Escalonamento Estático de Programas MPI baseado na Análise de Grafos

Raafel Ennes Silva, Nicolas Maillard, Tiarajú Diverio

PPGC - Instituto de Informática - UFRGS
Av. Bento Gonçalves 9500, Bloco IV, Porto Alegre/RS - Brasil
{resilva, nmaillard, diverio}@inf.ufrgs.br

Introdução

O escalonamento de tarefas pode ser dividido em dois tipos: estático e dinâmico. No escalonamento estático, as tomadas de decisões sobre a distribuição de tarefas ocorrem antes da execução da aplicação. Em contrapartida, no escalonamento dinâmico essas decisões são realizadas durante a execução.

Frequentemente, utiliza-se a representação através de grafos de uma aplicação para poder identificar os pontos críticos de comunicação e de processamento e, então, aplicam-se técnicas de escalonamento para otimizar o aproveitamento da arquitetura utilizada. Para um eficiente emprego dessas técnicas também se considera o modelo de programação utilizado - no caso, troca de mensagens.

A proposta do trabalho está situada no contexto de escalonamento estático de programas que utilizam troca de mensagens através da análise do grafo de fluxo de dados (DFG) extraído da aplicação. Para a troca de mensagens, o MPI é o padrão de *facto* e é usado mundialmente, porém a sua norma não prevê um serviço de escalonamento, deixando esta tarefa para o programador. As aplicações alvo desse tipo de abordagem são aquelas que mantém o mesmo padrão de comunicação após execuções sucessivas, por exemplo a fatoração LU implementada no Linpack (LIN 01).

O objetivo é fornecer uma ferramenta (β-MPI- Biblioteca de Escalonamento de Tarefas MPI) que produza automaticamente o grafo de fluxo de dados de programas MPI e que com base nas informações de volume de comunicação das arestas do grafo seja realizado o escalonamento.

Biblioteca β-MPI

A biblioteca β-MPI foi concebida para extrair o DFG de aplicações MPI e realizar o escalonamento baseando-se no volume de comunicação. O escalonamento em seu conceito mais amplo trata tanto de questões temporais como de questões de localidade. No caso da β-MPI, está sendo tratada apenas as questões de localização de execução das tarefas, ou seja, as tarefas são remapeadas com o intuito de otimizar o uso da arquitetura disponível. A β-MPI é estruturada em três blocos conforme mostra a Figura 1. O primeiro bloco trata da sobrecarga das primitivas MPI. Na implementação da β-MPI cada primitiva de comunicação do MPI foi sobrecarregada para que fosse possível filtrar as informações de processo origem, processo destino e o volume de mensagem trafegado.

Essas informações servem para forma a estrutura de dados, no segundo bloco, com todos os dados necessários para a montagem do DFG da aplicação.

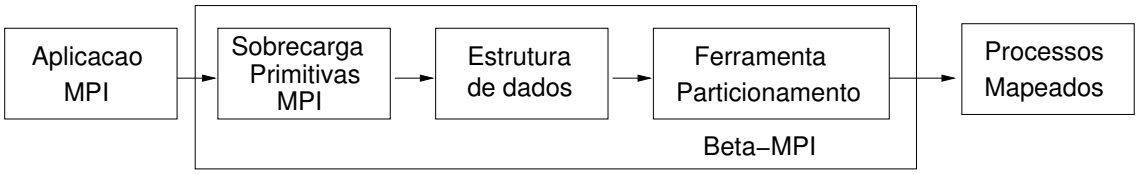

Figura 1: Estrutura da biblioteca.

A geração do DFG, ao final do segundo bloco da estrutura da biblioteca, acontecerá após uma execução prévia da aplicação. No terceiro bloco, o grafo será colocado no formato de arquivo de entrada de uma ferramenta de particionamento de grafos, no caso a ferramenta Metis (MET 01), a qual vai remapear as tarefas da aplicação. E ao final, o usuário realiza uma nova execução da aplicação com os processos remapeados.

O protótipo da biblioteca β-MPI foi avaliado em conjunto com a aplicação de fatoração LU implementada no *Highly Parallel Linpack* (HPL). Este *benchmark* foi concebido para computadores com memória distribuída. O HPL possui um arquivo que permite configurar uma série de parâmetros usados para otimizar o seu uso, tais como algoritmos de *broadcast*, grade de matriz e tamanho da matriz. Um dos resultados alcançados através do uso do HPL com e sem a β-MPI no cluster labtec da UFRGS foi para uma matriz com N=20000, grade P=4 Q=8 dividido em 16 partições (foram usados 16 nodos do cluster), obteve-se como resultado antes do remapeamento 8,484Gflops de com tempo de 628.68 segundos. E depois do remapeamento obteve-se 8,727Gflops e tempo de 611,20 segundos. O ganho final foi de 3% tanto em desempenho quanto em tempo de execução.

Considerações Finais

O protótipo da biblioteca β-MPI validou a idéia de remapeamento de processos através do uso em conjunto com o HPL. Outras aplicações estão tendo seus desempenhos avaliados através do uso com e sem a β-MPI. É importante notar que esse protótipo é válido para aplicações que mantenham o mesmo padrão de comunicação após execuções sucessivas.

Referências

[LIN 01] DONGARRA, J. et al. **The LINPACK Benchmark: Past, Present and Future**. December 2001: (Available in www.cs.utk.edu/ luszczek/articles/hplpaper.pdf).

[MET 01] Karypis, G. et al. **METIS:Unstrucured Graph Partitioning and Sparse Matrix Ordering System**. 1995: (Available in www-users.cs.umn.edu/ karypis/publications/partitioning.html).