Escalonamento estático de programas MPI usando particionamento de grafos: análise da decomposição LU

Rafael Silva, Guilherme Pezzi, Nicolas Maillard,
Tiarajú Diverio

Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves 9500
{resilva, pezzi, nmaillard, diverio}@inf.ufrgs.br

Resumo

Este trabalho apresenta uma análise sobre algoritmos em fases e particionamento em termos de custo de execução para o escalonamento de tarefas. Como estudo de caso foi escolhido o algoritmo de decomposição LU paralelo. O objetivo é verificar a viabilidade de disponibilizar uma biblioteca portável para o escalonamento estático de programas MPI, chamada β-MPI.

Introdução

O foco deste trabalho é o uso de métodos de particionamento de grafos para otimizar as comunicações entre os processos que fazem parte de uma computação paralela, alocando os processos de tal forma que o volume de comunicações entre processadores seja minimizado. Esta técnica já é clássicamente usada, por exemplo em [DOR 03]. No entanto, o uso das ferramentas de particionamento geralmente está embutido no programa, cujas estruturas de dados privadas estão usadas na contrução do grafo. A proposta é usar as ferramentas diretamente em programas MPI, utilizando apenas os recursos padrões da norma MPI 1.2, sendo assim inteiramente genérica. E como primeiro estudo de caso, foi analisado a aplicação de decomposição LU em paralelo, e verificado o custo que seria agregado a mesma no caso de um escalonamento de tarefas após a execução de uma fase.

Decomposição LU e o custo de particionamento

A decomposição LU serve para resolver sistemas de equações lineares algébricas ($Ax = b$). A decomposição ocorre da seguinte forma: Seja uma matriz A pode-se decompor esta matriz em duas, sendo que L é a triangular inferior e U a triangular superior. tal que $|A| = |L||U|$. A resolução do sistema $Ax = b$, dado por $|L||U|x = b \Leftrightarrow L(Ux) = b$ é obtida pela solução direta de $Ly = b$ e pela substituição reversa em $Ux = y$. O algoritmo de decomposição LU em paralelo é fortemente sincronizados por fases de $broadcast$ dos blocos pivô (método de eliminação de Gauss).

Para esta aplicação foi analisado o custo em termos de tempo de execução para realizar o particionamento em dois modelos de máquinas com p processadores. Para o

particionamento foi escolhido a ferramenta METIS a qual utiliza o método multinível k-way para realizar o particionamento de grafos [SCH 00].

O primeiro caso verificado foi o do *broadcast* em duas fases [BIS 97] através do modelo BSP (*Bulk Synchronous Parallel*). O broadcast é realizado em duas fases sendo que em uma primeira fase uma mensagem é enviada para um processo intermediário, escolhido randomicamente sendo a comunicação desbalanceada. E na segunda fase o processo intermediário se encarrega de enviá-la para os destinos, sendo a comunicação balanceada. Esta configuração fornece uma comunicação mais balanceada em termos de mensagens enviadas e recebidas entre os processos. Já com o *broadcast* executado em uma fase a diferença entre mensagens enviadas e recebidas entre os processos era muito grande, portanto a comunicação era muito desbalanceada. Com relação ao custo de particionamento sobre o volume de processamento, foi identificado que para um volume muito grande de dados a serem calculados, o custo de particionamento seria insignificante.

No segundo caso, o modelo do HPL(*Highly Parallel Linpack*)[DON 01] foi utilizado para realizar uma análise similar a anterior. O HPL foi desenvolvido para computadores com memória distribuída, rede estática com comunicação ponto a ponto entre eles. E os processadores são tratados de forma igual sendo que a taxa de comunicação depende apenas do processador utilizado. Para este modelo a relação de custo de particionamento sobre o volume de cálculo também indicou que o custo de particionamento seria mínimo frente a um grande volume de dados processados.

Conclusão e Trabalhos Futuros

Com as relações de custo obtidas nos dois modelos de máquina foi possível estabelecer o quanto a utilização de ferramentas e métodos de particionamento de grafos afetam no tempo de execução de uma aplicação como a do LU. Foi verificado que nesses modelos de máquina distribuída, o particionamento é desprezível frente ao volume de cálculos realizados pela aplicação.Logo, a intenção de disponibilizar uma biblioteca β-MPI portável para escalonamento estático de programas MPI torna-se viável. Atualmente, estão sendo analisados outros algoritmos que se encaixam neste modelo, tais como FFT e CG.

Referências

[DON 01] DONGARRA, J. The Linpack Benchmark: Past, Present and Future. Disponível em www.cs.utk.edu/ luszczek/articles/hplpaper.pdf, 2001.

[SCH 00] SCHLOEGEL, K. et al. Graph Partitioning for High Performance Scientific Simulations. Morgan Kaufman, 2000.

[DOR 03] DORNELES, R. Particionamento de Domínio e Balanceamento de Carga no Modelo HIDRA. UFRGS, 2003.

[BIS 97] BISSELING, R. Basic Techniques for Numerical Linear Algebra on Bulk Synchronous Parallel Computers. First Workshop on Numerical Analysis and Applications, Rousse, Bugaria, 1997.