

Paralelização de um Sistema de Simulação de Irrigação de Solos com Pthreads *

Cristiano A. Künas¹, Leandro P. Heck¹,
Douglas E. Hoffmann¹, Edson L. Padoin¹

¹ Departamento de Ciências Exatas e Engenharias (DCEENG)
Universidade Regional do Noroeste do Estado do Rio Grande do Sul – (UNIJUÍ)
Santa Rosa – RS – Brasil

crislianokunas@outlook.com, leandroph1992@hotmail.com,
douglas22.hoffmann@hotmail.com, padoin@unijui.edu.br

Resumo. *Este artigo apresenta a análise de desempenho da paralelização de um sistema de simulação de irrigação de solos. O objetivo do trabalho é analisar a viabilidade da utilização de Pthreads na paralelização de uma aplicação real. Os resultados alcançados apresentaram speed-up de até 6 vezes se comparado com a versão sequencial.*

1. Introdução

Nas inúmeras pesquisas realizadas em sistemas agroflorestais e agrícolas a ação da água no solo é de grande importância, uma vez que a deslocação dos nutrientes depende do movimento da água no solo. Assim, a descrição de como a água se desloca é fundamental nos projetos de irrigação, principalmente se considerado o sistema de irrigação por gotejamento, onde é preciso determinar a quantidade ótima de água necessária para o desenvolvimento das plantas em cada profundidade do solo [Elmaloglou and Diamantopoulos 2009, Arruda et al. 2015].

Para esses eventos surge a necessidade de implementar sistemas que, de alguma maneira simulem as operações ou ações do mundo real, ou seja, é preciso realizar simulações computacionais, por meio de técnicas matemáticas. Dessa maneira, é possível prever possíveis resultados, antes de aplicar na prática. Nesse sentido, existem diversos paradigmas que podem ser utilizados para implementar estas simulações. Dentre os quais tem-se o modelo de programação paralela com Pthreads que é uma API padrão para criação e manipulação de *threads*, para aplicações que fazem uso de *multithreaded*. A utilização desta API possibilita trabalhar mais em baixo nível, obtendo um controle maior sobre as *threads*.

Assim, o objetivo do trabalho é apresentar o resultado obtido com a utilização de programação paralela com Pthreads, em um sistema de irrigação de solos, avaliando a viabilidade de utilização do mesmo para resolução do problema.

O restante do trabalho está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta a metodologia utilizada na implementação e o ambiente de execução utilizado na realização dos experimentos. Resultados são discutidos na Seção 4, seguido das Conclusões e Trabalhos Futuros.

*Trabalho desenvolvido com recursos do edital MCTIC/CNPq - Universal 28/2018 sob número 436339/2018-8.

2. Trabalhos Relacionados

Na literatura inúmeros trabalhos apresentam estudos sobre a movimentação da água no solo. Borges *et al.* (2005) apresenta uma análise do comportamento da água do solo saturado e não saturado por meio da equação de Richards. A equação de Richards é uma equação diferencial parcial não linear que governa o processo de infiltração e escoamento em solos não-saturados [Männich 2008]. Para complementar o referido trabalho traz a implementação de um algoritmo utilizando o método de diferenças finitas.

No trabalho de Elmaloglou e Diamantopoulos (2009) é apresentado um modelo matemático que descreve o fluxo de água sob linhas de gotejamento subsuperficial levando em conta a absorção de água pela raiz, evaporação da água do solo e da superfície do solo.

No trabalho de Padoin *et al.* (2006), este sistema foi paralelizado utilizando a biblioteca *PVM*, o que possibilitou o uso de matrizes maiores gerando assim resultados mais precisos. Neste trabalho alcançou-se *speed-up* de 1,88 vezes com uma eficiência de até 38%. Já em Padoin *et al.* (2011) utilizou-se o algoritmo sequencial para analisar a eficiência energética de processadores e placas aceleradoras utilizando a paralelização por meio da plataforma *CUDA*.

Por outro lado, várias pesquisas buscam balancear as cargas das tarefas entre os processadores visando melhorar a eficiência da utilização dos sistemas paralelos [Kalé et al. 1998]. Neste trabalho foi implementado uma nova versão paralela do algoritmo utilizando *Pthreads*.

