

# Computação de Alto Desempenho: Adquirindo Competência e Disseminando a sua Utilização

Nelson L. Duarte<sup>1</sup>, Leonardo P. Silva<sup>1</sup>, Thiago P. Damas<sup>1</sup>, Rodrigo A. de Bem<sup>1</sup>, Luiz S. Laurino<sup>1</sup>, Eduardo C. Lopes<sup>1</sup>, Jorge D. Kautzmann<sup>1</sup>, Carlos F. Miranda<sup>1</sup>, Mauricio M. Mata<sup>2</sup>, Silvia C. Botelho<sup>2</sup>, Humberto C. Piccoli<sup>2</sup>, Glauber A. Gonçalves<sup>2</sup>

Fundação Universidade Federal do Rio Grande - FURG  
Grupo de Estudos em Processamento Paralelo e Distribuído - GEPPD  
geppd@ecomf.furg.br

<sup>1</sup> Membros do GEPPD      <sup>2</sup> Colaboradores do GEPPD

## Introdução

Várias têm sido as pesquisas realizadas sobre o assunto Computação de Alto Desempenho, entre elas: modelos de algoritmos paralelos; modelos de programação paralela; modelos de arquiteturas paralelas; modelos de comunicação; sistemas operacionais paralelos; ambientes de suporte à programação paralela; aplicações que exigem computação de alto desempenho, etc. Neste trabalho nos atemos aos problemas mais relacionados com os programadores e com os usuários dos sistemas. Ou seja, aos ambientes de suporte à programação paralela e às aplicações que exigem computação de alto desempenho. Mais especificamente, sobre *software* para arquiteturas multicomputadores, que é o caso das redes locais de computadores. Existem alguns conceitos que, interrelacionados, podem proporcionar a utilização mais amigável de um multicomputador: *thread*, *remote procedure call* (RPC), *socket*, *remote method invocation* (RMI), etc. E algumas ferramentas que, utilizando esses conceitos, se propõem a dar suporte à ambientes que visem tornar mais amigável a tarefa de programação paralela, por exemplo: *remote method invocation*, etc. E algumas ferramentas que, utilizando esses conceitos, se propõem a dar suporte à ambientes que visem tornar mais amigável a tarefa de programação paralela, por exemplo: *message passing interface* (MPI), *parallel virtual machine* (PVM). No entanto, a disseminação restrita desses conceitos, assim como a necessidade de aprimorar ainda mais essas ferramentas, dificultam a utilização plena dos recursos computacionais disponíveis no conjunto dos inúmeros computadores interligados em rede existentes nas instituições, particularmente na FURG. Assim, pretendemos com este projeto aprimorar os ambientes de programação paralela, tornando-os de utilização mais amigável.

## Objetivos

O trabalho tem início com a tentativa de uma perfeita e profunda apreensão dos conhecimentos existentes a respeito de Computação de Alto Desempenho, principalmente os relacionados à aspectos de *software*: *threads*, *remote procedure calls* (RPM), *remote*

*method invocation, sockets, message passing interface (MPI), parallel virtual machine (PVM)*, na forma em que se encontram implementados em sistemas abertos (*software livre*) sobre e no sistema operacional Linux. Essa apreensão dá-se através de estudos dirigidos e pela implementação da solução de exercícios paradigmáticos da área. Ao longo desses estudos teórico-práticos estão sendo identificadas melhorias que poderão ser realizadas nas soluções atualmente existentes, na direção acima assinalada. Tais melhorias serão incorporadas nos pacotes que vierem a ser utilizados. Caso o vulto dessas melhorias comporte, novos pacotes serão construídos. Concomitantemente, estão sendo identificados, junto aos pesquisadores que se tem acesso através dos registros da Instituição, problemas que requeiram o suporte de computação de alto desempenho. Dar-se-á retorno a esse levantamento, demonstrando a eventual possibilidade da solução dos problemas através de programação paralela.

## **Trabalhos em andamento**

Vários são os trabalhos que se pretende incluir no bojo do presente projeto. Alguns deles, em diferentes estágios de desenvolvimento, estão a seguir resumidos.

