes was performed in order to compare the relative performance and to identify the time-consuming routines. The profile of the execution time show that a significant fraction of the total time is spend in all codes to calculate the independent azimuthal modes.

Parallel versions of each code were written using calls to the message passing communication library MPI (Message Passing Interface) [3]. These versions were executed in a distributed memory parallel machine based on IA-32 architecture for a specific test case. The performance of the three parallel codes is then compared.

2. RADIATIVE TRANSFER EQUATION

The Radiative Transfer Equation (RTE) for radiances I is given by

$$\mu\frac{\partial}{\partial\tau}I(\tau,\mu,\varphi)+I(\tau,\mu,\varphi)=\frac{\varpi_0}{4\pi}\int_{-1}^{1}\int_{0}^{2\pi}\beta(\tau,\mu,\varphi;\mu',\varphi')I(\tau,\mu',\varphi')d\varphi'd\mu'+S(\tau,\mu,\varphi) \quad (1)$$

where τ is the optical variable. $\mu\in[-1,1]$ e $\varphi\in[0,2\pi]$ are the cosine of the incident polar angle θ and the incident azimuthal angle, respectively. ϖ_0 is the constant single scattering albedo. The scattering phase function $\beta(\tau,\mu,\varphi;\mu',\varphi')$, gives the scattering beam angular distribution and the source term is $S(\tau,\mu,\varphi)$.

The three RTE solvers used in this work, Hydrolight, PEESNA and LTSN are related to the following methods, respectively:

Invariant Imbedding: This method is used in the Hydrolight code and its description can be found in [4]. In this approach, the linear two point boundary condition is transformed into a non-linear initial value problem (matrix Riccati equation), solved by a well established integrator. The Hydrolight code allows to simulate the wind blown water surface.

Analytical $\mathbf{S}_N$ Method: In the PEESNA code, the radiance is split up into unscattered and scattered components. The solution for the former is given by a simple expression, while the latter is expanded by elementary solutions of the discrete ordinate equations. The coefficients of this expansion are obtained by solving a linear algebraic equation. The method was presented by Chalhoub and Garcia [1].

$\mathbf{LTS}_N$ Method: The LTSN scheme appeared in the early nineties in the neutron transport context [5], and was then extended to radiative transfer problems [6]. Its convergence was established using the C_0-semi group theory [7].

3. PARALLEL PROCESSING

The prevailing trend in the search for high performance is the use of parallel machines due to their good cost effectiveness. Two parallel architectures are usually considered: shared memory and distributed memory machines [8]. In the former class, the multiprocessors, all processors access a unique memory address space and there are scalability constraints.

In the latter class, off-the shelf machines called nodes are interconnected by a network composing a cluster or a MPP (massive parallel processors), a parallel machine with hundreds or thousands of nodes using a very fast interconnection scheme.

The processors of each node access only their local memories and data dependencies between node memories enforce communication by means of routines of a message passing library, like MPI (Message Passing Interface) or PVM (Parallel Virtual Machine). All current world top performance machines (supercomputers) are MPP's. Another point is the use of scalar or vector processors in parallel machines. The use of vector processors is restrained to specialized applications as wheather forecasting due to cost issues, but even in this area there is a migration from shared-memory vector-processor nodes to scalar-processor MPP's.

MPI supports the SPMD (single program multiple data) scheme in which each processor/node executes the same subset of instructions in a subdomain of data defined according to its rank. Data dependencies between processors require calls to the MPI library. On the other hand, a single processor/node may perform tasks like gathering partial results obtained by every node and broadcasting the global result to all other nodes, also by means of MPI calls.

An important issue is to maximize the amount of computation done by each processor and to minimize the amount of communication due to the MPI calls in order to achieve good performance. The speed-up is defined as the ration between the sequential and parallel execution times. A linear speed-up denotes that processing time was decreased by a factor of n when using n processors and can be thought as a kind of nominal limit. Efficiency is defined as the ration of the speed-up by n and thus it is 1 for a linear speed-up. Usually, communication penalties lead to efficiencies lesser than 1. Exceptionally, as data is partioned among processors, cache memory access can be optimized in such way that superlinear speed-ups can be attained, i.e. efficiencies greater than 1.

In this work, parallel execution was performed on a distributed memory parallel machine combining a low cost architecture and free software. The cluster is composed by 17 monoprocessed IA-32 scalar nodes running Linux and a Fast Ethernet switch. A Fortran 90 compiler and a message passing communication libray, the MPI (Message Passing Interface) were used to parallelize the code.