3. Metodologia

Considerando que o sistema de irrigação de solo já tenha sido modelado em outros trabalhos, para execução paralela, fez-se um estudo do algoritmo sequencial [Padoin et al. 2011]. Nesta nova versão paralela foi utilizado a biblioteca *Pthreads* almejando melhorar os tempos de execução das simulações.

Para a análise de desempenho, buscou-se a variação no número de *threads*, além do tamanho da matriz cúbica. Foram realizados experimentos a partir de 1 até 10 *threads*, além de tamanhos 72, 144 e 216. Os experimentos foram realizados utilizando sempre 300 iterações, sendo que em cada um deles, executou-se 10 vezes para obter a média aritmética e *speed-up*.

Na implementação paralela do algoritmo para arquiteturas Multi-Core de memória compartilhada utilizou-se o padrão (*Pthread*). Primeiramente a carga computacional - trabalho - é dividido entre as *threads* baseado no tamanho da matriz cúbica que representa o sistema simulado. Assim, cada *thread* terá uma parte ou fatia da matriz cúbica para realizar as operações.

```
void * propagacao(void *arga){
    .
    .
    .
    struct divisao *divisao = (int *)arga;
    for (i = (divisao->inicio); i < (divisao->final); i++) {
        for (j = 1; j < C-1; j++){
            for (k = 1; k < P-1; k++){
                b[i][j][k] = ((a[i][j][k] + a[i-1][j][k] + a[i][j-1][k] +
                    a[i][j][k-1] + a[i+1][j][k] + a[i][j+1][k] + a[i][j][k+1]) / 7);
            }
        }
    }
}
```

Figura 1. Pseudo-código do método de propagação.

O foco foi em utilizar a paralelização nos laços de repetição que demandam maior tempo e processamento para que fosse possível obter um melhor tempo de execução. A

paralelização foi aplicada ao método *propagação()* (Figura 1, pois este método é responsável por todos os cálculos, movimentações e atribuições de valores na matriz cúbica. Após realizados os cálculos, todas as *threads* são unidas, a partir do método *pthread join*, permitindo agrupar os resultados obtidos por cada uma delas.

3.1. Ambiente de Execução

O ambiente de execução possui um processador Intel Core i5 modelo 5200U. O processador possui 2 cores com 2.20 GHz de frequência. Este equipamento também possui 16 GB de memória RAM DDR3 de frequência 600 MHz. Para os experimentos, utilizou-se o sistema operacional Ubuntu versão 18.04.1 LTS, versão do kernel Linux é 4.15.0 – 39 – *generic*.

4. Resultados

O tempo médio de execuções da aplicação sequencial foi de $2,56 \pm 0,02$ s quando utilizado uma matriz cúbica de ordem (ou tamanho) 72 e 300 interações (Figura 2) para o problema de irrigação. Observa-se que o tempo de execução foi reduzido em 20,31 % quando executado a simulação com a versão paralela.

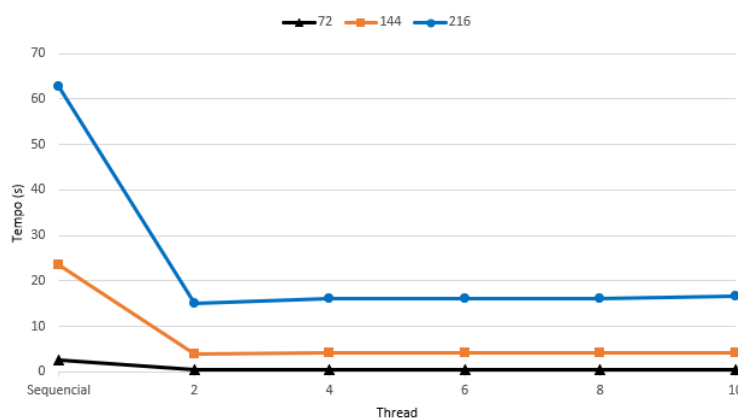


Figura 2. Tempo de execução (s) das simulações com diferentes tamanhos de sistema.

