

Processamento Paralelo

Aula 1 - Introdução

Adriano Wagner

17/01/2011



Objetivo

- ▶ Apresentar os conceitos básicos de processamento paralelo.
- ▶ Capacitar os alunos a implementar e executar um programa paralelo utilizando o padrão MPI (Message Passing Interface).
- ▶ Mostrar a aplicabilidade do conteúdo aprendido na migração sísmica.

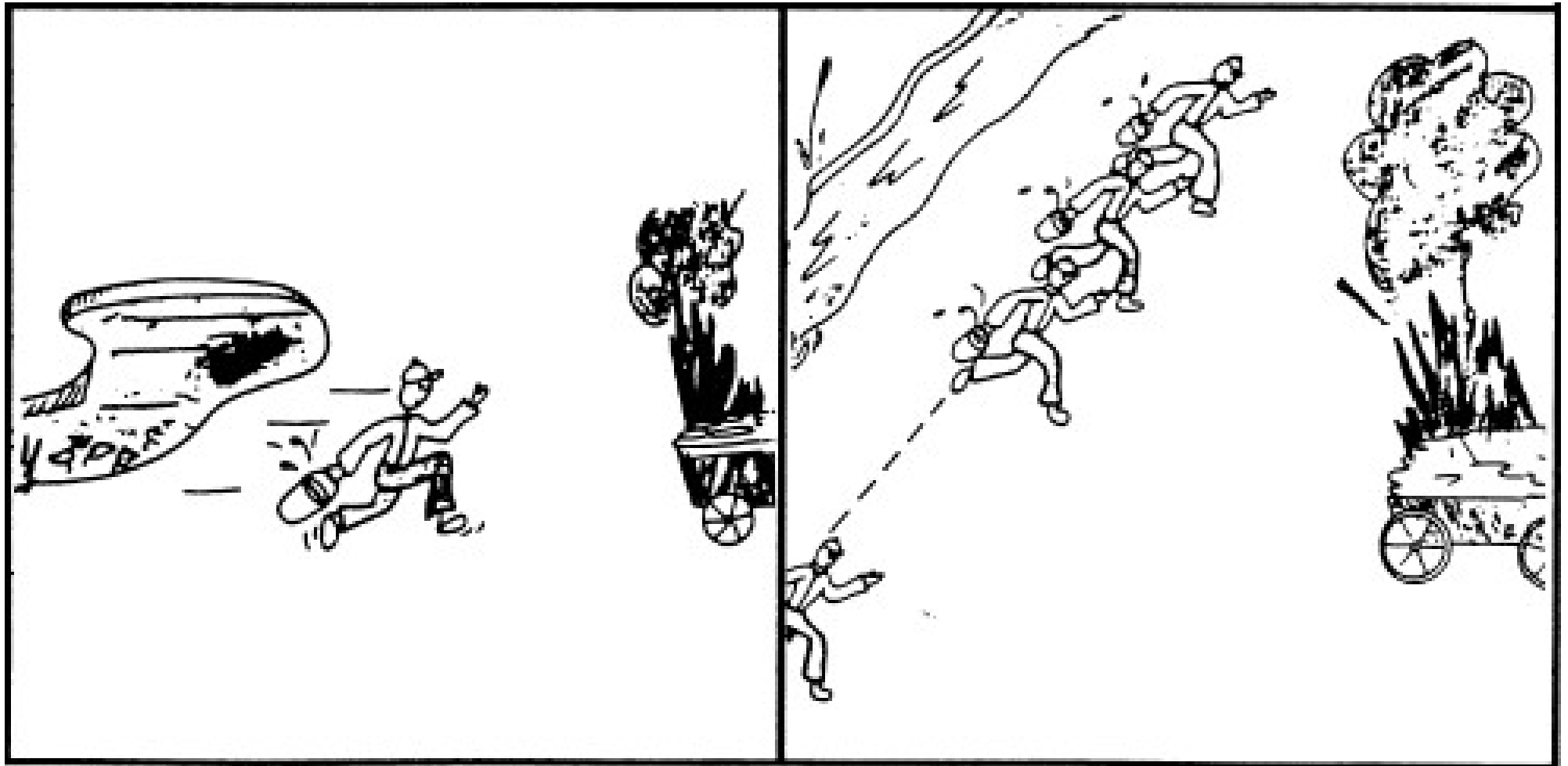
