

Programação Multi-thread em Arquitetura MultiCore

Samir J. Eichelt, Edson Luiz Padoin

Universidade Regional do Noroeste do Rio Grande do Sul – UNIJUÍ
Departamento de Tecnologia - DETEC - Campus Santa Rosa- RS
samir.eichelt@unijui.edu.br, padoin@unijui.edu.br

Introdução

Muitas das pesquisas no desenvolvimento de processadores não mais almejam simplesmente o aumento da frequência de *clock* para obtenção de melhores desempenhos. Com a utilização da tecnologia *multicore* tem-se a possibilidade de execução paralela de atividades dentro do mesmo processador. Porém, torna-se necessário desenvolver softwares que possam utilizar os recursos desta tecnologia.

Os projetos *multicore* consistem em colocar dois ou mais processadores no interior de um único *chip* [CAR 07]. Com isso, os sistemas operacionais dispõem de mais recursos para o escalonamento dos processos que agora são executados realmente de forma paralela. A grande motivação para esta pesquisa é que atualmente não se tem disponível no mercado programas que possam ser divididos e sejam executados de forma paralela conseguindo utilizar todos os benefícios dessa nova tecnologia.

Para tanto, foram desenvolvidas pesquisas sobre a evolução do processador, destacando as arquiteturas de Von Neumann, *pipeline* superescalar e vetorial e também as principais arquiteturas paralelas utilizadas bem como a tendência no projeto de novas máquinas. Tendo concluído o estudo da arquitetura passou-se a pesquisar formas de programação paralela de modo a dividir as tarefas para que possam ser executadas em paralelo dentro dos diversos núcleos existentes.

Evolução das Arquiteturas

As primeiras arquiteturas eram baseadas no padrão de Von Neumann e apresentavam como principal característica o processamento seqüencial, ou seja, há um único fluxo de instruções controlando um único fluxo de dados, também conhecido como SISD[COM 07]. Já as arquiteturas atuais implementam *pipeline* onde permite-se que haja sobreposição temporal das diversas fases de execução das instruções a serem executadas. Desta forma consegue-se realizar todo o trabalho em menos tempo, porém não consegue-se diminuir o tempo de latência de cada tarefa. Atualmente, a técnica de *pipeline* é a premissa utilizada para fazer com que os processadores executem tarefas mais rapidamente [SIL 07] [HEN 98].

Tendo como base a técnica de *pipeline*, foi desenvolvida a arquitetura de *pipeline* superescalar, que apresenta paralelismo em nível de instrução. Nela consegue-se aumentar o desempenho de um processador executando mais de uma instrução ao mesmo tempo [OLI 07].

Outra arquitetura é a denominada de *pipeline* vetorial. Sua característica é a sua unidade escalar *pipeline* típica e uma unidade vetorial que realizam operações de alto nível sobre *arrays* lineares [HEN 98] [SIL 07].

Já, dentre as arquiteturas paralelas pode-se citar a SMP, MPP, PVP, DSM, COW e NOW, as quais são brevemente descritas a seguir.

a) *Symmetric MultiProcessor* (SMP) consistem de múltiplos processadores conectados entre si e a uma memória por um barramento ou alguma outra forma de circuito de conexão interno, podendo também ser classificados como multiprocessadores UMA (*Uniform Memory Access*) [ALB 07];

b) *Massively Parallel Processor* (MPP), são máquinas formada por milhares de CPUs padronizadas que apresentam bom desempenho pela quantidade de processadores. Estes equipamentos geralmente utilizam uma rede de interconexão proprietária de desempenho muito alto [TAN 07].

c) *Parallel Vector Processor* (PVP) que são constituídas de poucos processadores, que possuem vários *pipelines* vetoriais de alto poder de processamento. A interconexão é feita, em geral, por uma matriz de chaveamento (*crossbar*) de alta vazão. A memória é compartilhada, e os sistemas também podem ser classificados como multiprocessadores UMA [REI 07];

d) *Distributed Shared Memory* (DSM) estas máquinas são semelhantes às SMP, onde um conjunto de CPUs em um multicomputador compartilha um espaço de endereço virtual paginado. Na versão mais simples, cada página é mantida na RAM de exatamente uma CPU [NAV 07][TAN 07].

e) *Cluster of Workstation* (COW), consiste basicamente num conjunto de Workstations ou Computadores Pessoais que não possuem teclado, mouse e monitor, interligados por uma tecnologia de rede.

f) *Net of Workstations* (NOW), consiste num conjunto de Computadores Pessoais completos com mouse, teclado e monitor, que são interligados por uma tecnologia de rede.

Em nível de evolução dos processadores destacam-se os *hyper-threading* (HT) e os *multi-core*. O primeiro possibilita o processamento paralelo através do uso de *threads*. Esta tecnologia simula em um único processador físico dois processadores lógicos. Cada processador lógico recebe seu próprio controlador de interrupção programável e um conjunto de registradores. Os outros recursos do processador físico, tais como, *cache* de memória, unidade de execução, unidade lógica e aritmética, unidade de ponto flutuante e barramentos, são compartilhados entre os processadores lógicos. Em nível de software, significa que o sistema operacional pode enviar tarefas para os processadores lógicos como se estivesse enviando para processadores físicos em um sistema de multiprocessamento [TEC 07].

O segundo, *multicore*, é o último conceito em tecnologia de processadores. Tecnologia esta, constituída de múltiplos núcleos, ou seja, duas ou mais unidades de execução no interior de um único chip. O sistema operacional trata esses núcleos como se cada um fosse um processador diferente, com seus próprios recursos de execução [CAR 07]. No sistema de múltiplos núcleos, como cada núcleo possui sua própria *cache*, os sistemas operacionais dispõem de recursos para trabalhar com processos que são executados em paralelo.