4. RESULTS

An inverse test case was chosen in order to test the performance of the three RTE solvers. This test case is related to typical coastal waters with parameters shown in table 1. The phase function was modeled by the Henyey-Greenstein function:

$$\beta(g;\psi) \equiv \frac{1}{4\pi}\frac{1-g^2}{(1+g^2-2\ gcos\psi)^{3/2}}, \tag{2}$$

where $g = 0.924$ for this case and show no significant loss of accuracy in comparison to a 174-term Legendre expansion.

Table 1. Water parameters for the chosen test case

Parameter	Meaning	Value
ϖ_0	single scattering albedo	0.5503
ζ	optical depth	11.94
μ_0	incident polar angle cosine	0.8
φ_0	incident azimuthal angle	0
g	Henyey-Greenstein parameter	0.924
L	anisotropy order	173
N	quadrature order S_N for PEESNA	132
N	quadrature order for LTSN	174

The timing of the three sequential codes for the chosen test case show significant differences. For instance, the PEESNA code was about 10 times faster than the LTSN code and Hydrolight was by far the fastest. It should be pointed out that this code is divided in two parts and the first one can be formerly executed in order to perform tasks like the phase function discretization and to store results in binary files. The second part, the actual RTE solver, then uses this "fixed" stored data and therefore runs very fast.

The analysis of the execution time profiles of these codes show that the the radiance calculation of the azimuthal modes account for 40% of the total time in the case of the PEESNA code, 88% in the LTSN code and 93% in Hydrolight. These profiles were obtained by means of the *gprof* Unix/Linux profiling tool. In all codes, as the azimuthal modes are independent, parallelization can be performed in order to assign a mode for each processor. A better speed-up was expected for the LTSN and Hydrolight codes as most of the processing time is spent in the azimuthal mode radiance calculation. This assumption was confirmed by the performance results shown in table 2 and in figures 1 and 2.

The LTSN speed-up scales better with the number of processors than the PEESNA one. The Hydrolight parallel performance was poor. The reason is that the it has to sum up 8 matrices for each azimuthal mode. This sum is done by the REDUCE MPI command, which causes a good amount of communication between processors for the current parallel version, that uses a straightforward and thus inefficient scheme to send these matrixes to the output processor. This can be observed in table 3: as the number os processors increases, the amount of time spent for the integration of the Riccati equations (denoted by *Riccati*) scales down, while the amount of time for the REDUCE command (denoted by *communic.*) scales up, precluding a good parallel performance.

Table 2. Speed-up and efficiency values of the three codes for p processors

	Hydrolight			PEESNA			LTSN		
p	Time(s)	Speed-up	Effic.	Time(s)	Speed-up	Effic.	Time(s)	Speed-up	Effic.
1	4.19			22.64			233.10		
2	3.26	1.29	0.643	11.63	1.95	0.973	116.96	1.99	0.996
4	3.16	1.33	0.331	6.07	3.73	0.932	59.27	3.93	0.983
8	3.34	1.25	0.157	3.30	6.85	0.857	29.94	7.79	0.973
16	3.75	1.12	0.070	1.92	11.79	0.737	15.12	15.42	0.964

Table 3. Hydrolight performance breakdown for p processors

	Time	Speed-up	Effic.	Riccati	communic.
1	4.19				
2	3.26	1.29	0.643	2.09	0.66
4	3.16	1.33	0.331	1.19	1.30
8	3.34	1.25	0.157	0.58	1.97
16	3.75	1.12	0.070	0.27	2.61

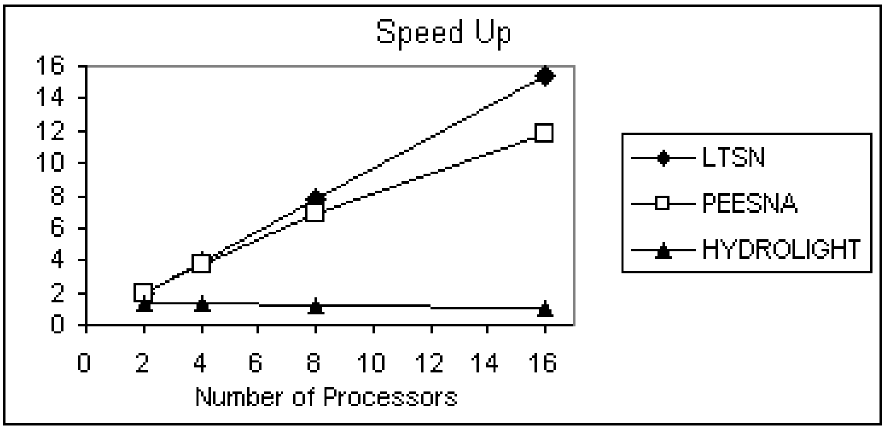

Figure 1. Speed-up of the three parallel codes.