Conhecimento prévio

- ▶ Lógica de programação
- ▶ Linguagem de programação
 - ✓ Fortran
 - ✓ C
- ▶ Estar familiarizado com o ambiente Linux

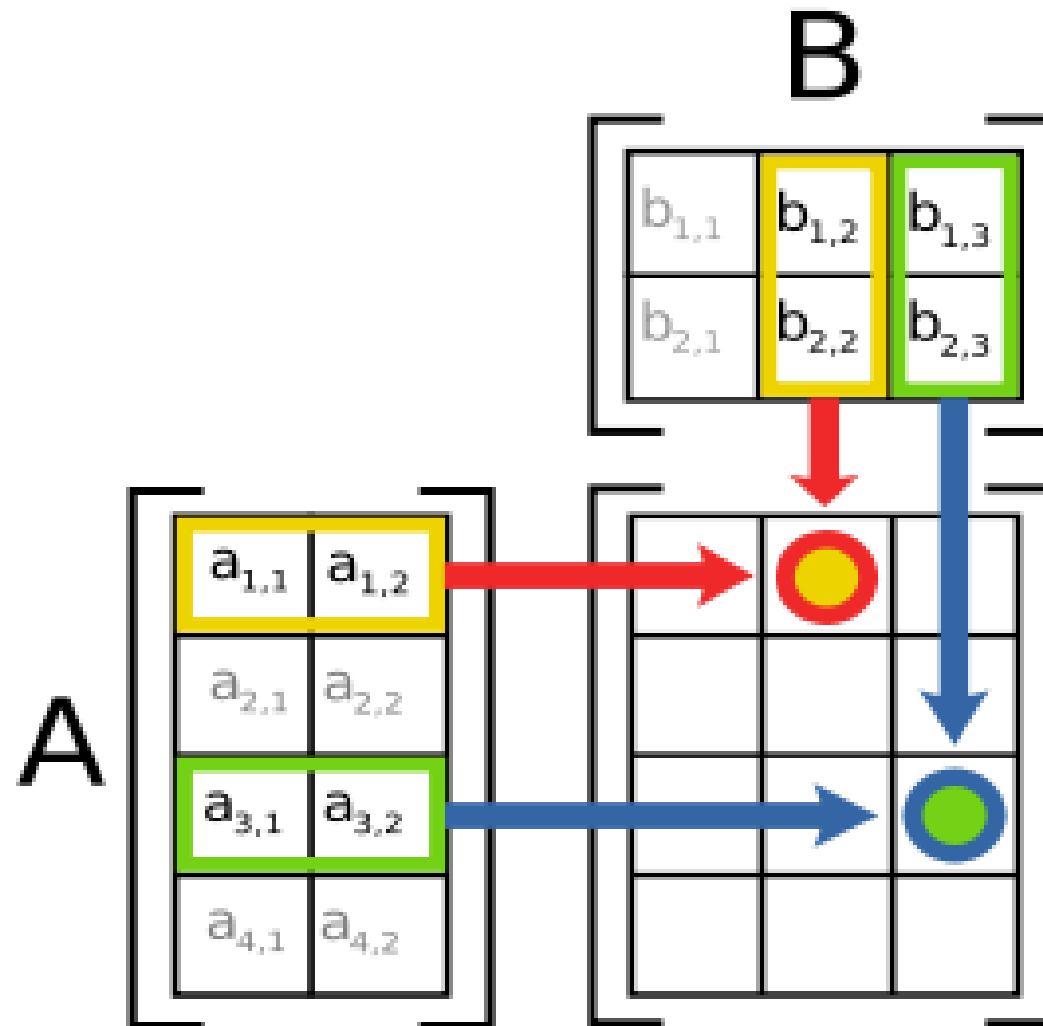
Processamento Paralelo

- ▶ Decomposição do problema em partes menores
- ▶ Processamento simultâneo das partes
 - ✓ Redução do tempo total de execução
- ▶ Reunião dos resultados parciais para obtenção do resultado final

