



AVALIAÇÃO DA ESTRATÉGIA DE IMPLEMENTAÇÃO CONCORRENTE DO ALGORITMO DE RANDOM WALKER PARA O MODELO DE POTTS CELULAR

LUZ, Leonardo Lobo da¹; CAVALHEIRO, Gerson Geraldo Homrich¹.

¹Departamento de Informática – IFM/UFPEl
Campus Universitário {loboluz161, gerson.cavalheiro}@gmail.com

1. INTRODUÇÃO

O modelo de *Potts* Celular permite simular a reorganização celular fruto da evolução das interações entre células dentro do ciclo evolutivo de um sistema (GRANER; GLAZIER, 1992). Ele é largamente utilizado em diversos tipos de simulações, tais como o crescimento de tumores, o desenvolvimento dos membros de aves e a interação entre espumas de sabão (CERCATO, 2005).

O modelo de *Potts* Celular é tradicionalmente implementado utilizando a técnica de Monte Carlo, apresentando um grande custo computacional devido ao elevado número de cálculos realizados. Conforme o tipo de experimento a ser gerado e o tamanho do sistema foco, a quantidade de processamento requerida em uma máquina mono-processada pode tornar inviável a simulação de determinados sistemas (CERCATO, 2005).

Alternativas têm sido apresentadas para melhorar o desempenho de execução de simulações baseadas no modelo de *Potts*. Nos trabalhos de Gusatto é apresentado o uso da técnica de *Random Walker* como mecanismo para realizar a evolução deste modelo, de forma a reduzir o tempo de processamento necessário (GUSATTO, 2004; GUSATTO et al, 2005). O ganho de desempenho obtido com a utilização do algoritmo de *Random Walker* paralelo em relação ao algoritmo tradicional de Monte Carlo foi bastante acentuado (CERCATO, 2005; CERCATO et al, 2006).

No entanto, os resultados das simulações apresentados nos trabalhos citados anteriormente ainda não foram validados em relação aos resultados oferecidos pela aplicação das técnicas convencionais de simulação, mais especificamente em relação ao algoritmo de Monte Carlo, utilizado como padrão para executar o modelo de *Potts* (CERCATO, 2005).

O objetivo do presente trabalho é apresentar resultados quantitativos gerados por simulações concorrentes do modelo de *Potts* Celular que permitam validar ou refutar o algoritmo concorrente de *Random Walker*, pela verificação de sua equivalência em termos práticos com a implementação tradicional do algoritmo de

Monte Carlo. Também procura-se explorar o potencial das arquiteturas *multi-core* com o auxílio de um ambiente extremamente propício para a escrita de programas paralelos: a ferramenta OpenMP (CHANDRA et al, 2001).

2. MATERIAL E MÉTODOS

A implementação do modelo de *Potts* realizada pelo algoritmo de Monte Carlo apresenta alto custo computacional, uma vez que o mesmo desperdiça potencial de processamento ao realizar um grande número de sorteios em posições que não estão sujeitas a trocas de rótulos (CERCATO, 2005). O algoritmo de *Random Walker*, por sua vez, concentra a seleção de rótulos às áreas de borda das células, rótulos estes sujeitos a troca de energia, reduzindo o tempo de execução de simulações e, conseqüentemente, melhorando o desempenho. O aspecto a ser verificado é se as propriedades estocásticas da técnica de Monte Carlo são mantidas utilizando uma implementação paralela do algoritmo de *Random Walker*. Esta verificação foi feita pela comparação dos resultados da simulação das implementações seqüencial e paralela do modelo de *Potts* Celular evoluída por Monte Carlo e por *Random Walker*.

Com o objetivo de reduzir o tempo de execução das simulações, foram utilizados mecanismos para explorar de forma eficiente arquiteturas *multi-core*. Cada núcleo corresponde a uma unidade de processamento completa. Para explorar o paralelismo neste tipo de arquitetura, foi utilizada a interface de programação *multithread* OpenMP (CHANDRA et al, 2001) e a linguagem C em ambiente *Linux*.

O *hardware* utilizado para executar as implementações foi um computador *Intel Pentium Core 2 Duo 6320*, com freqüência de *clock* 2.0 GHz e memória RAM de 2 GB. O sistema operacional *GNU-Linux kernel 2.6.22-14* e o compilador *icc (Intel)*, versão 10.1.015, para o suporte a sintaxe OpenMP, serviram de base para a escrita do código das aplicações. Maiores detalhes a respeito das implementações podem ser obtidos em Luz (2008).

