

Estudo Comparativo de Códigos Paralelos em Fortran, C e Java na Análise de uma Antena Monopolo, Utilizando Técnica Numérica de FDTD

Josivaldo de S. Araújo¹

Universidade Federal do Pará (UFPA), DEEC/LANE, CEP: 66075-90, Belém-PA.

Universidade da Amazônia (UNAMA), CCEET, CEP: 66060-902, Belém-PA

josivaldo@bcc.unama.br

Carlos Leônidas de S.S. Sobrinho; Rodrigo M. S. Oliveira; Carlos Renato L. Francês

Universidade Federal do Pará (UFPA), DEEC/LANE, CEP: 66075-90, Belém-PA.

rodrigo@deec.ufpa.br, leonidas@ufpa.br, rfrances@ufpa.br

Resumo - As técnicas numéricas são ferramentas extremamente poderosas utilizadas na solução de diversos problemas, pois são capazes de manipular grandes quantidades de dados, com complicadas geometrias, que são tão comuns na prática de simulações empregadas nas engenharias. Com a computação paralela, os resultados podem ser obtidos de forma mais rápida, uma vez que permite executá-las em um pequeno intervalo de tempo. Este trabalho realiza um estudo comparativo de códigos paralelos nas linguagens Fortran, C e Java em um cluster Beowulf com biblioteca MPI utilizando a técnica numérica de FDTD para simular a propagação de ondas eletromagnéticas em 2-D emitidas por uma antena monopolo simples.

Palavras-Chave – Técnicas Numéricas, FDTD, Antena Monopolo Simples, Cluster.

Abstract - Numerical Techniques are powerful tools for solving various problems, because they can handle very different and complicated structures, which are common in engineering. With the aid of parallel computing, results can be obtained in shorter periods. This work analyzes the performance of FDTD software written in C, FORTRAN and Java and ran in a Beowulf cluster with MPI. The numerical technique FDTD was used to simulate the electromagnetic scattering in 2-D. The wave is transmitted by a simple monopole antenna.

Keywords – Numerical Technical, FDTD, Simple Monopole Antenna, Cluster.

1. Introdução

Muitos cientistas e engenheiros utilizam diversas técnicas para solucionar problemas contínuos ou de campo. Estas técnicas podem ser experimentais, analíticas ou numéricas [1].

As experimentais são caras, consomem tempo, às vezes são perigosas e geralmente não possuem muita flexibilidade na variação de parâmetros. As analíticas podem ser utilizadas por somente uma limitada classe de problemas, pois muitas situações reais resultam em sistemas não-lineares e envolvem complexas formas e processos. Já as numéricas são simplificações analíticas, pois funções de métodos numéricos podem reduzir uma expressão matemática mais elevada às operações aritméticas mais básicas.

Na década de 40, com os computadores digitais veio a explosão no uso e desenvolvimento de técnicas numéricas [2], as quais são um veículo eficiente utilizado nas mais variadas áreas do conhecimento em simulações como, condições climáticas, propagação de ondas eletromagnéticas, aterramentos em descargas elétricas [3].

Com a necessidade de se obter respostas cada vez mais rápidas e melhorar, com isso, o desempenho na execução dos métodos numéricos, que exigem, em alguns casos, uma grande quantidade de memória, muitos pesquisadores têm se preocupado em desenvolver arquiteturas paralelas cada vez mais eficientes. Contudo, muitas delas, com o aumento da velocidade dos processadores acabam por esbarrar, além do custo, no limite da capacidade tecnológica para a obtenção de circuitos rápidos [4]. Uma das soluções mais utilizadas nos últimos anos, tanto no meio acadêmico quanto no meio comercial são os COW (Cluster Of Workstations) ou máquinas agregadas, devido à sua grande capacidade de processamento e baixo custo. Entre as variações existentes de COW, o de alto desempenho, também conhecido como Beowulf, é empregado em grande escala com o objetivo de executar grandes tarefas no menor tempo possível e com um custo bastante reduzido de hardware, ou seja, uma solução computacional mais viável financeiramente e tão eficiente quanto os chamados supercomputadores [5].

