Análise Comparativa do Produto Matriz-Vetor Entre MPI e MPI-2: Paralelização e Desempenho

Stéfano D. K. Mor, Márcia C. Cera, Nicolas Maillard

Instituto de Informática - UFRGS
{sdkmor, mccera, nicolas}@inf.ufrgs.br

Introdução

MPI (*Message-Passing Interface*) é, desde 1996, o padrão *de fato* para a implementação do modelo de comunicação por troca de mensagens em *Computação Paralela* (GROPP; LUSK; SKJELLUM, 1999). A especificação original foi projetada para trabalhar com um número fixo de processadores, o que restringe a sua flexibilidade.

O advento da Computação em *Grid*, as máquinas capazes de mudar o *hardware* durante a execução de aplicativos e as aplicações que, naturalmente, dependem da criação de processos em tempo de execução, acabaram por demandar uma extensão à norma MPI original (GROPP; LUSK; THAKUR, 1999). Surge, então, o MPI-2, especificação que, dentre outras inovações, oferece recursos para a *criação dinâmica de processos*.

A norma MPI-2 acaba por introduzir questões relevantes em relação à programação clássica com MPI. Duas delas são tratadas neste pôster: as diferenças entre a paralelização de algoritmos nos modelos com *criação estática* e *criação dinâmica* de processos, e o impacto disto no desempenho do programa.

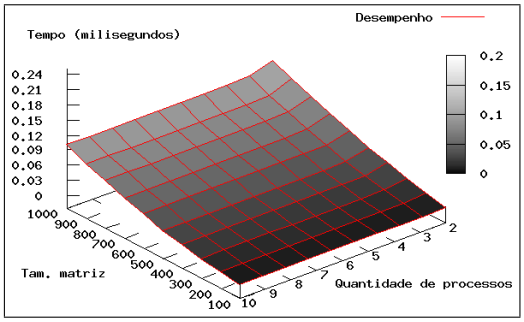
O algoritmo escolhido para realizar-se o experimento foi o *produto de uma matriz por um vetor* ($Ax = b$). Suas implementações em MPI-1.2 e MPI-2 são apresentadas em (GROPP; LUSK; SKJELLUM, 1999) e (GROPP; LUSK; THAKUR, 1999), respectivamente. Ambas as aplicações estão construídas no modelo de programação paralela "mestre-escravo" (DONGARRA et al., 2003).

Paralelização e Análise de Desempenho

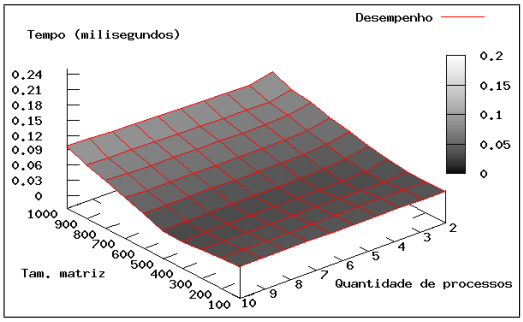
Na versão paralela, o mestre efetua um *broadcast* do vetor x para todos os escravos. Ele decompõe a matriz A em linhas, que ele distribui ciclicamente, à medida que os escravos requisitam trabalho.

No código com o MPI-2, o número de processos escravos dinamicamente criados é especificado como argumento. O processo mestre é quem dispara, através da primitiva "`MPI_Comm_spawn()`", os processo escravos.

As medições foram realizadas no *cluster* Labtec do GPPD (Grupo de Processamento Paralelo e Distribuição) da UFRGS, com a distribuição LAM-MPI. Para a avaliação de cada uma das duas versões (MPI 1.2 e MPI 2), são utilizados gráficos comparativos com relação ao número de nós, tamanho das entradas e tempo decorrido.



(a) MPI-1.2 Benchmark



(b) MPI-2 Benchmark

Figura 1: *Benchmarks* MPI $\times$ MPI-2 para o *Produto Matriz-Vetor*

As execuções são feitas para configurações utilizando de dois (2) a dez (10) nós. Para cada configuração foram executadas versões com vetores e matrizes quadradas (linhas e colunas) da ordem de cem (100) até mil (1000) elementos, incrementados de cem (100). O mesmo procedimento foi repetido cinco (5) vezes, extraindo-se, ao final, a média de tempo. Os desempenhos medidos podem ser visualizados na Figura 1. Os tempos começam a se equivaler a partir do tamanho quinhentos (500) de matriz e o comportamento é relativamente uniforme.

Conclusões

Essas medições ilustram o sobrecusto de se criar processos durante a execução do programa, no caso do MPI-2. À medida que se deve criar mais processos, o sobrecusto aumenta. Quando a matriz cresce, o custo de criação dos processos se torna pequeno em relação ao tempo de processamento em cada escravo; rapidamente o desempenho obtido com os dois MPI se torna semelhante. Uma vez que o MPI-2 é dinâmico, ele junta o desempenho do MPI 1.2 à possibilidade de prover funcionalidades dinâmicas.

Referências

DONGARRA, J.; FOSTER, I.; FOX, G.; GROPP, W.; KENNEDY, K.; TORCZON, L.; WHITE, A. **Sourcebook of Parallel Computing**. 1.ed. [S.l.]: Morgan Kaufmann Publishers, 2003.

GROPP, W.; LUSK, E.; SKJELLUM, A. **Using MPI - Portable Parallel Programming with Message-Passing Interface**. 2.ed. [S.l.]: The MIT Press, 1999.

GROPP, W.; LUSK, E.; THAKUR, R. **Using MPI-2**: advanced features of the message-passing interface. [S.l.]: The MIT Press, 1999.