

Paralelização do Método Procura em Rede com OPENMP*

Pablo J. Pavan¹, Sandra B. Neuckamp¹, Edson L. Padoin^{1,2}, Philippe O. A. Navaux²

¹Universidade Reg. do Noroeste do Estado do Rio G. do Sul (UNIJUI) - Ijuí - RS - Brasil

²Universidade Federal do Rio Grande do Sul (UFRGS) - Porto Alegre - RS - Brasil

{pablo.pavan, sandra.neuckamp, padoin}@unijui.edu.br, navaux@inf.ufrgs.br

***Resumo.** A utilização eficiente dos recursos computacionais é um fator indispensável na busca por desempenho. Com essa premissa técnicas de paralelização são cada vez mais empregadas. Este trabalho apresenta os resultados da paralelização do método Procura em Rede com a API OPENMP. Testes foram realizados em dois ambientes e demonstram uma redução de até 25,56 vezes no tempo de execução.*

1. Introdução

As aplicações de simulação que são utilizadas na resolução de problemas reais necessitam de elevados tempos de processamento computacional. Assim, sistemas computacionais de alto desempenho tem sido desenvolvidos para suprir tal necessidade. O processador é um dos componentes que tem apresentado grande avanço tecnológico, passando a proporcionar a execução paralela de aplicações. Deste modo, aplicações podem ser divididas em partes e executadas em paralelo nos núcleos das unidade de processamento [Pilla e Meneses 2015]. Devido a necessidade de se implementar sistemas que utilizam de maneira eficiente a capacidade computacional das máquinas modernas, novas tecnologias estão sendo desenvolvidas. Dentre estas tecnologias, destaca-se a API OPENMP.

Em modelos matemáticos, a estimação de parâmetros desconhecidos, a partir de dados experimentais constitui a área de pesquisa chamada de Problema Inverso (PI). Para resolução destes problemas [da Silva Neto e Neto 2005] propuseram o método Procura em Rede, este que busca a partir da estimativa mínima dos parâmetros, encontrar os parâmetros que representam os dados experimentais com o melhor coeficiente de determinação (R^2) para aquela estimativa. Assim, neste trabalho foi aplicada a API OPENMP na paralelização do método Procura em Rede, almejando reduzir o tempo computacional deste método, e utilizar de forma eficiente toda capacidade computacional disponível em um sistema paralelo.

O restante do trabalho está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve o estudo de caso. A metodologia é apresentada na Seção 4. Resultados são discutidos na Seção 5, seguidos das Conclusões e Trabalhos Futuros.

2. Trabalhos Relacionados

Diefenthaler utiliza o método Procura em Rede para estimar três parâmetros de uma função exponencial para o ajuste de curva referentes a dados coletados do resfria-

*Trabalho parcialmente apoiado por CNPq, CAPES, FAPERGS e FINEP. Pesquisa tem recebido recursos do programa EU H2020 e do MCTI/RNP-Brasil sob o projeto HPC4E de número 689772.

mento da água em diferentes ampolas de vidro [Diefenthaler et al. 2017]. Neste trabalho o método foi paralelizado utilizando OPENMP e apresentou ganhos de 90,82 % (utilizando 4 *cores*) na redução do tempo computacional se comparado ao método implementado em MATLAB.

Kang compara OPENMP, MPI e MapReduce aplicados sobre um benchmark [Kang et al. 2015]. Os resultados alcançados, indicam que quando o problema pode ser resolvido utilizando recursos de somente uma máquina o OPENMP é aconselhado já que não suporta memória distribuída. Para problemas que utilizam grande poder computacional, é aconselhado o uso do MPI já que este suporta memória distribuída, assim podendo ser executado em *clusters* ou *grids*. A utilização de MapReduce é aconselhada quando o número de dados utilizado seja grande, porém com poucas iterações.

3. Estudo de Caso

São diversas as situações do cotidiano em que observamos o resfriamento natural da água. Quando a água se encontra em um reservatório parado, ela tende a alcançar o equilíbrio térmico com o meio após um determinado tempo de exposição. Para este trabalho foram coletadas 9721 amostras de temperaturas de um ambiente de resfriamento natural da água. Neste estudo as temperaturas iniciaram em 79 graus Celsius e foram até 25 graus Celsius. As coletadas de temperatura foram feitas a cada 1 segundo utilizando-se de um equipamento de mensuração de temperatura ligado a um computador que armazenava os dados.

