



INSTITUTO SUPERIOR POLITÉCNICO DE TECNOLOGIAS E CIÊNCIAS

INSTITUTO SUPERIOR POLITÉCNICO DE TECNOLOGIAS E CIÊNCIAS

COMPUTAÇÃO PARALELA E DISTRIBUIDA

RELATÓRIO DO LAB 4 (PROGRAMAÇÃO COM MPI)

Clofía Loureiro – 20211438

Dério Queque – 20210634

William Azevedo – 20210217

Curso: Engenharia Informática

Turma : EINF – M1

3º Ano

Índice

1. Introdução	3
2. Sobre o desenvolvimento de um algoritmo de fatoração de matriz distribuída.....	4
2.1 Lendo dos dados de entrada.....	4
2.2 Atualização das Matrizes L e R.....	4
2.3 Identificação da Maior Predição para Cada Linha de B	4
3. Análise de Desempenho	4
4. Metodologia de Foster.....	5
2. Padrões de comunicação de mensagens:.....	5
3. Modo de memória compartilhada:.....	5
4. Modo de chamada de procedimento remoto (RPC):	5
5. Padrões de Comunicação em Grupo:	5
5. Conclusão	6

1. Introdução

Este relatório descreve a implementação e análise de programas MPI para resolver um problema específico: a factoração matricial distribuída. O objetivo é apresentar uma visão geral do desenvolvimento de programas paralelos usando MPI, desde a criação das estruturas de dados até a análise de desempenho.

O relatório abordará a implementação de um algoritmo de factoração matricial distribuída usando MPI, detalhando a estrutura do programa, as funções utilizadas e a análise de desempenho. Além disso, serão discutidos os resultados obtidos e as conclusões derivadas desses resultados. Este trabalho visa fornecer uma compreensão prática e aplicada da programação com MPI para resolver problemas computacionais complexos em ambientes distribuídos.

2.Sobre o desenvolvimento de um algoritmo de fatora  o de matriz distribu  da

MPI   usado na implementa  o do algoritmo de fatora  o de matrizes distribu  das para permitir que processos paralelos trabalhem juntos. Este algoritmo   dividido em v rias partes:

1. Leia os dados de entrada de um arquivo.
2. Inicialize as matrizes L e R com valores aleat rios.
3. Opera  es iterativas que atualizam as matrizes L e R atrav s de uma abordagem gradiente descendente.
- 4.Determinando a matriz B— que   obtida pelo produto das matrizes L e R.
- 5.Descobrimos a previs o m xima em cada linha da matriz B.
- 6.Salvando os resultados em um arquivo separado.

2.1 Lendo dos dados de entrada

Os dados de entrada s o lidos pelo software a partir de um arquivo que foi mencionado pelo usu rio como argumento na linha de comando. As informa  es consistem em: contagem de itera  es, valor alfa, n mero de recursos, dimens es da matriz e entradas diferentes de zero na matriz A.

2.2 Atualiza  o das Matrizes L e R

A atualiza  o das matrizes L e R envolve um procedimento de atualiza  o iterativo usando um algoritmo descendente de gradientes. Para a tarefa de lidar com se  es das matrizes L e R, cada processo MPI   respons vel; a comunica  o   facilitada entre esses processos para que as por  es apropriadas das matrizes possam ser atualizadas de forma eficaz.

2.3 Identifica  o da Maior Predi  o para Cada Linha de B

Para determinar a previs o mais precisa para cada linha em B,   realizado um processo de identifica  o, resultando na identifica  o da maior previs o.

Ap s a conclus o do processo de fatora  o, o programa passa a analisar cada linha da matriz B, determinando a maior previs o para cada usu rio e, assim, identificando a recomenda  o mais prov vel.

3. An lise de Desempenho

O tempo de execu  o global do programa   medido usando 'MPI_Wtime' para realizar an lises de desempenho. Isso permite a avalia  o do desempenho do algoritmo em v rios cen rios de entrada e com quantidades vari veis de processos MPI.

4. Metodologia de Foster

Em nossos esforços para implementar um algoritmo de fatoração de matriz distribuída usando MPI, planejamos incorporar a metodologia de Foster em nosso trabalho.

A metodologia de Foster foi implementada na execução da tarefa em questão.

1. SPMD Flow Pattern:

Seguindo o padrão SPMD, o programa em questão segue um modelo onde cada processo MPI executa um programa idêntico, mas executa operações em diferentes seções dos dados de entrada. A cada processo é atribuída a tarefa de gerenciar uma porção específica das matrizes L e R, trabalhando em conjunto para atualizar e realizar a multiplicação de matrizes nessas matrizes.

2. Padrões de comunicação de mensagens:

A comunicação entre processos ocorre usando a biblioteca de passagem de mensagens MPI . O MPI processa mensagens de troca para compartilhar dados, coordenar operações e sincronizar a execução de algoritmos.

3. Modo de memória compartilhada:

Embora os programas não usem o modo de compartilhamento de memória directamente, cada processo MPI tem acesso à sua própria memória local. No entanto, neste caso, a partilha de memória entre processos MPI não é necessária, uma vez que a comunicação ocorre apenas através da troca de mensagens.

4. Modo de chamada de procedimento remoto (RPC):

Este programa não usa explicitamente o padrão RPC porque não há chamadas de procedimento remoto entre processos. No entanto, o paradigma MPI pode ser pensado como uma forma de RPC, onde os processos MPI podem chamar funções remotas .

5. Padrões de Comunicação em Grupo:

Embora o programa não explore explicitamente os padrões de comunicação em grupo, ele pode ser considerado implicitamente presente. Os processos MPI são organizados em grupos de MPI_COMM_WORLD padrão, e as operações de comunicação são realizadas dentro do grupo.

Ao aplicar a abordagem de Foster ao trabalho, podemos obter uma melhor compreensão da natureza da comunicação entre os processos do IPM e como ela se encaixa em diferentes padrões de comunicação identificado por Foster. Isto permite-nos projetar e implementar sistemas paralelos e distribuídos de forma

mais eficiente, escolhendo o padrão de comunicação mais adequado às necessidades específicas de cada aplicação.

5.Conclusão

O desenvolvimento e análise do algoritmo de factoração matricial distribuída utilizando MPI demonstraram a eficácia da programação paralela para resolver problemas computacionais complexos em ambientes distribuídos. Ao longo deste trabalho, foram abordados os seguintes pontos: implementação do algoritmo, análise de desempenho, aplicação da metodologia de Foster.

Além do desenvolvimento do algoritmo, este trabalho também destacou a importância do trabalho em equipe para alcançar os objetivos com sucesso, a comunicação entre os elementos do grupo foi fundamental para garantir uma compreensão comum dos requisitos do projeto, trocar ideias e resolver problemas que surgiram ao longo do desenvolvimento, a distribuição eficiente de tarefas entre os elementos permitiu que cada um contribuísse com suas habilidades, maximizando a produtividade e a qualidade do trabalho final.

Em suma, tanto o desenvolvimento do algoritmo quanto a dinâmica de trabalho em equipe foram aspectos fundamentais para o sucesso deste projeto. A combinação de uma implementação eficaz do algoritmo com uma colaboração eficiente entre os membros da equipe resultou em um trabalho bem-sucedido e satisfatório.

LINK DO GITHUB: <https://github.com/depakas14/PP2-CPD.git>