No entanto, para que a computação executada em um cluster Beowulf seja bem sucedida é necessário considerar quatro distintas, mas inter-relacionadas

¹ Autor a quem toda correspondência deverá ser endereçada.

áreas que são [6]: estrutura de sistemas de hardware, onde todos os aspectos dos componentes físicos da topologia determinam a organização global do sistema; recursos administrativos e gerência de processos, são os softwares que viabilizam todas as fases de operação e manutenção das ferramentas do sistema, ou seja, instalação, configuração e inicialização, além da administração e gerência das tarefas, pelo status monitoramento, falhas, diagnósticos e manutenção; bibliotecas de programação e ferramentas paralelas determinam o paradigma de como o usuário irá coordenar os recursos de computação paralela para a execução simultânea e cooperada dos programas. Duas bibliotecas de comunicação se destacam no meio acadêmico por apresentarem suporte às linguagens de programação: Fortran, C e Java; MPI [7] e PVM [8]. Além das bibliotecas, pode-se conseguir a paralelização de programas através de compiladores paralelizantes, como o HPF [9], linguagens paralelas, etc. Finalmente, domínio e resultados de algoritmos paralelos provêm o modelo e o caminho para organizar uma aplicação para explorar um paralelismo intrínseco e obter um ganho efetivo de desempenho, além de validar os resultados gerados pelos códigos paralelos [4].

Outra tecnologia que vem ganhando espaço na área de computação paralela é a tecnologia em grids [10], que utiliza a idéia de computadores independentes e amplamente dispersos como plataforma de execução de aplicações paralelas, permitindo compartilhamento de seus serviços e recursos.

Este trabalho apresenta um estudo de desempenho, tendo em vista códigos paralelos, utilizando linguagens como Fortran, C e Java juntamente com a biblioteca de passagem de mensagem MPI em um cluster Beowulf. Para o estudo foram utilizados métodos numéricos, em particular, o denominado de FDTD (The Finite Differences in the Time Domain) [11]. Este método é muito utilizado para simular a propagação de ondas eletromagnéticas em ambientes indoor e outdoor, sendo por isso necessário uma grande quantidade de memória e um longo tempo de processamento para realizá-lo.

A avaliação de desempenho será demonstrada através do speedup e dos tempos de processamento de cada situação, verificando assim, as limitações e as vantagens na utilização paralela de cada linguagem.

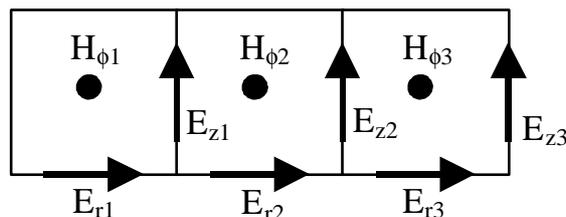
2. Método FDTD

O método das diferenças finitas no domínio do tempo (FDTD) foi introduzido por Kane Yee em 1966 [11] e representa uma forma simples e eficiente de solucionar as equações de Maxwell quando escritas na forma diferencial nos domínios temporal e espacial. Na proposta de Yee, as componentes do campo elétrico e magnético são intercaladas no espaço e no

tempo, de tal maneira que haja reciprocidade entre elas, como mostradas na Figura 1. Dessa forma, o método gera um número finito de conjuntos formados por uma quantidade finita de pontos a cada interação. Intuitivamente, isso é algo semelhante ao que ocorre na produção de vídeos digitais, onde cada quadro do filme contém um conjunto finito de pontos amostrados da imagem real e cada quadro corresponde a um instante discreto do processo [12].

Desde a primeira utilização do método FDTD na análise de antenas simples, antena monopolo cilíndrica e antena cônica [13], até os dias de hoje, houve uma grande evolução de técnicas numéricas e o surgimento de novas gerações de computadores que permitiram que o método FDTD pudesse ser utilizado na análise de estruturas complexas. Dentre as técnicas numéricas podemos citar: As condições de contorno absorvente, Perfectly Matched Layer ou PML, proposta inicialmente por Berenger [14], e o algoritmo FDTD em coordenadas não ortogonais [15].