Nesse trabalho buscou-se aplicar o método Procura em Rede paralelo sobre dados experimentais do resfriamento da água com o objetivo de realizar um ajuste de curva que representa tais dados. Para este ajuste são necessários estimar 4 parâmetros (a, b, c, d) de uma função exponencial dada por: $f(x) = a \cdot \exp(b \cdot x) + c \cdot \exp(d \cdot x)$.

O método Procura em Rede consiste em definir para cada parâmetro a ser estimado um intervalo com um valor mínimo e máximo que possivelmente contenha um valor ótimo do parâmetro considerado, construindo assim uma rede de intervalos. De posse desta rede, o Problema Direto é resolvido com todas as combinações dos valores que compõem a rede fazendo a busca pelo menor erro, onde este é a menor diferença entre o valor obtido pelo cálculo do Problema Direto e o valor coletado experimentalmente [da Silva Neto e Neto 2005].

4. Metodologia

Foram utilizados dois ambientes nos testes realizados. O primeiro nomeado **Máquina 1** é composto por uma tradicional desktop, com processador Intel Core-I7 4790 de 8 *cores* com frequência base de 3,6 GHz. Este equipamento possui 16 GB de memória RAM DDR3 de 1600 MHz. O segundo ambiente nomeado de **Máquina 2** possui dois processadores Intel Xeon E5-2640 v2 com 8 *cores* com 2 *SMT-cores*, totalizando 32 *cores* de 2,00 GHz. Este equipamento possui 64 GB de memória RAM DDR3 de 1600 MHz. Em ambas as máquinas utilizaram-se o sistema operacional Ubuntu com versão de *kernel* 3.16.0 – 70. A versão do OPENMP utilizada foi a 4.0 e do compilador gcc a 5.4.0.

Na implementação, procurou-se utilizar a diretiva de paralelização para laços de repetições, como neste problema existem 4 laços de repetições aninhados no cálculo inverso, a diretiva foi colocada no laço mais externo. Utilizando-se da opção *collapse* para

reduzir a granularidade do trabalho para cada segmento. Outros cálculos do problema foram paralelizados utilizando o *parallelfor* e *reduction*, buscando aumentar as regiões paralelas.

De modo a utilizar todos os recursos computacionais, o problema foi executado na Máquina 1 com 1, 2, 4 e 8 *cores* e na Máquina 2 com 1, 8, 16 e 32 *cores*. Em ambas as configurações foi utilizado a *flag* de otimização *-O2* do compilador *gcc*. Cada teste foi executado 10 vezes utilizando-se uma média aritmética para os resultados.

5. Resultados

A Figura 1 demonstra o ajuste de curva, os pontos em azul são os dados experimentais da temperatura da água e a linha em preto é a curva ajustada. Para este ajuste o valor de R^2 foi de 0,9963.

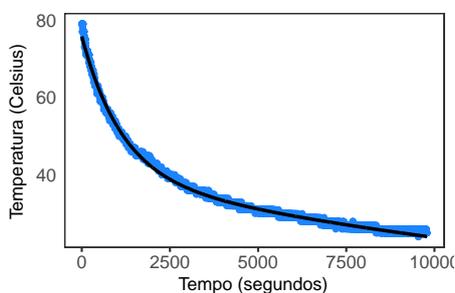


Figura 1. Ajuste da Curva em Relação aos Dados Experimentais.

Na Figura 2 são apresentados os tempos de execução mensurados com a execução da versão paralela na Máquina 1 e na Máquina 2 variando a quantidade de *cores*.

