Análise de desempenho: Eliminação Gaussiana utilizando *Message Passing Interface*

Alexandre Maros[1], **Nadyan Suriel Pscheidt**[1]

[1]Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – Joinville – SC – Brasil

alehstk@gmail.com, nadyan.suriel@gmail.com

1. Introdução

A eliminação Gaussiana é um famoso algoritmo para resolução de sistemas lineares. Neste trabalho foi implementado utilizando pivoteamento parcial e um vetor auxiliar para troca de linhas, evitando a troca física das mesmas, o que reduz o custo computacional significativamente.

O objetivo é executar o algoritmo em paralelo utilizando a biblioteca padrão de comunicação MPI (*Message Passing Interface*) [1], e analisar seu desempenho através dos tempos de execução obtidos com o algoritmo serial e o algoritmo em paralelo. Os resultados serão apresentados na seção seguinte, e uma breve conclusão sobre eles na seção final.

2. Testes de desempenho

Os testes foram executados em ambiente controlado onde haviam onze máquinas, número que baseou o teste com o MPI. A quantidade de variáveis foi escolhida pelo fato de ser inviável testar com uma quantidade maior, descrito a seguir.

Primeiramente foi executado o teste com a implementação em serial, obtendo um desempenho de **1 segundo e 712 milissegundos** para uma matriz de 1000 linhas (ou 1000 variáveis). Enquanto no teste de desempenho para a implementação utilizando MPI obteve um tempo de **12 minutos 56 segundos e 393 milissegundos** para a matriz de mesmo tamanho e 11 processos em execução.

Implementação	**Tempo**
Serial	0 minutos, 1 segundo e 712 milissegundos
MPI	12 minutos, 56 segundos e 393 milissegundos

Tabela 1: Desempenho com mil variáveis

Como pode ser visto, o desempenho utilizando MPI foi significativamente pior do que nos testes com a implementação em serial. Pode se dizer que o problema foi resolvido quase **454 vezes** mais rápido na versão serial.

3. Conclusão

Buscando uma explicação para tamanha discrepância entre os desempenhos de ambas as implementações, a mais plausível encontrada foi o fato de que o algoritmo possui

uma taxa de comunicação extremamente elevada enquanto o cálculo efetuado é bastante simples. Com isso boa parte (quase toda) do tempo de execução é dada pela comunicação entre processos, e uma pequena fração do tempo é de fato utilizado para o cálculo do problema. Então conclui-se que o gargalo está na comunicação dos processos.

Para estudar pouco mais a fundo o desempenho da implementação com MPI, foram executados testes com menos processos. Com 4 processos o tempo de execução foi o mesmo, cerca de 13 minutos. A explicação encontrada foi que mesmo reduzindo o número de processos comunicando-se entre si, a quantidade de informação trocada entre eles foi a mesma, não alterando o resultado final dos experimentos.

4. Referências

[1] BARNEY, Blaise. **Message Passing Interface (MPI)**. Disponível em: <https://computing.llnl.gov/tutorials/mpi/>. Acesso em: 31 out. 2016.

[2] THE OPEN MPI PROJECT. **Open MPI Documentation**. Disponível em: <https://www.open-mpi.org/doc/>.