A diferença entre a tecnologia *hyper-threading* e a *multicore*, é que a primeira é limitada a um único núcleo que utiliza os recursos de execução de forma mais eficiente para melhor habilitar os processos. Já a segunda proporciona dois ou mais conjuntos completos de recursos de execução que possibilitam o aumento do desempenho dos processos em execução [INT 07].

Programação Paralela

Um processo é constituído de vários atributos, como por exemplo, espaço de endereçamento, descritores de arquivos abertos, permissões de acesso, quotas, dentre outros. Também é associado ao processo um fluxo de execução, denominado de *thread* [OLI 04].

Threads são linhas de execução independentes dentro do código. Elas existentes dentro de um mesmo processo, compartilham os seus recursos, como o espaço de endereçamento (código e dados) e, devido a essa característica a gerência das *threads* (criação, destruição, troca de contexto, sincronização) é mais “leve” quando comparada com processos [OLI 04].

Sabendo que atualmente as empresas de processadores estão investindo em arquiteturas paralelas, a utilização da programação *multi-thread*, que consiste em colocar vários fluxos de execução (*threads*) dentro de um único processo, passa a ser uma alternativa muito interessante.

O desenvolvimento de programas depende do tipo de arquitetura onde o mesmo será executado. Nas arquiteturas onde a memória é distribuída precisa-se utilizar a técnica de troca de mensagens para que seja feita a comunicação entre processos, já nas arquiteturas que compartilham memória, as *threads* ou processos utilizam o mesmo espaço de endereçamento, assim a comunicação entre elas é feita através da própria memória.

Considerações Finais

Até o momento foram desenvolvidas pesquisas a cerca da evolução das arquiteturas de processadores, construção de máquinas paralelas e suas características. Também foi pesquisado sobre paradigmas de programação paralela no intuito de poder utilizar todo o processamento disponível por tais tecnologias. Atualmente, o trabalho encontra na fase de implementação de um estudo de casos onde aplicações que, utilizando a técnica de programação *multi-thread*, sejam executadas em arquiteturas HT e *multicore*.

Com isso, pretende adquirir um maior conhecimento e domínio sobre programação paralela e novas tecnologias de processadores. Desta forma podendo usufruir ao máximo dessas novas tecnologias que exploram o paralelismo para obter melhores desempenhos.

Referências

- [ALB 07] Albuquerque, Marcelo P. Alves, N. Giovanni, N. **Computação Distribuída de Alto Desempenho**. Disponível em: <http://mesonpi.cat.cbpf.br/e2006/graduacao/> Acessado em 20/06/2007.
- [BAL 07] Balthazar, José Manoel. **Arquitetura SIMD**. Disponível em: <http://demac.rc.unesp.br/gpacp/bibliografia> Acessado em 20/06/2007.
- [CAR 07] Cardoso, Bruno. Rosa, Sávio. Fernandes, Tiago. **Multicore**. Disponível em: <http://www.ic.unicamp.br/~rodolfo/Cursos/mc722/2s2005/Trabalho/g07-multicore.pdf> Acessado em 10/03/2007.
- [COM 07] **Computação Paralela e Arquiteturas Não Convencionais**. Disponível em: <http://www.inf.pucpcaldas.br/~morselli/pesquisa/compparal.htm> A. 31/05/2007.

- [HEN 98] Hennessy John L., Patterson David A.; **Organização e Projetos de Computadores A Interface Hardware/Software**. Berkeley,1998, p 249, 438.
- [INT 07] Intel. **Processadores**. Disponível em: <http://www.intel.com/support/pt/processors/mobile/coreduo/sb/CS-022131.htm> Acessado em 28/06/2007.
- [NAV 07] Navaux, P. O. A., Barreto, M. E., Avila, Rafael B., Oliveira, Fabio. **Execução de aplicações em ambientes concorrentes**. Disponível em: <http://www.inf.ufrgs.br/~avila/download/Navaux:EAA-ERAD01.pdf> Acessado em 21/06/2007.
- [OLI 04] Oliveira, Rômulo Silva. Carissimi, Alexandre da Silva. Toscani, Simão Sirineo. **Sistemas Operacionais**. Instituto de Informática da UFRGS, 2004, p 65 – 66.
- [OLI 07] Oliveira, Rafael Alves de. Lima, Rafael Henrique. Berri, Rômulo.Garcia, Tiago. **Arquitetura Superescalar**. Disponível em: <http://www.superescalar.hpg.ig.com.br/FINAL.htm> Acessado em 12/04/2007.
- [REI 07] Reis Jr, Neyval C. **Algoritmos Paralelos**. Disponível em: http://www.inf.ufes.br/~neyval/Processamento_paralelo2.pdf Acessado em 20/06/2007.
- [SIL 07] Silva, Gabriel P. Arquitetura de Computadores II. **Pipeline**. Disponível em: <http://equipe.nce.ufrj.br/gabriel/arqcomp2/Pipeline.pdf> A. 18/06/2007.
- [TAN 07] Tanenbaum Andrew S.; **Organização Estruturada de Computadores**. Quinta Edição, Pearson Education Inc., 2007 p 343 - 373.
- [TEC 07] **Tecnologia HyperThreading**. Disponível em: <http://www.clubedohardware.com.br/artigos/163> Acessado em 27/06/2007.