

Influência da ordem do sistema na paralelização de métodos numéricos

Alessandro Copetti¹, Cleverton Marlon Possani¹, Manuel Binelo¹,
Oleg Khatchatourian², Edson Luiz Padoin²

UNICRUZ¹ – Universidade de Cruz Alta
Rua Andrade Neves, 308 – 98025-810 – Cruz Alta, RS
{copetti, possani, mbinelo}@unicruz.edu.br
UNIJUI² – Universidade Regional do Noroeste do Estado do RS
Rua São Francisco, Cx. P. 560 – 98700-000 – Ijuí, RS
{padoin, olegkha}@unijui.tche.br

Introdução

A maioria dos problemas da Física, Técnica e Engenharia são modelados matematicamente pelos sistemas de equações diferenciais parciais. Mesmo para condições simples de contorno, a solução analítica destas equações é muito difícil de ser obtida. A maioria dos métodos numéricos (por exemplo, método de diferenças finitas, método de elementos finitos, método de volumes finitos e método de elementos de contorno) transformam estas equações em sistemas de equações lineares algébricas (SELA) de grande porte.

A escolha do método para resolução destes sistemas depende de propriedades da matriz do sistema (ordem, simetria, se esparsa, de banda, condicionamento, etc.) e do próprio estilo do pesquisador. Neste trabalho, para resolução destes sistemas foi estudada a aplicação dos métodos diretos, em especial o método de Householder, para os quais as características das matrizes (condicionamento da matriz) quase não influenciam seu desempenho. Entretanto, com o aumento da ordem do sistema, os erros de arredondamento limitam as aplicações destes métodos e a qualidade das soluções. Mesmo assim, uma análise comparativa preliminar realizada pelos autores, mostrou que alguns destes métodos possuem limite de estabilidade muito maior do que outros. Por exemplo, o método de Householder (método de reflexão) mostrou um desempenho melhor em comparação com outros métodos (foram estudados os métodos de Givens, Pivotação Completa e de Gauss).

A convergência dos métodos numéricos na resolução de problemas de evolução limita muito o passo temporal e consequentemente os passos espaciais que provocam um aumento enorme da ordem do sistema e o tempo de execução dos programas. Caso seja usado um cluster de computadores, a disponibilidade de memória e processamento pode ser aproveitada para alocação de grandes matrizes [EGE 95] [OST96].

Este trabalho tem como objetivo definir e implementar uma forma eficiente de execução paralela do método de Householder, explorando matrizes de grande ordem. A seção 2 apresenta a divisão de tarefas empregada no método de Householder. Na seção 3 são mostrados e analisados os testes efetuados no cluster. A seção 4 estuda o parâmetro tamanho da matriz na execução paralela. Na seção 5 são apresentadas as conclusões do trabalho.

Paralelização do Método de Householder

As transformações de Householder podem ser utilizadas como objeto de estudo por apresentarem um grande volume de processamento. Além da forte iteração durante o processamento, a complexidade do algoritmo é $2/3n^3$, o que representa um elevado tempo computacional. A paralelização apresentada em [COP03] utilizou o paradigma mestre-escravo como modelo de programação.

Tabela 1 - Divisão de tarefas no método HouseHolder

Mestre	Escravos
1- Divide a matriz em linhas	
2- Envia para os escravos	3- Recebe as linhas do mestre
Faça de 1 até ordem	
5- Calcula w e envia para escravos 7- Calcula $wt*B$ e envia para escravos	4- Envia sua primeira coluna p/ o mestre 6- Recebe w; calcula $wt*B$ parcial; envia 8- Faz a transformação $B=B-2*w*wt*B$
Desconsidera primeira linha e primeira coluna (diminuindo a ordem do sistema) Desativa dinamicamente os processadores não necessários; Fim do laço	
10- Recebe o vetor resultado do primeiro trabalhador.	9- Envia vetor x parcial p/escravo anterior Se for o 1º escravo, envia resultado p/ o mestre

O processo mestre faz a divisão da matriz inicial em linhas e distribui aos escravos. Durante o laço principal, duas comunicações entre processos são necessárias.

