

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE
COMPUTAÇÃO

SSC0143
PROGRAMAÇÃO CONCORRENTE - TURMA B

Jacobi-Richardson

PROFESSOR DR. JULIO ESTRELLA

Grupo 06:

Bruno Junqueira Adami
Lucas Junqueira Adami
Lucas Lobosque

Números USP:

6878762
6792496
6792645

25 de maio de 2012

1 Introdução

Sistemas lineares são descritos conforme a equação 1. A matriz A do sistema linear pode ser decomposta na forma descrita pela equação 2 e ser reduzida conforme a equação 3.

$$Ax = b \quad (1)$$

$$A = L + D + R \quad (2)$$

$$A^* = L^* + I + R^* \quad (3)$$

Usando essas decomposições, é possível apresentar o processo iterativo chamado de método de Jacobi-Richardson definido na equação 4. O processo de parada do método é definido na equação 5. Para garantir a convergência, a matriz A deve ser estritamente diagonalmente dominante, satisfazendo o critério das linhas e colunas.

$$x^{(k+1)} = -(L^* + R^*)x^{(k)} + b^* \quad (4)$$

$$\frac{\|x^{(k)} - x^{(k-1)}\|_\infty}{\|x^{(k)}\|_\infty} < \epsilon \quad (5)$$

Apresentado os conceitos necessários, o objetivo do trabalho é a implementação do método de Jacobi-Richardson utilizando uma abordagem sequencial e uma paralela usando OpenMP e MPI. Ao final, há uma discussão dos resultados obtidos pelas duas abordagens.

2 A abordagem sequencial

3 A abordagem paralela

O método pode ser considerado como um algoritmo iterativo paralelo. Isto é, a cada iteração, cada $x_i^{(k)}$ pode ser calculado independentemente das outras variáveis na mesma iteração. Portanto, é possível aplicar uma decomposição de dados, onde a cada passo, o vetor x é distribuído entre os nós para ser calculado, cada nó recebendo uma faixa de variáveis, conforme ilustra a figura 1.

O fluxo de dados segue a lógica da figura 2. O primeiro processo, o processo mestre é responsável pela administração de todo o processo. Ele carrega os arquivos de entrada, controla a cadeia de mensagens, envia aos outros processos pedaços do problema e verifica a condição de parada.

Em relação à matriz A , o processo principal contém todos os seus valores, obtidos durante o carregamento do problema. Quanto aos processos secundários, o mestre envia a eles somente a faixa necessária para que eles possam executar o processo. Isto é feito através da chamada *scatter* do MPI. Essa abordagem é possível pelo fato de que para o cálculo de x_i , somente a linha i da matriz A é necessária.

O mesmo ocorre com o vetor b . Para calcular x_i , somente o valor da posição

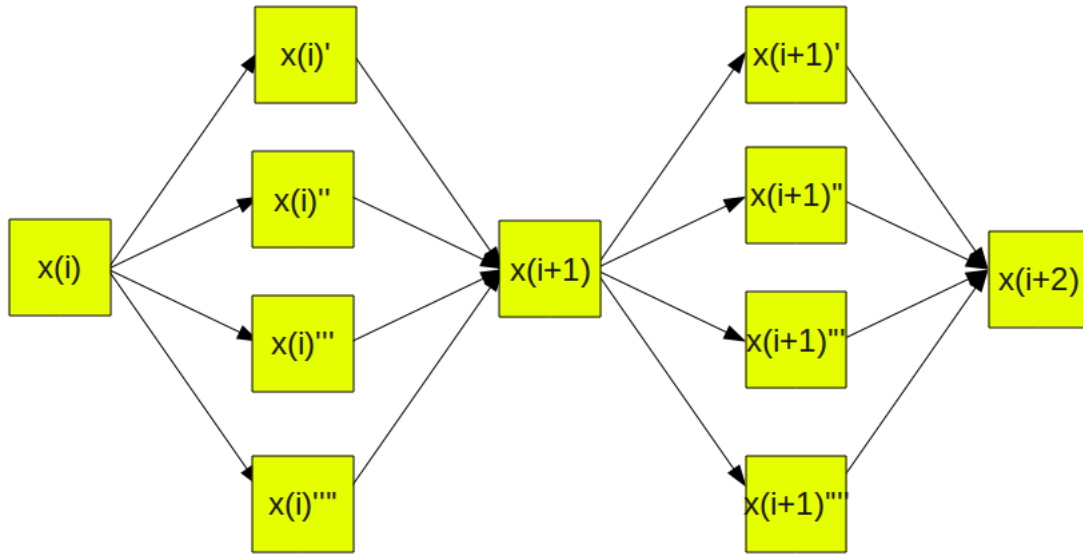


Figura 1: Decomposição de dados

i do vetor b é necessário. Uma chamada à função *scatter* é feita e cada processo secundário somente trabalha com a faixa que lhe diz respeito.

Para o vetor X , é necessário a cada iteração de todo o seu conteúdo completo. Portanto, um *broadcast* do MPI é feito pelo processo mestre a todos os processos antes que as iterações comecem ($x^{(0)}$).

