Paralelização do Algoritmo de Geração de Cadeias Aleatórias Contínuas de Carbono *

Gustavo Romano, Tiarajú A. Diverio, Nicolas Maillard

Instituto de Informárica - Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 - CEP 91.501-970 - Porto Alegre - RS
{gromano, diverio, nicolas}@inf.ufrgs.br

Resumo

Materiais amorfos têm sido intensivamente estudados devido a sua importância tecnológica. O método padrão para esse estudo é a Dinâmica Molecular (DM), que, embora seja bastante usado, é inerentemente pouco eficiente para materiais formados por cadeias fortes. Uma alternativa a esse método é o uso de cadeias aleatórias contínuas (CRN - do inglês *Continuous Random Network*). Essas cadeias têm sido utilizadas com sucesso para a descrição de semicondutores amorfos desde meados da década de 1930 [BARKEMA; MOUSSEAU(2000)].

O algoritmo mais popular para geração de CRN é o *Wooten Winer Weaire*, mais conhecido como WWW. Porém, embora o algoritmo WWW gere bons modelos de átomos, ele funciona apenas para modelos em que todos os átomos tenham hibridização sp^3. Na pesquisa de materiais de elevada dureza, os quais desempenham um importante papel na indústria moderna, o Instituto de Física da Universidade Federal do Rio Grande do Sul (UFRGS) propôs a geração de cadeias aleatórias contínuas formadas de átomos de carbono com hibridização mista sp^3/sp^2. Estas cadeias podem propor novos materiais (real ou hipotético) cuja dureza e módulo volumétrico sejam significativamente superiores aos do diamante. Embora sejam de grande importância, materiais desse tipo ainda não foram propostos pela literatura científica [PEROTTONI; JORNADA(2001)].

O algoritmo proposto para gerar essas cadeias consiste em posicionar de forma aleatória, em uma região do espaço, uma certa quantidade de átomos com hibridização sp^3 e sp^2. Após isso, através do método de *Simulated Annealing*, altera-se aleatoriamente as posições dos átomos até convergir em um sistema estável que atenda as restrições impostas. Nota-se que nesse processo, alguns átomos podem se deslocar além da célula de processamento sendo necessária uma comunicação entre processos.

A cada mudança de posição, cálculos locais são realizados para verificar se a nova posição do átomo é aceitável ou não. Para realizar esse cálculo, leva-se em conta a distância, o ângulo e o número de ligações que esse deslocamento criou. Se o movimento não for aceito, o átomo volta a sua posição anterior.

O sistema atual foi desenvolvido em linguagem C para funcionar em uma única máquina monoprocessada. Atualmente uma simulação com um universo de 128 átomos demora em média uma semana para ser realizada. Para um bom resultado e maiores chances de encontrar algum material com as propriedades desejadas, novas simulações devem ser realizadas com um universo superior a 1000 átomos. Porém, com o sistema

*Trabalho financiado pelo CNPq

atual esse tipo de simulação é inviável, pois seriam necessários meses para se obter um resultado, o qual poderia ainda não ser o desejado.

Proposta de Paralelização

A proposta do trabalho consiste em buscar formas de minimizar o tempo de execução dessas tarefas. A primeira abordagem será encontrar maneiras de otimizar essa implementação seqüencial. Em um segundo momento serão buscadas formas de paralelizar a aplicação tentando diminuir ainda mais o tempo de execução.

A primeira etapa de paralelização consistirá em adicionar primitivas OpenMP [CHANDRA et al.(2000)] no programa seqüencial para tentar explorar o paralelismo nas máquinas multiprocessadas e em novas máquinas multi-core. Essas primitivas serão introduzidas na etapa de cálculo local, que é a etapa que consome mais tempo de processamento e pode facilmente ser paralalelizada através da primitiva "*parallel for*". Espera-se, com essa alteração, uma diminuição no tempo de execução, porém, como apenas pequenas partes do programa poderão ser paralelizadas e as máquinas disponíveis para simulação são no máximo bi-processadas, acredita-se que o tempo de execução continuará ainda alto, necessitando assim, uma segunda etapa de paralelização.

Essa segunda etapa consistirá em buscar formas de paralelizar a aplicação através do uso da biblioteca de troca de mensagens MPI [PACHECO(1996)] (*Message Passing Interface*) e tem duas fases previstas. Na primeira, será realizada uma paralelização de forma simples, usando primitivas básicas do MPI. Nessa fase as cadeias serão geradas com o auxílio de algoritmos genéticos para tentar convergir mais rapidamente em um estado estável. Na segunda, além da utilização das primitivas básicas, tentar-se-á introduzir primitivas novas, presentes na norma MPI-2 [GROPP; LUSK; THAKUR(1999)]. Uma dessas primitivas é a *MPI_Comm_spawn*, que possibilita a criação dinâmica de novos processos. Esses processos serão criados para permitir uma maior concorrência na geração das cadeias. Com a utilização do MPI, pretende-se obter um bom desempenho, diminuindo significativamente o tempo de execução da aplicação, uma vez que um número bem maior de processadores poderá ser utilizado.

Referências

[BARKEMA; MOUSSEAU(2000)] BARKEMA, G. T.; MOUSSEAU, N. High-quality continuous random networks. **Phys. Rev. B**, [S.l.], v.62, p.4985–4990, 2000.

[CHANDRA et al.(2000)] CHANDRA, R.; MENON, R.; DAGUM, L.; KOHR, D.; MAYDAN, D. **Parallel Programming in OpenMP**. [S.l.]: Morgan Kaufmann, 2000.

[GROPP; LUSK; THAKUR(1999)] GROPP, W.; LUSK, E.; THAKUR, R. **Using MPI-2**. [S.l.]: MIT Press, 1999.

[PACHECO(1996)] PACHECO, P. **Parallel Programming With MPI**. [S.l.]: Morgan Kaufmann, 1996.

[PEROTTONI; JORNADA(2001)] PEROTTONI, C. A.; JORNADA, J. A. H. The carbon analogues of type-I silicon clatrates. **J. Phys.: Condens. Matter**, [S.l.], v.13, p.5981–5998, 2001.