Figura 1. Célula de Yee em 2-D em Coordenadas Cilíndricas



No entanto, apesar da facilidade na utilização e de todo o poder computacional proporcionado pelo método, têm-se dois grandes desafios a serem superados: a grande quantidade de memória requerida, que em alguns casos, a execução seqüencial torná-se impraticável e o longo tempo de processamento. Para superar esses desafios e evitar o estrangulamento da simulação, a solução utilizada foi a implementação paralela do código FDTD em um cluster do tipo Beowulf.

2.1 Antena Monopolo Simples

A antena monopolo é empregada em diversos seguimentos como celulares, carros, rádios e etc. A Figura 2 apresenta a geometria através de um plano-imagem a partir de uma linha de transmissão coaxial. Todos os condutores são considerados perfeitos. Os campos eletromagnéticos são independentes da coordenada cilíndrica ϕ , e as equações de Maxwell podem ser expressas como dois conjuntos independentes: Um que envolve somente as componentes E_r, E_z, H_ϕ , modo TM ou modo magnético transverso; e um que envolve somente as componentes E_ϕ, H_r, H_z , modo TE ou modo elétrico transverso. Como a excitação da antena é feita através de uma linha coaxial, somente

os modos TM rotacionalmente simétricos serão excitados. Dessa forma, esse modo é utilizado em nossa análise.

As equações de Maxwell para o modo TM na forma discreta e escritas em coordenadas cilíndricas são dadas por:

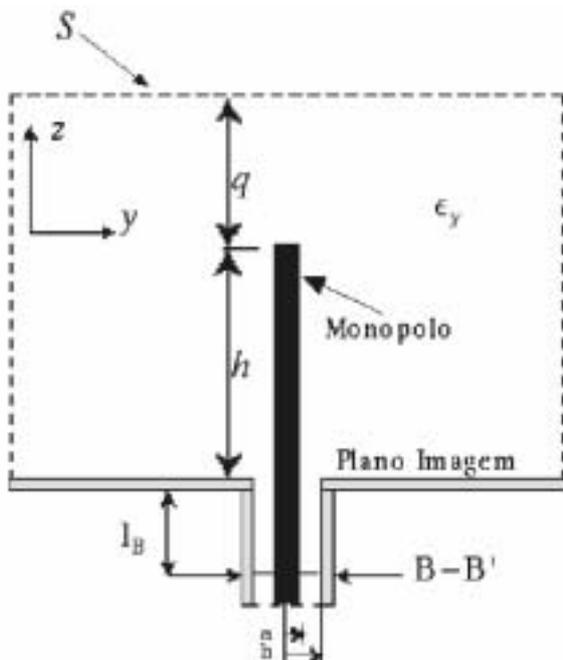
$$H_{\phi}^{n+0.5}(i,j) = H_{\phi}^{n-0.5}(i,j) + \frac{\Delta t}{\mu_0 \Delta r} [E_z^n(i+0.5,j) - E_z^n(i-0.5,j)] - \frac{\Delta t}{\mu_0 \Delta z} [E_r^n(i,j+0.5) - E_r^n(i,j-0.5)] \quad (1.a)$$

$$E_r^{n+1}(i,j-0.5) = E_r^n(i,j-0.5) - \frac{\Delta t}{\epsilon_0 \Delta z} [H_{\phi}^{n+0.5}(i,j) - H_{\phi}^{n+0.5}(i,j-1)] \quad (1.b)$$

$$E_z^{n+1}(i+0.5,j) = E_z^n(i+0.5,j) - \frac{\Delta t}{\epsilon_0 \Delta r} [r_{i+1} H_{\phi}^{n+0.5}(i+1,j) - r_i H_{\phi}^{n+0.5}(i,j)] \quad (1.c)$$

Os parâmetros que descrevem a antena monopolo são: a altura h e os raios a e b dos condutores da linha coaxial. A antena é caracterizada pelo tempo $\tau = h/c$, que é o tempo necessário para a luz percorrer o comprimento da antena. Na região de análise usou-se uma malha uniforme bidimensional de 1200×1200 , com

Figura 2. Geometria da antena monopolo cilíndrica alimentada através de um plano imagem, por uma linha de transmissão coaxial.



