Desempenho do Modelo de Ising Bidimensional em GPU e CPU

Wagner de Lima[1], **David Beserra**[2], **Alberto Araujo**[1]

[1]Universidade Federal Rural de Pernambuco - Unidade Acadêmica de Garanhuns
Avenida Bom Pastor - 55292-270, s/n - Boa Vista, Garanhuns - PE - Brasil

[2]Universidade Federal de Sergipe
Avenida Marechal Rondon, S/n - Jardim Rosa Elze, São Cristóvão - SE, 49100-000

{waglds, dw.beserra, albertoaepa}@gmail.com

Resumo. *Este trabalho utiliza implementa o Modelo de Ising através do algoritmo de Metropolis e compara o tempo de execução de duas versões, uma serial e outra paralela usando* CPU *e* GPU*, respectivamente. O* Speed-up *final da versão paralela foi de 22x em relação a serial. Este trabalho também preocupa-se em apresentar o gráfico da temperatura de transição.*

1. Introdução

De acordo com [Ising 1925], existe um modelo na física estatística que fornece uma descrição microscópica do ferromagnetismo, chamado de Modelo de Ising. Este modelo foi introduzido de forma a explicar a transição entre as fases ferromagnética e paraferromegnética. O Modelo de *Ising* bidimensional foi resolvido por Lars Onsager in 1944 [Onsager 1944] e é bem conhecido no campo da física estatística – assim como em áreas da computação (e.g. desempenho, simulações computacionais e modelos estocásticos) – tendo uma temperatura de trasição de fases $T_c \approx 2.269185°C$ [Onsager 1944]. De forma a validar cálculos matemáticos aplicados ao modelo, pesquisadores têm utilizado todo o poder computacional disponível, de modo a aproximarem-se de uma possível solução – dado que para uma solução ser considerada, é necessário empregar matrizes de spins computacionais de dimensões não convencionais, e.g 4096^2 (4096 linhas por 4096 colunas). Além do mais, uma Unidade Central de Processamento (CPU) não possui poder computacional suficiente para desempenhar critérios matemáticos baseadas em [Metropolis et al. 1953] sobre a matriz de *spins* em tempo hábil.

Desta maneira, utilizando-se uma *API* de processamento gráfico paralela (como *CUDA* ou *OpenCL*), junto com um processador gráfico, pode-se reduzir o tempo de processamento da simulação computacional escolhida e validar os atributos do Modelo de *Ising*.

2. Objetivos do Trabalho

Em 2006, a *NVIDIA* notou que além de disponibilizar placas gráficas para a indústria de games, poderia também direcionar *GPUs* à computação científica. Assim, os objetivos deste trabalho são: comparar o desempenho computacional entre duas simulações (em *CPU* e *GPU*) do algoritmo de *Metropolis* [Metropolis et al. 1953] aplicado ao modelo de *Ising*; e mostrar o speedup da aplicação desenvolvida em *GPU*.