Processamento Paralelo



Processamento Paralelo - Exemplo



Processamento Paralelo - Exemplo

- ▶ Processamento de dados sísmicos pré-empilhamento
 - ✓ Resultado de cada tiro não depende dos demais
 - ✓ Migração Reversa no Tempo (RTM)
 - ✓ 1 nó ~ 7 horas
 - ✓ 120 nós ~ 4min 30s

Processamento Paralelo

► Hardware paralelo

- ✓ Multicore
- ✓ Cluster
- ✓ FPGA
- ✓ GPGPU
- ✓ etc.

Processamento Paralelo - Cluster



Processamento Paralelo

- ▶ Software paralelo

- ▶ Threads

- ▶ MPI

- ▶ PVM

- ▶ CUDA

- ▶ Intel TBB

- ▶ etc.

Processamento Paralelo

- ▶ Hardware paralelo
 - ✓ Cluster Águia – CPGG/UFBA
- ▶ Software paralelo
 - ✓ MPI – Message Passing Interface
 - ✓ Fortran
 - ✓ C

Cluster Água

▶ ssh cursopp@agua

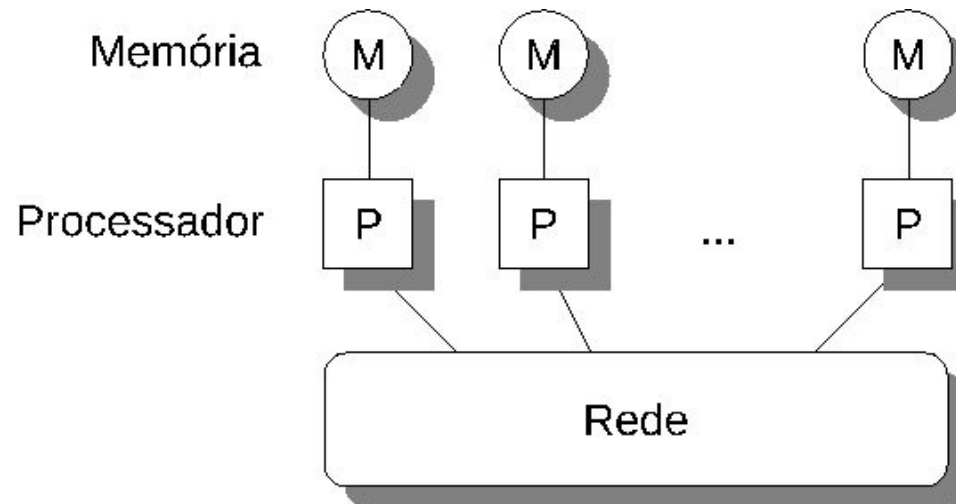
- ✓ Senha: cursopp
- ✓ Pasta individual

▶ 336 nós

- ✓ 24 por aluno

MPI

- ▶ *Message Passing Interface.*
- ▶ Padrão (protocolo) para comunicação de dados em computação paralela.



- ▶ Não é um programa.
- ▶ Possui várias implementações.
 - ▶ MPICH, Open MPI, Intel MPI...

MPI

- ▶ Orientado a mensagens.
- ▶ Todos os nós executam o mesmo programa.
 - ✓ Cada nó possui uma identificação única - rank
- ▶ Independe da linguagem
 - ▶ C e Fortran.

