Solução Paralela de Sistemas de Equações Lineares Através de Métodos de Decomposição de Domínio

Guilherme Galante[1], Rogério Luis Rizzi[1],
André Luis Martinotto[2], Tiarajú Asmuz Diverio[2]

[1]Universidade Estadual do Oeste do Paraná
GMCPAR – Grupo de Matemática Computacional e Processamento Paralelo
[2]Universidade Federal do Rio Grande do Sul
PPGC – Instituto de Informática
ggalante@bol.com.br, rogerio@unioeste.br, {almartin, diverio}@inf.ufrgs.br

Introdução

As soluções de sistemas de equação lineares estão entre os problemas mais comuns encontrados em computação científica [SAA1996]. Tais sistemas são gerados pela discretização de equações diferenciais parciais (EDPs), e são parte de grande parte dos modelos matemáticos de fenômenos físicos e tecnológicos.

Geralmente, tais sistemas de equações são de grande porte e esparsos necessitando de resoluções a cada passo de tempo [CAN2000]. Desse modo, sua solução requer abordagens e estratégias numérico-computacionais eficientes, de tal modo que atualmente é imprescindível o uso de computação de alto desempenho, que pode ser oferecida, por exemplo, pelo uso de computação paralela em arquiteturas de memória distribuída [RIZ2002].

O objetivo deste trabalho é a obtenção de soluções em paralelo para sistemas de equações lineares, de grande porte e esparsos. Para se obter a solução em paralelo, é utilizada a abordagem de decomposição de domínio. Nos métodos de decomposição de domínio obtém-se a solução global do problema combinando as soluções de subproblemas locais. Em particular, emprega-se neste trabalho, o método de decomposição de domínio aditivo de Schwarz.

Palavras-chave: Método de Decomposição de Domínio com Sobreposição, Solução Paralela de EDPs.

Métodos de Decomposição de Domínio

Para a solução paralela dos sistemas de equações gerados a partir da discretização de EDPs, deve-se empregar um conjunto de métodos matemáticos e computacionais. Dentre esses métodos estão os métodos de decomposição de domínio (MDDs). Os MDDs são baseados no particionamento do domínio computacional em subdomínios, de modo que a solução global do problema é obtida pela combinação apropriada das soluções obtidas em cada um dos subdomínios. Isto conduz a uma paralelização natural, particularmente adaptada às arquiteturas paralelas com memória distribuída, pois diferentes subdomínios podem ser tratados simultaneamente [SMI1996], [DEB1998].

De acordo com [SMI1996], os principais atrativos para a utilização de MDDs são:

- Uso de dados locais, necessitando de pouca comunicação, a qual fica restrita nas fronteiras dos subdomínios durante a sincronização da solução;
- Capacidade de trabalhar com distintos modelos de EDPs em regiões com geometria complexa, onde as EDPs podem apresentar diferentes comportamentos em diferentes partes do domínio;
- Podem ser utilizados como pré-condicionadores para acelerar métodos iterativos.

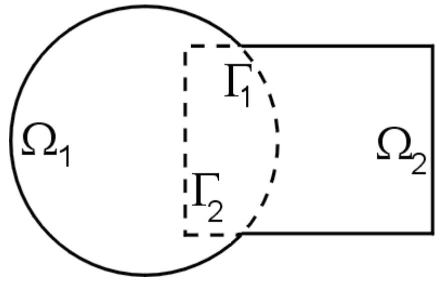

Figura 1 : Decomposição de um domínio com sobreposição

Os MDDs podem ser divididos em duas grandes classes: métodos de Schwarz, onde os subdomínios apresentam uma região de sobreposição, e métodos de Schur, onde os subdomínios não apresentam região de sobreposição [CHA1994].

