

# Viabilidade de Simulações Multi-corpos em Astrofísica usando um Ambiente de Grade Computacional

André Luiz Garkauskas, Airam Jonatas Preto, Stephan Stephany  
INPE - Instituto Nacional de Pesquisas Espaciais  
São José dos Campos, Brasil

E-mail: agarkauskas@hotmail.com, {airam, stephan}@lac.inpe.br

## Resumo

*Este artigo apresenta uma análise da viabilidade da utilização de computação em grade na execução de simulações astrofísicas multi-corpos. A computação em grade consiste de uma infra-estrutura de hardware e software que provê acesso confiável e consistente a recursos computacionais que estão distribuídos fisicamente. Ferramentas de software que fazem parte dessa infra-estrutura permitem a sua utilização, por exemplo, segundo um modelo de programação paralela baseado em comunicação por mensagens. A computação em grade requer o emprego de um escalonador que seja capaz de mapear tarefas a processadores dados os recursos computacionais disponíveis na grade.*

**Palavras-chave:** *computação em grade, simulação multi-corpos, astrofísica.*

## 1. Introdução

Instituições de pesquisa que possuem recursos computacionais para execução de algoritmos de simulação multi-corpos na área de astrofísica podem atingir rapidamente seus limites devido à escala e a complexidade dos experimentos.

Visando maximizar a utilização dos recursos computacionais, a computação em grade pode tornar-se uma alternativa obrigatória em um futuro próximo em função da crescente demanda de recursos e custos de infra-estrutura envolvidos nos problemas pesquisados.

A Computação em Grade difere da computação distribuída convencional devido ao fato de ser voltada para compartilhamento de recursos em larga escala (tais como supercomputadores, “clusters” de computadores, sistema de armazenamento de dados, ou equipamentos específicos) para aplicações orientadas, por exemplo, ao alto desempenho e/ou visualização científica, a exemplo as simulações astrofísicas multi-corpos [21].

O termo computação em grade foi escolhido em analogia com as redes de energia elétrica (*electric power grid*). A idéia é disponibilizar recursos computacionais

da mesma forma que hoje é disponibilizada a energia elétrica.

Uma descrição dos modelos numéricos de simulação multi-corpos em astrofísica é apresentada na seção 2. A seção 3 descreve a tecnologia de grades computacionais, que é baseada em protocolos específicos e está disponível no kit de ferramentas Globus, no caso da implementação proposta. Uma análise dos benefícios de um possível cenário de simulação em grade será apresentada na seção 4.

## 2. Simulações multi-corpos na astrofísica

A astrofísica é um campo muito adequado para aplicação de simulações computacionais porque os sistemas sob estudo (i.e. estrelas, galáxias, aglomerados) não são, obviamente, objetos que podem ser estudados em experimentos em laboratório. Com escalas típicas de tempo e espaço da ordem de  $10^8$  anos e  $10^{23}$  metros, respectivamente, astrofísicos e astrônomos podem somente agir como observadores dos sistemas em estudo. A única forma que os experimentos podem ser conduzidos em ambiente controlado é por meio de simulações computacionais, as quais demandam alto poder de processamento dadas escalas espacial e temporal envolvidas.

Dentre todas as simulações computacionais em astrofísica, as simulações multi-corpos são as mais utilizadas no estudo de astrofísica em geral, abrangendo diversos tópicos, tais como formação planetária, dinâmica de aglomerados de estrelas, interação entre galáxias e modelagem cosmológica [1].

### 2.1. O problema gravitacional multi-corpos clássico

Numa simulação computacional multi-corpos em astrofísica, usa-se um modelo de partículas. Assim, uma estrela é representada por uma partícula pontual desprovida de outras propriedades a não ser a massa, devido à qual sofre atração gravitacional das demais partículas. Outros aspectos, como queima do combustível nuclear, irradiação e convecção, que podem ser importantes no estudo da evolução estelar, não são

considerados. O problema de determinar o comportamento dinâmico de um sistema de partículas sujeito apenas à atração gravitacional é chamado de problema gravitacional multi-corpos clássico (*N-Body problem*). A evolução de aglomerados de estrelas ou galáxias (cada um considerado uma massa pontual) e o desenvolvimento de estruturas em espiral são problemas que podem ser aproximados desta forma [2].

