

# Análise Teórica da Performance de um Algoritmo para Dinâmica Molecular

Juliana Zacharias Paukowski<sup>1</sup>, Claudio Schepke<sup>2</sup>, Nicolas Maillard<sup>2</sup>

<sup>1</sup>Instituto de Física – Universidade Federal do Rio Grande do Sul (UFRGS)  
Caixa Postal 15.051 – CEP 91.501-970 – Porto Alegre – RS – Brasil

<sup>2</sup>Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)  
Caixa Postal 15.064 – CEP 91.501-970 – Porto Alegre – RS – Brazil

jupaukowski@if.ufrgs.br, {cschepke,nicolas}@inf.ufrgs.br

**Resumo.** A modelagem numérica muitas vezes requer um grande número de operações a serem executadas, tornando o tempo de computação grande demais para ser viável. Neste trabalho, analisa-se teoricamente uma implementação paralela para um algoritmo de Dinâmica Molecular, mostrando que o tempo de comunicação entre os processadores (custo inevitável da implementação) é cerca de duas ordens de grandeza menor que o tempo de processamento da simulação entre comunicações, o que torna a paralelização do algoritmo uma solução atrativa.

## 1. Introdução

Modelos científicos utilizam a matemática para descrever sistemas e processos existentes na natureza. Porém, nem todos os modelos são analíticos, necessitando de uma abordagem numérica para obter os resultados desejados. Entre esses, vários envolvem muitas operações, tornando o tempo de simulação grande demais para ser viável. Uma solução para isso é a implementação paralela do algoritmo, que pode diminuir o tempo computacional e tornar aceitável a simulação proposta. Uma análise de performance e custo de comunicação dá a base para começar a implementação do algoritmo com ideia do *speed-up* a ser atingido.

### 1.1. Motivação e Modelo

A Dinâmica Molecular é muito utilizada na modelagem de sistemas físicos, químicos e biológicos, tendo como princípio o cálculo das leis de Newton para cada partícula, as quais interagem através de um potencial (ou campo de forças). O objetivo é utilizar mecânica estatística para relacionar o comportamento das partículas que constituem o sistema com propriedades termodinâmicas macroscópicas do sistema em equilíbrio.

Como exemplo, é considerado um sistema de  $N$  partículas de raio  $r_p \sim 10^{-10}m$  inicialmente a uma velocidade distribuída aleatoriamente de acordo com a temperatura atribuída inicialmente ( $\frac{3}{2}k_bT = \frac{1}{2}m \langle v^2 \rangle$ ), colocadas em um cubo de volume  $V$  ( $L^3$ ), sendo definida uma densidade média  $\rho = \frac{N}{V}$  regulável de acordo com o volume. As partículas interagem por um potencial efetivo esfericamente simétrico  $U(r)$  de curto alcance, para fins de simplificação. A força que age sobre a partícula  $i$  pela interação com a partícula  $j$  é calculada de acordo com a equação

$$\vec{F}_{ij}(r_{ij}) = -\nabla U(r_{ij}) \quad (1)$$

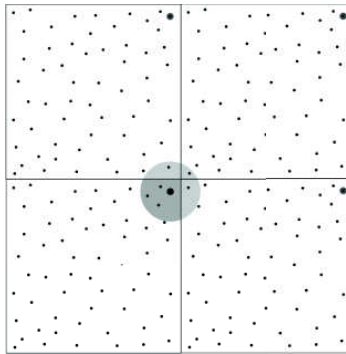
onde  $r_{ij}$  é a posição do centro da partícula **j** tendo o centro da partícula **i** como origem. Calcula-se, então, a Segunda Lei de Newton na forma

$$m_i \frac{d\vec{v}_i}{dt} = \sum_j^N \vec{F}_{ij}(r) \quad (2)$$

$$\frac{d\vec{r}_i}{dt} = \vec{v}_i \quad (3)$$

usando algum método de integração numérica - geralmente Verlet [Verlet 1967] ou *leapfrog* - para obter o incremento na posição da partícula **i**. O peso computacional da simulação está no número de partículas (de  $10^3$  a  $10^6$  em simulações médias), passos de tempo muito curtos ( $\Delta t \sim 10^{-15}$  s) e a necessidade de muitos passos de tempo para atingir o equilíbrio termodinâmico (por volta de  $10^8$  passos, dependendo do sistema). Para diminuir o número de equações, é levado em conta a forma do potencial para justificar o cálculo das forças de interação da partícula **i** apenas em relação às partículas que estejam a uma distância menor que um raio de corte ( $r_c$ ), de duas a sete vezes maior que o raio da partícula, visto que além dessa distância a interação é praticamente nula.