3. RESULTADOS E DISCUSSÃO

Foram realizadas, no total, três simulações baseadas no modelo de *Potts* Celular. Em cada uma delas, três algoritmos foram utilizados como parâmetro para a construção dos gráficos: Monte Carlo seqüencial, Monte Carlo paralelo e *Random Walker* paralelo. Na primeira simulação, os algoritmos paralelos utilizaram duas (2) *threads* de serviço. Na segunda simulação, os algoritmos paralelos utilizaram quatro (4) *threads* e na terceira simulação, os algoritmos paralelos de Monte Carlo e *Random Walker* utilizaram oito (8) *threads* de serviço. Outros resultados encontram-se documentados em Luz (2008).

As três simulações evoluíram dentro de um número total de 1100 passos, utilizando o mesmo tamanho de matriz: 2000 x 2000. Importante observar que todas as execuções consideraram a mesma matriz inicial. Os resultados apresentados consideram o número de passos realizados na simulação. Tanto no Monte Carlo quanto no *Random Walker* um passo corresponde ao número de sorteios necessários para que todos os rótulos passíveis de sorteio possam ser escolhidos. No Monte Carlo, o número de sorteios corresponde ao tamanho da matriz. No *Random Walker*, este número corresponde ao número de rótulos em borda de

células – o passo no *Random Walker* é, portanto, computacionalmente menos oneroso que o passo no Monte Carlo.

A primeira simulação, mostrada neste artigo, divide-se em dois tipos de resultados: variação do número de células (bolhas) presentes no sistema e variação da área média das células como um todo. Abaixo, são mostrados dois tipos de gráficos que representam estas duas situações. O primeiro gráfico, mostrado na Fig. 1, mostra a variação do número de células no sistema no decorrer da simulação, de acordo com o número de passos executados. Inicialmente, o número total de células presentes no sistema é de aproximadamente 10850, em todas as três simulações. Os algoritmos de Monte Carlo e *Random Walker* paralelos, representados no gráfico da Fig. 1, foram executados com duas *threads*. Conforme pode ser observado, o número de células nesta primeira simulação decresce conforme o aumento do número de passos ao longo do tempo. Próximo do fim do experimento, as células começam a atingir um determinado ponto de equilíbrio e o sistema torna-se mais estável.

Os algoritmos seqüencial e paralelo de Monte Carlo apresentaram resultados muito semelhantes em praticamente todos os passos de simulação. O algoritmo paralelo de *Random Walker* mostrou resultados bastante próximos dos outros dois.

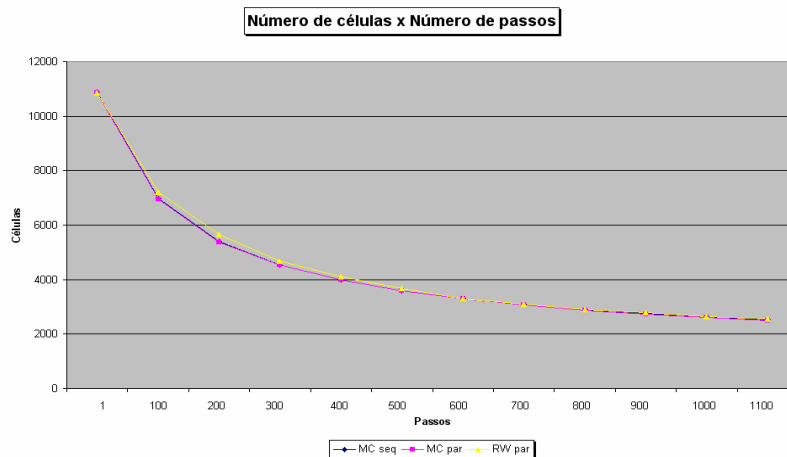


Figura 1 – Gráfico ilustrando a variação do número de células de acordo com o número de passos simulados.

O segundo tipo de resultado, apresentado no gráfico da Fig. 2, mostra a variação do valor da área média das células presentes no sistema ao longo da simulação, de acordo com o número de passos executados. Este valor depende do número de células presentes no sistema em determinado passo de execução e é obtido através da razão entre o número total de elementos presentes na matriz pelo número total de células em cada passo da simulação. Antes da simulação, o valor da área média é de aproximadamente 368 UA (unidades de área).