## 2.2 – Sistemas colisionais e não-colisionais

As técnicas de solução de problemas multi-corpos dividem-se em duas classes, de acordo com a importância da taxa de colisão de estrelas binárias na evolução do sistema. A taxa de colisões binárias mede a extensão que a órbita de uma estrela em um campo gravitacional é alterada.

Se o sistema é composto de poucas estrelas, a órbita de qualquer estrela em particular depende primariamente da posição precisa de cada vizinho e o sistema é chamado de colisional, dominado pelas colisões entre estrelas binárias. A evolução de um sistema colisional é altamente dependente da massa individual de cada estrela.

Se o sistema é composto por um grande número de estrelas, como galáxias, onde este número situa-se acima de  $10^{11}$ , este sistema é descrito como sistema não-colisional, já que sua evolução somente será influenciada por encontros próximos após 20 bilhões de anos, escala na qual atualmente estima-se a idade do universo [2].

Neste último caso, os parâmetros de interesse nas simulações multi-corpos são as médias dos valores de densidade de massa, potencial e campo gravitacionais computados com base em centenas de milhares de separações estelares. Formações de estrelas binárias certamente constituem uma proporção considerável de estrelas na nossa galáxia, mas o detalhe de seu movimento orbital não é representado em uma simulação com grande número de corpos celestes, que constitui um sistema não colisional. Portanto, o problema associado com a integração de estrelas binárias deve ser estudado em uma simulação de pequenos sistemas colisionais.

A evolução de um sistema não-colisional é determinada pela densidade de massa média e não pela massa individual de estrelas individuais. Simulações deste tipo de sistema são usualmente executadas, por conveniência, com um grande número de partículas de mesma massa, mesmo que uma galáxia possua uma distribuição característica das massas. Num sistema não-colisional a distribuição de massa não influi na evolução das quantidades médias em estudo [2].

## 2.3 – Algoritmos de simulação multi-corpos na astrofísica

Atualmente têm-se três classes de algoritmos para simulações multi-corpos em astrofísica:

- Exatos,
- Com relaxação partícula-partícula,
- Mistos

Esta classificação baseia-se nas técnicas de integração utilizadas para evolução do sistema.

### 2.3.1 – Algoritmos exatos

Algoritmos exatos (ou diretos) são os mais simples e os mais caros em termos de carga computacional. Entretanto, atualmente consistem na melhor ferramenta para estudo da evolução de sistemas colisionais. As equações de movimento são dadas por:

$$\ddot{\vec{r}}_i = \sum_j^n \frac{m_j}{r_{ij}^3} \left( \vec{r}_i - \vec{r}_j \right); \quad r_{ij} = \left| \vec{r}_i - \vec{r}_j \right|$$

Acima,  $n$  é o número de partículas,  $m$  a massa e a constante gravitacional ( $G$ ) não aparece, pois seu valor normalizado é igual à unidade. Estas equações são integradas diretamente. Expandindo a solução em série de Taylor em termos dos valores já calculados das derivadas da posição têm-se para uma dimensão:

$$x_{n+1} = x_n + \dot{x}_n dt + \frac{\ddot{x}_n}{2} dt^2 + \sum_{k=1}^N \frac{F_n^{(k)}}{m_{n+1}} \cdot \frac{dt^k}{(k+2)!}$$

$F_n^{(k)}$  significa derivada de ordem  $k$  da força e o índice  $n$  refere-se ao último cálculo da posição.

Este método não requer nenhuma hipótese *a priori* para funcionar e pode, a princípio, ser tão preciso quanto se queira tomando um intervalo de integração ( $dt$ ) pequeno suficiente ou expandindo a série em  $k$  até uma ordem adequada. Existem códigos que utilizam expansão até a décima ordem, contudo, normalmente não há vantagens em expandir a série em uma ordem superior à quarta ordem [1].

