

389593

matemática aplicada - SB
Análise numérica
cluster
Threads

Paralelizações de Métodos Numéricos em Clusters Empregando as Bibliotecas MPICH, DECK e Pthread

CNPq 1.01.04.00-3

Delcino Picinin Júnior¹
André Luis Martinotto²
Rogerio Luis Rizzi³
Ricardo Vargas Dorneles⁴
Tiarajú Asmuz Diverio⁵

Resumo

No GMC PAD (Grupo de Matemática da Computação e Processamento de Alto Desempenho) da UFRGS vem sendo desenvolvido um modelo da hidrodinâmica e do transporte de massa 2D e 3D chamado HIDRO. A discretização das equações do HIDRO gera sistemas de equações lineares simétricos positivos e definidos e não simétricos.

Neste trabalho são apresentadas paralelizações de métodos iterativos do subespaço de Krylov, bem como a estrutura de armazenamento desenvolvida para essas paralelizações, fazendo-se uso de diferentes ferramentas e abordagens.

Os métodos paralelizados foram o Gradiente Conjugado (GC) e o Resíduo Mínimo Generalizado (GMRES), que podem ser adotados, respectivamente, para solução de sistemas de equações lineares simétricos positivos e definidos e não simétricos.

Essas paralelizações foram projetadas para serem executadas em uma arquitetura formada por um agregado de máquinas independentes e multiprocessadas (*cluster*).

1 Introdução

Através de modelos computacionais é possível simular complexos processos naturais, pois a análise dos resultados de um modelo e de suas propriedades, viabilizada pelo avanço

¹picinin@inf.ufrgs.br

²almartin@inf.ufrgs.br Bolsista CNPq

³rizzi@inf.ufrgs.br

⁴cadinho@inf.ufrgs.br

⁵Apoio: CNPq, LabTeC UFRGS/Dell

da tecnologia dos computadores, permite entendimentos e predições de situações reais (CUMINATO; MENEGETT, 2000).

Tais modelos são, geralmente, constituídos de um sistema de equações diferenciais parciais (EDPs), que são definidas em um espaço contínuo e infinito de pontos. Para resolver numericamente essas EDPs, faz-se necessário transformar esse espaço em um espaço discreto e finito de pontos, sendo o processo chamado de discretização, o responsável por essa transformação.

O processo de discretização gera um sistema de equações que deve ser resolvido a cada passo de tempo. Existem, basicamente, duas classes de métodos que os resolvem: a dos métodos diretos e a dos métodos iterativos. Nesse trabalho, emprega-se os métodos iterativos, pois as matrizes dos coeficientes a serem resolvidas são esparsas e de grande porte e, o emprego de métodos diretos destruiria essa esparsidade, dificultando a otimização no armazenamento. Além disso, os métodos diretos oferecem dificuldades adicionais no processo de paralelização, dada a sua intrínseca dependência de dados.

Dada a necessidade de grande quantidade de memória e de velocidade de processamento para a solução de sistemas de equações com essas características, que geralmente têm milhares ou milhões de incógnitas, o emprego de processamento paralelo é uma alternativa viável para sua solução, quando se considera necessário que a simulação computacional seja feita em um tempo suficientemente rápido para ser útil.

Nesses casos, uma boa alternativa é o emprego de ambientes de alto desempenho. Entre os ambientes de alto desempenho, os *clusters* têm se mostrado uma opção acessível e eficiente, com uma boa relação custo/benefício para obter um alto desempenho computacional.

2 Ambiente de Desenvolvimento

A ambiente de desenvolvimento adotado neste trabalho pode ser classificado seguindo dois critérios, hardware e software. O hardware se refere a parte física da máquina, já o software se refere as bibliotecas de programação paralela que podem ser adotadas.

2.1 Hardware

Um *cluster* é uma arquitetura baseada na união de um conjunto de máquinas independentes, interconectadas por uma rede de comunicação rápida, formando uma plataforma de alto desempenho para execução de aplicações paralelas (BUYA, 1999).

O *cluster* adotado neste trabalho é constituído por vinte e um PCs, onde vinte nodos são dedicados exclusivamente para processamento e um nodo é servidor. Na tab. 1 são apresentados mais detalhes referentes a arquitetura deste *cluster*.

