

# MAC 0219/5742 – Introdução à Computação Concorrente, Paralela e Distribuída

Prof. Dr. Alfredo Goldman  
MiniEP1 - Contenção versão 1.0

**Monitores:** Giuliano Belinassi e Matheus Tavares

## 1 Introdução

Em computação paralela, contenção (ou `thread contention`) é o nome dado para a situação em que uma ou mais thread(s) se encontra(m) impossibilitada(s) de continuar pois aguarda(m) um recurso compartilhado que atualmente está sendo utilizado por uma outra thread. O tempo que uma thread passa em contenção trata-se de um tempo perdido, que poderia estar sendo usado para alguma tarefa útil.

Um exemplo de contenção é o caso em que muitas threads chegam em uma seção crítica ao mesmo tempo. Como apenas uma pode executar a região, simultaneamente, todas as demais ficarão aguardando até que a região seja liberada. Este caso também é chamado de `lock contention`.

Neste miniEP, tentaremos empregar uma técnica para redução de contenção em um programa de exemplo, contendo uma seção crítica.

## 2 Software fornecido

### 2.1 Compilar e Rodar

Foi fornecido no PACA o código `contention.c`. Este código aloca e preenche aleatoriamente um vetor com  $M$  `doubles`, aplica uma função custosa<sup>1</sup> (em termos de tempo) sobre cada elemento e então, calcula o valor máximo do vetor. Todo o trabalho em cima do vetor é dividido entre  $N$  threads, da forma mais uniforme possível.

---

<sup>1</sup>Esta função tem única finalidade de ocupar as threads por mais tempo, para que melhor possamos observar os efeitos das mudanças que vamos aplicar.

Compile o código com:

```
$ make
```

E execute com:

```
$ ./contention <num_threads> <array_size>
```

Onde `num_threads` é a quantidade de threads a serem criadas para o trabalho (que deve ser  $\geq 1$ ) e `array_size` é o tamanho do vetor. A saída do programa será o tempo de parede transcorrido desde a criação das threads até a finalização, em segundos.

Certifique-se de que você entendeu o código e execute algumas vezes antes de prosseguir.

## 2.2 Técnica para reduzir contenção

Note, que para cada iteração em `thread_work`, as threads precisam adquirir o `lock` antes de checar se o elemento daquela iteração é maior que o máximo encontrado até o momento. Sabemos, porém, que a razão de vezes em que essa checagem resulta em `true`, não é necessariamente muito alta. Sendo assim, muitas threads acabam tendo que esperar, na contenção, sendo que nem sequer irão alterar a variável global `max`.

Se fizermos, porém, a checagem antes de entrar na seção crítica, podemos evitar entrar nela quando o resultado for falso. Assim, evitamos que a thread fique na contenção desnecessariamente e ela pode prosseguir para a próxima iteração. É possível inserir essa checagem compilando com `make IF=1`. Essa opção inserirá um `if`, tal como o presente na seção crítica, imediatamente antes da chamada de `pthread_mutex_lock`.

Rode com e sem o `IF=1`. Houve diferença de tempo? E se inserisemos mais `ifs` encadeados antes da seção crítica? Será que o tempo de contenção seria mitigado mais ainda? Você pode compilar com `IF=K`, trocando `K` pelo número de `ifs` que deseja inserir.

## 3 Relatório

Você deverá produzir um relatório em formato `.txt` ou `.pdf` contendo os seguintes itens:

1. Faça testes variando o tamanho do vetor, a quantidade de threads e a quantidade de `ifs` encadeados, mostrando medias e intervalos de confiança dos tempos impressos na saída.

2. Dê um parecer do que você observou nos testes do item anterior. Porque você acha que ocorreu o observado?
3. Explique porque não podemos eliminar o `if` de dentro da seção crítica quando adicionamos o `if` de fora.
4. **[Bonus]** Este item é optativo e valerá **até 2 pontos a mais** no miniEP: Pense em uma forma que poderíamos alterar o código `contention.c` para imprimir ao final, não só o tempo transcorrido, mas também uma estimativa da porcentagem deste tempo que foi gasto em contenção. Escreva sua ideia no relatório e envie junto o código alterado, se você implementar a ideia.

Seu relatório pode (e é aconselhável) conter imagens, gráficos e/ou tabelas.

## 4 Entrega

Deverá ser entregue um pacote no PACA com uma pasta com o nome e o sobrenome do estudante que o submeteu no seguinte formato: `nome_sobrenome.zip`. Essa pasta deve ser comprimida em formato ZIP e deve conter:

- O relatório em formato `.txt` ou `.pdf`. Arquivos em formato `.doc`, `.docx` ou `.odt` não serão aceitos.
- (opcional) Quaisquer imagens que você queira anexar
- (opcional) O código alterado do item 4., se você desejar

Em caso de dúvidas, use o fórum de discussão do Paca. A data de entrega deste Exercício Programa é até às **08:00h do dia 09 de abril**. Não se esqueça que o professor irá sortear um aluno para explicar sua implementação do miniEP neste mesmo dia. Não é necessário preparar slides, apenas falar brevemente sobre o código e as decisões de projeto tomadas.

*[/Boa Sorte!*