

Paralelização do Método GMRES com MPI e Pthreads

André Luis Martinotto¹, Emilia Juliane Frizzo¹, Ricardo Vargas Dorneles²

Universidade de Caxias do Sul
Rua Francisco Getúlio Vargas, 1130 Caxias do Sul – RS
{[almartin](mailto:almartin@ucs.br), [ejfrizzo](mailto:ejfrizzo@ucs.br), rvdornel@ucs.br}

Introdução

Alguns problemas físicos podem ser modelados através de equações diferenciais parciais (EDPs), que trabalham com domínios contínuos e precisam ser discretizadas para serem tratadas computacionalmente. A discretização das EDPs pode resultar em Sistemas de Equações Lineares (SELAS), que de maneira geral são grandes e esparsos. Sua solução é obtida através da utilização de métodos numéricos, que podem ser divididos em duas classes: métodos diretos e métodos iterativos. Os métodos diretos são impraticáveis na resolução de SELAS grandes e esparsos, sendo mais apropriada a utilização de métodos numéricos iterativos, já que eles trabalham apenas sobre os elementos não nulos da matriz de coeficientes, e não destroem a esparsidade desta. Além disto, permitem a utilização de formatos alternativos para o armazenamento da matriz de coeficientes.

O método de discretização adotado, que emprega um estêncil de 5 pontos, gerou SELAS cujas matrizes são do tipo banda, e apresentam os elementos não-nulos distribuídos regularmente em um pequeno número de diagonais próximas à diagonal principal. As matrizes banda foram armazenadas utilizando o formato diagonal [SAA96].

Note-se que métodos iterativos são formados por operações de álgebra linear, e a paralelização destes métodos pode ser feita paralelizando suas operações. Assim, para o desenvolvimento deste trabalho optou-se pela paralelização do método GMRES (Generalized Minimum Residual Method), já que a matriz de coeficientes para o problema em questão, um modelo de hidrodinâmica e transporte de massa, é grande, esparsa e não simétrica.

Método GMRES

O GMRES é um método iterativo desenvolvido por Saad e Schultz (1986), para solução de sistemas lineares com matrizes não simétricas, possuindo como principal característica a construção de uma base ortonormal no subespaço de Krylov. O subespaço de Krylov de dimensão m é definido como $\text{span} \{ r_0, Ar_0, A^2 r_0, \dots, A^{m-1} r_0 \}$.

A base ortonormal no subespaço de Krylov pode ser obtida através do processo de Gram-Schmidt. O processo de Gram-Schmidt apresenta problemas na aritmética de ponto flutuante, resultando em vetores não ortogonais. Uma alternativa para contornar esse problema é a substituição do processo de Gram-Schmidt pelo processo de Gram-Schmidt modificado, que é um processo matematicamente equivalente mas computacionalmente superior [RIZ01].

O processo de Gram-Schmidt modificado, além de uma base no subespaço de Krylov, gera uma matriz H denominada de Matriz de Hessemberg, que representa a matriz de coeficientes no subespaço de Krylov [SAA86]. Essa matriz apresenta um formato quase triangular superior, necessitando apenas eliminar a diagonal abaixo da

¹ Bolsistas de Iniciação Científica FAPERGS

² Prof. MSc. do Departamento de Informática da UCS

diagonal principal. A eliminação dessa diagonal pode ser executada pelo processo de rotação de Givens em cada vetor adicionado à base pelo processo de Gram-Schmidt Modificado. O processo de Givens baseia-se na rotação de vetores no subespaço para a eliminação de elementos da matriz, e a rotação do plano é feita através da multiplicação do vetor a ser rotacionado por uma matriz rotação.

Após a transformação da matriz de Hessemberg em uma matriz triangular superior, o sistema pode ser resolvido por retrossubstituição e um novo resíduo é calculado. Se a convergência não for atingida, a solução mais recente é utilizada para iniciar uma nova iteração [RIZ01].

O algoritmo do GMRES padrão pode ser impraticável computacionalmente pois, a cada iteração, os produtos matriz por vetor precisam ser armazenados, o que pode ocasionar problemas de falta de memória quando é necessário um grande número de iterações para resolver o sistema linear. Uma alternativa para contornar esse problema é limitar a dimensão m do subespaço. Essa alternativa recebe o nome de GMRES(m) e possui a desvantagem de não ter a robustez do GMRES tradicional.

Paralelização do GMRES

A implementação feita neste trabalho foi desenvolvida para ser executada em *clusters* de PCs, com sistema operacional Linux. Devido ao fato de muitos *clusters* serem formados por máquinas multiprocessadas, esta implementação explora o paralelismo intra-nodos, através de *threads*, e entre-nodos utilizando a biblioteca de troca de mensagens MPI.

O algoritmo do GMRES é composto por operações de álgebra linear envolvendo matrizes e vetores. A paralelização do GMRES consistiu basicamente na paralelização destas operações. Grande parte delas pode ser executada em paralelo sem a necessidade de comunicação entre os processos, o que possibilita um aumento significativo no desempenho do algoritmo [CAN00].

Foram paralelizadas as operações de subtração de vetores, adição de vetores, multiplicação de vetor por um escalar, produto escalar entre vetores e multiplicação de matriz esparsa por vetor. Na subtração e na soma de vetores, cada nodo executa a operação sobre as partes dos vetores que lhe cabem, sem que haja necessidade de comunicação entre os nodos envolvidos. A multiplicação de um vetor por um valor escalar apresenta um funcionamento semelhante ao das operações de subtração e soma de vetores. Nela, cada nodo multiplica o valor escalar pela parte do vetor que possui, sem a necessidade de nenhum tipo de comunicação. No produto escalar entre dois vetores é necessário a comunicação entre os nodos envolvidos. Cada nodo calcula o produto escalar sobre os dados que possui, e depois é feita uma operação de redução em todos os processos. Na operação de redução o produto total é calculado somando os produtos escalares de todos os nodos.