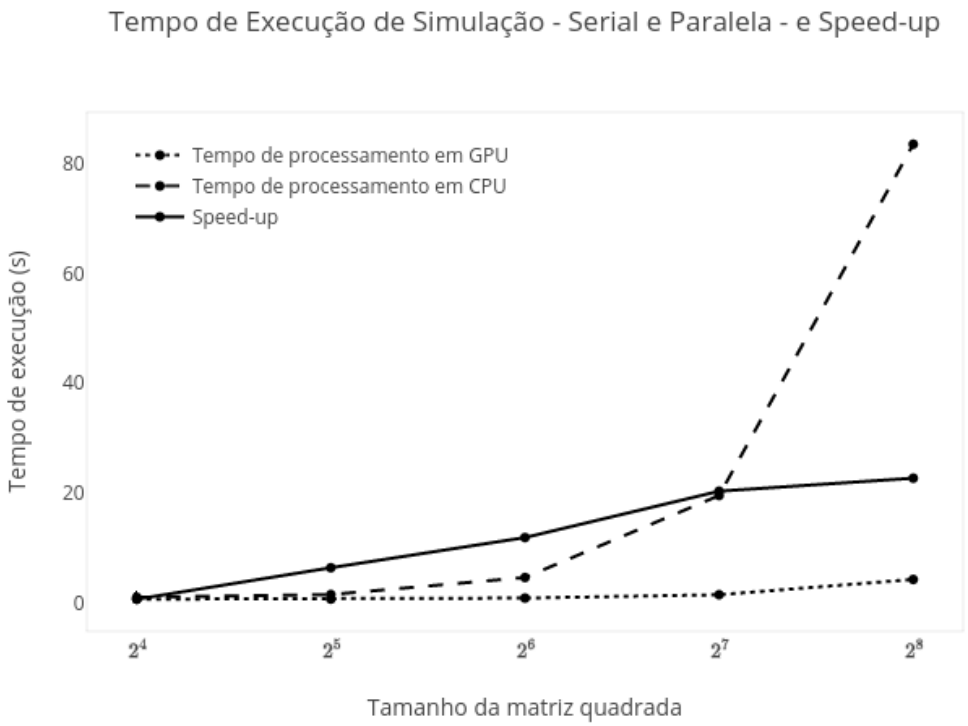

Figura 1. Tempo de processamento total das simulações. As linhas tracejada e pontilhada estão para as simulações serial e paralela, respectivamente.

3. Material e Métodos

Durante a implementação foi utilizando a linguagem de programação *C* para o desenvolvimento do código serial; e *C+CUDA* para o código paralelo. *CUDA (Compute Unified Device Architecture)* é uma interface de programação, que ao ser combinada com *C*, oferece recursos de computaçao de alto desempenho. Assim, para ser utilizada (CUDA) é necessário uma placa gŕafica NVIDIA instalada no computador em uso. Para questões de comparação de desempenho, a simulação serial foi executada num processador *Intel Core i7*, com 8 núcleos de *2.40GHz* de frequência; e *8GB (gigabytes)* de memória *RAM*. A simulação paralela foi executada em 1 *GPU* do modelo *GeForce GT 740M*, com *2GB* de memória e 384 núcleos de processamento. A matriz de spins foi particionada de maneira bidimensional, em quatro segmentos, como mostra [Preis et al. 2009]. Os segmentos de matrizes foram designados a blocos quadrados de *threads* paralelas (com 32 linhas por 32 colunas) que constituem *grids* computacionais quadrados de 16 linhas por 16 colunas. Assim, o problema que uma vez fora dividido, será reagrupado após seu processamento. Na simulação serial, a matriz de spins foi processada por inteiro, já que não havia necessidade de divisão, devido ao fato de não haver paralelismo.

4. Resultados e Discussão

De acordo com [Harvey and Fabritiis 2009], uma GPU possui precisão dupla muito comparável à dupla precisão de uma CPU, para uma larga gama de aplicações. Assim, fazer uso de GPUs é uma forma de resolver problemas computacionais (ou de outras áreas

da ciência) complexos com menor tempo e custo. Deste modo, comparações de desempenho de simulações Ising em CPU e GPU têm sido desenvolvidas utilizando diferentes algoritmos e métodos almejando encontrar a melhor forma de se chegar à uma aplicação mais rápida. A Figura 1 mostra a diferença no tempo de processamento para as simulações serial e paralela e o *speed-up*, em redes de spins que variam de 16^2 até 256^2. Na implementação serial houve um aumento exponencial de tempo, enquanto que na paralela o tempo de processamento permaneceu quase constante. O *speed-up* da aplicação cresce a medida que a rede de spins aumenta. Desta maneira, o speedup final da simulação paralela obtido foi de $22,20x$ em relação a serial, como mostra a Tabela 1 (tomada a maior rede de spins utilizada, 256^2, já que o tempo de execução permanece quase inalterado). A Figura 2 exibe o estado de magnetização para os diversos tamanhos da matriz quadrada após passar por uma temperatura crítica Tc. Percebe-se que quanto maior a matriz de spins, maior a semelhança com o modelo original descrito por [Ising 1925], o que comprova o impacto deste trabalho. Ambos os gráficos podem ser vistos de forma interativa na página web do autor[1].