De posse das equações de cálculo de força e movimento, aplica-se o algoritmo abaixo:

- 1) Acumula-se as forças resultantes do cálculo de todas as  $F_{ij}$  entre as  $N$  partículas –  $O(N^2)$
- 2) Integram-se as partículas de acordo com as equações de movimento com as forças resultantes –  $O(N)$
- 3) Atualiza-se o contador de tempo ( $dt$ )
- 4) Volta para 1

Fica claro que para computarmos todas as  $F_{ij}$  para todas as  $N$  partículas, necessitamos de  $O(N^2/2 - N)$  passos, lembrando que  $F_{ij} = -F_{ji}$ .

Alguns representantes importantes desta classe de algoritmos são NBODY2[1] e Kyrá. Uma abordagem

muito interessante para resolução de algoritmos exatos pode ser encontrada em [18].

### 2.3.2 – Algoritmos com relaxação partícula-partícula

Ao invés de calcular a força gravitacional de todos os corpos do sistema com todos os outros  $N-1$  corpos restantes, este método calcula a força sobre um corpo em função do potencial gravitacional gerado pelas partículas do sistema. No caso de sistemas gravitacionais, isto é feito resolvendo-se a equação de Poisson. Para a solução numérica, divide-se o espaço em células que contém  $n$  corpos e é calculado o potencial gerado por estas células. Tendo o potencial gerado por cada uma, podemos calcular a força sobre uma dada partícula e extrapolar sua posição e velocidade para um instante seguinte (como no método exato).

Os principais representantes desta categoria são os algoritmos *PM – Particle Mesh*[12], *Fast Multipole*[13] e *Self-Consistent Field*.

Este método é muito mais rápido que o anterior devido à forma como é calculada a força, atingindo complexidade  $O(N)$  na maioria dos casos, mas, por outro lado, tem uma resolução espacial limitada e não é adequado para o estudo de sistemas colisionais; só é possível resolver características do sistema que sejam maiores que as células utilizadas no cálculo do potencial. Não se pode, por exemplo, seguir o movimento de apenas uma partícula individualmente. O que pode ser feito é seguir a evolução da densidade de uma célula no tempo [4].

### 2.3.3. Algoritmos mistos

Algoritmos mistos ou hierárquicos usam as forças de atração entre as partículas em princípio. Entretanto, as partículas são agrupadas em uma estrutura de árvore juntamente com sua massa total e seu centro de massa. Desta forma, determina-se a força resultante entre as partículas e os centros de massa agrupados, utilizando-se de alguma relaxação, de acordo com a distância e a massa total envolvida.

Este algoritmo é o mais utilizado atualmente em simulações de um grande número de partículas ( $10^9$ ), por sua alta eficiência  $O(N * \log N)$ , boa resolução numérica e por sua alta capacidade de paralelização.

Descrição algorítmica:

- 1) Ordenação das partículas –  $O(N * \log N)$
- 2) Construção do agrupamento em árvore das partículas, calculando seu raio crítico e o centro de massa –  $O(N)$
- 3) Para cada partícula, percorre-se a árvore procurando a altura que é mais precisa para cálculo da componente de força resultante através de uma aproximação sem relaxação –  $O(N)$ .

- 4) Para cada partícula, calcula a força resultante –  $O(N)$
- 5) Para cada partícula atualiza posição e velocidade

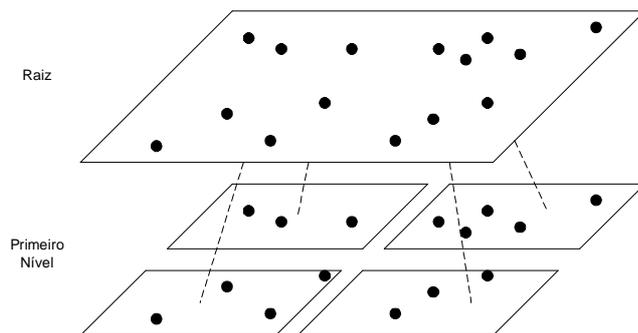


Figura 1 – Árvore de um algoritmo hierárquico

Contudo, para obter-se alta precisão nos cálculos das forças comparável com os algoritmos exatos exigida pelos sistemas colisionais deve-se forçar o passo 3 do algoritmo a procurar a altura mais próxima possível das folhas da árvore e isto acaba levando a uma complexidade  $O(N^2)$ , reduzindo drasticamente o desempenho do algoritmo.

