

Estudo de Técnicas de Mapeamento de Processos em Arquiteturas Multiprocessadas

Manuela K. Ferreira, Felipe L. Madruga, Marco A. Z. Alves, Philippe O. A. Navaux

Grupo de Processamento Paralelo e Distribuído (GPPD)
Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
{mkferreira, flmadruga, mazalves, navaux}@inf.ufrgs.br

1. Introdução

Atualmente, os processadores com dois ou quatro ou mais núcleos são cada vez mais utilizados, tanto em servidores de grande porte, como em computadores pessoais. As previsões para os próximos anos são de que o número de núcleos por *chip* aumente drasticamente [Azanovic 2006]. Para garantir o aproveitamento dos diversos núcleos presentes em um *Chip* Multiprocessado (CMP), a tendência é que aumente o paralelismo das aplicações.

Neste contexto, o mapeamento de processos é o ato de escolher a distribuição dos diversos processos de uma aplicação paralela para os processadores de um determinado sistema. O objetivo mais comum do mapeamento de processos é a diminuição do tempo de execução das aplicações paralelas, buscando um melhor balanceamento de carga ou diminuir o tempo de comunicação entre processos.

O compartilhamento de memória *cache* entre os núcleos de um CMP surge como uma nova possibilidade de comunicação. Assim, ao mapear processos paralelos que se comunicam frequentemente, é importante considerar se os processadores possuem ou não compartilhamento de *cache*, o que pode levar a uma comunicação mais rápida. Em um *cluster* de CMPs, além da comunicação entre núcleos de um mesmo *chip*, que podem compartilhar memória *cache*, é necessário considerar a comunicação entre *chips* dentro da mesma máquina e ainda entre os diferentes nodos do *cluster*.

O objetivo deste artigo é apresentar uma introdução sobre o mapeamento de processos, enfatizando a necessidade de considerar também a comunicação dentro do CMP ao efetuar o mapeamento dos processos, além disso, serão descritas as próximas etapas do trabalho que está sendo desenvolvido sobre o tema.

2. Técnicas de Mapeamento de Processos e Trabalhos Futuros

O mapeamento de processos em aplicações paralelas é um problema NP-Completo, sendo necessária para sua solução a utilização de algoritmos heurísticos que levam em consideração as características da aplicação paralela e da organização do sistema para se aproximar ao mapeamento ideal [Koppler 99].

As características da aplicação que são comumente consideradas são relacionadas ao volume de comunicação entre processos ou recursos específicos necessários para um determinado processo. Em relação à organização, normalmente são

consideradas a disponibilidade de recursos em um determinado processador ou a velocidade de comunicação entre os processadores. Quanto maior o número de características consideradas, tanto da aplicação como da organização, mais difícil se torna o problema de mapear os processos para os processadores.

Além disso, o mapeamento de processos pode ser dividido em dois grupos: *i) Mapeamento estático* que é feito somente uma vez no início da execução da aplicação e se mantém o mesmo até o final desta execução; *ii) Mapeamento dinâmico* que pode se modificar ao longo da execução da aplicação, conforme os padrões de comunicação entre os processos ou a disponibilidade dos recursos se modifique. Neste caso, surge ainda mais um desafio, a migração de processos em tempo de execução.

Algoritmos heurísticos distintos que mapeiam um grafo que representa a quantidade de comunicação entre processos em um grafo representando a velocidade de comunicação entre os processadores são propostos em [Chen 2006] e [Rodrigues 2009]. Em ambos os trabalhos, os mapeamentos foram estáticos com a necessidade de uma execução prévia da aplicação para descobrir as características de comunicação entre os seus processos. A diferença entre esses dois trabalhos está no algoritmo de mapeamento utilizado, em [Chen 2006] foi proposto um algoritmo que inicia com um mapeamento randômico dos processos e realiza iterações trocando o mapeamento de pares de processos e calculando o ganho na velocidade de comunicação até não haver mais ganho. Em [Rodrigues 2009] foi utilizado o algoritmo de Particionamento Recursivo Duplo que recursivamente divide em dois o conjunto de processos e faz o mesmo com o conjunto de processadores, utilizando para isso a quantidade e a velocidade de comunicação, respectivamente, realizando a cada divisão o mapeamento do conjunto de processos para o conjunto de processadores, também com o objetivo de minimizar o custo a comunicação.

Após uma pesquisa inicial feita sobre o estado da arte do mapeamento de processos atuais verificamos que a maioria dos trabalhos correlatos não leva em consideração a comunicação entre núcleos dentro de um mesmo *chip* (*intra-chip*). Planejamos, então, avaliar essas técnicas de mapeamento de processos através da execução de *benchmarks* em um *cluster* de CMPs. Com a avaliação, poderemos propor a modificação ou combinação das técnicas para que a comunicação *intra-chip* seja levada em consideração e possamos trazer benefício ao desempenho das aplicações.

Referências

- Asanovic, R. et al (2006) “The Landscape of Parallel Computing Researches: A View from Berkeley”, Electrical Engineering and Computer Sciences, University of California at Berkeley, Technical Report N° UCB/EESC-2006-183.
- Chen, H. et al (2006) “MPIPP: an automatic profile-guided parallel process placement toolset for SMP clusters and multiclusters”, In Proceedings of the 20th annual international conference on Supercomputing, p. 353-360, New York, USA.
- Koppler, R. (1999) “Geometry-Aided Rectilinear Partitioning of Unstructured Meshes”, Lecture Notes in Computer Science, p. 450-459.
- Rodrigues, E. R., Madruga, F. L. e Navaux, P. O. A. (2009) “Multi-core Aware Process Mapping and its Impact on Communication Overhead of Parallel Applications”, IEEE Symposium on Computers and Communications (ISCC).