Esse *cluster* faz parte do Laboratório de Tecnologia em *Clusters* (LabTeC), que é um laboratório de pesquisa do Instituto de Informática da UFRGS em conjunto com a Dell do Brasil.

Tabela 1: Arquitetura do *cluster* (LabTeC)

	CPU	RAM	Rede
Servidor	Pentium IV Dual 1.8 GHz	1 GByte	<i>Gigabit Ethernet</i>
Nodos	Pentium III Dual 1.1 GHz	1 GByte	<i>Fast Ethernet</i>

2.2 Software

Dependendo da arquitetura das máquinas que formam o *cluster*, podem existir dois níveis de paralelismo a serem explorados. No caso específico do *cluster* do LabTeC, por tratar-se de um *cluster* de máquinas multiprocessadas, tem-se os níveis de paralelismo intra-nodal (memória compartilhada) e inter-nodal (memória distribuída).

Para cada nível de paralelismo existem diferentes bibliotecas que podem ser adotadas :

- intra-nodal (memória compartilhada) - na exploração do paralelismo intra-nodal, as alternativas são o uso de bibliotecas de *threads* ou de bibliotecas de troca de mensagens;
- inter-nodal (memória distribuída) - na exploração do paralelismo inter-nodal uma alternativa é o uso de bibliotecas de troca de mensagens.

3 Paralelizações do GC e GMRES

Na solução de um sistema de equações lineares através de uma abordagem paralela, os dados devem estar distribuídos entre os processos. Uma abordagem adotada no HIDRA é baseada na geração dos sistemas de equações paralelamente após o particionamento do domínio e a resolução destas via decomposição de dados.

O algoritmo para particionamento de domínios não retangulares que mostrou melhor resultado para o domínio apresentado como estudo de caso foi o RCB (um exemplo pode ser visto na fig. 1).

3.1 Estrutura de Dados para Domínios Não Retangulares

Na abordagem adotada, o domínio do problema é particionado, sendo as células de cada subdomínio resultante enumeradas seguindo uma enumeração conforme ordenação natural. Após a enumeração são gerados os sistemas de equações de forma paralela. Um exemplo de dois subdomínios com suas dependências e células enumeradas pode ser visto na fig. 2.

Baseando-se nessa necessidade de comunicação e visando proporcionar um melhor desempenho na execução dos métodos paralelizados, está sendo adotado um formato de armazenamento que identifica e separa as informações a serem enviadas e recebidas.

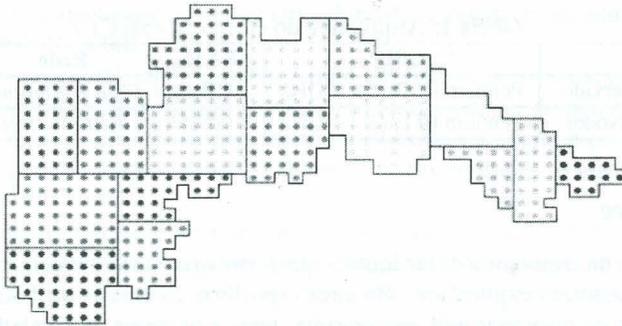


Figura 1: Particionamento em subdomínios irregulares com o algoritmo RCB

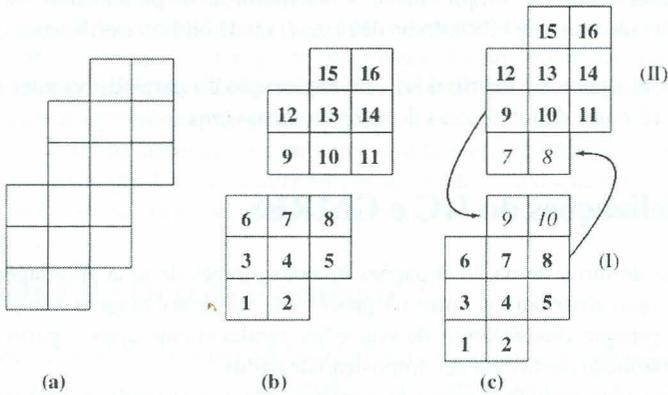


Figura 2: Exemplo para particionamento com enumeração

4 Avaliação e Resultados

As figs. 3 e 4 apresentam, respectivamente, os gráficos de *speedup* dos tempos de execução dos métodos CG e GMRES.

