Instructions for Authors of SBC Conferences Papers and Abstracts

Luciana P. Nedel[1], Rafael H. Bordini[2], Flávio Rech Wagner[1], Jomi F. Hübner[3]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

[2]Department of Computer Science – University of Durham
Durham, U.K.

[3]Departamento de Sistemas e Computação
Universidade Regional de Blumenal (FURB) – Blumenau, SC – Brazil

{nedel,flavio}@inf.ufrgs.br, R.Bordini@durham.ac.uk, jomi@inf.furb.br

Resumo. *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

1. Introdução

2. Desenvolvimento

2.1. Implementação do Algorítimo

2.1.1. Código Sequencial

O código sequencial implementa a solução númerica da equação de difusão usando uma abordagem serial. Utilizando-se do método de diferenças finitas, é simulado a dispersão de uma substância em uma matriz bidimensional. Cada célula da matriz representa a concentração de uma substância em um ponto do espaço.

O cálculo é realizado em um laço de repetição que itera sobre todas as células da matriz. A atualização de cada célula depende da média das concentrações dos seus vizinhos imediatos e de parâmetros físicos como coeficiente de difusão, o intervalo de tempo DELTA_T e o espaçamento espacial DELTA_X.

2.1.2. Código Paralelo

O código paralelo implementa uma versão desenvolvida em OpenMP do mesmo algorítimo sequencial, para distribuir o trabalho entre mútiplos núcleos do processador. Isso é alcançado utilizando-se das diretivas específicas da biblioteca na estrutura do código sequencial.

- #pragma omp parallel for collapse(2): É a diretiva que divide automaticamente os loops de cálculo entre mútiplos threads. Cada thread é responsável por atualizar uma parte distinta da matriz, permitindo que várias partes do cálculo sejam realizadas simultaneamente. O uso de collapse(2) indica que o compilador deve tratar múltíplos laços aninhados com um único para fins de paralelismo. No caso do código de difusão os dois laçõs de repetição aninhados serão linearizados.
- omp_set_num_threads(int num_threads): Permite configurar dinamicamente o número de threads a serem utilizados pelo código.
- #pragma omp parallel for reduction(+:difmedio) collapse(2): Essa diretiva assim como a anterior executa as iterações do laço de repetição de maneira paralela, com o adicional de reduções, que garantem que a operação de somatório que monitora a convergência do algorítimo seja realizada de maneira segura entre as diferentes threads que acessam as variáveis dessa operação.

2.1.3. Interface Python e ferramenta CMake

O sistema de compilação do projeto é gerenciado pelo CMake, uma ferramenta de automação de compilação. Por meio de um arquivo de configuração o processo de construção do projeto é realizado de forma modular e eficiente. Nesse arquivo são definidas as dependências dos programas como a biblioteca OpenMP. Além disso, uma particularidade do uso do CMake e do projeto é o uso de flags de compilação que desativam as otimizações do compilador, dado que nas primeiras versões do código, as diferenças de desempenho entre as implementações sequencial e paralela não eram significativas. Supôs-se que isso ocorria devido a vetorização automática realizada pelo compilador de C no código sequencial, otimizando as operações a ponto de reduzir a vantagem do paralelismo. Ao desativar essas otimizações, forçou-se o compilador a seguir uma execução mais direta e sem otimizações agressivas, permitindo que a diferença de desempenho entre as abordagens fosse evidenciada.

Ademais foi implementada uma interface Python para integrar os códigos em C com funcionalidades de alto nível. Essa interface utiliza o módulo ctypes para carregar a biblioteca de difusão escrita em C e mapear suas funções. Essa interface descreve funções que executam os códigos sequencial e paralelo em C, proporcionando a flexibilidade para execução das simulações, facilitando a integração com os métodos de analise desenvolvidos em notebooks Jupyter, nos quais há a implementação do código de visualização dos resultados.

