Avaliação da Estratégia de Implementação Concorrente do Algoritmo de Random Walker para o Modelo de Potts Celular

Leonardo Lobo da Luz, Gerson Geraldo H. Cavalheiro

Instituto de Informática – Universidade Federal de Pelotas (UFPEL)
Caixa Postal 354 – 96.010-900 – Pelotas – RS – Brasil

loboluz161@gmail.com, gersonc@anahy.org

Resumo. *Este artigo apresenta a utilização de dois algoritmos bastante difundidos na área de simulação de sistemas complexos: Monte Carlo e Random Walker. O objetivo do trabalho é aplicar ambos a um modelo da Física, denominado Potts Celular, que tem como finalidade simular a interação entre partículas microscópicas, e apresentar um comparativo entre os resultados quantitativos das simulações geradas. As execuções dos algoritmos se dão de forma concorrente com o auxílio da ferramenta de programação paralela OpenMP.*

1. Introdução

O modelo de *Potts* Celular pertence ao campo da Biologia e tem por finalidade simular a reorganização celular. Atualmente, ele vem sendo largamente utilizado para realizar diversos tipos de simulações, dentre os quais a evolução de células cancerígenas. Com relação a algoritmos para simular este modelo, o de Monte Carlo é um dos mais utilizados e mais aceitos pela comunidade acadêmica e científica, pois seus resultados são bastante próximos daqueles esperados. Porém, ele é bastante ineficiente em termos computacionais [Cercato 2005].

O algoritmo de *Random Walker* [Gusatto 2004] apresenta-se como uma alternativa viável e eficiente para solucionar este problema. O ganho de desempenho obtido com a execução paralela foi bastante acentuado, como pode ser observado nos trabalhos de Cercato (2005) e Gusatto (2004).

No entanto, os resultados das simulações geradas pelo algoritmo de *Random Walker* 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 de simulações geradas por esses dois algoritmos. 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].

O restante deste artigo está organizado como segue. A seção 2 apresenta o modelo de *Potts* Celular. A seção 3 apresenta os algoritmos de Monte Carlo e *Random Walker*. A seção 4 descreve o ambiente para programação paralela em *OpenMP* e a seção 5 analisa os resultados quantitativos gerados pelas simulações dos dois algoritmos. Por fim, a seção 6 conclui o trabalho.

2. O Modelo de Potts Celular

O modelo de *Potts* Celular é um modelo computacional desenvolvido por Glazier e Graner com a finalidade de simular a reorganização celular fruto da evolução das interações entre células no ciclo evolutivo de um sistema [Graner and Glazier 1992]. Este modelo é largamente utilizado em diversos tipos de simulações, dentre eles o crescimento de tumores, desenvolvimento dos membros de aves e a interação entre espumas de sabão. Entretanto, em sua forma tradicional, o modelo de *Potts* apresenta um custo computacional bastante elevado [Cercato 2005].

3. Os Algoritmos de Monte Carlo e Random Walker

O algoritmo de Monte Carlo é um método numérico utilizado para encontrar soluções para problemas matemáticos, os quais podem ter muitas variáveis, e que não podem ser facilmente resolvidos através de outros métodos numéricos conhecidos. Devido ao seu grau probabilístico, serve de auxílio a diversas simulações de fenômenos naturais. O algoritmo de Monte Carlo, quando aplicado ao modelo de *Potts* Celular, permite selecionar células de um sistema para fazê-lo evoluir. Um passo de simulação através deste algoritmo corresponde ao sorteio de tantas células quantas houver no sistema, caracterizando, desta forma, seu custo computacional [Newman and Barkema 1999].

O algoritmo de *Random Walker*, por sua vez, concentra a seleção de rótulos às áreas de borda das células, rótulos estes sujeitos a trocas de energia. Tal fato reduz o tempo de execução de simulações e, conseqüentemente, melhora o desempenho [Gusatto 2004]. Esta característica representa uma vantagem significativa sobre o Monte Carlo.

4. Ambiente de Execução

Para explorar o paralelismo das aplicações, foi utilizada a interface de programação *multithread OpenMP* [Chandra et al 2001], juntamente com a linguagem C, em ambiente *GNU/Linux*. O *hardware* utilizado foi um computador *Intel Pentium Core* 2 *Duo* 6320, com freqüência de *clock* 2.0 GHz e memória RAM de 2 GB. O compilador icc da *Intel*, para o suporte a sintaxe *OpenMP*, serviu de base para a compilação dos códigos-fonte.

5. Análise e Discussão dos Resultados

Foram utilizados três algoritmos para a simulação do modelo de *Potts* Celular: Monte Carlo seqüencial, Monte Carlo paralelo e *Random Walker* paralelo. Na simulação, os algoritmos paralelos utilizaram duas (2) *threads* de serviço. Resultados mais detalhados encontram-se documentados em Luz (2008).

Os três algoritmos 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, representando unidades de tempo. 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.