O representante mais conhecido dos algoritmos mistos é o algoritmo Barnes-Hut[14].

## 3. Computação em grade

As aplicações de alto desempenho que estão emergindo atualmente necessitam de características únicas que não estão presentes em uma única estação de trabalho. Ainda, a complexidade das aplicações cresce em conjunto com a sua necessidade por performance. No ritmo atual, pesquisadores interessados em estudar fenômenos científicos em ambiente de simulação necessitam cada vez mais explorar o hardware e o software disponíveis, distanciando-se do seu real objetivo.

Os recursos de hardware disponíveis como rede, CPU e armazenamento, por exemplo, precisam ser negociados com administradores de sistemas para que não haja comprometimento da qualidade dos experimentos estudados.

Endereçando estas questões, visando o compartilhamento de recursos de hardware e software distribuídos geograficamente com escalonamento automático de processos e políticas de autenticação de uma maneira transparente para os usuários, nasce o termo computação em grade[9].

### 3.1. Metacomputação

O termo metacomputação antecede o termo computação em grade e emerge como evolução intermediária à computação em grade.

O termo é usado para denotar a um supercomputador virtual em rede, construído dinamicamente através de

recursos computacionais distribuídos geograficamente conectados por uma rede de alto desempenho.

Por exemplo, uma máquina de memória distribuída pode ser responsável por processar os dados provenientes de uma estação conectada a algum tipo de sensor, que por sua vez entrega os dados processados a uma outra estação com recursos de visualização em tempo real. As três entidades envolvidas possuem características únicas conectadas através de uma rede de alto desempenho, ou bases de dados extremamente rápidas, para permitir a análise em tempo real dos dados coletados. Ainda podemos ter recursos de armazenamento com características únicas em cada etapa do processo. A figura 6 ilustra este exemplo de metacomputação.

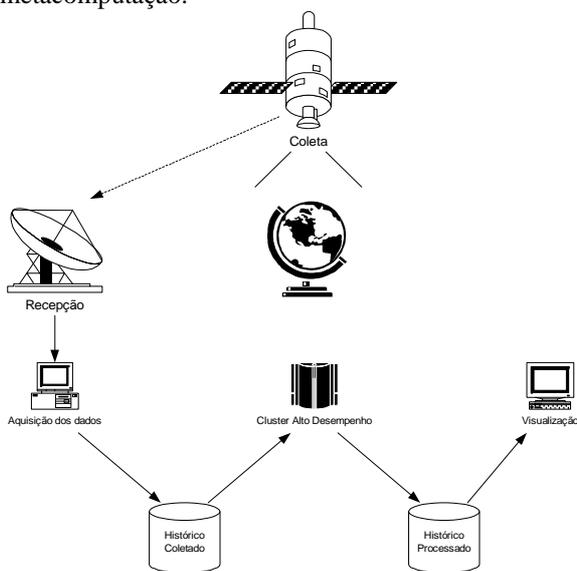


Figura 6 – Exemplo de Supercomputador Virtual em Rede - Metacomputador

Metacomputadores possuem muito em comum com sistemas distribuídos e sistemas paralelos, mas também diferem destas duas arquiteturas em importantes conceitos. Como um sistema distribuído, um supercomputador virtual em rede deve integrar recursos de várias naturezas de forma abrangente, possivelmente conectado em diferentes domínios administrativos. Entretanto, a necessidade de alta performance pode requerer modelos de programação e interfaces radicalmente diferentes daquelas usadas em sistemas distribuídos. Como na computação paralela, aplicações de metacomputadores geralmente precisam sincronizar sua comunicação com muito cuidado para poder serem compatíveis com os requisitos de performance [16].

