Programação em Computação Paralela e Distribuída

Laboratório, ERBASE 2009

Esquemas algoritmos

Domingo Giménez

Universidad de Murcia, Spain

Grupo de Computación Paralela

http://dis.um.es/~domingo

Murilo Boratto

UFBa

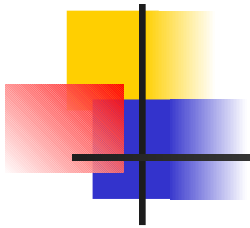
Leandro Coelho

UNEB

Esquemas algoritmos paralelos

- Paralelismo de dados
- Compartilhamento de dados
- Esquemas em árvore
- Computação em Pipeline
- Mestro-Escravo
- Computação síncrona

Para a cada tema, veremos idéias gerais e praticaremos com alguns exemplos



Paralelismo de dados

- Trabalha-se com muitos dados que se tratam de uma forma igual ou similar
- Algoritmos numéricos nos quais os dados estão armazenados em vectores ou matrices e se trabalha com eles realizando um mesmo processamento
- Apropriada para memória compartilhada. Obtém-se o paralelismo dividindo o trabalho das repetições entre os threads (#pragma omp for)
- Repetições onde não há dependências de dados, ou se podem evitar

Paralelismo de dados: exemplos

- **Soma de números:**

  codigo6-1.c (paralelismo impícito); codigo6-2.c (paralelismo explícito: programa-se a divisão do trabalho)

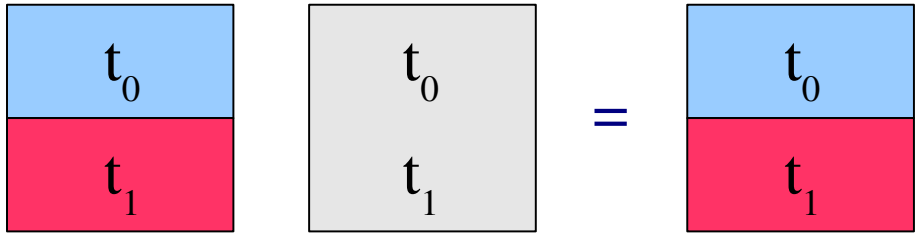
- **Ordenação por grau:**

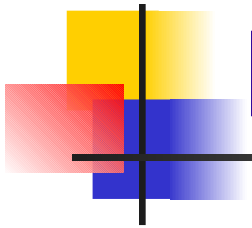
  codigo6-3.c (implícito); codigo6-4.c (implícito, um grupo de threads); codigo6-5.c (explícito)

- **Multiplicação de matrices:**

  codigo6-6.c (implícito);

  codigo6-7.c (explícito)





Partilha de dados

- Trabalha com volumes grandes de dados que estão em vectores ou matrices
- Obtém-se paralelismo dividindo o trabalho em zonas diferentes dos dados em diferentes processos
- A distribuição de dados nos processadores determina o trabalho que cada um faz
- Divide-se o trabalho em regiões e há comunicações entre regiões vizinhas
- Há comunicações para aceder a dados que têm outros processos

Partilha de dados: exemplos

- **Soma de números:**

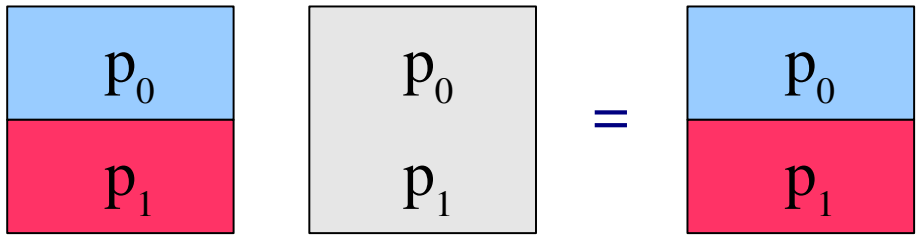
  codigo4-3+4+5.c (o processo 0 envia um número igual de dados a somar à cada processo)

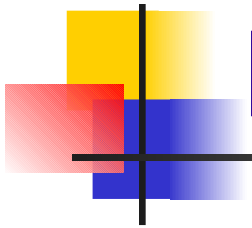
- **Ordenação por grau:**

  codigo6-8.c (o processo 0 envia dados); codigo6-9.c (os dados estão distribuídos, pelo que há passos de comunicação)

- **Multiplicação de matrices:**

  codigo6-10.c (o processo 0 envia as matrices)



