

CMP154 – Introdução a Processamento Paralelo e Distribuído
Carlos Eduardo Benevides Bezerra
Projeto de arquitetura, para simulações com o SIMICS

Foco: HPC

Melhorias esperadas: reduzir o tempo de execução de um conjunto de tarefas fortemente dependentes (que se comunicam muito entre si), através da definição de uma hierarquia de memória que beneficie essa comunicação inter-processos.

Inovação: será feita uma comparação de seis arquiteturas diferentes, que são as arquiteturas SPARC multi-core (Niagara) e x86 multi-core (Core 2 e Phenom), cada uma com dois tipos de hierarquia de memórias. Embora não se saiba o quanto é nova essa comparação, os resultados poderão ser uma contribuição ao estado da arte.

Estudo necessário: revisar a estrutura da arquitetura ultrasparc niagara e das arquiteturas core 2 e athlon, de maneira a entender o que será necessário modificar para atingir os objetivos definidos para o projeto. Além disso, será necessário ver como essas arquiteturas podem ser simuladas através do SIMICS, além de como as alterações podem ser feitas no que tange à especificação da hierarquia de memória.

Descrição com etapas:

1. Estudo das arquiteturas utilizadas nos processadores das famílias ultrasparc niagara, core 2 e athlon/phenom.
2. Estudo do simulador SIMICS e execução de pequenos testes, com prototipação rápida, até chegar ao modelo definitivo do projeto;
3. Especificar com detalhes as mudanças arquiteturais a simular no SIMICS:
 - Para cada arquitetura (sparc, core 2 e amd multi-core), serão especificadas, a princípio, duas hierarquias de memória: cache compartilhada e cache privada. Espera-se que, para um conjunto de tarefas muito dependentes, uma cache compartilhada desempenhe melhor.
4. Após a especificação das arquiteturas, implementar um algoritmo ipc-intensive (multiplicação de matrizes, por exemplo)
5. Simular em cada uma das arquiteturas, com diferentes números de threads e verificar o speed-up com cada uma.

Possível aplicação: os resultados obtidos a partir das simulações realizadas neste trabalho poderão ser úteis para definir arquiteturas de propósito específico, i.e. para aplicação multi-thread com alta dependência de dados entre as threads.

Dificuldade prevista do projeto: 3

Dificuldade prevista do SIMICS: 5

Diagrama:

