

Aplicação de um Modelo de Programação com Paradigma de Memória Compartilhada na Espacialização do Relevo utilizando Threading Building Blocks (TBB)

Deleisson Santos¹, Edmilson Brito¹, Elmo Libório¹, Tiago Machado¹,
Murilo Boratto¹, Brauliro Leal¹, Leandro Coelho²

¹ Colegiado de Engenharia de Computação
Universidade Federal do Vale do São Francisco (UNIVASF)
Avenida Antônio Carlos Magalhães, 510 – 48.902-300
Juazeiro – Bahia – Brasil

{deleissonariel, jb.es, elmo.leo, tiago_otti}@hotmail.com,

{murilo.boratto, brauliro.leal}@univasf.edu.br

²Núcleo de Arquitetura de Computadores e Sistemas Operacionais (ACSO)
Universidade do Estado da Bahia (UNEB)
Rua Silveira Martins, 2555, Cabula, 41195-001
Salvador – Bahia – Brazil

leandrocoelho@uneb.br

Resumo. *A Espacialização do Relevo é um instrumento muito utilizado na simplificação da representação do relevo e sua grande importância se deve ao fato de possibilitar a descrição de fenômenos por meio de modelos matemáticos. Graficamente, equivale a identificar a curva ou superfície matemática que melhor se ajusta aos pontos de um diagrama de dispersão. A Computação de Alto Desempenho vem ampliar o poder computacional da espacialização possibilitando o desenvolvimento da representação do relevo na região do Sub-Médio no Vale São Francisco. Nesse trabalho apresentamos uma metodologia para representação do relevo utilizando o método de regressão polinomial bidimensional otimizado através de um Modelo Paralelo de Computação de Alto Desempenho.*

1. Introdução

Alguns fatos recentes têm proporcionado o desenvolvimento de aplicações que representem recursos geofísicos através de uma visão computacional de forma eficiente, e dentre essas representações destaca-se a Espacialização de Relevo através de polinômios bidimensionais [Nogueira et al. 2008]. O problema de representação do relevo através de um polinômio já havia sido estudado anteriormente [Bajaj et al. 1993], porém sem a abordagem na distribuição de processamento. Este fato limitou que um polinômio de alto grau fosse estimado e o tamanho da área a ser representada fosse limitada, pois quanto maior for a informação a ser representada, maior será o poder computacional exigido, sendo que além disso, uma representação com relativa fidelidade requer um polinômio de alto grau.

Dentre as inúmeras justificativas para a realização da Espacialização do Relevo nos centramos na busca de respostas para mensurações em áreas agrícolas, tendo no dimensionamento de plantações, na otimização de recursos hídricos, na logística de estocagem da produção e minimização dos efeitos erosivos, fatores preponderantes. Logo o processo de Espacialização do Relevo torna-se uma ferramenta fundamental e indispensável na exploração eficiente da agricultura, principalmente no pólo agrícola situado as margens da região do Sub-Médio do Vale do Rio São Francisco, que destaca-se como uma das maiores zonas de vinicultura e fruticultura para exportação do país. E nesta zona um dos principais problemas que dificultam uma maior eficiência dos fatores produtivos agrícolas deve-se a problemas relacionados com a erosão dos solos, associado ao uso da terra de maneira inadequada. Neste contexto, insere-se a realização deste estudo também possam contribuir com a caracterização dos processos de degradação do solo.

O objetivo deste trabalho é apresentar uma metodologia para representação do relevo do Sub-Médio do Rio São Francisco através do método de regressão polinomial bidimensional utilizando a Computação de Alto Desempenho. O trabalho está estruturado da seguinte forma: Na Seção 2 abordaremos o modelo matemático da Espacialização do Relevo. Na Seção 3 apresentaremos o Modelo de Programação de Memória Compartilhada utilizando a biblioteca Threading Building Blocks (TBB). Na Seção 4 são apresentados os Resultados Experimentais e finalizaremos com Conclusões e Trabalhos futuros.

2. Modelagem Matemática do Relevo

Um Modelo Matemático do Relevo é uma representação matemática computacional da distribuição de um fenômeno espacial que ocorre dentro de uma região da superfície terrestre [Namikawa et al. 2003]. Uma das técnicas para realizar essa representação do relevo é através do Modelo de Grade Regular [Rufino et al. 2009], onde o mapeamento da superfície é feita com um ajuste global a uma superfície polinomial através da técnica de regressão polinomial. Essa técnica que ajusta um polinômio bidimensional que melhor descreva a variação dos dados de uma amostra torna-se limitada devido o alto poder computacional demandado exigido para realizar a regressão em um conjunto de dados muito grande. Quando se utiliza modelos matemáticos de regressão, o método de estimação dos parâmetros mais amplamente utilizado é o método dos mínimos quadrados ordinários [Golub and Loan 1989] que consiste em estimar uma função para representar um conjunto de pontos minimizando o quadrado dos desvios. Considerando um conjunto de coordenadas geográficas (x, y, z) , tomando a altitude estimada como função estimadora destes pontos, um polinômio de grau r em x e de grau s em y pode ser dado conforme a Equação 1, com o erro ε_{ij} estimado pela Equação 2.