Nos gráficos de *speedup* apresentados acima pode ser observado que o método do GC teve um melhor aumento de desempenho se comparado com o GMRES. Uma possível razão para isso é o fato do GMRES possuir maior *overhead* de comunicação. Esse *overhead* se deve ao fato de que no processo de Gram-Schmidt ocorre um grande número de operações

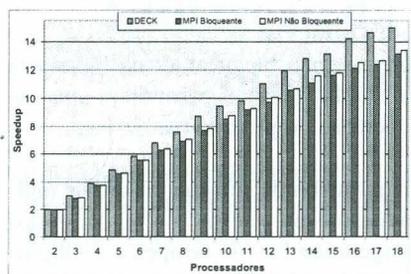


Figura 3: Speedup do GC

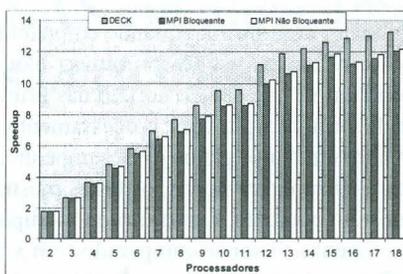


Figura 4: Speedup do GMRES

de produtos escalares que necessitam de comunicação global.

As figs. 5 e 6 apresentam uma comparação entre o uso de apenas múltiplos processos e o uso de múltiplos processos em conjunto com o uso de múltiplas *threads*.

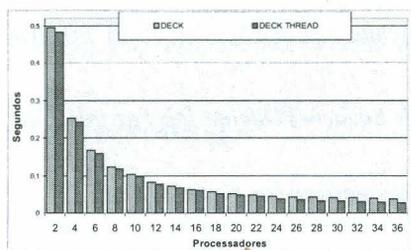


Figura 5: Processo vs. thread no GC

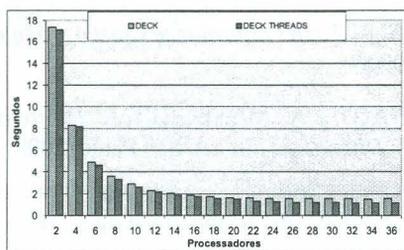


Figura 6: Processo vs. thread no GMRES

Nos resultados apresentados podemos observar que nas paralelizações dos dois métodos a exploração do paralelismo intra-nodal fazendo uso de múltiplas *threads* teve um melhor desempenho. O ganho de desempenho no uso de múltiplas *threads* em relação ao uso de apenas múltiplos processos se torna mais significativo de acordo com o aumento gradativo do número de processadores. Uma justificativa para esse aumento é a diminuição da carga de processamento e o aumento da comunicação quando se usa apenas múltiplos processos.

5 Conclusões

Na exploração do paralelismo inter-nodos, pode-se concluir que a biblioteca DECK proporciona um bom desempenho, relativo a bibliotecas MPICH, e portanto, é uma boa ferramenta para exploração do paralelismo, não somente inter-nodos mas também intra-nodos,

obtendo um maior desempenho quando comparada com o MPICH.

No que se refere ao uso de primitivas bloqueantes ou não bloqueantes na biblioteca MPICH, concluiu-se que devido ao fato das primitivas não bloqueantes permitirem uma sobreposição de comunicação com processamento na operação de produto matriz-vetor, elas proporcionaram um maior ganho de desempenho.

Já no paralelismo intra-nodos, os testes mostraram que a abordagem que faz uso de múltiplas *threads* em conjunto com o uso de múltiplos processos em *clusters* multiprocessados tem um desempenho maior se comparado com a abordagem que faz uso de apenas múltiplos processos. Baseando-se nos testes, concluiu-se que o uso de múltiplas *threads* em conjunto com múltiplos processos não apresentou uma diferença de desempenho muito significativa quando comparada com o uso de apenas múltiplos processos em situações com grande carga de trabalho, no entanto, apresentou uma diferença mais significativa com a diminuição da carga de trabalho, característica que permitiu aumentar a escalabilidade das paralelizações.

Referências

BUYAYA, R. *High Performance Cluster Computing: Achitecture and Systems*. [S.l.]: Prentice Hall, 1999.

CUMINATO, J. A.; MENEGETT, A. *Discretização de Equações Diferenciais Parciais: Técnicas de Diferenças Finitas*. São Paulo, 2000.