3. Resultados

Nesta seção, apresentamos os resultados obtidos de nossa implementação. Inicialmente, analisamos a equivalência lógica entre os códigos sequencial e paralelo, considerando possíveis erros que podem surgir na paralelização, como condições de corrida ou inconsistências de sincronização. Em seguida, ilustramos, por meio de mapas de calor, a atualização dos valores da matriz ao longo do tempo. Por fim, realizamos uma análise comparativa dos tempos médios de execução, *speedup* e eficiência entre as duas versões.

3.1. Validação da Implementação

Para assegurar a correção das duas implementações, verificamos em cada iteração se os valores presentes em cada célula da matriz são idênticos. Dessa forma, o resultado final na última iteração deve ser o mesmo em ambas as versões.

Por meio desse procedimento, utilizando a interface Python em conjunto com um Jupyter Notebook, comprovamos que as duas soluções produzem resultados idênticos. Isso era esperado, pois no código paralelo não ocorrem condições de corrida, uma vez que a escrita não é realizada na mesma região de memória das leituras, tornando o processamento de cada célula pelas *threads* independente.

3.2. Validação da Implementação

Para ilustrar o funcionamento da implementação, foram gerados mapas de calor, representado pela Figura 1, nos quais cada ponto de uma matriz 50x50 é representado por uma cor distinta. Cores escuras correspondem a valores próximos de um, indicando alta concentração do contaminante, enquanto cores claras representam valores próximos de zero, indicando baixa presença de contaminação.

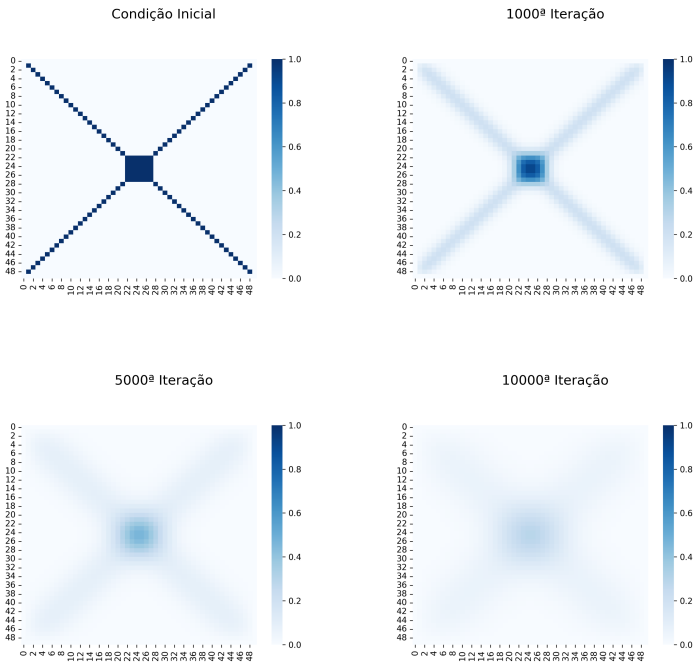

Figura 1. Mapa de calor em quatro instantes distintos da simulação.

Analisando a progressão dos mapas de calor, observamos que o comportamento faz sentido no contexto da solução proposta. Inicialmente, o contaminante é adicionado com alta concentração nas diagonais e no centro da matriz, evidenciado pelas regiões azuis escuras. Com o avanço das iterações, o contaminante começa a se difundir para as regiões adjacentes, aumentando gradativamente a luminosidade nessas áreas e dimi-

nuindo nos pontos de concentração inicial. Na última iteração, a concentração se distribui uniformemente pela matriz, com valores próximos entre si.

3.3. Análise de Desempenho

A análise de desempenho foi realizada em um computador *desktop* com as seguintes especificações. Além disso, as especificações dos parâmetros do problema foram incluídas na tabela correspondente.

