

# **Computação Concorrente (ICP117)**

## **Laboratório 2**

Aluno: Pedro Henrique Figueiredo Von Zuben  
DRE: 119055699

**Os testes foram feitos na máquina virtual "Oracle VM VirtualBox" utilizando 2 processadores e 2 threads (1 por processador).**

### **Conclusão dos testes mostrados abaixo:**

Como podemos perceber, a utilização de apenas uma thread faz com que o tempo da multiplicação concorrente seja bem parecido com o tempo da multiplicação sequencial ou até mesmo maior que ele. Isto acontece, pois não há ganho de desempenho ao utilizarmos o método concorrente se não houver ao menos mais que uma thread, já que não haverá como realizar o paralelismo. Além disso, o tempo de criar as threads e alocar elas acaba sendo mais custoso que apenas realizar a multiplicação sequencial.

Desta forma, vemos que ao utilizarmos duas threads, o tempo de execução da multiplicação concorrente praticamente se torna metade do tempo de execução da multiplicação sequencial. Isto porque efetivamente dobramos a quantidade de threads e com isto conseguimos utilizarmos do ganho de desempenho vindo do paralelismo (aceleração).

**Dimensão = 500, número de threads = 1:**

- **Teste 1:**

- Tempo da multiplicação sequencial: 0.697288
- Tempo da multiplicação concorrente: 0.694384
- Ganho de desempenho com a versão concorrente (aceleração): 1.004182

- **Teste 2:**

- Tempo da multiplicação sequencial: 0.686850
- Tempo da multiplicação concorrente: 0.680237
- Ganho de desempenho com a versão concorrente (aceleração): 1.009722

- **Testa 3:**

- Tempo da multiplicação sequencial: 0.695290
- Tempo da multiplicação concorrente: 0.674030
- Ganho de desempenho com a versão concorrente (aceleração): 1.031542

**Dimensão = 500, número de threads = 2:**

- **Teste 1:**

- Tempo da multiplicação sequencial: 0.680663
- Tempo da multiplicação concorrente: 0.398355
- Ganho de desempenho com a versão concorrente (aceleração): 1.708683

- **Teste 2:**

- Tempo da multiplicação sequencial: 0.682784
- Tempo da multiplicação concorrente: 0.444241
- Ganho de desempenho com a versão concorrente (aceleração): 1.536969

- **Teste 3:**

- Tempo da multiplicação sequencial: 0.700313
- Tempo da multiplicação concorrente: 0.432945
- Ganho de desempenho com a versão concorrente (aceleração): 1.617557

**Dimensão = 1000, número de threads = 1:**

- **Teste 1:**

- Tempo da multiplicação sequencial: 5.541872
- Tempo da multiplicação concorrente: 5.996969
- Ganho de desempenho com a versão concorrente (aceleração): 0.924112

- **Teste 2:**

- Tempo da multiplicação sequencial: 5.538862
- Tempo da multiplicação concorrente: 5.667341
- Ganho de desempenho com a versão concorrente (aceleração): 0.977330

- **Teste 3:**

- Tempo da multiplicação sequencial: 5.552184
- Tempo da multiplicação concorrente: 5.589028
- Ganho de desempenho com a versão concorrente (aceleração): 0.993408

**Dimensão = 1000, número de threads = 2:**

- **Teste 1:**

- Tempo da multiplicação sequencial: 5.568556
- Tempo da multiplicação concorrente: 3.056071
- Ganho de desempenho com a versão concorrente (aceleração): 1.822129

- **Teste 2:**

- Tempo da multiplicação sequencial: 5.572304
- Tempo da multiplicação concorrente: 3.074976
- Ganho de desempenho com a versão concorrente (aceleração): 1.812146

- **Teste 3:**

- Tempo da multiplicação sequencial: 5.601981
- Tempo da multiplicação concorrente: 2.972738
- Ganho de desempenho com a versão concorrente (aceleração): 1.884452

**Dimensão = 2000, número de threads = 1:**

- **Teste 1:**

- Tempo da multiplicação sequencial: 84.059840
- Tempo da multiplicação concorrente: 88.191477
- Ganho de desempenho com a versão concorrente (aceleração): 0.953152

- **Teste 2:**

- Tempo da multiplicação sequencial: 80.067713
- Tempo da multiplicação concorrente: 86.674785
- Ganho de desempenho com a versão concorrente (aceleração): 0.923772

- **Teste 3:**

- Tempo da multiplicação sequencial: 83.170765
- Tempo da multiplicação concorrente: 88.299393
- Ganho de desempenho com a versão concorrente (aceleração): 0.941918

**Dimensão=2000, número de threads = 2:**

- **Teste 1:**

- Tempo da multiplicação sequencial: 82.452058
- Tempo da multiplicação concorrente: 44.617256
- Ganho de desempenho com a versão concorrente (aceleração): 1.847986

- **Teste 2:**

- Tempo da multiplicação sequencial: 80.549788
- Tempo da multiplicação concorrente: 43.275922
- Ganho de desempenho com a versão concorrente (aceleração): 1.861307

- **Teste 3:**

- Tempo da multiplicação sequencial: 83.428389
- Tempo da multiplicação concorrente: 44.207262
- Ganho de desempenho com a versão concorrente (aceleração): 1.887210