Estas considerações sugerem que enquanto a metacomputação pode ser construída sobre as tecnologias de software distribuídas e paralelas, também implica em avançados mecanismos, técnicas e ferramentas.

### 3.2.1. Aplicações em metacomputação

Cientistas e engenheiros estão apenas começando a explorar as novas possibilidades permitidas por metacomputadores. Alguns experimentos [16] identificaram quatro classes distintas de aplicações:

- **Supercomputação pessoal (Desktop supercomputing):** Estas aplicações acoplam as estações gráficas de alta capacidade com supercomputadores ou bases de dados remotas. Este acoplamento conecta os usuários de uma forma mais eficiente com as capacidades computacionais eventualmente disponíveis à distância.
- **Instrumentos inteligentes (Smart Instruments):** Estas aplicações conectam os usuários a instrumentos como microscópios, telescópios ou satélites que estão acoplados com supercomputadores remotos. Este incremento computacional pode habilitar interação quase em tempo real com a saída de dados destes instrumentos.
- **Ambientes colaborativos (Collaborative Environments):** Um terceiro grupo de aplicações acopla múltiplos ambientes virtuais para que os usuários possam interagir de diferentes localidades com simulações distribuídas em supercomputadores.
- **Supercomputação distribuída (Distributed Supercomputing):** Esta classe de aplicações acopla múltiplos computadores para resolução de problemas que são muito grandes para um computador ou que pode beneficiar-se de executar diferentes partes do problema em diferentes arquiteturas. Podemos distinguir modos agendados e não-agendados de operação. No modo agendado recursos uma vez adquiridos são dedicados à aplicação. No modo não-agendado aplicações ou usam recursos disponíveis que podem ser compartilhados com outras aplicações. O Condor [17] é um sistema que suporta este modo de operação. No geral, o modo agendado é utilizado para simulações que são altamente restritivas ao tempo de execução enquanto o modo não-agendado é apropriado para simulações que podem adaptar-se a disponibilidade de recursos computacionais.

### 3.2.2. Características de sistemas de metacomputação

Enquanto alguns experimentos proveram os caminhos de como os futuros sistemas de metacomputação poderão se assemelhar e como deverão ser usados, a definição e aplicação destes sistemas ainda são tema de pesquisa [16]. O que pode ser feito a partir dos estudos atuais é observar as características destes sistemas:

- **Escala e necessidade de seleção:** Até agora a maioria dos experimentos de metacomputação foram feitos com poucos componentes e aplicações. Num futuro próximo podemos esperar muito mais estruturas interligadas, das

quais os recursos serão selecionados para aplicações em particular de acordo com os critérios de conectividade, custo, segurança e confiabilidade.

- Heterogeneidade em múltiplos níveis: Todos os recursos computacionais utilizados para construção de supercomputadores virtuais e as redes que conectam estes recursos são altamente heterogêneos. Heterogeneidade pode acontecer em diversos níveis, desde dispositivos físicos a políticas de escalonamento e armazenamento.
- Estrutura imprevisível: Tradicionalmente, aplicações de alta performance são desenvolvidas para uma classe específica de sistemas com suas características muito bem conhecidas, até mesmo para um computador em particular [18].
- Comportamento dinâmico e imprevisível: Sistemas de alta performance tradicionais usam políticas de administração de recursos de armazenamento e processamento para prover acesso exclusivo e previsível para suas aplicações. Em ambientes de metacomputação, recursos, especialmente redes, são mais propícios ao compartilhamento. Uma consequência deste compartilhamento é que o comportamento e a performance podem variar no tempo. Sistemas de metacomputação em larga escala podem ainda sofrer com falhas de rede e outros recursos. Em geral, não é possível garantir até mesmo os mínimos requisitos de qualidade de serviço[16].

### 3.3. Da metacomputação para a computação em grade

A Computação em Grade difere da metacomputação devido ao fato de possibilitar compartilhamento coordenado de recursos em larga escala (como supercomputadores, “clusters”, sistema de armazenamento, dados, instrumentos e pessoas) e permitir a resolução de problemas computacionais em organizações virtuais multi-institucionais, em alguns casos orientados ao alto desempenho.