Tabela 1. Tabela de especificação de Hardware

Especificações	Detalhes
Processador	Intel i7-4790 @ 3.60GHz
Núcleos / Lógicos	4 / 8
Memória RAM	8 GB
Sistema Operacional	Ubuntu 22.04.05 (via WSL)

Tabela 2. Tabela de especificação da Simulação

Especificações	Detalhes
Dimensão da Matriz (N x N)	2000 x 2000
Número de Iterações	500
Distribuição Inicial	Alta concentração no centro
Coeficiente de Difusão	0.1
dt	0.01
dx	1.0

Para obter valores mais consistentes e minimizar influências externas, como outros programas em execução, cada teste foi executado quinze vezes e, assim, calculamos o tempo médio gasto e seu desvio padrão. O *speedup* é calculado dividindo-se o tempo de execução sequencial pelo tempo de execução paralelo correspondente, enquanto a eficiência é determinada ao dividir o *speedup* pelo número de *threads* utilizados.

Tabela 3. Tabela de comparação de desempenho entre o código sequencial e o paralelo utilizando OpenMP.

Nº Threads	Tempo	Speedup	Eficiência
1	26.22 ± 2.09	1.0	100%
2	14.59 ± 1.49	1.80	89,88%
4	10.60 ± 1.43	2.47	61.82%
8	8.52 ± 1.32	3.08	38.48%
16	8.29 ± 1.29	3.16	19.77%
32	8.67 ± 1.43	3.03	9.45%

O gráfico de *speedup* (Figura 2) mostra uma tendência de estabilização em torno do valor três, enquanto o *speedup* linear ideal apresenta valores significativamente maiores, o que pode sugerir que a escalabilidade da aplicação é limitada, possivelmente devido

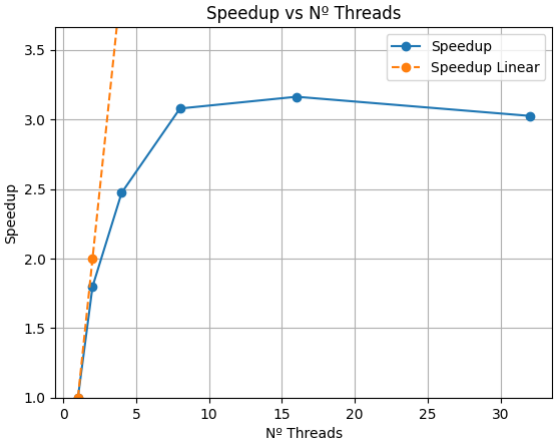

Figura 2. Gráfico do *speedup* por número de *threads*

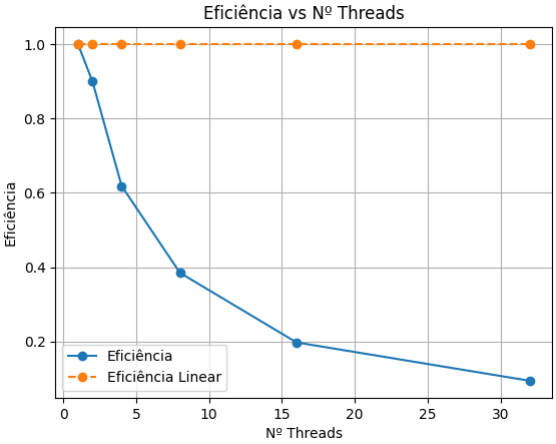

Figura 3. Gráfico da eficiência por número de *threads*

a *overheads* de sincronização ou ao fato de que o problema não é suficientemente grande para aproveitar eficientemente um número maior de *threads*. Além disso, o desempenho pode estar sendo afetado pela latência de memória ou pela arquitetura do processador utilizado e, consequentemente, não há grandes vantagens em utilizar um número elevado de *threads* para resolver este problema específico, já que aumentaria excessivamente o consumo de energia pela máquina, que pode ser observado pela grande queda no gráfico de eficiência da Figura 3.

