

Laboratório 2 – Computação Concorrente

Aluno: Carlos Cozzolino

DRE: 115086800

- **Medições para a função sequencial:**

Para 500:

Primeira

Inicialização: 4.716661

Multiplicação: 1.001695

Finalização: 0.000218

Segunda

Inicialização: 9.603151

Multiplicação: 1.065125

Finalização: 0.000288

Terceira

Inicialização: 2.586934

Multiplicação: 1.1275074

Finalização: 0.000167

Quarta

Inicialização: 1.858710

Multiplicação: 0.911724

Finalização: 0.000216

Quinta

Inicialização: 1.546993

Multiplicação: 1.152818

Finalização: 0.000223

Para 1000:

Primeira

Inicialização: 3.178810

Multiplicação: 7.674710

Finalização: 0.001615

Segunda

Inicialização: 1.610613

Multiplicação: 8.126327

Finalização: 0.001588

Terceira

Inicialização: 1.699137

Multiplicação: 6.928427

Finalização: 0.001110

Quarta

Inicialização: 1.483163

Multiplicação: 6.946913

Finalização: 0.001643

Quinta

Inicialização: 1.350665

Multiplicação: 6.409893

Finalização: 0.001639

Para 2000:

Primeira

Inicialização: 7.650834

Multiplicação: 83.084084

Finalização: 0.005645

Segunda

Inicialização: 8.152788

Multiplicação: 91.649027

Finalização: 0.005979

Terceira

Inicialização: 2.304954

Multiplicação: 91.335211

Finalização: 0.005626

Quarta

Inicialização: 1.931017

Multiplicação: 83.262091

Finalização: 0.005827

Quinta

Inicialização: 1.344357

Multiplicação: 85.359164

Finalização: 0.007890

- **Medições para função concorrente:**

1 thread para 500:

Primeira

Inicialização: 3.070094

Multiplicação: 1.028372

Finalização: 0.000303

Segunda

Inicialização: 2.828393

Multiplicação: 1.070525

Finalização: 0.000296

Terceira

Inicialização: 2.251155

Multiplicação: 1.424595

Finalização: 0.000218

Quarta

Inicialização: 3.562648

Multiplicação: 1.139149

Finalização: 0.000287

Quinta

Inicialização: 2.986929

Multiplicação: 1.200584

Finalização: 0.000288

1 thread para 1000:

Primeira

Inicialização: 6.034264

Multiplicação: 9.026593

Finalização: 0.002110

Segunda

Inicialização: 3.162495

Multiplicação: 9.057536

Finalização: 0.002145

Terceira

Inicialização: 2.001228

Multiplicação: 8.217202

Finalização: 0.001217

Quarta

Inicialização: 1.369860

Multiplicação: 8.787150

Finalização: 0.001531

Quinta

Inicialização: 1.601809

Multiplicação: 7.880349

Finalização: 0.002069

1 thread para 2000:

Primeira

Inicialização: 2.160622

Multiplicação: 93.320828

Finalização: 0.007273

Segunda

Inicialização: 1.832509

Multiplicação: 99.094566

Finalização: 0.007956

Terceira

Inicialização: 2.361244

Multiplicação: 102.106016

Finalização: 0.007799

Quarta

Inicialização: 1.656839

Multiplicação: 97.563582

Finalização: 0.007604

Quinta

Inicialização: 1.856621

Multiplicação: 100.794840

Finalização: 0.007705

2 threads para 500:

Primeira

Inicialização: 2.178955

Multiplicação: 0.837082

Finalização: 0.000223

Segunda

Inicialização: 1.290934

Multiplicação: 0.726880

Finalização: 0.000219

Terceira

Inicialização: 1.267038

Multiplicação: 0.792824

Finalização: 0.000217

Quarta

Inicialização: 1.346936

Multiplicação: 0.754492

Finalização: 0.000200

Quinta

Inicialização: 1.114668

Multiplicação: 0.750811

Finalização: 0.000219

2 threads para 1000:

Primeira

Inicialização: 1.914059

Multiplicação: 5.076184

Finalização: 0.001923

Segunda

Inicialização: 1.762183

Multiplicação: 5.277849

Finalização: 0.001926

Terceira

Inicialização: 2.954263

Multiplicação: 5.424799

Finalização: 0.002030

Quarta

Inicialização: 2.378558

Multiplicação: 5.566919

Finalização: 0.001315

Quinta

Inicialização: 1.618633

Multiplicação: 4.628956

Finalização: 0.001342

2 threads para 2000:

Primeira

Inicialização: 3.480844

Multiplicação: 52.740490

Finalização: 0.007890

Segunda

Inicialização: 2.958970

Multiplicação: 56.106439

Finalização: 0.006009

Terceira

Inicialização: 1.880946

Multiplicação: 58.613352

Finalização: 0.007766

Quarta

Inicialização: 1.840421

Multiplicação: 59.389807

Finalização: 0.006806

Quinta

Inicialização: 1.785535

Multiplicação: 61.319489

Finalização: 0.007642

4 threads para 500:

Primeira

Inicialização: 1.475326

Multiplicação: 0.413115

Finalização: 0.000220

Segunda

Inicialização: 1.387118

Multiplicação: 0.389298

Finalização: 0.000219

Terceira

Inicialização: 1.107207

Multiplicação: 0.398832

Finalização: 0.000218

Quarta

Inicialização: 1.443004

Multiplicação: 0.420423

Finalização: 0.000228

Quinta

Inicialização: 1.203081

Multiplicação: 0.455365

Finalização: 0.000228

4 threads para 1000:

Primeira

Inicialização: 2.194117

Multiplicação: 2.969063

Finalização: 0.001961

Segunda

Inicialização: 1.802123

Multiplicação: 3.101613

Finalização: 0.001528

Terceira

Inicialização: 1.826379

Multiplicação: 2.909364

Finalização: 0.001306

Quarta

Inicialização: 1.699057

Multiplicação: 2.995121

Finalização: 0.001378

Quinta

Inicialização: 1.682725

Multiplicação: 2.947458

Finalização: 0.001952

4 threads para 2000:

Primeira

Inicialização: 1.680146

Multiplicação: 25.902066

Finalização: 0.007622

Segunda

Inicialização: 5.968596

Multiplicação: 26.313187

Finalização: 0.007617

Terceira

Inicialização: 1.736255

Multiplicação: 27.452027

Finalização: 0.007016

Quarta

Inicialização: 1.704571

Multiplicação: 25.923455

Finalização: 0.007625

Quinta

Inicialização: 1.785187

Multiplicação: 26.593637

Finalização: 0.007718

Com todos os dados coletados, podemos calcular o ganho de desempenho entre os tempos da função sequencial e da concorrente para N threads.

Os dados utilizados nos cálculos são os tempos em que a função foi mais rápida para cada situação:

Para 500:

1 thread: $(2.7)/(3.6) = 0.75$

2threads: $(2.7)/(1.9) = 1.42$

4threads: $(2.7)/(1.5) = 1.8$

Para 1000:

1 thread: $(7.8)/(9.4) = 0.83$

2threads: $(7.8)/(6.2) = 1.26$

4threads: $(7.8)/(4.6) = 1.7$

Para 2000:

1 thread: $(85.2)/(95.5) = 0.89$

2threads: $(85.2)/(59) = 1.44$

4threads: $(85.2)/(27.5) = 3.1$

Os resultados estão dentro do esperado, visto que a parte onde utilizamos as threads (multiplicação) temos uma aceleração praticamente proporcional ao nº de threads utilizadas, quanto maior o trabalho dado ao programa, melhor podemos ver essa diferença.

Computador do aluno possui um Xeon E5 2630L (8 núcleos, 16 threads).

