Simulação de Redes Contínuas de Carbono Utilizando a Plataforma BOINC

A. Lorenzatti[1], C. K. Abreu[1], F. H. Jornada[2], G. Romano[3], M. Passos[1], A. L. Martinotto[1], C. A. Perottoni[2]

[1] Departamento de Informática, CCET, UCS, Caxias do Sul, RS, Brasil
[2] Instituto de Física, UFRGS, Porto Alegre, RS, Brasil
[3] Instituto de Informática, UFRGS, Porto Alegre, RS, Brasil

Introdução

Materiais de elevada dureza desempenham um importante papel na indústria moderna. Apesar dos esforços despendidos pela comunidade científica internacional não existem evidências para a existência de algum composto cuja dureza exceda a do carbono diamante. No entanto, ao menos em princípio, é possível que uma rede aleatória contínua (CRN -*Continuous Random Network*) [BAR 2000], formada de átomos de carbono com hibridização mista sp^3/sp^2 seja mais dura que a estrutura do diamante. O estudo experimental é extremamente dificultado pela impossibilidade de se exercer um controle rigoroso sobre vários parâmetros físicos importantes, incluindo o grau de hibridização sp^3/sp^2, a distribuição espacial dos átomos e a conectividade da rede em longo alcance. Esses parâmetros, no entanto podem ser controlados em simulações computacionais, permitindo assim inferir acerca da influência dos mesmos sobre as propriedades mecânicas de CRNs.

Dentro desse contexto, desenvolveu-se um algoritmo capaz de gerar CRNs formadas por átomos com hibridização sp^3/sp^2. O algoritmo proposto consiste em posicionar de forma aleatória no espaço uma certa quantidade de átomos, e através do método de *Simulated Annealing* alterar aleatoriamente as posições dos átomos até convergir em um sistema estável e que atenda as restrições impostas. É digno de nota que, uma simulação com um universo de 128 átomos demora em média uma semana para ser realizada e para que existam maiores chances de encontrar um material que atenda as propriedades desejadas são necessárias novas simulações com um universo superior a 1000 átomos, o que levaria meses de simulação e poderia gerar um resultado não desejado.

Solução Adotada

Para contornar o problema descrito na seção anterior, uma solução é desenvolver uma aplicação distribuída baseada na plataforma BOINC *(Berkeley Open Infrastructure for Network Computing)* [AND 2004]. Embora o BOINC fora criado para que aplicações de larga escala possam utilizar o tempo ocioso de milhões de computadores pessoais distribuídos em todo o mundo, as suas propriedades permitem que seja possível utilizá-lo em escalas menores, como por exemplo, aproveitando o tempo ocioso de recursos computacionais disponíveis em instituições ou departamentos [KAC 2005].

Nesses ambientes fechados, o BOINC pode ser uma alternativa, pois, além de facilitar o desenvolvimento das aplicações através da API BOINC, fornece mecanismos

para o controle de segurança e de tolerância a falhas. Com o BOINC é necessário apenas definir a forma como os dados serão tratados pelo servidor e como serão processados pelo cliente, os demais aspectos relacionados a um sistema são abstraídos pela plataforma [AND 2004].

Uma implementação BOINC baseia-se no particionamento de um problema em uma série de subproblemas menores (chamadas workunits - WUs) que podem ser processadas em diferentes computadores que executam um mesmo programa (no caso o programa para geração das CRNs) [KAC 2005]. Cada WU representa os parâmetros de entrada para a aplicação cliente, desta forma, elas podem conter diferentes parâmetros permitindo a execução de simulações distintas em cada cliente. Essa forma de divisão torna-se atrativa uma vez que pode-se realizar diversas simulações simultaneamente afim de buscar novos materias. Porém, alguns problemas surgem uma vez que o processamento de cada WU continua sendo elevado e, por exemplo, torna-se complicado detectar se um computador falhou ou os dados de entrada ainda não foram processados.

Para contornar esse problema, após parte do processamento dessa CRN, o cliente BOINC retorna essa WU com um resultado parcial, para que seja feito um processamento, a partir do qual, são eliminadas CRNs que dificilmente convergiriam para um resultado satisfatório e as restantes são redistribuídas aos clientes BOINC para a continuação do processamento.

É importante destacar ainda que uma vez que a aplicação CRN utilizará apenas o tempo ocioso dos computadores dos laboratórios de informática da Universidade de Caxias do Sul tem-se a necessidade da utilização de *checkpoints*, pois o processamento da WU poderá ser interrompido pelo usuário por diversas vezes. Além disso, a API BOINC possui funções que oferecem suporte a *checkpoint* bem como suporte gráfico para desenvolvimento de uma proteção de tela (OpenGL) [AND 2004].

Conclusões

Como o BOINC proporciona um alto grau de abstração em relação ao desenvolvimento, permitindo ignorar aspectos que não estão diretamente relacionados a aplicação e por oferecer facilidades que podem ser utilizadas em projetos de menor porte verificamos que o BOINC e sua API proporcionarão maior flexibilidade à aplicação CRN. Flexibilidade essa, presente na facilidade com que se pode dividir um problema em sub-problemas menores, assim como solucioná-los utilizando o tempo ocioso dos computadores presentes nos laboratórios da Universidade de Caxias do Sul.

Referências

[AND 2004] ANDERSON, D. P. BOINC: A System for Public-Resource Computing and Storage. In: IEEE/ACM INTERNATIONAL WORKSHOP ON GRID COMPUTING,, 5., 2004, Pittsburgh, USA. **Anais...** [S.l.: s.n.], 2004.

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

[KAC 2005] KACSUK, P.; PODHORSZKI, N. **Scalable Desktop Grid System**. [S.l.]: European Research Network on Foundations, 2005. Technical Report. (TR-0006).