

Exploração do paralelismo

# Processamento Paralelo e Distribuído

Marcelo Trindade Rebonatto

Formas de exploração do paralelismo

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Roteiro

- Formas de exploração do paralelismo
  - Implícito
  - Explícito
- Formas de anotação do paralelismo em programas

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto 2/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Formas

- Implícito
  - Compiladores especiais paralelizantes
  - Exploração automática do paralelismo
- Explícito
  - Paralelismo de Dados
  - Trocas de mensagens (passagem)
  - Variáveis compartilhadas

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto 3/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Implícita

- Programador não necessita explicitar o paralelismo através de comandos
- Escrita de programas: metodologia igual a sequencial
- Compilador realiza a paralelização do código

```

graph LR
    A[Código sequencial] --> B[Compilador Paralelo]
    B --> C[Executável Paralelo]
            
```

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto 4/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Implícita

- O compilador deve:
  - Analisar e compreender as dependências de dados
  - Particionar o programa em blocos e analisar dependências entre os blocos
  - Transformar cada um dos blocos em tarefas a serem executadas em paralelo
- Qualidade do código
  - Depende da aplicação
  - Depende (muito) da qualidade do compilador utilizado

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto 5/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Implícita

- Vantagens
  - Facilidade por parte do programador
  - O trabalho fica a cargo do compilador
  - Possível reaproveitamento de programas sequenciais existentes
- Desvantagens
  - Não exploração total do paralelismo
  - Não podem ser utilizados em qualquer tipo de aplicação (formas específicas do paralelismo)
  - Códigos (geralmente) menos eficientes

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto 6/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Explícita

- Programador deve explicitar o paralelismo através de comandos
  - Cooperação para resolver os problemas
  - Comunicações
- Cuidados na escrita de programas
  - Fluxos de execução diferentes
  - Sincronizações
  - Dependência de dados

Processamento Paralelo e Distribuído - Prof.: Marcelo Triunfo e Rebonatto
7/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Explícita

- Compilador: tarefa simples
  - Linguagens especiais
  - Bibliotecas
  - Diretivas

```

graph LR
    A[Código paralelo] --> B[Compilador]
    B --> C[Executável Paralelo]
  
```

Processamento Paralelo e Distribuído - Prof.: Marcelo Triunfo e Rebonatto
8/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Explícita

- O programador deve:
  - Analisar e compreender as dependências de dados
  - Particionar o programa em blocos e analisar dependências entre os blocos
  - Transformar cada um dos blocos em tarefas a serem executadas em paralelo
- Qualidade do código
  - Depende da aplicação
  - Depende (muitíssimo) do programador

Processamento Paralelo e Distribuído - Prof.: Marcelo Triunfo e Rebonatto
9/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Explícita

- Vantagens
  - Exploração total (quase) do paralelismo
  - Permite exploração onde fontes de paralelismo não são passíveis de detecção automática
- Desvantagens
  - O trabalho fica a cargo do programador
  - Programação difícil de implementar e depurar
  - Conhecer profundamente a aplicação

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto 10/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Modelos de exploração Explícita

- Paralelismo de dados
  - Mesmo programa
  - Atuação sobre dados diferentes
  - Execução em unidades de processamento diferentes
- SPMD - *Single Program Multiple Data*
  - Primitivas de identificação

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto 11/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Modelos de exploração Explícita

- Trocas de Mensagens ou Passagem de mensagens (Message Passing)
  - Múltiplos processos
  - Espaço de endereçamento separados
  - Comunicação com Send e Receive
  - Suporta paralelismo
    - ✧ Dados: SPMD
    - ✧ Controle: MPMD

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto 12/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Modelos de exploração Explícita

- Variáveis compartilhadas (*shared memory*)
  - Semelhante ao paralelismo de dados
  - Comunicação
    - ✧ Leitura e escrita em variáveis compartilhadas
    - ✧ Controle da Sincronização
  - Acesso a dados

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto
13/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Explícita

- Formas de anotação do paralelismo
  - Diretivas ao compilador
  - Linguagem seqüencial com chamadas de sistemas (*system calls*)
  - Linguagem seqüencial com chamadas de bibliotecas (*library calls*)
  - Linguagem de programação paralela

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto
14/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Explícita

- Diretivas ao compilador
  - O programador insere no código fonte diretivas para indicar ao compilador que ações devem ser executadas em paralelo
  - O compilador deve suportar diretivas especiais
  - Divide o trabalho
    - ✧ Simplifica o trabalho do compilador
    - ✧ Simplifica o trabalho do programador
  - Ex. Power C
    - ✧ Linguagem
    - ✧ Compilador