MPI – Estrutura Básica

▶ Inclusão da biblioteca MPI

▶ Inicialização da MPI

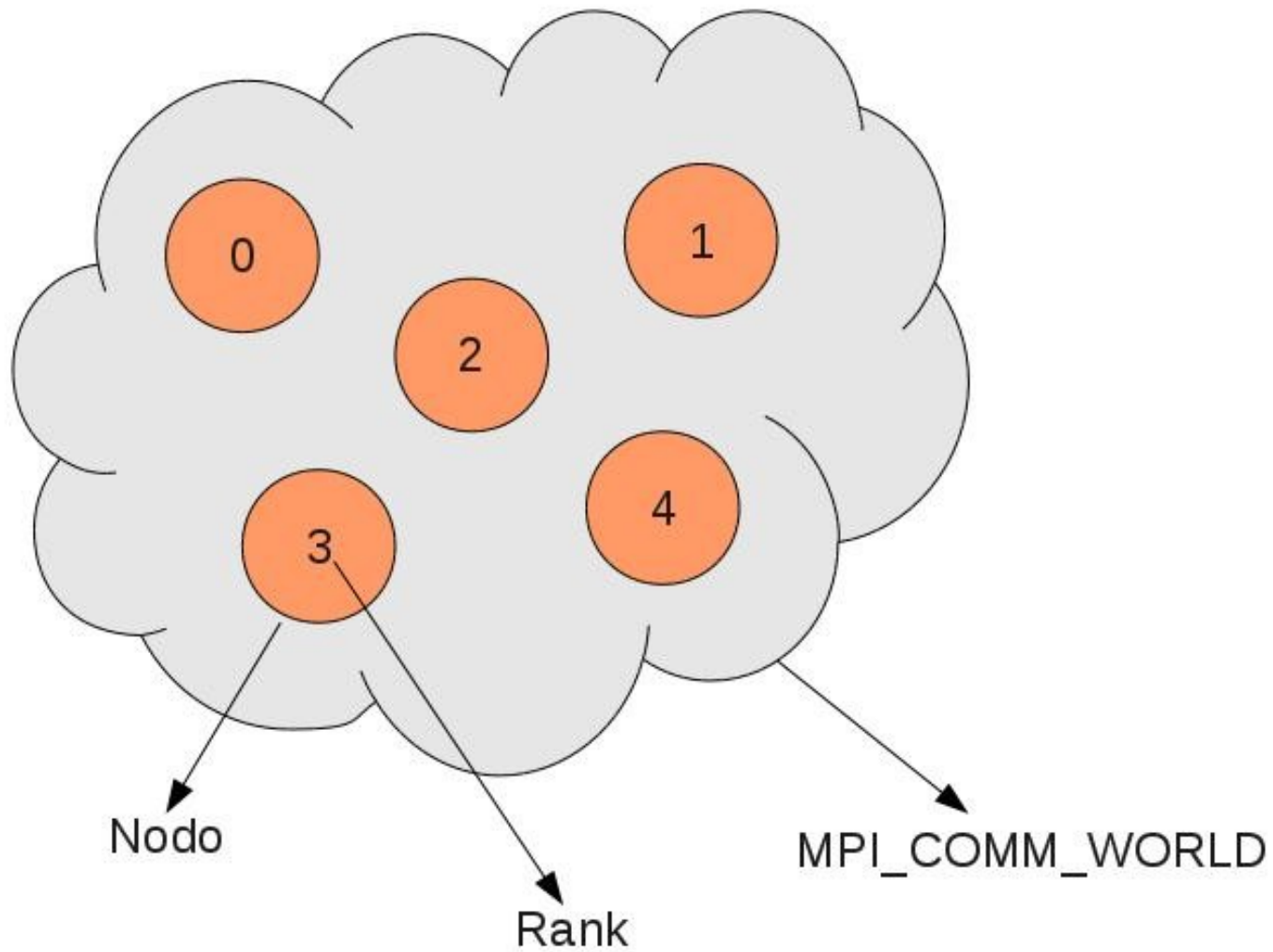
- ✓ `MPI_Init` - Reconhece todos os nós participantes e os reúne em um grupo: `MPI_COMM_WORLD`.
- ✓ `MPI_Comm_rank` - Atribui um identificador a cada nó do grupo
- ✓ `MPI_Comm_size` - Recupera o número de participantes do grupo