A simulação sequencial com uma matriz cúbica na ordem de 144 com 300 iterações teve um tempo médio de execuções de $23,63 \pm 0,09$ s (Figura 2). Com o aumento do tamanho do sistema, é possível ver que o tempo teve uma redução mais significativa se comparado com a versão sequencial. Na simulação com uma matriz deste tamanho os tempos foram reduzidos para $3,84 \pm 0,08$ s, o que representa ganhos de até 6,15 vezes.

Além disso, constatou-se que a variação de execuções por número de *threads* não obteve grandes diferenças de tempos, ficando consideravelmente semelhantes às execuções com apenas 2 *threads*. Um dos motivos que limitaram a escalabilidade é a arquitetura utilizada que possui apenas dois cores. Outra limitante é a dependência de dados entre cada iteração (passo temporal) da implementação.

Como forma de obter uma maior precisão na simulação, foram também executados experimentos com matriz de ordem 216. Os tempos médios da versão sequencial foram de $62,64 \pm 0,18$ s. Na versão paralela, semelhante aos experimentos com cubos de ordem 72 e 144, os tempos obtidos foram proporcionalmente reduzidos para $15,18 \pm 0,54$ s, alcançando o melhor *speed-up* com 2 *threads*, o qual representa ganhos de até 4,13 vezes sobre o algoritmo sequencial.

5. Conclusões

Analisando os resultados, percebe-se que a utilização da programação paralela com Pthreads apresentou ganhos de desempenho se comparados com a versão sequencial da aplicação. Percebeu-se que, conforme é aumentado a ordem da matriz, obtêm-se um aumento equivalente no tempo de execução da aplicação. Deste modo tem-se *speed-up* semelhante nas execuções paralelas independente da ordem de matriz simulada.

Para trabalhos futuros pretende-se executar a aplicação de modo a simular um ambiente real com matrizes maiores. Por exemplo, tendo como base uma matriz de ordem 294.912, a versão sequencial executada com 300 interações levaria em torno de 13,65 horas para chegar ao resultado. Com a versão paralela implementada em Pthreads, executando com o mesmo tamanho em 2 *threads*, o tempo seria proporcionalmente reduzido para 2,18 horas, o que ainda é um valor elevado, para pesquisas científicas na área de irrigação de solos em sistemas agroflorestais e agrícolas.

Nesse sentido, como trabalhos futuros, pretende-se rodar a simulação paralela em sistemas computacionais que disponham de uma quantidade maior de cores. Também, pretende-se mudar a implementação do código, e trabalhar com a biblioteca OpenMP, onde a divisão do trabalho entre as *threads* fica a cargo da biblioteca e não o programador.

Referências

- Arruda, G., Padoin, E. L., Pilla, L. L., Navaux, P. O. A., and Mehaut, J.-F. (2015). Proposta de balanceamento de carga para redução de migração de processos em ambientes multi-programados. In *XVI Simpósio de Sistemas Computacionais (WSCAD-WIC)*, pages 1–8, Florianópolis, RJ.
- Borges, P. A., Coelho, G., and Buligon, S. (2005). Análise do comportamento da água em solos saturados e não saturados. In *XVIII Congresso Nacional de Matemática Aplicada e Computacional, CNMAC, Santo Amaro, SP*.
- Elmaloglou, S. and Diamantopoulos, E. (2009). Simulation of soil water dynamics under subsurface drip irrigation from line sources. *Agricultural Water Management*, 96(11):1587–1595.
- Kalé, L. V., Bhandarkar, M., and Brunner, R. (1998). Load balancing in parallel molecular dynamics. In *International Symposium on Solving Irregularly Structured Problems in Parallel*, pages 251–261. Springer.
- Männich, M. (2008). Desenvolvimento de soluções analíticas e numéricas da equação de richards. Master's thesis, UFPR.
- Padoin, E. L., DILL, S. L., BORGES, P. A., and BORGES, R. S. (2006). Paralelização de métodos computacionais aplicados à análise das variações do teor de umidade de solos saturados e não saturados. In *VII Workshop em Sistemas Computacionais de Alto Desempenho*, pages 97–104.
- Padoin, E. L., Pilla, L. L., Boito, F. Z., Kassick, R. V., and Navaux, P. O. (2011). Análise do consumo energético do algoritmo de irrigação de solos em arquiteturas heterogêneas. *Conferencia Latino Americana de Computación de Alto Rendimiento, Colima, Mexico. CLCAR*, pages 1–7.