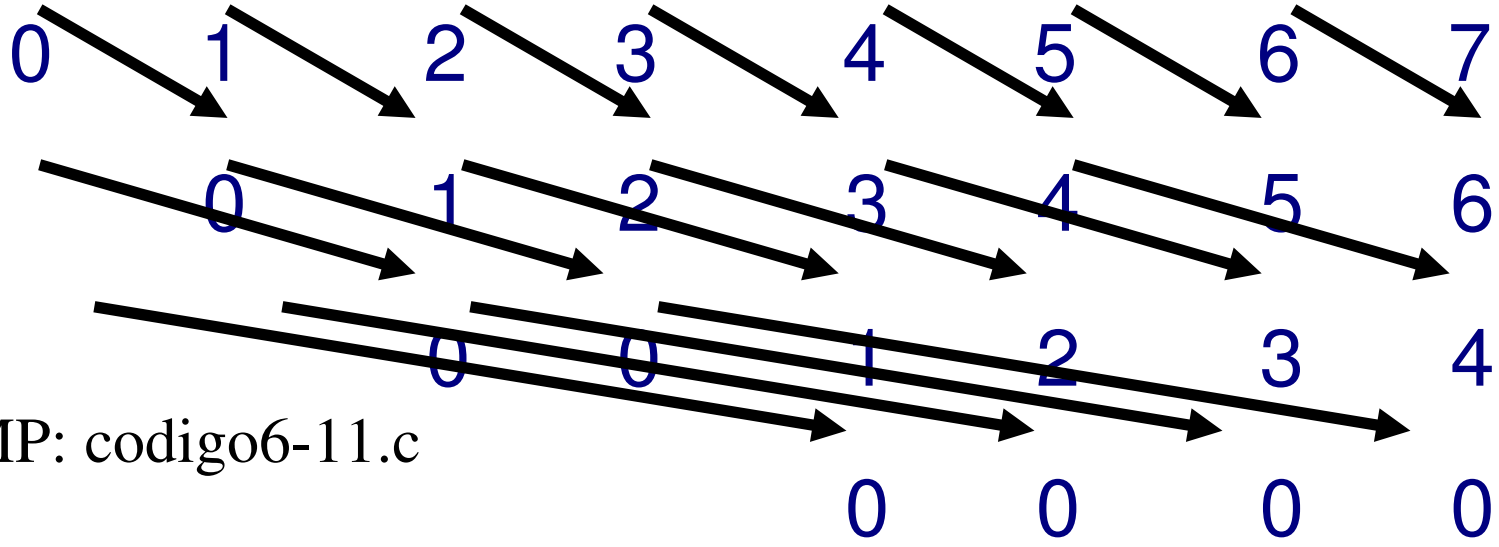


Esquemas em árvore

- Problemas que têm representação em forma de **árvore** ou **grafo**, e se resolvem basicamente percorrendo a árvore ou grafo e realizando as computações que aparecem
- Constrói-se um **grafo** de dependências para representar as computações (nodos) e as dependências de dados (aristas)
- Decide-se a atribuição de tarefas aos processos e as comunicações entre eles, que virão dadas pelas aristas
- A atribuição dos nodos aos processos faz-se equilibrando a carga, e minimizando as sincronizações e comunicações

Esquemas em árvore: exemplos

- **Soma prefixa**: dada uma série de números $x_0, x_1, \ldots, x_{n-1}$ obter todas as somas $x_0+\ldots+x_i$, com $i=0,\ldots,n-1$