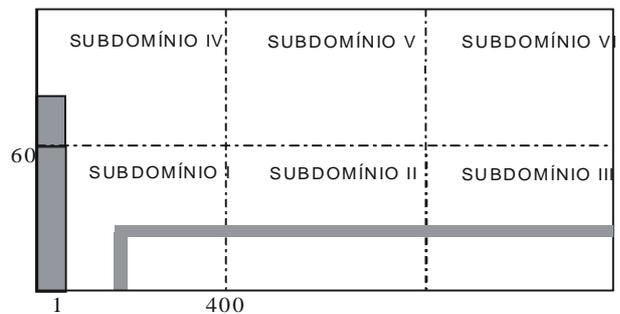
O monopolo cilíndrico foi excitado por um pulso gaussiano de 1V. O ponto de excitação foi a seção transversal B-B' localizada na linha de alimentação, Figura 2.

3. Implementação Paralela do Método FDTD

A antena monopolo foi analisada utilizando-se três diferentes linguagens de programação: Fortran, C e Java juntamente com a biblioteca de passagem de mensagem MPI em uma arquitetura paralela de um cluster Beowulf. A malha utilizada e a divisão realizada nos domínios foram a mesma para as três linguagens. Partiu-se de um processamento seqüencial que possuía uma dimensão de 1200×1200 elementos e, a partir daí, os domínios foram sendo divididos para serem processados por duas, três, quatro e seis máquinas. Por exemplo, para a divisão em seis subdomínios, resultaram matrizes de 400×600 , e a divisão pode ser visualizada na Figura 3.

O principal objetivo da utilização de diferentes linguagens é, justamente, verificar qual delas melhor se adapta a uma estrutura paralela de cluster, levando em consideração fatores como compiladores, comunicação via troca de mensagens e a própria estrutura de cada linguagem para o caso em estudo.

Figura 3. Divisão do domínio em seis subdomínios.



Para realizar a atualização dos campos elétricos e magnéticos, cada matriz foi armazenada e calculada por somente um processador, sendo que cada um irá calcular as componentes dos campos elétrico e magnético confinadas no seu domínio. A atualização das componentes é realizada somente nas interfaces de cada domínio, como mostrado na Figura 4.

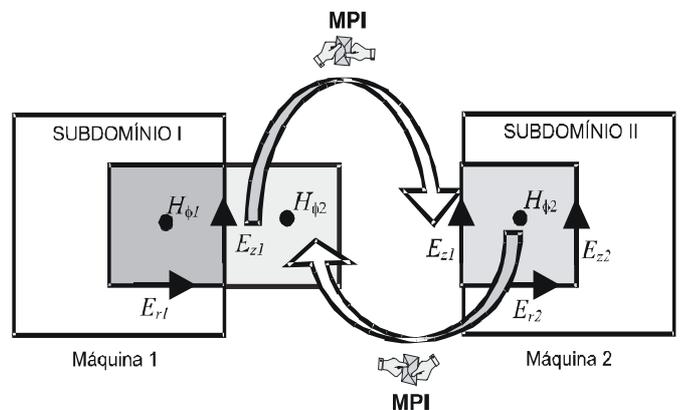


Figura 4. Atualização das componentes de campo nas interfaces dos subdomínios.

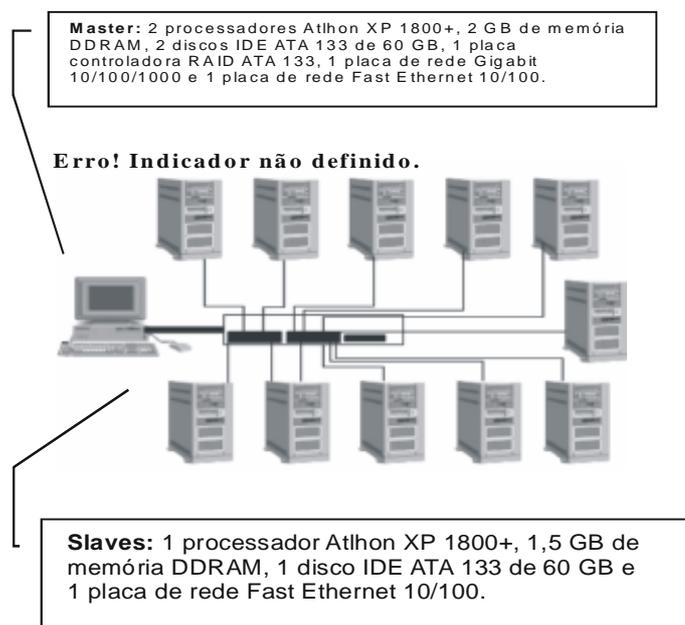
Como as interfaces são constituídas por várias células, muitos valores precisam ser trocados entre as máquinas durante cada iteração. Dessa forma, também foi empregado o empacotamento das informações com o recurso MPI_PACK da MPI. Isso significa que todos os valores são agrupados sequencialmente em uma única string que é enviada ao processador adjacente, reduzindo o número de aberturas da porta de comunicação, e, portanto, aumentando o desempenho global.

