ENGENHARIA AEROESPACIAL E MECÂNICA

MECATRÔNICA

CE-265 PROCESSAMENTO PARALELO

Docente: Jairo Panetta

Discente: Lucas Kriesel Sperotto

Exercício 8 – SUMÁRIO E CRÍTICA DO ARTIGO "STENCIL COMPUTATION OPTIMIZATION AND AUTO-TUNING ON STATE-OF-THE-ART MULTICORE ARCHITECTURES"

O autor explora as mais recentes arquiteturas HPC mostrando suas principais diferenças e influências na aplicação de algoritmos "auto-tuning". O autor argumenta que entender o design de cada arquitetura permite determinar uma melhor utilização do "auto-tuning", bem como uma melhor adaptação da arquitetura para determinados problemas. Outro fator é obter a portabilidade do "auto-tuning" tanto para as atuais arquiteturas, como prever as tendências do desenvolvimento e garantir portabilidade com futuras arquiteturas. Já que a tendência é tornar o software cada vez mais responsável pelo desempenho quando se tem uma grande diversidade de hardware.

O autor também demonstra como se comporta a eficiência dessas arquiteturas em relação ao consumo de energia e ao numero de Flop's atingido, de modo que a definição da eficiência da arquitetura está migrando de "desempenho sustentável" para uma noção de "desempenho sustentável por watt". O que demonstra a preocupação com o custo energético envolvido no processamento.

O comparativo dos desempenhos e peculiaridades de cada arquitetura aplicando algoritmo "auto-tuning" foram testados em aplicações cientificas, no caso a resolução de uma edp em domínio tridimensional. Mostrando as vantagens e desvantagem de novas arquiteturas para aplicação cientifica. Em particular uma estratégia interessante esta na decomposição do domínio em quatro níveis de forma a explorar o paralelismo no nível dos dados.

Os resultados mostram que para algoritmos com paralelismo suficientes, o emprego de um grande número de processadores mais simples oferece maior potencial de desempenho do que um pequeno número de processadores mais complexos. Isto é demonstrado tanto para o desempenho em unidades de Flop's como para o desempenho por Watt.

Entretanto chips que empregam um grande número de núcleos mais simples oferecem um desempenho significativo e vantagens na eficiência energética. Por exemplo, arquiteturas CUDA. Os resultados gerais mostram que o "auto-tuning" é essencial para obter o máximo desempenho de uma grande variedade de arquiteturas, embora o tempo de otimização "auto-tuning" seja longo para algumas arquiteturas.

Concordo com o autor e gostei muito do artigo, considero o desenvolvimento do "auto-tuning" algo extremamente complexo, visto a grande variedade de processadores e suas diferenças. E claro, diante desta variedade, temos que adequar ao máximo o software para que ele obtenha desempenho e portabilidade. O mais interessante é que a arquitetura CUDA teve um bom desempenho por watt, aperar de algumas limitações.

SUMÁRIO E CRÍTICA DO ARTIGO “ROOFLINE: AN INSIGHTFUL VISUAL PERFORMANCE MODEL FOR MULTICORE ARCHITECTURES”

O autor afirma que a diversidade de processadores e também a mudança radical da computação sequencial para a computação paralela (multicore) dificulta de certa forma o trabalho dos programadores e desenvolvedores de compiladores. Diante disso ele propõe que um modelo que ofereça orientações de desempenho e de fácil entendimento é crucial para o auxilio na otimização do trabalho em HPC. Uma alternativa é a “analise pelo estrangulamento”, mostrando fatores relacionados ao gargalo da memória. Interessante neste ponto do texto é o surgimento da “velha” Lei de Amdahl como exemplo dos casos relacionados.

O autor explica que se deve medir o tráfego entre a memoria principal e a memoria cache. Este sim é o principal gargalo, definindo a intensidade operacional para prever a largura de banda de memoria necessária para um núcleo em um computador.

O modelo “Roofline” fornece um limite superior para o desempenho. De forma a ajudar a identificar quais os sistemas seriam bons para determinadas aplicações ou mesmo como alterar o código de forma a extrair maios desempenho com relação ao uso da memória. A intensidade operacional indica que o numero de operações de ponto flutuante por byte transferido da memória, sendo esta uma métrica muito importante para a correta análise de sistemas multicore.

Os gargalos de memória pode ser reduzidos com a reestruturação dos loops de acessos a unidades de memoria. Adequando o tamanho das unidades acessadas por paço do loop para um tamanho que caiba na memória rápida e que se adeque a banda de acesso a memoria lenta, aumentando significativamente a largura de banda de memória.

O autor faz alguns comparativos entre processadores e arquiteturas, demonstrando o teto atingível em cada uma, e principalmente estas peculiaridades nos dizem quanto esforço pode ser desprendido para que tenhamos uma melhor eficiência de algoritmos otimizados para esta arquitetura. Claro que auxilia na própria estratégia de otimização do código. O autor também testa alguns problemas científicos para as arquiteturas apresentadas demonstrando o teto máximo da aplicação e o teto atingido nos testes.

Acredito que a principal contribuição deste trabalho esta em definir o desempenho computacional não em termos de limite da computação, mas sim em defini-lo em limites de acesso a memória. Concordo que o processador não deve ficar ocioso, dessa maneira os programas devem ser adequados a quantidade de memoria rápida existente na maquina. De maneira que o teto máximo da computação é definido pela banda de acesso a memoria lenta, ou mesmo quantos flop’s são executados por ida a memória lenta.

Considero ambos os artigos complementares, os dois demonstram a importância e a dificuldade em se obter considerável desempenho em computação de alto desempenho. Cada artigo leva em conta métricas apropriadas, como tempo de execução, consumo de energia, enfim, o curioso é que os problemas apresentados desde a primeira aula, e mesmo conceitos como o da CM5 aparecem ainda na literatura atual.