OpenMP: codigo6-11.c

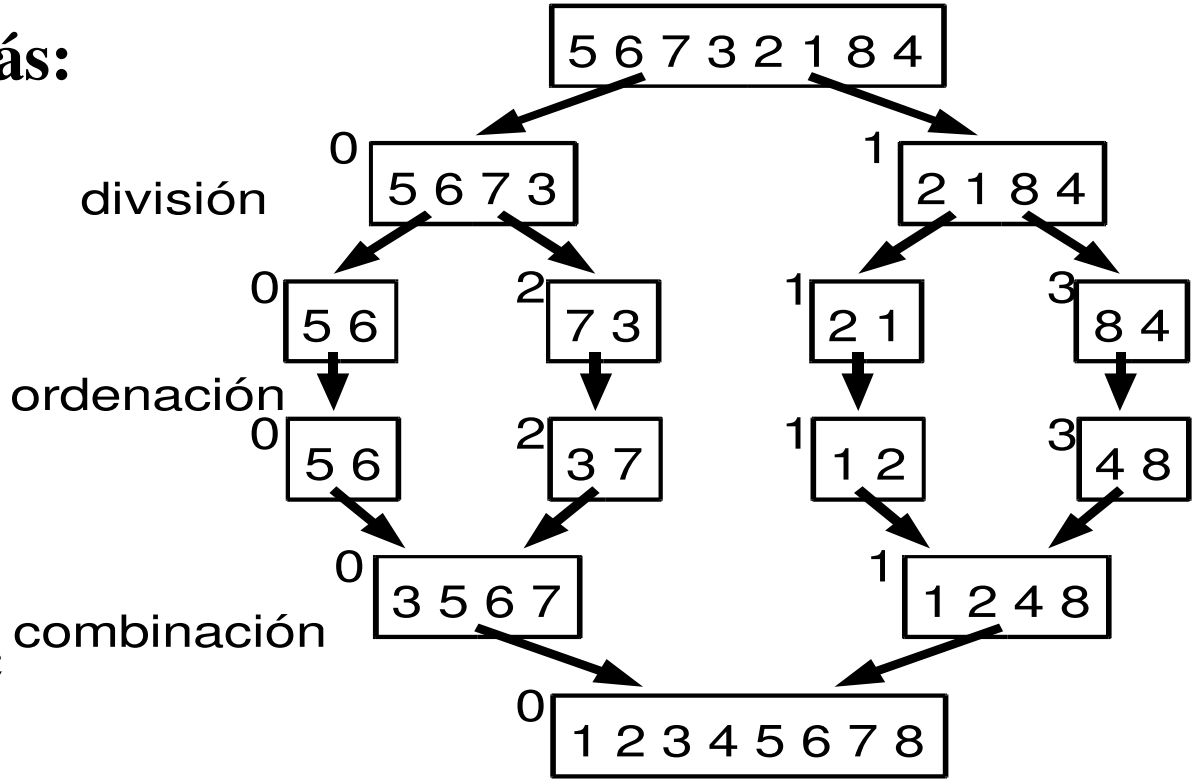
MPI: codigo6-12.c

tirar restrição de que o número de processadores seja n, e potência de 2

Esquemas em árvore: exemplos

- **Divide e Vencerás:**

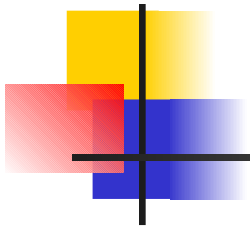
mergesort



OpenMP: codigo6-15.c

MPI: codigo6-16.c

eliminar as restrições de tamanho, programar outros D&V (quicksort...)



Pipeline

- Decompõe-se o problema numa série de tarefas sucessivas
- Os dados viajam numa direcção pela estrutura de processos
- Tem estrutura de passagem de mensagens: entre tarefas consecutivas comunicam-se os dados
- É interessante utilizar quando:
  - não há somente um conjunto de dados, isto é, os dados entram para computarse um depois de outro
  - não é necessário que uma tarefa esteja acabada para começar a seguinte

Pipeline: exemplo

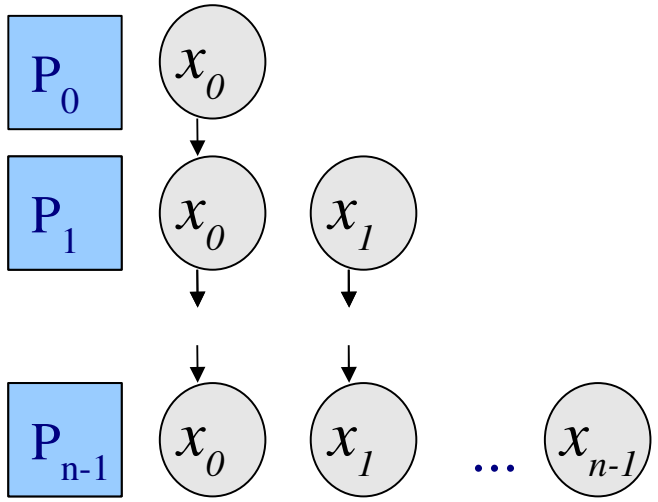
- **Sistema de equações**

$$a_{0,0}x_0 = b_0$$

$$a_{1,0}x_0 + a_{1,1}x_1 = b_1$$

$$\ldots$$

$$a_{n-1,0}x_0 + a_{n-1,1}x_1 + \ldots + a_{n-1,n-1}x_{n-1} = b_{n-1}$$

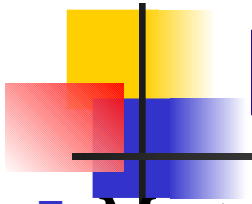


MPI: codigo6-17.c

eliminar as restrições de tamanho

codigo6-18.c , sistema de equações em OpenMP (paralelismo de dados)