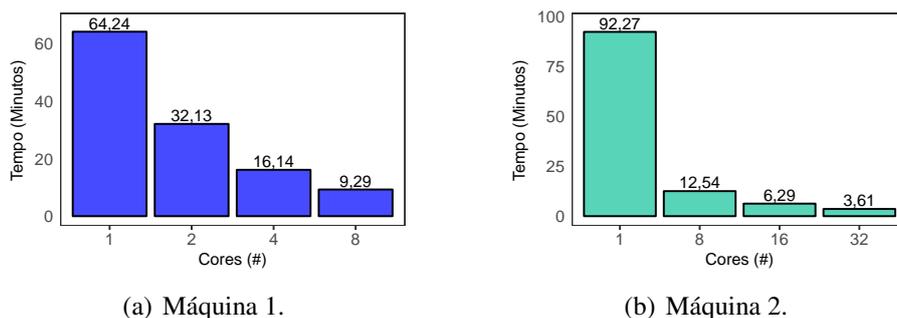
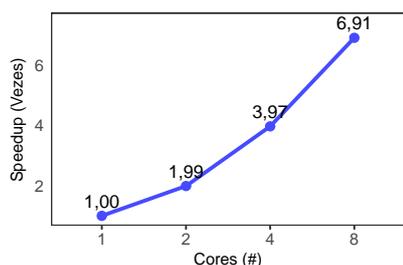


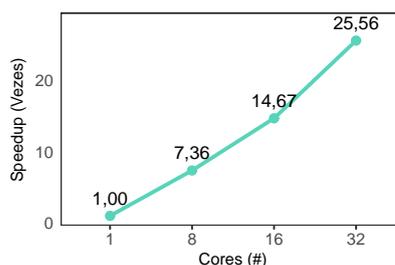
Figura 2. Tempo de Execução do Método Procura em Rede Implementado em OPENMP.

O tempo de execução da versão sequencial foi de 64,24 minutos na Máquina 1 e de 92,27 minutos na Máquina 2. O tempo é maior na Máquina 2 pois seu clock base é menor que na Máquina 1. Este tempo foi reduzido para 9,29 minutos na Máquina 1 e para 12,54 minutos na Máquina 2 quando utilizado oito *cores*.

Com o uso da versão paralela, em ambas as máquinas obteve-se ganhos de desempenho (Figura 3). Para 8 *cores* na Máquina 1 o ganho foi de 6,91 vezes enquanto que na Máquina 2 foi de 7,46 vezes. O melhor *speedup* ganho foi de 25,56 vezes na



(a) Máquina 1.



(b) Máquina 2.

Figura 3. Speedup da Implementação com OPENMP em diferentes cores.

Máquina 2 utilizando 32 *cores*. Isso representa em tempo de execução uma redução de 99,27 minutos para 3,61 minutos.

Com o uso da versão paralela, o emprego do método Procura em Rede passa ser viável em pesquisas com dados reais. Devido a grande demanda de processamento ele não tem sido utilizado mesmo apresentando resultados eficientes. Neste nosso estudo, a função exponencial e os parâmetros ajustados pelo método Procura em Rede, foi encontrado uma curva que representa os dados experimentais do resfriamento da água com uma acurácia de 99,63%.

6. Conclusões e Trabalhos Futuros

O método Procura em Rede pode ser utilizado para encontrar parâmetros de diversas funções, neste trabalho o mesmo foi utilizado para encontrar os parâmetros de uma função exponencial para o ajuste de curva de dados experimentais do resfriamento da água. Para este problema o método tornou-se eficaz apresentando uma acurácia de 99,63%. O uso de paralelismo para o método Procura em Rede, apresentou uma redução no tempo de execução de 25,56 vezes, reduzindo o tempo de 92,27 minutos para 3,61 minutos.

Em trabalhos futuros, pretende-se executar esta aplicação em ambientes com maior disponibilidade de *cores*, como também realizar uma otimização nos acessos a memória. Pretende-se também paralelizar este método para outros problemas, buscando a redução no tempo de execução.

Referências

- da Silva Neto, A. J. e Neto, F. D. M. (2005). *Problemas inversos: conceitos fundamentais e aplicações*. EdUERJ.
- Diefenthäler, A. T., Avi, P. C., e Padoin, E. L. (2017). Processamento paralelo na determinação da curva de resfriamento da Água pelo método de procura em rede. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 6(1):1–2.
- Kang, S. J., Lee, S. Y., e Lee, K. M. (2015). Performance comparison of openmp, mpi, and mapreduce in practical problems. *Advances in Multimedia*, 2015:7.
- Pilla, L. L. e Meneses, E. (2015). Programação paralela em charm++. *ERAD/RS, Gramado, RS, Brasil*.