### **Implementação paralela SPMD do algoritmo gradiente conjugado**

O algoritmo Gradiente Conjugado (GC) tem por objetivo resolver numericamente um sistema de equações lineares do tipo Poisson em um domínio quadrado unitário, dado um conjunto de condições de contorno e a solução exata do problema. Utilizou-se aproximação por diferenças finitas de quarta ordem e condições de contorno de Dirichlet. O algoritmo itera até que seja atingida uma tolerância determinada pelo usuário. Quando resolvendo problemas que exigem alta precisão, o tempo de obtenção da resposta final do algoritmo GC torna-se extenso, sendo indicada utilização de computadores com alto poder de processamento. Com objetivo de utilizar o poder de processamento disponível na instituição, propôs-se uma implementação paralela SPMD (Single Program, Multiple Data) para o algoritmo GC, baseada no paradigma de divisão e conquista, dividindo igualmente o domínio do problema entre as máquinas participantes do processamento. De forma inerente à aproximação por diferenças finitas de quarta ordem, determina-se o valor de cada ponto da matriz, a partir dos valores obtidos na iteração anterior, de seus oito pontos adjacentes. Assim sendo, o cálculo dos valores dos pontos localizados nas bordas de um subdomínio dependem dos valores de pontos localizados nos subdomínios adjacentes. São criados processos cooperantes e esses, ao final de cada iteração, comunicam-se entre si, repassando os valores dos pontos da borda para aqueles em que existe relação de dependência de dados. Ainda ao final de cada iteração são calculados os erros acumulados em cada subdomínio, sendo esses erros somados para obtenção do erro de todo o domínio. Para determinar o fim da computação o erro de todo domínio é confrontado com a tolerância dada. Essa implementação foi realizada utilizando a linguagem de programação C e a biblioteca de comunicação MPI-LAM.

### **Análise de dados climáticos através de programação paralela**

Durante a execução do projeto PAVAO (Padrões da Variabilidade Oceânica de Me-

soescale no Sul do Brasil: uso de funções empíricas ortogonais e redes neurais supervisionadas), foram identificados dois aspectos nos quais o processamento paralelo poderia ser aplicado, visando sanar algumas limitações existentes. O primeiro desses aspectos diz respeito ao método das funções empíricas ortogonais (EOFs), também conhecido como análise das componentes principais (PCA), que é um importante e conhecido método utilizado na análise de dados climáticos. Os dados são apresentados na forma de uma matriz, com cada uma de suas linhas podendo ser encarada como um mapa contendo, por exemplo, os valores instantâneos de temperatura em pontos de uma área do oceano. A análise consiste em, a partir da matriz, determinar padrões espaciais de variabilidade, suas variações no tempo e a importância de cada um desses padrões. Tais informações são obtidas através do cálculo dos autovalores e autovetores, que é realizado sobre uma matriz de covariância ( $R = F^t \cdot F$ ) da matriz de dados ( $F$ ). Existem implementações bem sucedidas de todas essas funções, mas sem o uso de paralelização, o que causa a inconveniência da enorme demanda de tempo para análise de grandes matrizes. Já o segundo aspecto, consiste na implementação de uma rede neural artificial (RNA), com o objetivo de introduzir a não linearidade à análise, que se transforma em análise não linear das componentes principais (NLPCA). Hsieh [HSI 2001] propõe e implementa uma RNA, onde podemos perceber que, além da grande demanda de processamento, uma nova limitação surge, agora a respeito da rede neural propriamente dita, que, devido ao número de entradas fornecidas (matriz muito grande), poderia ficar saturada. Dessa maneira, a utilização de processamento paralelo, em ambos os casos, possibilitaria a análise de um volume de dados bem maior do que o atualmente realizado.

