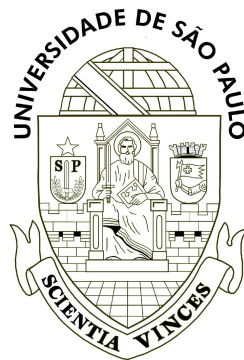


**Universidade de São Paulo**  
Instituto de Matemática e Estatística

MAC0219 - Programação Concorrente e Paralela  
2023



*Mini EP 7 - Contenção*

Beatriz Viana Costa

## Conteúdo

1. Questão 1	3
2. Questão 2	4
3. Questão 3	4

## 1. Questão 1

Para a realização dos testes o programa foi compilado de 5 maneiras:

- (1) *make*
- (2) *make IF=1*
- (3) *make IF=2*
- (4) *make IF=5*
- (5) *make IF=10*

Dessa maneira, pudemos observar as simulações sem nenhum *if* para além do presente na seção crítica, com 1, 2, 5 e 10 *ifs*.

Os testes foram feitos para a quantidade de *threads* variando entre 10 e 10.000 e o tamanho do *array* variando entre 10 e 100.000.000. A cada teste aumentávamos estes valores em uma potência de 10. O tempos apresentados são resultado da média de 10 execuções, em segundos.

RODANDO NORMAL (SEM IF)							
NUM THREADS / TAM ARRAY	10	100	1000	10000	100000	1000000	10000000
10	0,00435	0,00454	0,00679	0,07084	0,31673	1,88486	22,18982
100	0,03375	0,03377	0,0381	0,06958	0,18309	1,86277	28,86521
1000	0,27595	0,24067	0,17072	0,26963	0,51404	1,93493	32,67997
10000	1,29251	1,37083	1,254	1,34479	2,23066	4,29194	30,73628

RODANDO COM IF=1							
NUM THREADS / TAM ARRAY	10	100	1000	10000	100000	1000000	10000000
10	0,00397	0,00171	0,00223	0,00576	0,06332	0,76255	8,07702
100	0,03177	0,01209	0,01925	0,01725	0,11091	0,39093	9,41263
1000	0,25031	0,13204	0,15144	0,25921	0,20727	0,49631	10,73419
10000	1,29794	1,4228	1,40149	1,39269	1,59176	2,80652	12,07325

RODANDO COM IF=2							
NUM THREADS / TAM ARRAY	10	100	1000	10000	100000	1000000	10000000
10	0,00384	0,00152	0,00273	0,00576	0,06303	0,65766	11,19101
100	0,03779	0,01195	0,0297	0,01587	0,10805	0,44353	10,66027
1000	0,25381	0,19805	0,22856	0,301	0,22924	0,57639	10,04099
10000	1,35905	1,49074	1,39501	1,5313	1,75514	3,22064	11,6003

RODANDO COM IF=5							
NUM THREADS / TAM ARRAY	10	100	1000	10000	100000	1000000	10000000
10	0,00434	0,00161	0,00242	0,00577	0,07492	0,77265	11,88026
100	0,03037	0,01726	0,01325	0,01761	0,115	0,40679	10,86873
1000	0,25161	0,18354	0,15652	0,22414	0,35351	0,49663	10,16301
10000	1,5735	1,28891	1,31189	2,10621	1,86815	2,31463	10,7567

RODANDO COM IF=10							
NUM THREADS / TAM ARRAY	10	100	1000	10000	100000	1000000	10000000
10	0,00515	0,00182	0,00263	0,00566	0,06498	0,86338	10,6506
100	0,03528	0,01423	0,03042	0,02055	0,11365	0,42099	10,39222
1000	0,20606	0,12475	0,18509	0,26361	0,38491	0,6602	10,42446
10000	1,33901	1,40953	1,52074	1,2846	1,91339	2,74761	11,38326

## 2. Questão 2

Foi possível observar uma grande melhora dos tempos ao adicionarmos um *if* ao nosso código, a partir daí, comparando o desempenho dos códigos com 1, 2, 5 e 10 *ifs* a mais, os tempos de execução foram muito similares, havendo no máximo diferença de um segundo nos casos em que o vetor era de tamanho na ordem de  $10^8$ .

Assim, é possível notar que ao adicionarmos um *if*, Já diminuímos consideravelmente a contenção em nosso código, isso porquê este *if* exterior já é o suficiente para liberar *threads* que carregam valores que são menores que *max*.

## 3. Questão 3

Pois o *if* exterior serve como uma "pré-seleção" (ou uma verificação dupla) dos maiores valores do vetor, então já nesse *if* algumas *threads* são barradas, logo são liberadas para acessarem outros espaços do vetor e assim continuar o programa. Ou seja, esta verificação dupla permite o aumento da concorrência e diminui a invalidação de caches.

O *if* da região crítica não pode ser retirado na presença de um *if* exterior, pois não é garantido pela condicional de fora que somente uma *thread* irá tentar alterar o valor de *max*. Isto significa, pode ser que mais de uma *thread* pelos *ifs* exteriores, e ao chegar na seção crítica, o valor correto de *max* seja atribuído, porém depois ele seja alterado novamente pela *thread* que ficou em espera para o valor incorreto.