4. Cluster Amazônia

O Laboratório de Análise Numérica em Eletromagnetismo – LANE, juntamente com o Departamento de Engenharia Elétrica e de Computação da Universidade Federal do Pará desenvolveram, em 2001, o primeiro cluster do tipo Beowulf da instituição, para realizar estudos sobre processamento paralelo aplicado a simulação de antenas e a propagação de ondas eletromagnéticas utilizando o método FDTD. O primeiro cluster desenvolvido foi o LANE1 que era formado por três máquinas, sendo uma servidora (master) e duas escravas (slaves).

Hoje o laboratório possui o cluster AMAZÔNIA, que foi adquirido através de uma parceria entre o DEEC/UFPA e as empresas Ericson e Amazônia Celular. Este é constituído por 11 máquinas, sendo o servidor constituído por 2 processadores, conforme Figura 5. E assim como o cluster LANE1, o AMAZÔNIA também utiliza softwares de livre distribuição como o sistema operacional Linux, distribution Red Hat 9.0, kernel 2.4.20-8 e a biblioteca de passagem de mensagem, MPI 2.0, versão lam-mpi 7.0.6. Os compiladores utilizados foram o gcc versão 3.4.0, GNU Fortran 3.4.0 e para o mpiJava foi utilizada a versão 1.2.5, js2sdk 1.4.2-rc1.

Figura 5. Cluster AMAZÔNIA – UFPA/LANE

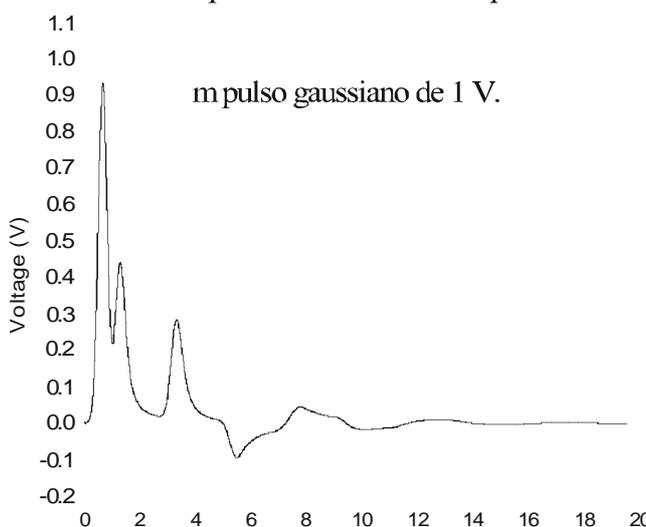


5. Resultados Obtidos

Com a implementação do método numérico FDTD em uma plataforma paralela, alguns resultados puderam ser observados, principalmente, no que diz respeito aos tempos de processamento das linguagens utilizadas.

Os resultados encontrados com a utilização das diferentes linguagens estão em plena concordância com os encontrados por Maloney [13], como se pode observar na Figura 6, pela voltagem total (voltagem incidente mais voltagem refletida).

