

Aplicação de Processamento Paralelo na Solução do Problema da Soma dos Subconjuntos

Fernando Concatto¹

¹Bacharelado em Ciência da Computação – Universidade do Vale do Itajaí (UNIVALI)
Caixa Postal 360 – CEP 88302-202 – Itajaí – SC – Brasil

fernandoconcatto@edu.univali.br

Resumo. *O Problema da Soma dos Subconjuntos é um problema computacionalmente difícil, demandando tempo exponencial para ser resolvido. Este trabalho buscou aplicar técnicas de processamento paralelo na busca de soluções para o problema, com a intenção de identificar o ganho de desempenho por thread utilizada. //TODO finalizar*

1. Introdução

Um problema computacional pode ser interpretado como uma questão a ser respondida, geralmente possuindo *parâmetros*, ou *variáveis*. O problema deve ser definido a partir da descrição de todos os seus parâmetros e do estabelecimento de quais propriedades a resposta ou *solução* deve ser composta para ser considerada uma resposta válida para o problema. Uma *instância* do problema é obtida ao atribuir valores a todos os seus parâmetros [Garey and Johnson 1979].

Algoritmos são procedimentos passo-a-passo que resolvem problemas. Dado um problema, um algoritmo *resolve* tal problema se ele sempre produz uma solução para qualquer uma de suas instâncias. Um objetivo bastante comum na busca de soluções para um problema é o desenvolvimento de um algoritmo eficiente, que resolve o problema no menor tempo possível. O campo da Teoria da Complexidade Computacional busca estudar e classificar os algoritmos, identificando a quantidade de recursos computacionais necessária para executar um algoritmo. Geralmente, a eficiência de um algoritmo é definida pela quantidade de operações básicas que o mesmo demanda para resolver o problema. Esta quantidade é usualmente estabelecida em termos do tamanho da instância do problema, denotado pelo símbolo n [Arora and Barak 2009, Garey and Johnson 1979].

Uma das principais classes de problemas identificadas pela Teoria da Complexidade é a classe NP-completo, um subconjunto da classe NP, que significa *polinomial não determinístico*. A classe NP contém todos os problemas cujas soluções podem ser verificadas em tempo polinomial, enquanto a classe NP-completo é composta por problemas onde todos os problemas em NP podem ser reduzidos em tempo polinomial para eles [Garey and Johnson 1979]. Redução, nesse contexto, significa transformar um problema em outro de forma com que uma solução para o segundo problema também possa ser utilizada para resolver o primeiro [Sipser 1996].

Apesar de que as soluções para problemas NP-completos podem ser verificadas em tempo polinomial, nenhum algoritmo para resolver um problema NP-completo em tempo polinomial foi encontrado até hoje; todos demandam tempo exponencial ou superior. Apesar disso, não há nenhuma prova de que não existe um algoritmo eficiente para

resolver problemas NP-completo. Esta condição é um dos principais pontos da questão “ $P \neq NP$?”, um dos maiores problemas abertos no campo da Ciência da Computação [Sipser 1996].

Entre os problemas pertencentes à classe NP-completo está o Problema da Soma dos Subconjuntos. Sua NP-completude foi comprovada por Richard Karp, juntamente com diversos outros problemas, em seu artigo de 1972, intitulado “Reducibility Among Combinatorial Problems” [Karp 1972]. Por ser um problema NP-completo, não se conhece um algoritmo capaz de resolvê-lo em tempo polinomial. Este trabalho se propôs a analisar este problema aplicando técnicas de processamento paralelo, com a intenção de acelerar a velocidade de busca pela solução do problema utilizando um algoritmo de força bruta, que demanda tempo exponencial, e analisar o ganho de desempenho obtido em função da quantidade de unidades de processamento executando simultaneamente.

2. 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.

3. 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).

4. 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.

4.1. Subsections

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

5. Figures and Captions

Figure and table captions should be centered if less than one line (Figure 1), otherwise justified and indented by 0.8cm on both margins, as shown in Figure 2. The caption font must be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.



Figura 1. A typical figure

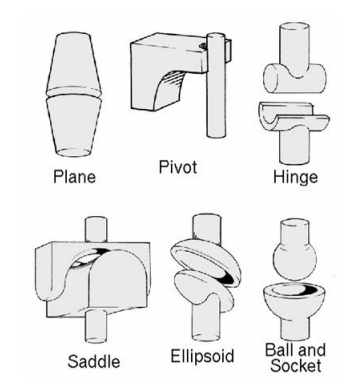


Figura 2. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 5.

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.

Tabela 1. Variables to be considered on the evaluation of interaction techniques

	Value 1	Value 2
Case 1	1.0 ± 0.1	$1.75 \times 10^{-5} \pm 5 \times 10^{-7}$
Case 2	0.003(1)	100.0

6. 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.

7. 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.

Referências

Arora, S. and Barak, B. (2009). Computational Complexity: A Modern Approach. *Annals of Physics*, (January).

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.

Garey, M. R. and Johnson, D. S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences). *Computers and Intractability*, page 340.

Karp, R. M. (1972). *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA.

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

Sipser, M. (1996). *Introduction to the Theory of Computation*, volume 27.

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.