Mestre-Escravo



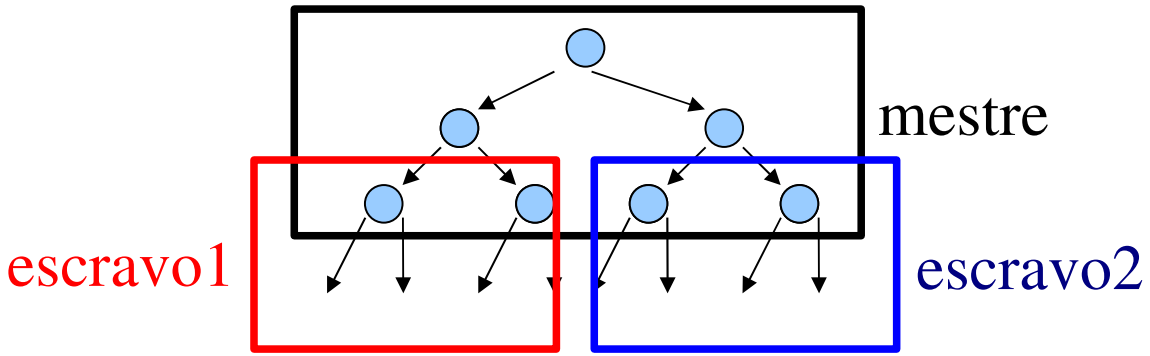
- Mestre-Escravo: um processo mestre inicializa os processos escravos, atribuindo trabalho e recolhe as soluções parciais geradas
- Granja de processos: um conjunto de processos trabalha de maneira conjunta mas independente na resolução de um problema
- Trabalhadores replicados: os trabalhadores actuam de forma autónoma, resolvem tarefas que podem gerar novas tarefas que as resolve o mesmo ou outro trabalhador
  - bolsa de tarefas
  - problema de terminação

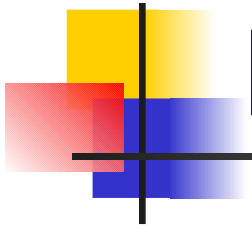
Mestre-Escravo: problema das rainhas

codigo6-25.c, sequencial com bolsa de tarefas

codigo6-26.c, en OpenMP
codigo6-27+28.c, en MPI
} acesso à bolsa é "cuello de botella"

Possível solução: o mestre gera até um nível e atribui trabalhos de maneira estática aos escravos (codigo6-29.c, en OpenMP)

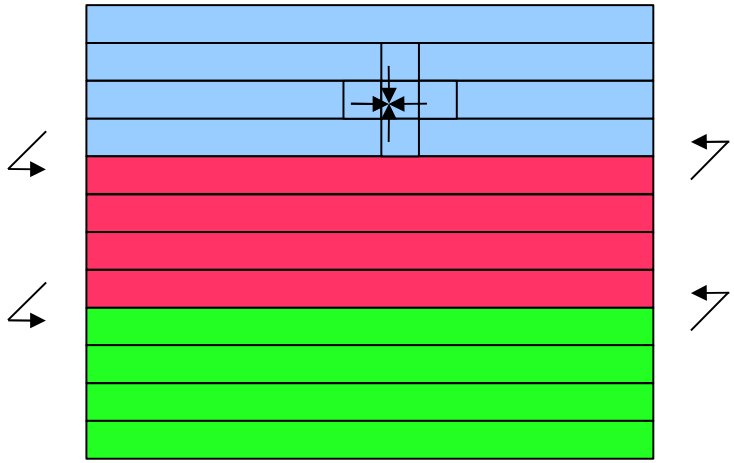




Paralelismo síncrono

- Resolve-se um problema por **iteraciones sucessivas**
- Cada processo realiza o mesmo trabalho sobre uma parte dos dados
- Resultados de uma iteración utilizam-se na seguinte: ao final da cada iteración há sincronização ou comunicação
- Acaba quando se cumpre algum critério de convergencia:
  - precisa-se comunicação global
  - ou um número fixo de repetições

Paralelismo síncrono: Relaxação de Jacobi



codigo6-19.c, OpenMP implícito

codigo6-20.c, OpenMP explícito

codigo6-21.c, MPI

Paralelismo síncrono: multiplicação de Cannon

codigo6-22+23.c, MPI

$p=q^2$ processos

As matrices distribuem-se desde o processo 0, depois $q-1$ pasos