

Introdução ao Processamento Paralelo e Distribuído



Videoaula

Notas dos slides

APRESENTAÇÃO

O presente conjunto de slides pertence à coleção produzida para a disciplina *Introdução ao Processamento Paralelo e Distribuído* ofertada aos cursos de bacharelado em Ciência da Computação e em Engenharia da Computação pelo Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas.

Os slides disponibilizados complementam as videoaulas produzidas e tratam de pontos específicos da disciplina. Embora tenham sido produzidos para ser assistidos de forma independente, a sequência informada reflete o encadeamento dos assuntos no desenvolvimento do conteúdo programático previsto para a disciplina.



2

Programação Multithread

Tratando questões de contenção

Notas da videoaula

DESCRIÇÃO

Esta videoaula trata de questões relacionadas à contenção promovida pelo uso de mecanismos de sincronização.

OBJETIVOS

Nesta videoaula o aluno compreenderá impacto, no desempenho, do uso de mecanismos de sincronização e conhecerá alternativas para mitigar estes efeitos no desempenho final de execução de seu programa.

4

“ **Stop in the name of love
Before you break my heart**

Diana Ross, The Supremes

5

Coordenação é necessária, mas a sincronização é?

Controlar o acesso a dados compartilhados é vital para que os fluxos de execução concorrente de um programa manipulem corretamente os dados. Os mecanismos de sincronização, como os que promovem execuções em regime de exclusão mútua ou controlam o fluxo cumprem seu papel em efetivar este controle.



6

Coordenação é necessária, mas a sincronização é?

Controlar o acesso a dados compartilhados é vital para que os fluxos de execução concorrente de um programa manipulem corretamente os dados. Os mecanismos de sincronização, como os que promovem execuções em regime de exclusão mútua ou controlam o fluxo cumprem seu papel em efetivar este controle.



7

Entendendo um pouco mais sobre sincronização

Usualmente, operações de sincronização, como aquelas sobre um mutex ou um semáforo, requerem chamadas de serviços oferecidos pelo SO e implicam em atraso na execução da ut que solicita este serviço devido:

- A mudança do modo de execução (modo usuário para modo sistema e então modo usuário novamente);
- Ao tempo que a ut solicitante permanece bloqueada aguardando a liberação do recurso solicitado;
- A troca de contexto do fluxo de execução responsável pela ut que requer a sincronização.



8

Alternativas

Spin lock

Thread-aware

Instruções atômicas

Algoritmos não bloqueantes

“No more waiting, no more, aching...
No more fighting, no more, trying...”

The Cor

9

Alternativas: Instruções Atômicas

Instruções atômicas executam manipulam uma palavra em regime de exclusão mútua. Exemplos:

```
{
    int x = 1, y;
    y = fetch_and_add( &x, 10 );
    // x == 11 e y == 1
}

{
    int x = 1, y;
    y = add_and_fetch( &x, 10 );
    // x == 11 e y = 11
}
```

Fetch add
Fetch sub
Fetch or
Fetch and
Fetch xor
Fetch nand
Compare and Swap
Test and Set
Lock release

10

Alternativas: Instruções Atômicas

Instruções atômicas executam manipulam uma palavra em regime de exclusão mútua. Exemplos:

```
{
    int antigo, novo, teste;
    bool estado;
    estado = cas(&antigo, teste, novo);
}
```

Compara o valor antigo com teste. Se forem iguais, antigo recebe novo e a função retorna true. Caso sejam diferentes, o valor antigo permanece e a função retorna false.

Fetch add
Fetch sub
Fetch or
Fetch and
Fetch xor
Fetch nand
Compare and Swap
Test and Set
Lock release

11

Alternativas: Thread-aware

Um processo multithread é dito **thread-safe** quando o controle do acesso aos dados é promovido de forma satisfatória. A execução multithread é segura, portanto.

Um processo multithread é dito **thread-aware** quando, além de da execução ser segura, o projeto do algoritmo considera a existência de múltiplos threads, sendo reduzido o potencial de dependências entre eles.



12

Alternativas: Thread-aware

Exemplo

Considere uma biblioteca que realiza operações de E/S, com manipulação de arquivos ou comunicação entre processos. As operações de E/S são essencialmente mutuamente exclusivas no acesso aos recursos. Para evitar que uma chamada aos serviços desta biblioteca bloqueie um thread no aguardo do tratamento da solicitação de outro thread, os pedidos podem ser enfileirados como requisições para posterior tratamento.



13

Alternativas: Algoritmos não Bloqueantes

Outra alternativa é ter uma brilhante ideia para conceber uma estratégia de implementação não bloqueante. As técnicas utilizadas fazem uso de instruções atômicas e, muitas vezes, consideram que o custo de recalcular um dado compensa os tempos de latência economizados com as menores taxas de sincronização.



14

Alternativas: Algoritmos não Bloqueantes

Soluções deste tipo são reais. Vários algoritmos podem ser adaptados e apresentar resultado correto, ou, ao menos satisfatório, utilizando algoritmos adaptados para serem livres de bloqueio. Existem vários algoritmos já desenvolvidos para manipulação de estruturas de dados, como listas encadeadas e tabelas has.



15



16