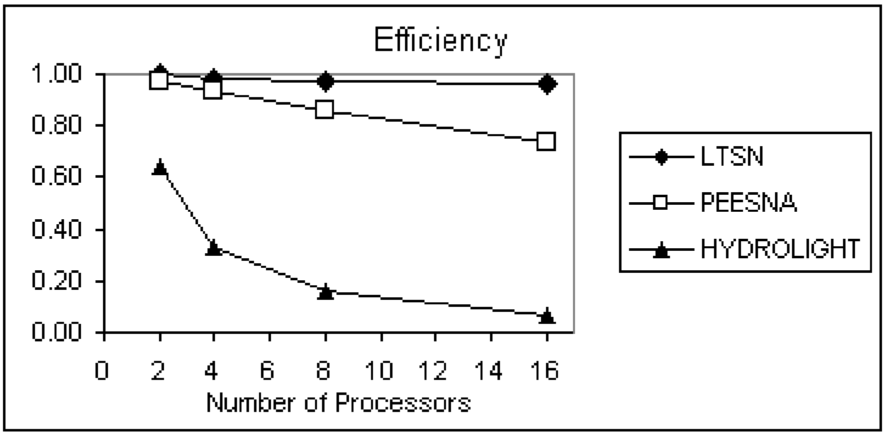

Figure 2. Efficiency of the three parallel codes.

5. CONCLUSIONS

The analysis of the performance of the sequential codes show that although the PEESNA and Hydrolight are faster than the LTSN, the latter spends most of the azimuthal mode radiance calculation time in the inversion of the mode-associated LTSN matrix. Recent advances of the LTSN method include the diagonalization of this matrix [9] and therefore a significant performance gain can be expected for its sequential version. As a further step, it is intended to evaluate this new version for the current test case in order to repeat the comparison of its sequential performance with the PEESNA and Hydrolight codes. In addition, a new parallel version of the LTSN method would be also generated.

The analysis of the parallel versions show that the LTSN scales better with the numer of processors than PEESNA and that Hydrolight parallel performance was poor for this initial version. Another further step would be to rewrite this particular part of the parallel code in order to improve its performance.

However, it is dificult to make a conclusive performance comparison of the three codes as there are many particular issues and only one test case was focused. Perhaps the improved parallel versions of LTSN and Hydrolight may change the results presented in this work.

It seems that the current trend is to improve the spatial resolution of simulations that require solving the RTE. Another point is that inverse-problem iterative solvers demands hundreds or even thousands of iteractions, with the RTE being solved at every one. Therefore, performance is a significant issue for RTE solvers and the use of parallel version of the existing solvers seems to be a requirement. On the other hand, new RTE algorithms may be chosen or designed devising the development of parallel versions.

ACKNOWLEDGMENTS

Author S. Stephany thanks FAPESP, São Paulo State Foundation for Research Support, for the support given to this study through a Research Project grant (process 01/03100-9). Author R. P. Souto acknowledges finantial support by CNPq-Brazil (2001).

REFERENCES

[1] E. S. Chalhoub, R. D. M. Garcia. "The equivalence between two techniques of angular interpolation for the discrete-ordinates method", *J. Quant. Spec. & Rad. Transfer*, **v. 64**, pp. 317–535 (2000).

[2] C. D. Mobley *Hydrolight 3.0 user's guide*, SRI Intl., Menlo Park, USA (1995).

[3] W. Gropp, E. Lusk, A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, The MIT Press, Cambridge, USA (1999).

[4] C. D. Mobley, *Light and Water: Radiative Transfer in Natural Waters*, Academic Press, San Diego, USA (1994).

[5] L. B. Barichello, M. T. Vilhena, “A General Approach to One Group One Dimensional Transport Equation, *Kerntechnik*, **58**, pp. 182-184 (1993).

[6] C. F. Segatto, M. T. Vilhena, “Extension of the LTS_N Formulation for Discrete Ordinates Problem without Azimuthal Symmetry”, *Annals of Nuclear Energy*, **21**, pp. 701-710 (1994).

[7] R.P. Pazos, M.T. Vilhena, “Convergence in Transport Theory”, *Applied Numerical Mathematics*, **30**, pp. 79-92 (1999).

[8] K. Dowd, C. Severance, *High Performance Computing*, O'Reilly & Associates Inc., Sebastopol, USA (1998).

[9] C. F. Segatto, M. T. Vilhena, M. G. Gomes, “The one-dimensional LTSN solution in a slab with high degree of quadrature”, *Annals of Nuclear Energy*, **26**, pp. 925-934 (1999).

[10] S. Stephany, R. V. Correa, C. L. Mendes, A. J. Preto, “Identifying Performance Bottlenecks in a Radiative Transfer Application”, *Applications of High-Performance Computing in Engineering*, WIT Press, Southampton, UK, pp. 51-60 (2000).