Ao término de cada iteração, os valores obtidos estão armazenados no vetor X , cada processo contendo sua faixa de valores. Para prosseguir com a próxima iteração, o *allgather* do MPI é chamado sobre o vetor X , garantindo a atualização desse vetor a todos os nós.

Os macros OpenMP foram utilizados com 4 threads. Há um macro para paralelizar o processo de normalização da matriz A antes do início do algoritmo. Há também um macro que paraleliza o cálculo de todos os x_i pertencentes à faixa do processo. Para cada x_i , outro macro é utilizado na multiplicação feita entre o vetor x e a linha i da matriz A . Um macro foi criado para checar a condição de parada do algoritmo (definido na equação 5). Finalmente, um último macro é usado para obter a solução do problema.

4 Resultados e conclusões

Referências

[Jacobi-Richardson] Franco, Neide Bertoldi; Cálculo Numérico - São Paulo: Pearson Prentice Hall, 2006

[OpenMP] <http://bisqwit.iki.fi/story/howto/openmp/>

[OpenMP] <http://openmp.org/wp/>

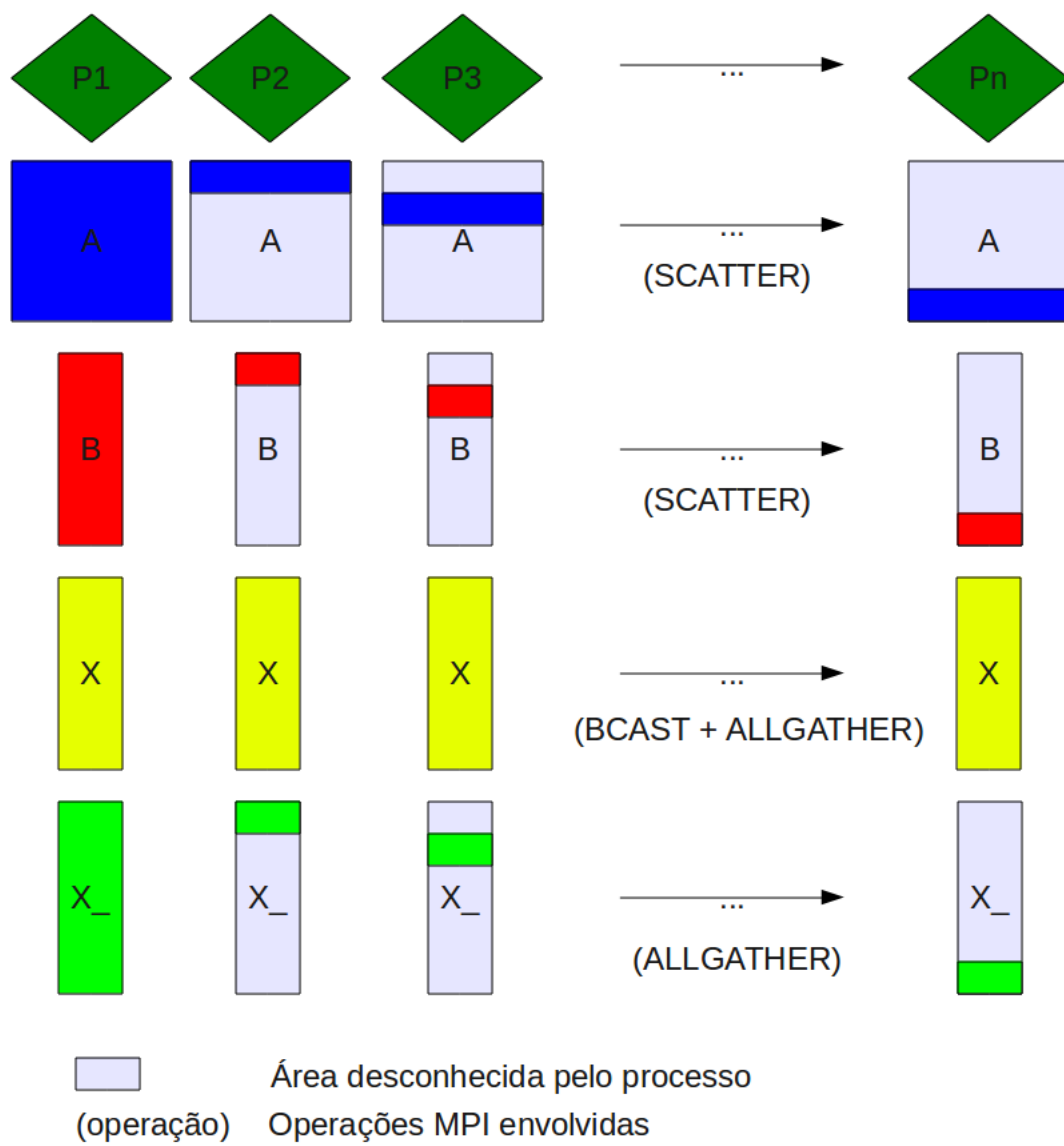


Figura 2: Fluxo de dados

[MPI] <http://www.slac.stanford.edu/comp/unix/farm/mpi.html>

[MPI] <http://www.eecis.udel.edu/~saunders/courses/372/01f/manual/manual.html>