O compartilhamento deve necessariamente ser altamente controlado, com os fornecedores e consumidores de recursos definindo claramente e cuidadosamente apenas o que é compartilhado, quem é permitido compartilhar e as condições nas quais o compartilhamento ocorre. Para atingir este objetivo a palavra chave é interoperabilidade que em um ambiente onde todos estão conectados significa protocolos comuns. Desta forma, uma arquitetura de grade nada mais é que uma arquitetura de protocolos nos quais usuários e fornecedores de recursos negociam, estabelecem, administram e exploram as relações de compartilhamento[21].

A arquitetura de Grade, proposta por Foster, segue os princípios do modelo ampulheta. Neste modelo, reproduzido na figura 7, diferentes comportamentos de alto nível (topo da ampulheta) são mapeados em um

pequeno conjunto de abstrações centrais (“core abstractions”) e protocolos (parte mais estreita da ampulheta), que por sua vez são mapeados em diferentes tecnologias fundamentais (base da ampulheta) [21].

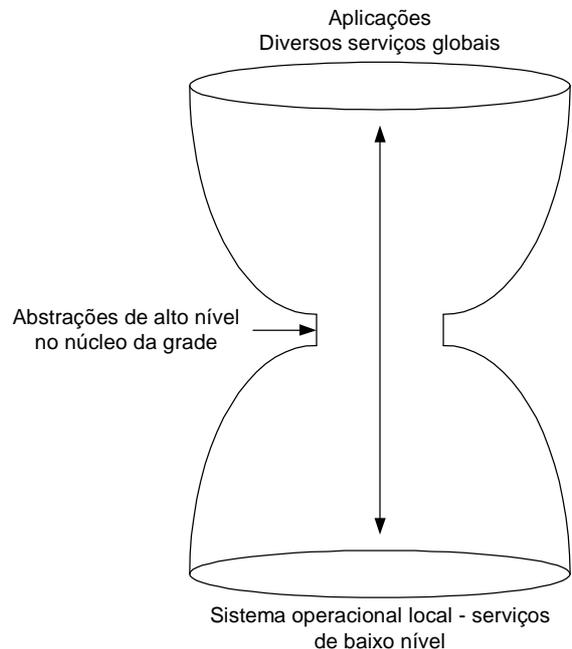


Figura 7 – Modelo ampulheta da grade

Atualmente não existe um padrão para tecnologia de Computação em Grade, mas o kit de ferramentas Globus (Globus Toolkit) tem se tornado bastante aceito [16].

### 4. A utilização da grade para simulações multi-corpos na astrofísica

A importância de grades computacionais na resolução de diversos problemas computacionais torna-se evidente a cada dia. As tecnologias de grades atuais, apresentadas nas seções anteriores, permitem a agregação de recursos computacionais de alto desempenho.

Devido à necessidade crescente de recursos computacionais, a área de simulação em multi-corpos é uma candidata potencial para usar esta tecnologia. De acordo com o apresentado anteriormente na seção 3, existe utilidade desta tecnologia nas áreas de computação sob demanda, colaborativa, intensiva de dados e supercomputação distribuída para o ambiente de simulação multi-corpos na área de astrofísica.

Os conceitos de supercomputação distribuída aliados à computação colaborativa apresentados na seção 3 podem escalar os recursos disponíveis de simulação alimentando a crescente demanda por processamento dos algoritmos apresentados na seção 2. As instituições desprovidas de recursos de simulação podem usufruir de uma infraestrutura de simulação de grade geograficamente distribuída.

Do outro lado, os resultados obtidos podem ser compartilhados e visualizados através dos conceitos de instrumentos inteligentes.

As simulações podem ser executadas e armazenadas de forma que futuros cenários de simulação podem ser pesquisados antes de serem executados, evitando-se que a mesma simulação seja executada duas ou mais vezes para o mesmo conjunto de parâmetros de entrada.