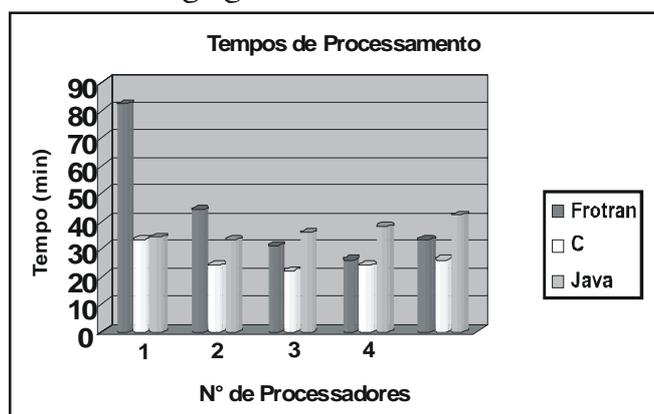
Figura 6. Voltagem na linha coaxial para uma antena monopolo cilíndrica excitada por u



Os códigos foram executados em caráter dedicado, sendo obtidas as médias das 15 execuções realizadas para cada caso, apresentando um desvio padrão próximo da média.

A Figura 7 apresenta os tempos de processamento em cada uma das linguagens e das configurações utilizadas em cada domínio.

Figura 7. Tempos de Processamento nas linguagens e domínios utilizados.



O estudo comparativo das linguagens pode, também ser observado com o speedup, como mostra a Figura 8.

Observa-se, nos gráficos 7 e 8, que a linguagem C obteve o melhor resultado para este caso, sendo que no caso serial o tempo obtido por esta linguagem chegou a quase 1/3 do tempo serial do código executado em Fortran.

Figura 8. Speedup das linguagens e domínios utilizados.

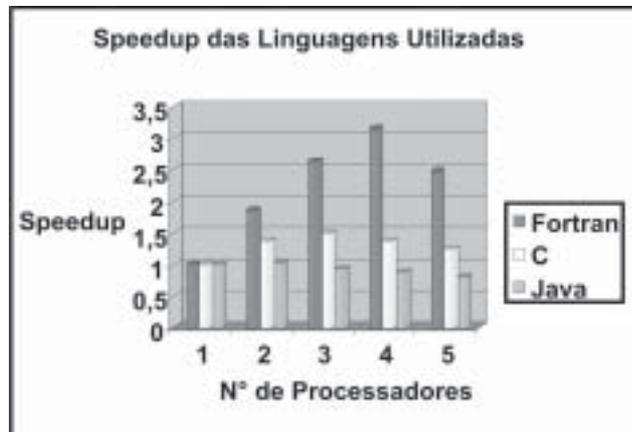


Figura 9. Tempos da Linguagem Fortran Utilizando e não-Utilizando a Função MPI_PACK.



Os resultados obtidos com a utilização das três linguagens além de serem os mesmos apresentados na Figura 6, também se apresentaram compatíveis com o da Figura 10, onde se pode observar a propagação da onda para 1200 iterações, em 8 máquinas.

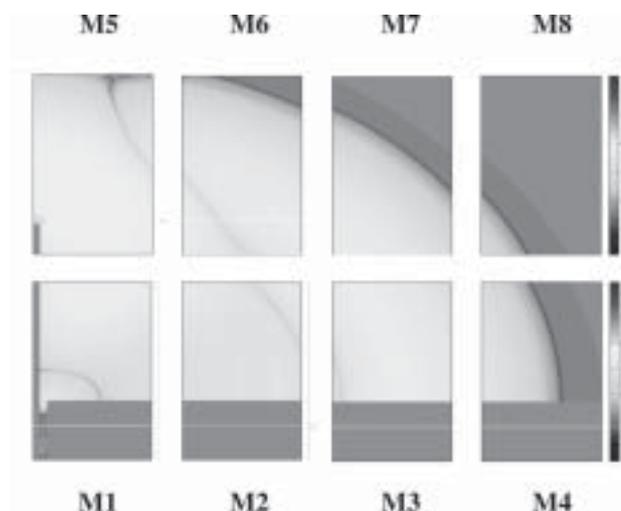
6. Conclusão

O principal objetivo deste trabalho foi realizar um estudo comparativo na utilização de códigos paralelos utilizando diferentes linguagens de programação – Fortran, C e Java – que podem ser suportadas pela biblioteca de passagem de mensagem MPI em um cluster Beowulf. O código utilizado foi de uma antena monopolo simples aplicando o método das diferenças finitas no domínio do tempo (FDTD). A intenção foi avaliar qual das linguagens utilizadas tem um melhor comportamento e com isso, uma melhor resposta para ser utilizada com processamento paralelo utilizando o método numérico em estudo.