$$\hat{z} = f(x_i, y_j) = \sum_{k=0}^r \sum_{l=0}^s a_{kl} x_i^k y_j^l \quad (1)$$

$$\varepsilon_{ij} = z_{ij} - \hat{z}_{ij} \quad (2)$$

3. Modelo de Programação com Paradigma de Memória Compartilhada - Threading Building Blocks (TBB)

O modelo de Programação com Paradigma de Memória Compartilhada [Kumar 2002] caracteriza-se pela existência de uma porção de memória que possa ser acessada diretamente por todos os elementos de um conjunto de processos. Esta memória será utilizada para transferência de informação entre os mesmos. Este tipo de modelo corresponde a sistemas que possuem um conjunto de memória compartilhada com todos os processadores envolvidos, onde a memória estaria distribuída no sistema, entre os distintos processadores. Existem ferramentas específicas de programação em Memória Compartilhada. As mais conhecidas são: pthreads [Dongarra et al. 1998], OpenMP [Dagum and Menon 1998] e Threading Building Blocks [TBB 2011].

Threading Building Blocks é uma biblioteca desenvolvida pela Intel, voltada para a área de processamento paralelo. É uma biblioteca construída sobre *templates*, o que a torna somente utilizável sobre a linguagem C++. Seu objetivo é facilitar a utilização de *threads*, minimizando as preocupações recorrentes na programação de Alto Desempenho, como a busca dos melhores parâmetros de execução.

Algoritmo 1 Pseudo-Código com a Otimização TBB.

```
class Matrizes {
public:
void operator()(const blocked_range2d<size_t>& r) const{
    double *A = my_A,*B = my_B, *x = my_x, *y = my_y, *z = my_z;
    int      N = my_N, R = my_R, s = R, n = my_n;
    for(size_t l=r.rows().begin();l!=r.rows().end();++l)
        for( size_t c=r.cols().begin(); c!=r.cols().end();++c){
            A[l+c*N] = 0.0;
            for(size_t i=0;i<n;++i)
                A[l+c*N] += pow(x[i], (int)(1/(s+1)) +(int)(c/(s+1)))
                    *pow(y[i], 1%(R+1)+c%(R+1));}
            for(size_t l=r.rows().begin();l!=r.rows().end();++l){
                B[l] = 0.0;
                for(size_t i = 0; i < n; i++)
                    B[l] += z[i] * pow(x[i], (int)(1/(s+1)))* pow(y[i], 1%(R+1) );}}
    Matrizes(double *A, *B, *x, *y, *z, int N, R, n):
    my_A(A), my_B(B), my_x(x), my_y(y), my_z(z), my_N(N),
    my_R(R), my_n(n) {};}
void ParallelMatrizes(double *A, *B, *x, *y, *z, int N, R, n){
    parallel_for(blocked_range2d<size_t>(),Matrizes());}
```

No Algoritmo 1 mostra-se que para o problema proposto foi adotada a estratégia de paralelismo baseado na classe *template tbb::parallel_for*, que executam a distribuição, o gerenciamento e a execução automática das tarefas, expressas em ciclos de repetições com baixo desempenho no algoritmo de Espacialização de Relevância. O método *blocked_range2d<T>* é provida pela biblioteca TBB, que descreve um paralelismo para estruturas bidimensionais sobre um tipo T de dado. Sua inicialização

representa a subdivisão das tarefas dos ciclos de repetições de 0 a $n - 1$, alocando-as em subespaços para cada processador. O gerenciador de tarefas do TBB analisa automaticamente o número ótimo de *threads*, executando o melhor balanceamento de carga de trabalho entre os demais processadores. Como resultado, a aplicação com *threads* utilizando TBB tem uma maior escalabilidade, já que se adapta de maneira automática a aplicação aos melhores parâmetros do sistema de execução.

4. Resultados Experimentais

Esta seção mostra os resultados experimentais do algoritmo implementado para o Modelo de Computação utilizando o Paradigma de Memória Compartilhada para a Espacialização do Relevô. Utilizou-se uma máquina Linux, com arquitetura Intel Xeon, 2 Processadores, quadcore, 2 GB de Memória, denominada **SOL**, sol.inf.um.es, estando localizada no Laboratório de Computação Científica na Universidad de Murcia (UM) [LABCOCI 2011], Espanha.