Portais colaborativos podem facilitar a comunicação e distribuição dos dados e o acesso a diferentes versões dos algoritmos apresentados.

Uma proposta de trabalho futuro seria adaptar os algoritmos apresentados ao ambiente de grade computacional, acessíveis através de um portal colaborativo onde os resultados obtidos seriam compartilhados. Este portal dispararia a execução de uma simulação em algum cluster da grade, alocando os recursos conforme a demanda e as diretivas de prioridades ou cotas de acesso.

A usabilidade e produção de resultados a partir deste ambiente colaborativo poderia então ser comparada aos métodos tradicionais de simulação em cluster no que tange ao desempenho e eficiência de execução, demonstrando assim os benefícios diretos e indiretos da computação em grade.

## 6. Referências

- [1] Aarseth, S. J. *NBODY2: A direct multi-corpos integration code*, New Astronomy 2, p. 277, 2001, 7.
- [2] Hockney, R.W.; Eastwood, R. W. *Computer Simulation Using Particles*, McGraw Hill, New York, 1988.
- [3] Modi, J. J. *Parallel Algorithms and Matrix Computation*, Oxford University Press, New York, 1988.
- [4] Spurzem, R. *Astrophysical N-Body Simulations: Algorithms and Challenges*, Proceedings for International Symposium on Supercomputing, 1997, 9.
- [5] Foster, I. *The Grid: A New Infrastructure for 21st Century Science*, Physics Today, 2002, 2.
- [6] Snir, M.; Otto, S.; Huss-Lederman, S.; Walker, D.; Dongarra, J. *MPI: The Complete Reference*, Cambridge: The MIT Press, 1996.
- [7] Karonis, N.; Toonen, B.; Foster, I. *MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface*, Journal of Parallel and Distributed Computing, 2003.
- [8] Russel, M.; Allen, G.; Foster, I.; Seidel, E.; Novotny, J.; Shalf, J.; Laszewski, G.; Daus, G. *The Astrophysics Simulation Collaboratroy: A Science Portal Enabling Community Software Development*, Proceedings of High-Performance Distributed Computing 10, San Francisco, CA, 2001, 7.
- [9] Foster, I. *The Grid: Computing Without Bounds*, Scientific American, 2003, 4.
- [10] Correa; Varela, R.; Preto, A.; Stephany, S.; Wuensche de Souza, C. A. *Análise quantitativa dos custos de comunicação para programas usando MPI executados em máquinas paralelas de memória distribuída*. II Worcap, São José dos Campos, SP, 2002.
- [11] Stephany, S.; Preto, A.; Granato; E.; Ribeiro, J. A. *A parallel implementation for a driven molecular dynamics algorithm*, Proceedings of the XXIV Iberian Latin-American Congress on Computational Methods in Engineering. Ouro Preto, Brasil, 2003.
- [12] Merz, H.; Pen, U.; Trac, H. *Towards Optimal Parallel PM Multi-corpos Codes: PMFAST*, New Astronomy 10, p 393-407, 2005.
- [13] Cheng, H.; Greengard, L.; Rokhlin, V. *A fast adaptative multipole algorithm in three dimension.*, Journal of Computational Physics 155, p. 468-498 1999.
- [14] Barnes, J.; Hut, P. *A Hierarchical  $O(N\log N)$  Force-Calculation Algorithm*, Nature, Vol. 324, p. 446-449, 1986, 12.
- [15] Almeida, E. *Avaliar a viabilidade do uso de Grade Computacional para a execução do BRAMS como modelo meteorológico de área limitada*. Proposta de trabalho do curso de doutorado em Computação Aplicada, INPE, São José dos Campos, 2003.
- [16] Foster, I.; Kesselman, C. *Globus: A Metacomputing Infrastructure Toolkit*. International Journal of Supercomputer Applications, 11(2), p. 115-128, 1997.
- [17] Frey, J.; Tannenbaum, T.; Livny, M.; Foster, I.; Tuecke, S. *Condor-G: A Computation Management Agent for Multi-Institutional Grids*. Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, 2001, 7.