Os MDDs de Schwarz caracterizam-se pela decomposição do domínio Ω em ***n*** subdomínios sobrepostos Ω_k, tal que $\Omega \subseteq \cup^n \Omega_k$, onde os contornos internos são denotados por $\Gamma_{ij}=\partial\Omega_i \cap \Omega_j$. onde $\Omega=\Omega^{\circ}\cup\partial\Omega$ denota o fecho do domínio e $\partial\Omega$ denota a fronteira de Ω A fronteira artificial Γ_i é parte de Ω_i que é interior de Ω, e $\partial\Omega_i \backslash \Gamma_i$ são os pontos de $\partial\Omega_i$ que não estão em Γ_i [RIZ2002].

Implementação

A Figura 2 mostra o esquema geral do MDD aditivo de Schwarz. Inicialmente tem-se o domínio do problema (malha computacional). Este, por sua vez, deve ser particionado para que possa ser resolvido em paralelo. Atualmente, existem diversas ferramentas de particionamento, dentre elas podemos citar Metis, Chaco, Scotch e Jostle [DOR2003]. É importante salientar que as ferramentas citadas não geram as sobreposições necessárias para os métodos de Schwarz, portanto estas sobreposições devem ser geradas em um processo à parte.

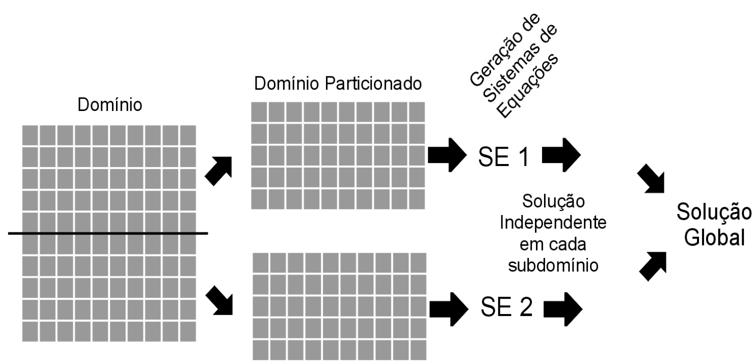

Figura 2: Esquema geral dos métodos de decomposição de domínio

Uma vez particionado, o domínio é discretizado através de métodos de discretização e de aproximação que podem ser representados por um estêncil. Este processo resulta na geração de um sistema de equações para cada subdomínio. Em geral, na discretização são gerados sistemas são de grande porte e esparsos.

Levando este fato em consideração pode-se economizar um espaço significativo de memória se apenas os termos diferentes de zero forem armazenados. Isso foi feito criando-se um arquivo de entrada para cada subdomínio utilizando formato CSR [SAA1996]. A estrutura CSR é baseada na criação de quatro vetores, um contendo a diagonal principal, um vetor contendo os demais elementos da matriz, um vetor contendo as colunas onde os elementos estão posicionados na matriz e um vetor contendo *n + 1* posições, onde é armazenado o número de elementos da matriz até a linha anterior. A identificação a qual linha cada valor armazenado no vetor de dados pertence é baseada no quarto vetor, onde a primeira posição armazena o valor zero, a segunda posição o total de elementos até o fim da primeira linha, a terceira posição é o total de elementos até o fim da segunda linha e assim sucessivamente. Juntamente com os dados do sistema de equações, são armazenadas informações para troca de dados entre os processos.

O particionamento de domínios e o processo de geração dos arquivos de dados referente a cada subdomínio são feitos pelo HIDRA. HIDRA é o modelo computacional paralelo com balanceamento dinâmico de carga para a hidrodinâmica e para o transporte de substâncias bi e tridimensional desenvolvido por Rogério Luis Rizzi [RIZ2002] e Ricardo Vargas Dorneles [DOR 2003].