Os melhores parâmetros de execução foram encontrados automaticamente através do TBB, tendo em vista a obtenção do máximo desempenho frente as características do ambiente de experimentação. Podemos observar um aumento gradativo no desempenho do algoritmo, ou seja, uma redução no tempo de execução, tendo como referência o algoritmo sequencial. A eficiência da biblioteca utilizada aumentam em uma razão diretamente proporcional a complexidade do problema, sendo que para problemas pequenos o ganho de desempenho é insatisfatório. A Figura 1 e a Tabela 1 mostram uma crescente melhora no desempenho do sistema. Para uma observação mais precisa do comportamento do problema de relevô é necessário a realização de execuções com altos Graus do Polinômio estimados.

Tabela 1. Representação tabular dos tempos de execução (em segundos) e speedup.

| Grau do Polinômio | Sequencial | TBB | Speedup |
|-------------------|------------|----------|---------|
| 2 | 597,67 | 564,95 | 1,15 |
| 4 | 2983,6 | 2936,98 | 1,33 |
| 6 | 10506,64 | 10510,59 | 1,78 |
| 8 | 27930,37 | 17999,93 | 2,53 |
| 10 | 59585,84 | 24596,85 | 3,65 |
| 12 | 107916,87 | 31193,78 | 5,17 |

5. Conclusões e Trabalhos Futuros

Os resultados experimentais obtidos neste trabalho indicam que o modelo proposto é eficiente, tendo como base a classificação de tarefas, tempo de resposta e distribuição da carga de trabalho, para o problema proposto. O modelo construído demonstra qual a mensuração adequada do poder de computação deve ser combinado com o paralelismo intrínseco, o qual deve ser utilizado na execução do algoritmo. Não obstante, existem outros modelos de programação distribuída que podem explorar a temática abordada neste trabalho tornando-o mais completo. Neste sentido novas experimentações estão sendo desenvolvidas para validar o modelo matemático proposto. Finalmente, o nosso objetivo para o futuro é estender o algoritmo paralelo a

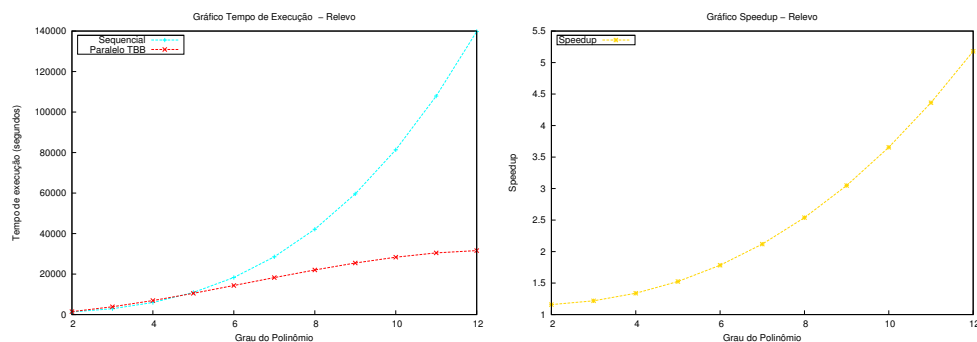


Figura 1. Representação gráfica dos tempos de execução (em segundos) e speedup.

outras plataformas mesclando ferramentas mais poderosas para lidar com arquitetura específicas.

Referências

- Bajaj, C., Ihm, I., and Warren, J. (1993). Higher-order interpolation and least-squares approximation using implicit algebraic surfaces. *ACM Trans. Graph.*, 12:327–347.
- Dagum, L. and Menon, R. (1998). OpenMP: An industry-standard API for shared-memory programming. *IEEE Comput. Sci. Eng.*, 5(1):46–55.
- Dongarra, J. J., Duff, I. S., Sorensen, D. C., and der Vorst, H. A. V. (1998). *Numerical Linear Algebra for High-Performance Computers*. SIAM.
- Golub, G. H. and Loan, C. F. V. (1989). *Matrix Computations*. JohnsHopkinsPress, Baltimore, MD, USA, second edition.
- Kumar, V. (2002). *Introduction to Parallel Computing*. Addison-Wesley Longman Publishing Co., Inc.s, Boston, MA, USA, 2nd edition.
- LABCOCI (2011). Laboratório de Computación Científica (Universidad de Murcia). Available in (2011) June 16: <http://www.um.es/pcgum/>.
- Namikawa, L., Felgueiras, C., Mura, J., and Lopes, E. (2003). Modelagem numérica de terreno e aplicações. *INPE*, 1:158.
- Nogueira, L., Abrantes, R. P., and Leal, B. (2008). A methodology of distributed processing using a mathematical model for landform attributes representation. In *Proceeding of the IADIS International Conference on Applied Computing*.
- Rufino, I., ao, C. G., Rego, J., and Albuquerque, J. (2009). Water resources and urban planning: the case of a coastal area in brazil. *journal of urban and environmental engineering*, 3:32–42.
- TBB (2011). Threading Building Blocks. Available in (2011) June 16: <http://www.threadingbuildingblocks.org>.