As implementações seqüencial e paralela de Monte Carlo apresentaram resultados muito semelhantes, como indica o comportamento das curvas mostradas no gráfico. Apesar de diferenciar-se um pouco em alguns passos iniciais da simulação, o algoritmo paralelo de *Random Walker* apresentou resultados finais bastante próximos dos resultados apresentados pelos algoritmos de Monte Carlo.

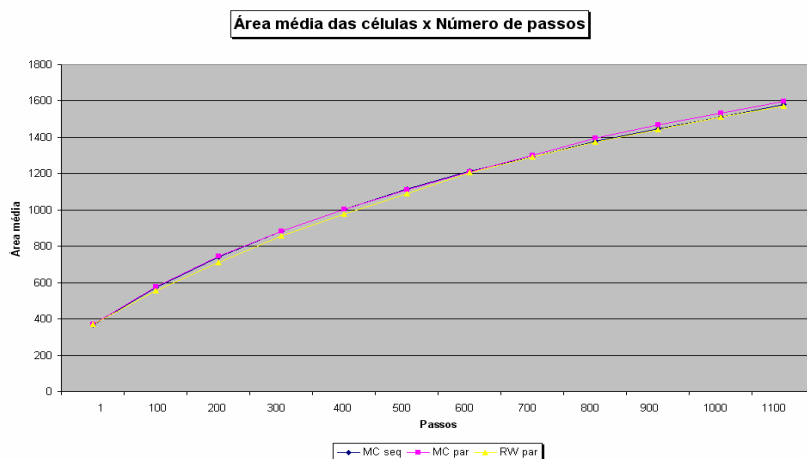


Figura 2 – Gráfico ilustrando a variação do valor numérico da área média das células de acordo com o número de passos simulados.

4. CONCLUSÕES

O presente trabalho apresentou duas implementações paralelas, utilizando o ambiente de programação *multithreaded* OpenMP, dos algoritmos de Monte Carlo e *Random Walker*, juntamente com um comparativo das simulações apresentadas pela execução destes dois algoritmos. Foi possível observar que os resultados obtidos pelas simulações de *Potts* Celular utilizando a técnica de Monte Carlo e o algoritmo de *Random Walker* foram bastante próximos. O ganho na utilização do algoritmo de *Random Walker* é a obtenção de resultados em um tempo menor de processamento.

Entretanto, faz-se necessário realizar uma análise criteriosa dos resultados quantitativos apresentados, de forma a validar o algoritmo de *Random Walker*. Tal avaliação deve ser feita, preferencialmente, por profissionais ligados às áreas de Bioinformática, Física e Biologia, com maior legitimidade em compreender os processos envolvidos na evolução das simulações.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- CERCATO, Fernando Piccine. **Um algoritmo de alto desempenho para evoluir o modelo de Potts Celular**. São Leopoldo, 2005. 83 p. Dissertação (Mestrado em Ciência da Computação). Universidade do Vale do Rio dos Sinos, 2005.
- CERCATO, F. P., MOMBACH, J. C. M., CAVALHEIRO, G. G. H. **High performance simulations of the cellular Potts model**. In: HPCS 2006 – International Symposium on High Performance Computing Systems and Applications, 2006, Saint Johns. XX International Symposium on High Performance Computing Systems and Applications. Los Alamitos: IEEE Computer Society, 2006.
- CHANDRA, R., DAGUM, L., KOHR, D., MAYDAN, D., McDONALD, J., MENON, R. **Parallel Programming in OpenMP**. Morgan Kaufmann, San Francisco, 2001.
- GRANER, F., GLAZIER, J. A. **Simulation of biological cell sorting using a twodimensional extended potts model**. *Physical Review Letters*, v. 1, n. 69, p. 2013 – 2016, 1992.
- GUSATTO, Éder. **Uma nova abordagem na implementação do modelo de Potts Celular buscando eficiência e explorando paralelismo**. São Leopoldo, 2004.

Monografia (Graduação em Ciência da Computação). Universidade do Vale do Rio dos Sinos, 2004.

GUSATTO, E., MOMBACH, J. C. M., CERCATO, F. P., CAVALHEIRO, G. G. H. **An efficient parallel algorithm to evolve simulations of the cellular Potts model.** *Parallel processing letters*. V. 1, p. 199 – 208, 2005.

LUZ, Leonardo Lobo da. **Avaliação da estratégia de implementação concorrente do algoritmo de Random Walker para o modelo de Potts Celular.** Pelotas, 2008. 85f. Trabalho acadêmico (Graduação em Ciência da Computação) – Instituto de Física e Matemática, Universidade Federal de Pelotas.