Estudo exploratório sobre computação de alto desempenho com ênfase em programação paralela

Daniel Centeno Einloft, Dalvan Griebler (co-orientador), Luiz Gustavo Leão Fernandes (orientador)

Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Faculdade de Informática (FACIN), Grupo de Modelagem de Aplicações Paralelas (GMAP)

Resumo

O projeto tem por finalidade estudar computação de alto desempenho com ênfase em programação paralela e posteriormente realizar um estudo de caso em uma arquitetura multi-core. A idéia é contribuir futuramente no desenvolvimento de uma interface de programação paralela, denominada Domain Specific Language for Patterns-Oriented Parallel Programming, que tem como objetivo oferecer ao programador uma interface com mais alto nível de abstração, para diminuir o esforço de programação sem comprometer o desempenho.

Neste contexto, foi realizada uma pesquisa geral sobre a programação paralela, onde foram estudados não só conceitos intrínsecos sobre a programação paralela, como padrões de paralelização, seção crítica, etc, mas também formas de classificação de arquiteturas parelelas. O estudo das arquiteutras foi feito tanto de forma mais genérica, como a taxonomia de Flynn, quanto mais específicas, como multi-core, cluster, grid e GPU.

A aplicação dos conhecimentos mencionados foi colocada em prática através da paralelização do algoritmo de multiplicação de matriz. Neste estudo de caso, foram elaboradas duas formas de paralelizar o algoritmo, ambas utilizando o padrão mestre-escravo: paralelismo dinâmico e estático. Os experimentos foram realizados no computador-servidor do GMAP, avaliando o *Speedup* e a Eficiência usando a biblioteca *Pthread*, de 2 a 8 threads. Além disso, está em andamento o desenvolvimento do mesmo algoritmo usando processos, com a chamada de sistema *fork.*

Com o término dos testes, notou-se que existe um aumento de desempenho considerável com a utilização de threads quando o algoritmo é paralelizado corretamente. Dessa forma, o programa utiliza melhor os recursos computacionais presentes em uma

arquitetura multi-core e a eficiência do algoritmo aumenta. Para o desenvolvimento do algoritmo em Pthreads, foi necessário criar as threads explicitamente e garantir que o mestre esperasse as threads escravas, porém a sincronização de dados foi feita de forma automática, pois compartilham a memória. Já nos processos, também é feita a criação explícita e a espera do mestre, porém a sincronização e compartilhamento de dados é feita de forma manual, aumentando a complexidade do programa. Com as diferentes formas de paralelização de um algoritmo e as dificuldades encontradas no desenvolvimento, nota-se a necessidade de uma interface de mais alto nível para facilitar a programação paralela, tornando o desenvolvimento de novas tecnologias mais rápido e eficiente.