São aplicadas condições de contorno periódicas - como se a caixa fosse replicada repetidamente e partículas próximas à borda interagissem com partículas na borda oposta. As partículas interagentes estão representadas de maneira simplificada na Figura 1.



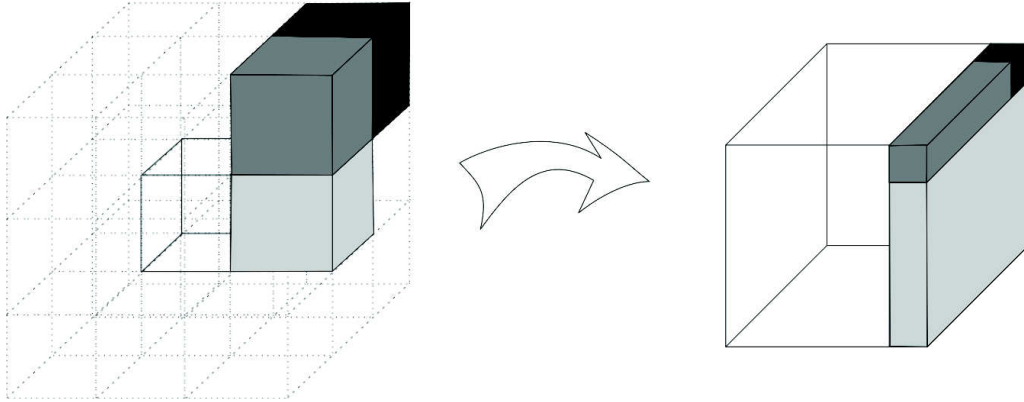
**Figura 1. Representação bidimensional do mapeamento do raio de corte para uma partícula perto do contorno da caixa.**

Para paralelizar numa arquitetura com  $P$  processadores, divide-se o volume  $V$  em  $P$  partes cúbicas - cada parte atribuída a um processador diferente. Cada cubo menor (que agora terá lado  $l$ ) deverá receber e enviar dados referentes à posição de partículas contidas em seus 26 cubos vizinhos (Figura 2).

Na média:

- 6 cubos recebem  $\rho(r_c l^2)$  partículas (faces vizinhas);
- 12 cubos recebem  $\rho(r_c^2 l)$  partículas (arestas vizinhas);
- 8 cubos recebem  $\rho(r_c^3)$  partículas (vértices vizinhos).

A lista de vizinhos das partículas é atualizada usualmente a cada 10-20 passos de tempo, assim como a lista das posições das partículas dentro do volume considerado na vizinhança dos cubos menores - o que requer comunicação entre os processadores.



**Figura 2.** À esquerda, tomando o cubo central transparente como referência, exemplos de cubos vizinhos por face (cinza claro), aresta (cinza escuro) e vértice (preto). À direita, o volume ocupado pelas partículas que têm sua posição compartilhada com cada tipo de vizinho

## 2. Análise Teórica de Custo do Processamento e Comunicação

Para a análise teórica do custo de comunicação foi adotado o modelo **LogP** [Culler et al. 1993], que trata explicitamente do desempenho da rede e é mais realista ao tratar de sistemas com memória distribuída que outros modelos. Este modelo tem como parâmetros a latência de comunicação ( $L$ ), o sobrecusto de envio e recebimento das mensagens (do inglês *overhead* -  $o$ ), o intervalo mínimo entre dois envios ou dois recebimentos de mensagens (do inglês *gap* -  $g$ ) - equivalente ao inverso da vazão da rede - e o número de processadores ( $P$ ). Por ser pouco significativo, o *overhead* foi ignorado. O tempo gasto para o envio de uma mensagem de tamanho  $N$  é dado por:

$$\tau = L + Ng \quad (4)$$

O custo dependerá do número de partículas a ser enviado - apenas a posição das partículas, três dimensões com 32 bits cada (precisão simples). Assim, o custo para envio de dados entre cubos vizinhos será:

$$\tau_{faces} = 6L + 6g(\rho r_c l^2) \cdot 96 \quad (5)$$

$$\tau_{arestas} = 12L + 12g(\rho r_c^2 l) \cdot 96 \quad (6)$$

$$\tau_{vértices} = 8L + 8g(\rho r_c^3) \cdot 96 \quad (7)$$

O tempo total é dado pela soma dos custos de envio entre cubos vizinhos:

$$\tau_{total} = 26L + g\rho[6(r_c l^2) + 12(r_c^2 l) + 8(r_c^3)] \cdot 96 \quad (8)$$

Para dar uma ideia da ordem de grandeza do tempo de performance, utilizou-se os dados de uma implementação sequencial de Dinâmica Molecular [Plimpton 1995], executado em um Cray C90. Plimpton usou, em unidades reduzidas escritas em função de um parâmetro do potencial ( $\sigma$ ),  $\rho^* = 0,8442$ ,  $T^* = 0,72$ ,  $r_c = 2,5\sigma$ . Para um sistema com 500.000 partículas cada passo de tempo demorou 2,86 segundos para ser processado.

Como a performance de pico do Cray C90 é 952 MFLOPs e computadores atuais como o Intel Core i7 têm sua performance de pico em torno de 100 GFLOPs, espera-se uma diferença de 2 ordens no tempo de execução, estimando um tempo de  $10^{-2}$  segundos por passo de tempo -  $10^6$  segundos para uma simulação média com  $10^8$  passos no total.

Foi usada para latência o tempo de  $42\mu s$  e para a vazão (inverso do *gap*) o valor de  $941.4Mb/s$ , medidos em clusters existentes no Instituto de Informática da UFRGS com conexão *Gigabit*. Para 64 processadores ( $l = 21$  nas unidades reduzidas), obtém-se

$$\begin{aligned}\tau_{total} &= 26 \cdot 42 \cdot 10^{-6} + \frac{0,8442 \cdot 96}{941.4 \cdot 10^6} \cdot [6 \cdot (2,5 \cdot 21^2) + 12 \cdot (2,5^2 \cdot 21) + 8 \cdot (2,5^3)] \\ \tau_{total} &= 1.1ms,\end{aligned}$$

uma ordem de grandeza menor que o tempo de processamento de um passo de tempo. Como a comunicação acontece apenas a cada 20 passos de tempo, a diferença sobe para duas ordens de grandeza.

### 3. Conclusão e Perspectivas

Neste trabalho analisou-se teoricamente o custo de comunicação para uma paralelização de um algoritmo de Dinâmica Molecular e comparou-se com uma estimativa do tempo de processamento para cada iteração, encontrando uma diferença de duas ordens de grandeza a menos no tempo de comunicação. Isso torna a implementação paralela uma boa solução para diminuir o tempo de simulação desse tipo de problema.

Com as estimativas do tempo de processamento extrapoladas a partir do tempo medido em um Cray C90 pode-se afirmar apenas sobre as ordens de grandeza. Para melhor aproximação do desempenho e ganho real, novas medidas em uma arquitetura atual serão feitas.

Também poderiam ser levados em conta efeitos de congestionamento de rede, armazenamento temporário de mensagens (como no protocolo TCP), quebra de mensagens grandes, máquinas heterogêneas e aqueles provenientes da topologia da rede, visto que tais parâmetros não são considerados no modelo LogP.

### Referências

- Culler, D., Karp, R., Patterson, D., Sahay, A., Schauser, K. E., Santos, E., Subramonian, R., and von Eicken, T. (1993). Logp: Towards a realistic model of parallel computation. *ACM SIGPLAN symposium on Principles and practice of parallel programming - PPOPP*, pages 1–12.
- Plimpton, S. (1995). Fast parallel algorithms for short range molecular dynamics. *Journal of Computational Physics*, 117:1–19.
- Verlet, L. (1967). Computer experiments on classical fluids. *Physical Review*, 159(1):100–103.