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto
15/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Diretivas ao compilador

```
#pragma parallel
#pragma shared (total_pi)
#pragma local (i, local_pi)
{
    #pragma pfor iterate (i=0; 1000000; 1){
        for (i=0; i<1000000; i++) {
            local_pi = local_pi + (4.0 / (1.0 + i * i));
        }
    }
    #pragma critical {
        total_pi = total_pi + local_pi;
    }
}
```

Processamento Paralelo e Distribuído - Prof.: ~~Tristão~~ Marcelo Tristão e Rebonatto 16/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Explícita

- Linguagem seqüencial com chamadas de sistema (*system calls*)
  - Utilização da linguagem C
  - Utilização do comando fork
    - ✧ Criação de um processo filho
- Difícil controle e depuração
- Apenas uma camada de software: bom desempenho
- Pouco utilizado em programação paralela

Processamento Paralelo e Distribuído - Prof.: ~~Tristão~~ Marcelo Tristão e Rebonatto 17/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## System calls

```
pid = fork();
if (pid == 0)
    printf("Eu sou um processo filho\n");
else
    if (pid > 0)
        printf("Eu sou um pai, e possuo um filho com pid %d\n",pid);
    else
        printf("Erro na criacao de processo filho");
```

Processamento Paralelo e Distribuído - Prof.: ~~Tristão~~ Marcelo Tristão e Rebonatto 18/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Explícita

- Linguagem seqüencial com chamadas de bibliotecas (*library calls*)
  - Utilização da linguagem C (Fortran)
  - Chamadas a primitivas de bibliotecas específicas
    - ✧ SEND, RECEIVE
    - ✧ LOCK, UNLOCK
  - Duas camadas de software
    - ✧ Bibliotecas fazem mapeamento com *systems calls*
    - ✧ Maior ineficiência

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto 19/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Explícita

- Linguagem seqüencial com chamadas de bibliotecas (*library calls*)
  - Muito utilizadas em programação paralela distribuída
  - Portabilidade

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto 20/25

---

---

---

---

---

---

---

---

Exploração do paralelismo

## Exploração Explícita

- Linguagem seqüencial com chamadas de bibliotecas (*library calls*): Exemplos
  - Com trocas de mensagens (MP)
    - ✧ PVM (Parallel Virtual Machine)
    - ✧ MPI (Messaging Passing Interface)
  - Memória Compartilhada/Distribuída
    - ✧ pthreads
    - ✧ Treadmarks (SDSM)
  - Com MP e DSM
    - ✧ Athapscan0

Processamento Paralelo e Distribuído - Prof.: Marcelo Trindade Rebonatto 21/25

---

---

---

---

---

---

---

---

## Library Calls - MPI

```
#include <stdio.h>
#include <mpi.h>

int main (int argc, char **argv) {
    MPI_Status status;

    MPI_Init(&argc, &argv);
    . . .
    MPI_Send(msg, strlen(msg)+1, MPI_CHAR, 0, tag,
             MPI_COMM_WORLD);
    . . .
    for (i = 1; i < np; i++)
        MPI_Recv(msg, 100, MPI_CHAR, i, tag,
                 MPI_COMM_WORLD, &status);
    MPI_Finalize();
}
```

Processamento Paralelo e Distribuído - Prof.: *Marcelo Trindade* e Rebonatto 22/25

## Library Calls - pthreads

```
#include <stdio.h>
#include <pthread.h>

int x;          /* variavel global compartilhada */
pthread_mutex_t tranca;

int main() {
    int a;
    pthread_t thid[3];

    pthread_mutex_lock( &tranca );
    . . .
    a = x;
    pthread_mutex_unlock( &tranca );
}
```

Processamento Paralelo e Distribuído - Prof.: *Marcelo Trindade* e Rebonatto 23/25

## Exploração Explícita

- Linguagem de programação paralela
  - A própria linguagem possui primitivas para criar e gerenciar o paralelismo
  - Deve possuir compatibilidade com ambiente paralelo
    - ✧ Hardware
    - ✧ S.O.
    - ✧ Linguagem
    - ✧ Compilador
  - Ambiente adequado a programação paralela
  - Pouca portabilidade

Processamento Paralelo e Distribuído - Prof.: *Marcelo Trindade* e Rebonatto 24/25



## Linguagem Paralela - CPAR

```
shared double total_pi = 0;
void main() {
    int i;
    double local_pi;
    semaforo semaforo;

    create_sem (&semaforo, 1);
    forall i = 0 to 100000 {
        local_pi = local_pi + (4.0 / (1.0 + i * i));
        lock (&semaforo);
        total_pi = total_pi + local_pi;
        unlock (&semaforo);
    }
}
```

Processamento Paralelo e Distribuído - Prof.: *Marcelo Trindade Rebonatto* 25/25

---

---

---

---

---

---

---

---