Ambiente de Execução e Medidas de Desempenho

Efetuuou-se a execução da aplicação paralela em um cluster composto de 18 computadores Athlon XP de 2.4 Ghz com 256 MB de RAM (DDR, 266 MHz). Rodando SO Linux/Redhat 9.0, kernel 2.4.20-8 e compilador gcc com a biblioteca de comunicação PVM 3.4.4-6. A rede utilizada foi uma Fast Ethernet (100 base T). Um outro computador com as configurações anteriores foi destinado como servidor/mestre, mas com 512 MB de memória RAM e conexão Ethernet de 1 Gbps.

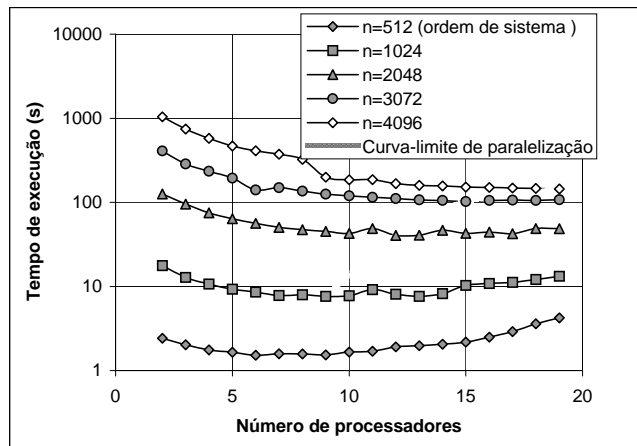


Figura 1. Execução paralela do método de HouseHolder

A curva limite de paralelização mostra os pontos onde foram obtidos os menores tempos. Nota-se que quanto maior a ordem da matriz, mais justifica-se o acréscimo de nodos processadores.

Tamanho da Matriz e a Execução Paralela

A execução do método paralelo de Householder em [COP03] utilizou um cluster com máquinas homogêneas com somente 128 MB de RAM, o que limitou a ordem da matriz. Naquele ambiente foram experimentados sistemas de ordem até 2048. Assim, algumas questões surgiram: Quem estabelece o limite de memória que pode ser alocada pelo processo? Como fazer para aumentar o tamanho do sistema (ordem da matriz)? Onde está o gargalo de memória durante a execução: nos clientes ou no servidor? O programa paralelo pode ser adequado às restrições de memória?

A partir disso, foi desenvolvido um programa piloto para alocar um ponteiro do tipo double de tamanho: $\text{ordem}^2 * 8$. O processo normalmente aloca a memória livre do sistema e, a medida que necessita, passa a utiliza a área de swap.

Quando se utiliza máquinas com 256 MB de RAM e partição de swap de 256 MB, têm-se disponíveis aproximadamente 200 MB de RAM para a alocação dos processos. Nesse exemplo, pode-se resolver sistemas com ordem de, no máximo 7168, conforme mostra a figura 2. No entanto, devido ao uso de swap, o tempo de execução passa a aumentar.

Para avaliar esta elevação de tempo, foi desenvolvido um novo programa piloto que efetua alguns cálculos matemáticos sobre os elementos da matriz. O programa aloca uma matriz de 4096 sobre os quais são efetuadas três diferentes execuções: a primeira execução utilizou a memória RAM disponível e obteve um tempo de execução de 1,64 segs; na segunda, induzimos a necessidade do programa utilizar 80 MB de swap e o tempo de execução passou para 5,33 segs; numa terceira execução foi utilizado somente área de swap e o tempo total foi de 55,75 segs.