Observou-se que a linguagem C teve um melhor comportamento e com isso, um tempo de resposta mais curto quando comparada às outras linguagens, para o caso em estudo. Isso devido a própria estrutura da linguagem, bem como à implementação de um compilador em C (gcc) mais eficiente que os utilizados em Fortran (g77) e Java (javac). Certamente o fato de o Fortran ter apresentado um desempenho inferior em comparação ao C é o fato de o compilador g77 ser um frontend do compilador gcc, ou seja, o código Fortran é primeiramente convertido em C e depois propriamente compilado [17].

A troca de informações por pacote mostrou-se bastante eficiente, já que reduz o número de chamadas à porta de comunicação. Certamente a utilização de compressão dos dados do pacote daria um ganho ainda superior implementado na linguagem C.

Figura 10. Propagação da antena monopolo cilíndrica para 1200 iterações em 8 máquinas.



7. Referências

- [1] Chapra, Steven C. & Canale, Raymond P., “Numerical Methods for Engineers”, Second Edition, McGraw-Hill International Editions, New York, 1990.
- [2] Sadiku, Matthew N. O., “Numerical Techniques in Electromagnetics”, Second Edition, CRC Press, Florida, 2001.
- [3] Zaneta, Luiz C.J., “Transitórios Eletromagnéticos em Sistemas de Potência”, Editora EdUsp, São Paulo, 2003.
- [4] De Rose, César A. F. & Navaux, Philippe O. A., “Arquiteturas Paralelas”, Instituto de Informática da UFRGS, Editora Sagra Luzzatto, Porto Alegre, 2003.
- [5] Vrenios, Alex, “Linux Cluster Architecture”, SAMS, Indianapolis, 2002s.
- [6] Sterling, Thomas, “Beowulf Cluster Computing with Linux”, Massachusetts Institute of Technology, Massachusetts, 2002.
- [7] <http://www.mpi.nd.edu/lam>

- [8] <http://netlib.org/pvm3>
- [9] Koelbel, Charles H. et al, "The High Performance Fortran Handbook", 1994.
- [10] Foster, I. & Kesselman C., "The Grid: Blueprint for Future Computing Infrastructure, 1999.
- [11] K.S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," IEEE Trans. Antennas Propagat., vol. AP-14, pp. 302-307, May 1966.
- [12] Oliveira, Rodrigo M.S., "Método FDTD Aplicado na Análise da Propagação Eletromagnética em Ambientes Indoor e Outdoor", Trabalho de Conclusão de Curso, UFPA/DEEC, Belém-Pará, 12/02/2003.
- [13] Maloney, J. G. S. Smith, and W. R. Scott, Jr., "Accurate computation of the radiation from simple antennas using the finite-difference time-domain method," IEEE Trans. Antennas and Propagation, Vol.38, 1990, pp.1059-1068
- [14] Berenger, J.P., "A perfectly matched layer for the absorption of electromagnetic waves," J. Computational Physics, vol. 114, 1994, pp. 185-200.
- [15] Lee J.F., and Mittra R., "Finite difference time domain algorithm for non-orthogonal grids," Technical Rep. Electromagnetic Communication Lab., Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 1991.
- [16] Araújo, J.S., Sobrinho, C.L.S.S., Santos, R.O., Rocha, J.M., Guedes, L.A., Kawasaki, R.Y., "Análise de Antenas em 2-D Utilizando o Método das Diferenças Finitas no Domínio do Tempo com Processamento Paralelo e MPI", INATEL, Volume 06, Número 01, Junho de 2003.
- [17] www.gnu.org/software/fortran

Responsabilidade de autoria

As informações contidas neste artigo são de inteira responsabilidade de seu autor. As opiniões nele emitidas não representam, necessariamente, pontos de vista da Instituição e/ou do Conselho Editorial.