### **Tratamento de dados climáticos com uso de redes neurais**

Vêm-se trabalhando em tratamento de dados climáticos, com o uso de redes neurais. Porém, devido à não utilização de um método capaz de identificar qual seria a arquitetura ótima da rede neural a ser aplicada a uma determinada situação, essa identificação tem sido feita de forma artesanal, baseada apenas na experiência adquirida em situações anteriores. Esse processo tem-se mostrado ineficiente, pois o número de iterações do tipo construção-treinamento-teste, até a obtenção de uma arquitetura adequada, pode consumir muito tempo e, dependendo do problema, esse tempo se torna proibitivo. Visando resolver esse problema, passaremos a fazer uso de algoritmos genéticos, com o intuito de automatizar o processo de determinação de qual arquitetura seria ótima para cada situação. Para que essa solução seja viável deverão ser adotadas ferramentas da programação paralela e distribuída.

### **Tratamento de imagens de satélites**

O Laboratório de Sensoriamento Remoto da FURG recebe diariamente, através de uma estação de recepção de imagens de satélites, oito imagens com mais de 100Mb cada uma. Essas imagens devem ser processadas antes de sua utilização prática. Esse processamento envolve duas etapas bastante distintas. Inicialmente, é necessário que as imagens passem por um processo de calibração. Logo após, sobre as imagens calibradas são aplicados uma série de algoritmos, gerando assim o produto de interesse para pesquisa. A imagem final pode representar a temperatura da região abrangida, incluindo

solo, mar e nuvens. Outro produto de muito interesse para pesquisa é a concentração de clorofila nos oceanos. Cada imagem possui uma dimensão de aproximadamente 2048 por 5000 pontos, com cinco bandas distintas, onde cada ponto representa um índice de radiação variando de 0 a 1024 (10 bits). Cada ponto da imagem representa uma área de 1  $Km^2$  analisada pelo satélite. Durante o processo de calibração, esta imagem bruta deve ser convertida de forma que cada ponto passe a ser representado por um número de ponto flutuante (geralmente 32 bits). A segunda etapa é a aplicação de uma série de algoritmos sobre determinadas bandas da figura, de acordo com o objetivo a ser analisado através da imagem final. Devido à dimensão destas imagens e ao grande número de cálculos realizados para cada ponto das mesmas, esta aplicação exige um alto poder computacional. Devido ao intrínseco potencial paralelo da aplicação, o objetivo do trabalho é realizar uma implementação SPMD de todo o processo envolvido na obtenção das imagens finais.

## Conclusões

Foi apresentado um dos trabalhos que atualmente vem sendo desenvolvidos pelo Grupo de Estudos em Programação Paralela e Distribuída da FURG (GEPPD). Trata-se da aplicação de técnicas de programação paralela à solução de problemas que exigem computação de alto desempenho. Mais especificamente, da utilização de *software* para arquiteturas multicomputadores. Como o suporte de *hardware* utilizado para os testes não possui mecanismo de chaveamento de mensagens de alta velocidade (rede Myrinet, p.e.), foram desconsiderados os tempos de resposta das soluções. Ao ter que considerar esses tempos, conclui-se que o projeto é viável apenas para aplicações que não exijam baixos tempos de resposta. Para as demais aplicações a sua viabilidade fica restrita à simulações, o que, de qualquer forma, proporciona a condição de se analisar a possibilidade dessas aplicações virem a utilizar a tecnologia em foco, quando estiver disponível um mecanismo de chaveamento de mensagens de alta velocidade.

## Referências

- [ANA 2001] ESCOLA REGIONAL DE ALTO DESEMPENHO, 1., 2001, Porto Alegre. **Anais...** [S.l.: s.n.], 2001.
- [HSI 2001] HSIEH, W. W. **Nonlinear principal component analysis by neural networks**. Tellus, (in press).
- [PAC 97] PACHECO, P. S. **Paralell programming with mpi**. [S.l.]: Morgan Kaufmann, 1997.
- [SIL 95] SILBERSCHATZ, A.; GALVIN, P. B. **Operating system concepts**. Reading, USA: Addison-Wesley, 1995. 780p.
- [TAN 92] TANENBAUM, A. S. **Modern operating systems**. New Jersey: Prentice Hall PTR, 1992. 728p.
- [TAN 96] TANENBAUM, A. S. **Computer networks**. New Jersey: Prentice Hall PTR, 1996. 814p.