Os algoritmos visam simular estruturas celulares representadas em um espaço discreto. Uma célula é constituída por um grupo de rótulos (posições de uma matriz) espacialmente conexos e de mesmo valor numérico. Como o sistema apresenta muitas células, e todas elas aumentam ou diminuem de tamanho ao longo da simulação, o registro deste tipo de informação torna-se um fator importante para a determinação do resultado de uma simulação. Os gráficos a seguir ilustram o número de células e a área média das mesmas em uma determinada unidade de tempo.

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 10800. Como pode ser observado, o número de células na simulação decresce conforme aumenta o 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.

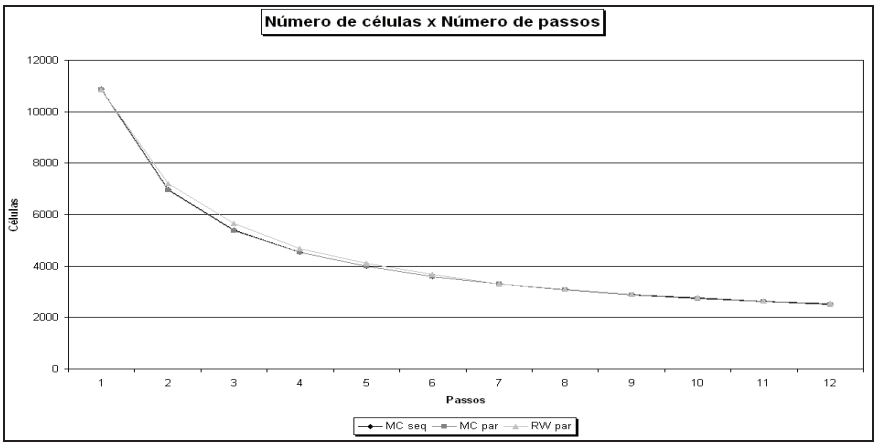

Figura 1. Gráfico ilustrando a variação do número de células de acordo com o número de passos simulados.

Como pode ser constatado na figura, os três algoritmos apresentaram resultados muito semelhantes em praticamente todos os passos de simulação.

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). Conforme pode ser observado, os três algoritmos apresentaram resultados muito semelhantes, como indica o comportamento das curvas mostradas no gráfico.

Tal semelhança entre os resultados apresentados nas duas figuras sugere que o algoritmo de *Random Walker* é tão confiável quanto o de Monte Carlo para simulações de sistemas complexos.

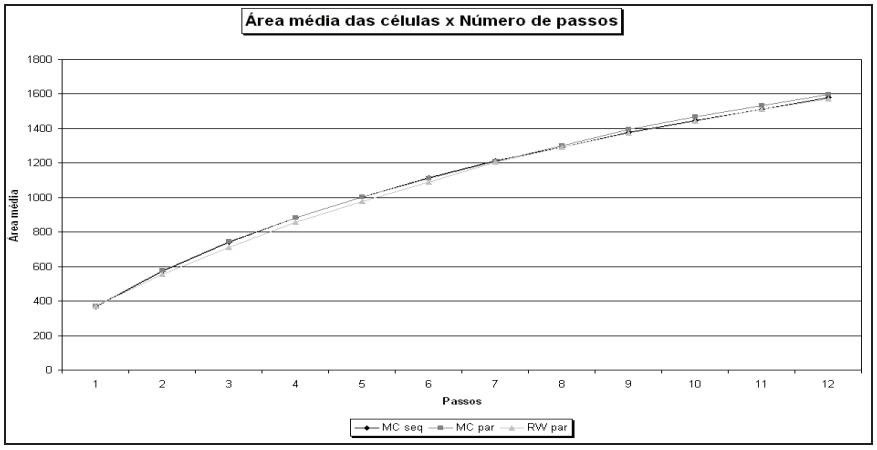

Figura 2. Gráfico ilustrando a variação do valor numérico da área média das células de acordo com o número de passos simulados.

6. Conclusões

Este trabalho apresentou um comparativo entre três implementações (duas paralelas e uma seqüencial) dos algoritmos de Monte Carlo e *Random Walker*, utilizando o ambiente de programação *multithreaded* OpenMP. Foi possível observar que os resultados obtidos pelas simulações foram bastante próximos.

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.

Referências

Cercato, F. P. (2005) "Um Algoritmo de Alto Desempenho para Evoluir o Modelo de Potts Celular". Dissertação de Mestrado. São Leopoldo: UNISINOS.

Chandra, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J., and Menon, R. (2001) "Parallel Programming in OpenMP". Morgan Kaufmann, San Francisco.

Graner, F.; Glazier, J. A. (1992) "Simulation of Biological Cell Sorting Using a Twodimensional Extended Potts Model". *Physical Review Letters*, v. 1, n. 69, p. 2013-2016.

Gusatto, E. (2004) "Uma Nova Abordagem na Implementação do Modelo de Potts Celular Buscando Eficiência e Explorando Paralelismo". Monografia (Graduação). São Leopoldo: UNISINOS.

Luz, L. L. (2008) "Avaliação da Estratégia de Implementação Concorrente do Algoritmo de Random Walker para o Modelo de Potts Celular". Monografia (Graduação). Pelotas: UFPEL.

Newman, M. E. J.; Barkema, G. T. (1999) "Monte Carlo Methods in Statistical Physics". 1ª ed.: Oxford University Press.