A operação mais custosa nas diversas variantes do algoritmo do GMRES é a multiplicação matriz por vetor. No caso das matrizes banda, a multiplicação matriz por vetor apresenta algumas particularidades. Na multiplicação de matrizes densas por vetor, é necessário que o vetor esteja inteiro em todos os processos; já com matrizes banda isso não é necessário, cada processo necessita de apenas uma parte do vetor. Essa parte corresponde à sua porção do vetor e mais alguns elementos adicionais dos nodos vizinhos. A reorganização do vetor utilizado na multiplicação matriz por vetor é feita através da comunicação par-ímpar definida em [CAN00].

O processo de Givens representa uma porcentagem muito pequena de processamento no método GMRES e, portanto, optou-se em não paralelizar esse processo, pois o *overhead* superaria o ganho de desempenho.

Em arquiteturas multiprocessadas, além do MPI, está sendo investigada a utilização de *threads*. Com isto, pretende-se tirar melhor proveito dos diversos processadores de um nodo. Para cada operação de álgebra linear foram utilizadas n threads, onde cada *thread* manipula uma parte dos dados. O valor de n corresponde ao número de processadores existentes em cada máquina. Teoricamente, isto proporciona um ganho de desempenho proporcional ao número de processadores disponíveis.

Resultados obtidos

Os testes foram realizados usando sistemas com três ordens: 60000, 120000 e 180000 incógnitas. Nos testes das implementações que utilizam apenas MPI foram utilizados de um a oito nodos do *cluster*. Já em implementações que utilizam *threads* foram utilizados quatro nodos com dois processadores cada, e em cada nodo foram utilizadas duas *threads*.

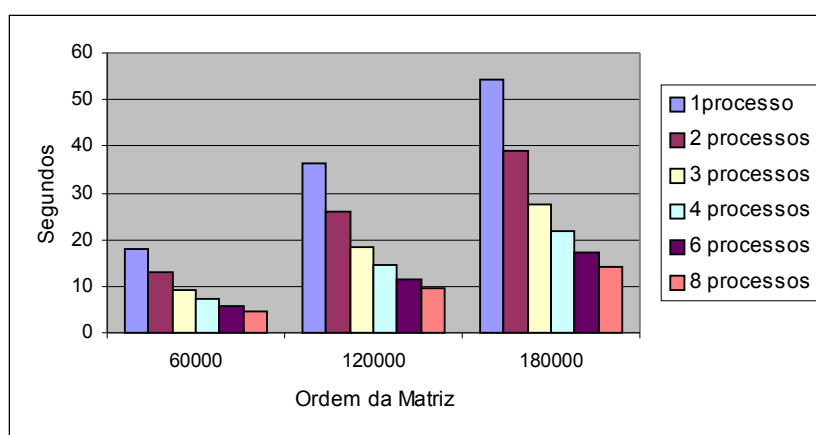


Figura 1 – Desempenho do Método GMRES com MPI

No gráfico da fig. 1 pode ser visto o desempenho obtido com a paralelização do método GMRES utilizando apenas MPI. Estes valores foram obtidos utilizando-se o subespaço de Krylov com $m = 5$. Através de testes verificou-se que o desempenho tende a cair com o aumento do valor de m . Com uma diminuição do valor m diminui-se o número de produtos internos a serem calculados no método de Gram-Schmidt a cada iteração do GMRES.

Tabela 1 TEMPOS DE EXECUÇÃO (SEGUNDOS) - MPI E MPI COM THREADS

			Número de Processadores			
			1	2	3	4
Dimensão da Matriz	60.000	I	17,990463	12,935076	12,338789	9,778672
		II	15,465573	9,686285	10,793103	8,812451
	120.000	I	36,15015	25,662403	24,975449	19,909774
		II	31,218046	19,135093	20,706693	17,151144
	180.000	I	54,271824	38,433421	37,663329	29,775225
		II	46,722838	28,634371	30,965176	24,919374
I – MPI			II – MPI com Threads			

Na tab. 1 pode-se observar os resultados obtidos com a paralelização do Método GMRES com MPI e *threads*.

Conclusões

A principal dificuldade enfrentada na paralelização do GMRES é o alto custo de comunicação (overhead) existente no método de Gram-Schmidt modificado. Formas alternativas estão sendo pesquisadas com o objetivo de diminuir o overhead existente no método paralelizado.

Os resultados obtidos com a utilização de *threads* foram menos expressivos do que o esperado. A obtenção de desempenho com *threads* é influenciada por diversos fatores, por exemplo, os mecanismos de escalonamento e sincronismo do sistema operacional. Estudos estão sendo realizados com o objetivo de detectar a causa do baixo desempenho obtido com *threads*.

Agradecimentos

A FAPERGS pelo suporte econômico, através das bolsas concedidas, para o desenvolvimento dessa pesquisa.

Referências

- [CAN00] CANAL, Ana Paula. **Paralelização de Métodos de Resolução de Sistemas Lineares Esparsos com Deck em um Cluster de PCs**. Porto Alegre: UFRGS, 2000. Dissertação de Mestrado.
- [RIZ01] RIZZI, Rogério L. **Modelo Computacional Paralelo para a Hidrodinâmica e para o Transporte de Massa Bidimensionais e Tridimensionais**. Porto Alegre: UFRGS, 2001. Proposta de Tese para Doutorado.
- [SAA86] SAAD, Yousef, SCHULTZ. **A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems**. New Have: Yale University, 1986.
- [SAA96] SAAD, Yousef. **Iterative Methods for Sparse Linear Systems**. Boston: PWS Publishing Company, 1996.