▶ Processamento

▶ Finalização da MPI

- ✓ `MPI_Finalize` - Finaliza o processo

MPI – Estrutura Básica



Exemplo – Fortran 90

```
program firstmpi
! inclusao da biblioteca MPI
  use mpi
  integer :: mpierror, mpisize, mpirank

! inicializacao da MPI
  call MPI_Init(mpierror)
  call MPI_Comm_size(MPI_COMM_WORLD, mpisize, mpierror)
  call MPI_Comm_rank(MPI_COMM_WORLD, mpirank, mpierror)

! processamento
  print*,mpirank,mpisize

! finalizacao da MPI
  call MPI_Finalize(mpierror)
end program firstmpi
```

Exemplo - C

```
/*inclusao da biblioteca MPI*/  
#include <mpi.h>  
#include <stdio.h>  
  
int main(int argc,char **argv){  
    int mpirank,mpisize;  
  
    /*inicializacao da MPI*/  
    MPI_Init(&argc, &argv);  
    MPI_Comm_rank(MPI_COMM_WORLD, &mpirank);  
    MPI_Comm_size(MPI_COMM_WORLD, &mpisize);  
  
    /*processamento*/  
    printf("%d %d\n",mpirank,mpisize);  
  
    /*finalizacao da MPI*/  
    MPI_Finalize();  
  
    return(0);  
}
```

Compilação

- ▶ Compiladores específicos
- ▶ `<compilador> <arquivo> -o <executável>`
 - ✓ Ex: `mpif90 exemplo0.f90 -o exemplo0`
- ▶ Fortran 90 – `mpif90`
- ▶ Fortran 77 – `mpif77`
- ▶ C – `mpicc`

Executando no cluster Água

► Script de submissão

```
#!/bin/bash
```

```
#$ -cwd
```

```
#$ -j y
```

```
#$ -S /bin/bash
```

```
#$ -pe orte 8
```

```
#$ -N job-exemplo
```

```
#$ -q o8n16-336.q
```

```
/opt/openmpi/bin/mpirun -v -np $NSLOTS ./exemplo0
```

Executando no cluster Água

▶ qsub script

- ✓ Submete um job na fila, atribuindo um identificador.
- ✓ Mensagens do processo ficam no arquivo `<jobname>.o<id>.`

▶ qstat

- ✓ Mostra o status dos jobs submetidos.

▶ qdel id

- ✓ Remove o job com o identificador especificado.

Exercício

- ▶ Alterar o exemplo visto acima
- ▶ Se o rank for igual a zero
 - ✓ Imprimir a mensagem “Mestre”
- ▶ Se não
 - ✓ Imprimir a mensagem “Escravo”
- ▶ if – else
- ▶ Compilar
- ▶ Executar com 2, 4 e 8 nós

Exercício – Fortran 90

```
program firstmpi
! inclusao da biblioteca MPI
  use mpi
  integer :: mpierror, mpisize, mpirank

! inicializacao da MPI
  call MPI_Init(mpierror)
  call MPI_Comm_size(MPI_COMM_WORLD, mpisize, mpierror)
  call MPI_Comm_rank(MPI_COMM_WORLD, mpirank, mpierror)

! processamento
  if (mpirank == 0) then
    print*, "Mestre"
  else
    print*, "Escravo"
  endif

! finalizacao da MPI
  call MPI_Finalize(mpierror)
end program firstmpi
```

Links úteis

- ▶ <http://www.cs.mtu.edu/~shene/COURSES/cs201/NOTES/fortran.html>
- ▶ http://www-teaching.physics.ox.ac.uk/Unix+Prog/hargrove/tutorial_77/
- ▶ <http://www.ead.cpdee.ufmg.br/cursos/C/>
- ▶ <https://computing.llnl.gov/tutorials/mpi/>