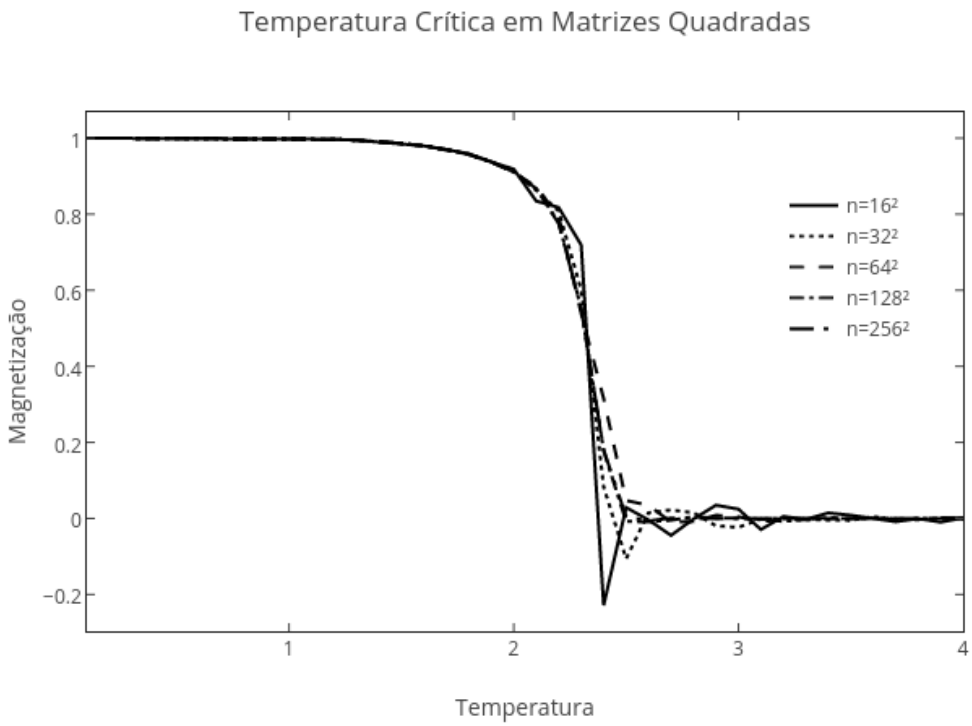

Figura 2. Temperatura crítica em redes de spins quadradas. Após passar pela temperatura crítica, a magnetização tende a zero.

5. Conclusão

Este artigo apresentou uma versão bidimensional acelerada do Modelo de Ising. A computação em GPU foi utilizada como uma nova plataforma de desenvolvimento voltada para aplicações paralelas. A interface de programação CUDA que foi utilizada,

[1]Os endereços para visualização dos gráficos são: https://goo.gl/LzJcMM e https://goo.gl/1GMmTz

Tabela 1. *Speedup* e tempo de execução (em segundos) por redes de *spins*.

	Tempo de execução (em segundos)		
Tamanho da rede	CPU	GPU	Speed-up
2^4	0.5	0,15	0,15
2^5	1,0	0,16	5,88
2^6	4,1	0,35	11,38
2^7	19,04	0,96	19,83
2^8	83,05	3,74	20,22

permite implementar algoritmos paralelos usando extensões da linguagem C. Com a implementação do algoritmo bidimensional do Modelo de Ising, foram obtidos resultados com *speed-up* de $22x$ mais rápidos em GPU quando comparados com os alcançados em CPU. Para trabalhos futuros, serão considerados variações do Modelo de Ising a serem implementadas por diferentes algoritmos de cluesterização.

Referências

Harvey, M. J. and Fabritiis, G. D. (2009). An implementation of the smooth particle mesh ewald method on gpu hardware. *Journal of Chemical Theory and Computation*, 5(9):2371–2377.

Ising, E. (1925). Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6).

Onsager, L. (1944). Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Phys. Rev.*, 65:117–149.

Preis, T., Virnau, P., Paul, W., and Schneider, J. J. (2009). {GPU} accelerated monte carlo simulation of the 2d and 3d ising model. *Journal of Computational Physics*, 228(12):4468 – 4477.