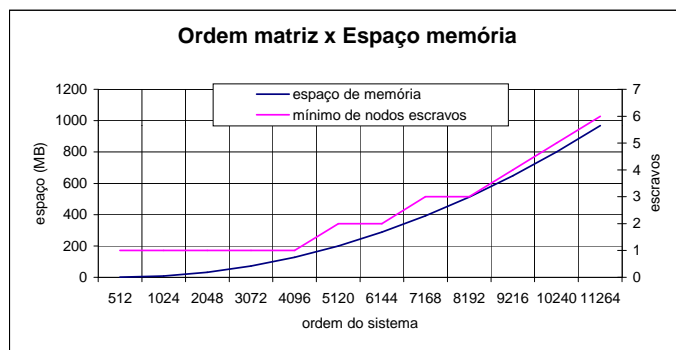


Figura 2. Memória e número de escravos necessários para cada ordem da matriz

Devido ao baixo desempenho quando se utiliza área de swap, faz-se necessário desenvolver um sistema que somente utilize a memória RAM dos equipamentos. Assim, no algoritmo paralelo do método de Householder foi colocado o requisito do número

mínimo de nodos necessários à execução. Por exemplo: para um sistema de ordem 8192 com o modelo mestre/escravo, onde o mestre não atua no processamento, e dispondo de 2 escravos, cada escravo recebe uma matriz de 4096×8192 (o método de Householder divide a matriz em linhas). Desta forma, é necessário alocar 256 MB que é superior aos 200 MB de RAM livre (valor aproximado). Nesse caso, a situação ideal é 1 mestre e 3 escravos. Cada escravo aloca 2730×8192 resultando em 170 MB de alocação.

Conforme a figura 2, onde é apresentada a linha correspondendo ao número mínimo de escravos necessários para execução de matrizes de grande porte, a condição utilizada para efetuar a paralelização otimizada é: $(\text{ordem} / \text{numeroDeEscravos}) * \text{ordem} * 8$, onde o resultado não deve ultrapassar a memória RAM disponível nos nodos escravos.

Conclusões e Trabalhos Futuros

O aumento da ordem do sistema, erros de arredondamento, a falta de memória, bem como outros fatores, muitas vezes limitam a resolução de sistemas ou até mesmo influenciam na qualidade das soluções encontradas. O método de reflexão - Householder tem apresentado um desempenho melhor, se comparado com outros, por exemplo, Givens, Pivotação Completa e de Gauss.

Este trabalho, apresentou uma análise de desempenho da versão paralela do método de *Householder* para resolução de SELAs usando PVM em aglomerados de computadores. Implementou-se um modelo paralelo que leva em consideração o tamanho da matriz e o numero de nodos escravos envolvidos na resolução do sistema. Avaliou-se quanto a utilização de swap prejudica o desempenho do sistema quando os nodos escravos precisam utilizar áreas de memória maiores do que a memória principal disponível. Com isso, foi definida uma nova metodologia para resolver sistemas com um número mínimo de nodos escravos para a resolução de sistemas de grande ordem.

Como trabalho futuro, pretende-se implementar um escalonador que inicialmente consulte alguns parâmetros de cada nodo como a quantidade de memória disponível e a velocidade do processador, a fim de determinar o tamanho da tarefa de cada escravo.

Referências

- [COP 03] COPETTI, Alessandro, PADOIN, Edson L, Khatchatourian, Oleg **Resolução Paralela de Sistemas de Equações Lineares Algébricos**, ERAD – Escola Regional de Alto Desempenho – Santa Maria –RS. 2003.
- [EGE 95] EGECIOGLU Ö., SRINIVASAN, A., **Givens and Householder Reductions for Linear Least Squares on a Cluster of Workstation**, Proc. Int. Conf. on High Performance Computing (HiPC), New Delhi, pp. 734-739, 1995
- [OST 96] OSTROMSKY, T., HANSEN P.C., ZLATEW, Z., **A Parallel Sparse QR-Factorization Algorithm**. 1996, In: Applied Parallel Computing. Springer, pages 462-472.