4. Conclusão

5. General Information

All full papers and posters (short papers) submitted to some SBC conference, including any supporting documents, should be written in English or in Portuguese. The format paper should be A4 with single column, 3.5 cm for upper margin, 2.5 cm for bottom margin and 3.0 cm for lateral margins, without headers or footers. The main font must be Times, 12 point nominal size, with 6 points of space before each paragraph. Page numbers must be suppressed.

Full papers must respect the page limits defined by the conference. Conferences that publish just abstracts ask for **one**-page texts.

6. First Page

The first page must display the paper title, the name and address of the authors, the abstract in English and "resumo" in Portuguese ("resumos" are required only for papers written in Portuguese). The title must be centered over the whole page, in 16 point boldface font and with 12 points of space before itself. Author names must be centered in 12 point font, bold, all of them disposed in the same line, separated by commas and with 12 points of space after the title. Addresses must be centered in 12 point font, also with 12 points of space after the authors' names. E-mail addresses should be written using font Courier New, 10 point nominal size, with 6 points of space before and 6 points of space after.

The abstract and "resumo" (if is the case) must be in 12 point Times font, indented 0.8cm on both sides. The word **Abstract** and **Resumo**, should be written in boldface and must precede the text.

7. CD-ROMs and Printed Proceedings

In some conferences, the papers are published on CD-ROM while only the abstract is published in the printed Proceedings. In this case, authors are invited to prepare two final versions of the paper. One, complete, to be published on the CD and the other, containing only the first page, with abstract and "resumo" (for papers in Portuguese).

8. Sections and Paragraphs

Section titles must be in boldface, 13pt, flush left. There should be an extra 12 pt of space before each title. Section numbering is optional. The first paragraph of each section should not be indented, while the first lines of subsequent paragraphs should be indented by 1.27 cm.

8.1. Subsections

The subsection titles must be in boldface, 12pt, flush left.

9. Figures and Captions

Figure and table captions should be centered if less than one line (Figure 4), otherwise justified and indented by 0.8cm on both margins, as shown in Figure 5. The caption font must be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

In tables, try to avoid the use of colored or shaded backgrounds, and avoid thick, doubled, or unnecessary framing lines. When reporting empirical data, do not use more decimal digits than warranted by their precision and reproducibility. Table caption must be placed before the table (see Table 1) and the font used must also be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

Figura 4. A typical figure

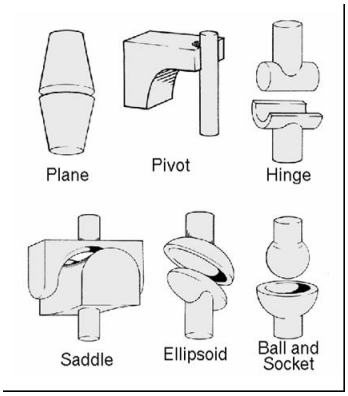

Figura 5. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 9.

10. Images

All images and illustrations should be in black-and-white, or gray tones, excepting for the papers that will be electronically available (on CD-ROMs, internet, etc.). The image resolution on paper should be about 600 dpi for black-and-white images, and 150-300 dpi for grayscale images. Do not include images with excessive resolution, as they may take hours to print, without any visible difference in the result.

11. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991], and [Smith and Jones 1999].

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

Tabela 4. Variables to be considered on the evaluation of interaction techniques

	Chessboard top view	Chessboard perspective view
Selection with side movements	6.02 ± 5.22	7.01±6.84
Selection with in-depth movements	6.29±4.99	12.22±11.33
Manipulation with side movements	4.66± 4.94	3.47±2.20
Manipulation with in-depth movements	5.71 ±4.55	5.37 ±3.28

Referências

Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley & Sons ltd.

Knuth, D. E. (1984). *The TEX Book*. Addison-Wesley, 15th edition.

Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, *Advances in Computer Science*, pages 555–566. Publishing Press.