Após o processo de particionamento de domínio, que é feito de modo seqüencial, cada processo obtém, paralelamente, seus dados iniciais do arquivo que corresponde ao seu domínio. Na seqüência, cada processo gera seu próprio sistema de equações, e calcula independentemente a solução dos mesmos através de iterações sucessivas, como mostrado na figura 3.

$$\begin{cases} L_i u_i^n = f_i, u \in \Omega_i \\ u_i^n = g, \ u \in \partial\Omega_i \setminus \Gamma_i \\ u_i^n = g^{n-1}, \ u \in \Gamma_i \end{cases}$$

Figura 3: Método Aditivo de Schwarz [SMI1996]

Este método é altamente paralelizável, pois as resoluções nos subdomínios podem ser feitas simultaneamente, exigindo comunicações somente para as trocas de dados das condições de contorno que são as soluções obtidas pelos subdomínios vizinhos na iteração *n*-1.

A implementação foi desenvolvida de modo a utilizar *clusters* de PCs como plataforma, explorando o paralelismo através da biblioteca de troca de mensagens MPICH. O *cluster* utilizado nos testes é o *cluster* LabTec-FRGS, formado por 20 nodos Dual Pentium III 1.1 GHz, com 1 Gb de RAM e placa de rede Gigabit-Ethernet.

Resultados

Os resultados do MDD aditivo de Schwarz em termos de tempo de execução são apresentados na Figura 4. Para este trabalho foram resolvidos sistemas de equações originados pelo modelo HIDRA com 11506 incógnitas. Dependendo da resolução desejada, podem ainda ser gerados sistemas na ordem de 46024 e 184096 incógnitas.

É importante ressaltar que foram utilizados nos testes vinte nodos do *cluster*, onde um processo foi alocado para cada nodo, não havendo contenção de memória nem de barramento.

Conclusões

Conforme pode ser observado no gráfico da Figura 4 houve um ganho considerável de desempenho em relação à versão monoprocessada. Pode-se observar também a presença de picos causados pelo aumento no número de iterações em alguns casos (4, 9 e 14 processos).

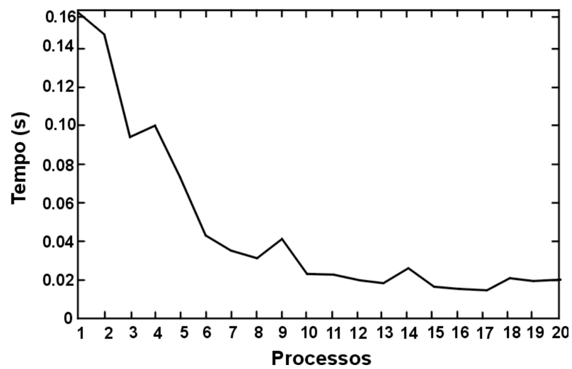

Figura 4: Tempo de execução do MDD aditivo de Schwarz

Como trabalho futuro será acrescentado múltiplas *threads* para explorar o paralelismo intra-nodal através da biblioteca OpenMP, com o propósito de aproveitar melhor os recursos de arquiteturas multiprocessadas.

Referências

[CAN2000] CANAL, A. P. **Paralelização de Métodos de Solução de Sistemas Lineares Esparsos com o DECK em um Cluster de PCs**. 117 p. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre. 2000.

[CHA1994] CHAN, T. F.; MATHEW, T. P. **Domain Decomposition Algorithms**. Acta Numerica, Los Angeles, p. 61-143, Aug. 1994.

[DEB1998] DEBREU, L.; BLAYO, E. **On the Schwarz Alternating Method for Oceanic Models on Parallel Computers**. Journal of Computational Physics, v. 141, p. 93-111, 1998.

[DOR2003] DORNELES, R. V. **Particionamento de Domínio e Balanceamento de Carga no Modelo HIDRA.** 2003. Tese (Doutorado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre.

[RIZ2002] RIZZI, R. L. **Modelo Computacional Paralelo para a Hidrodinâmica e para o Transporte de Substâncias Bidimensional e Tridimensional**. 2002. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.

[SAA1996] SAAD, Y. **Iterative Methods for Sparse Linear Systems**. PWS Publishing Company. 1996.

[SMI1996] SMITH, B.; BJORSTAD, P.; GROPP, W. **Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations**. Cambridge: Cambridge University, 1996.