



## PROGRAMAÇÃO CONCORRENTE E DISTRIBUÍDA

### ATIVIDADE 3 - MÉTRICAS

Aluna: Rosangela Miyeko Shigenari

RA: 92334

Turma: Integral

A partir da solução paralela para o problema de *N-Body*, desenvolvido no problema 2 da tarefa 2, faça as seguintes medidas:

a) A partir da versão serial do código fornecido, meça a fração (percentual) de tempo que o código executa apenas tarefas sequenciais, ou seja, tempo decorrido para o trecho do código que *\*não\** é concorrente, e meça também a fração (percentual) de tempo que o código executa tarefas concorrentes.

b) construa uma tabela com as estimativas de speedup para execuções paralelas do código, a partir da fórmula da lei de *Amdhal*, para 1, 2, 4, 8, 16, 32 e 64 processadores

c) A partir das medidas de tempo em execuções paralelas efetuadas na mesma atividade 2 - problema 2, construa outras duas colunas da tabela iniciada no item b, contendo o speedup e eficiência paralela encontradas nas execuções com 1, 2 e 4 threads.

d) Compare os valores teóricos esperados de Speedup, a partir da lei de *Amdhal* (item b), com os valores medidos (item c). Analise os resultados.

e) Modifique, a versão concorrente desenvolvida do código, alterando a função original: *double Random(void)* para que a mesma possa ser executada concorrentemente. Descreva também a principal limitação encontrada na versão original da função que impede que a mesma possa ser executada concorrentemente.

f) Refaça as medidas da tabela desenvolvida no item d com a nova versão do código. Implica em novas execuções paralelas da versão desenvolvida no item e.

Primeiramente, foram realizadas algumas alterações em relação ao código do n-body (em anexo como nbody.c), portanto, as medidas foram feitas novamente, em uma máquina com as seguintes especificações:

- *MacBook Air* Processador Intel Core i5 *dual core* de 1,8 GHz
- 8 GB de memória LPDDR3 de 1600 MHz
- Armazenamento SSD de 128 GB

a)

	Tempo na parte sequencial (s)	Tempo na parte paralela (s)

1 thread	4,158 (8,1%)	47,128 (91,9%)
2 threads	3,7937 (15,2%)	21,223 (84,8%)
4 threads	3,634 (21,4%)	13,383 (78,6%)
8 threads	3,688 (27,8%)	9,597(72,2%)

b) Cada célula representa o *Speedup* com uma determinada quantidade de processadores em relação a um número  $n$  de *threads*.

o *SpeedUp* é dado por:  $Speedup = 1 / [s + (1-s)/P]$

	1 thread	2 threads	4 threads	8 threads
1 processador	1	1	1	1
2 processadores	1.85	1.74	1.65	1.56
4 processadores	3.22	2.75	2.44	2.18
8 processadores	5.11	3.87	3.2	2.72
16 processadores	7.22	4.88	3.8	3.09
32 processadores	9.11	5.6	4.19	3.33
64 processadores	10.48	6.05	4.42	3.46

c) A eficiência é dada por:  $Eficiência(P) = Speedup(P) / P$

- Para 1 processador

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
1 thread	4,158 (8,1%)	1	1
2 threads	3,7937 (15,2%)	1	1
4 threads	3,634 (21,4%)	1	1
8 threads	3,688 (27,8%)	1	1

- Para 2 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
1 thread	4,158 (8,1%)	1,85	0,925
2 threads	3,7937 (15,2%)	1,74	0,87

4 threads	3,634 (21,4%)	1,65	0,825
8 threads	3,688 (27,8%)	1,56	0,78

- Para 4 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
1 thread	4,158 (8,1%)	3,22	0,805
2 threads	3,7937 (15,2%)	2,75	0,6875
4 threads	3,634 (21,4%)	2,44	0,61
8 threads	3,688 (27,8%)	2,18	0,545

- Para 8 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
1 thread	4,158 (8,1%)	5,11	0,639
2 threads	3,7937 (15,2%)	3,87	0,484
4 threads	3,634 (21,4%)	3,2	0,4
8 threads	3,688 (27,8%)	2,72	0,34

- Para 16 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
1 thread	4,158 (8,1%)	7,22	0,451
2 threads	3,7937 (15,2%)	4,88	0,305
4 threads	3,634 (21,4%)	3,8	0,238
8 threads	3,688 (27,8%)	3,09	0,193

- Para 32 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
1 thread	4,158 (8,1%)	9,11	0,285

2 threads	3,7937 (15,2%)	5,6	0,175
4 threads	3,634 (21,4%)	4,19	0,131
8 threads	3,688 (27,8%)	3,33	0,104

- Para 64 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
1 thread	4,158 (8,1%)	10,48	0,164
2 threads	3,7937 (15,2%)	6,05	0,094
4 threads	3,634 (21,4%)	4,42	0,069
8 threads	3,688 (27,8%)	3,46	0,054

d) Pela Lei de *Amdal* o *Speedup* é limitado pela fração serial do procedimento, o que pôde ser observado pelas tabelas do item b e c, já que com o aumento do número de *threads* se houve um aumento do tempo de execução da parte sequencial, e mesmo com o aumento do número de processadores, não foi possível aumentar a eficiência.

e) Código modificado em anexo (em main.c).

A principal limitação se dá pela variável global *seed*. Ao criar várias *threads*, cada uma modifica o valor dessa variável, podendo ocasionar uma inconsistência no resultado final da execução do programa.

f)

- Para 1 processador

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
1 thread	2,289(4,8%)	1	1
2 threads	2,323(9,8%)	1	1
4 threads	2,581 (16,5%)	1	1
8 threads	2,574 (21,2%)	1	1

- Para 2 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
--	-------------------------------	----------------	------------

1 thread	2,289(4,8%)	1,908	0,954
2 threads	2,323(9,8%)	1,821	0,91
4 threads	2,581 (16,5%)	1,717	0,859
8 threads	2,574 (21,2%)	1,65	0,825

- Para 4 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
1 thread	2,289(4,8%)	1,908	0,477
2 threads	2,323(9,8%)	1,821	0,455
4 threads	2,581 (16,5%)	1,717	0,429
8 threads	2,574 (21,2%)	1,65	0,413

- Para 8 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
1 thread	2,289(4,8%)	1,908	0,238
2 threads	2,323(9,8%)	1,821	0,228
4 threads	2,581 (16,5%)	1,717	0,215
8 threads	2,574 (21,2%)	1,65	0,052

- Para 16 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
1 thread	2,289(4,8%)	1,908	0,119
2 threads	2,323(9,8%)	1,821	0,114
4 threads	2,581 (16,5%)	1,717	0,108
8 threads	2,574 (21,2%)	1,65	0,026

- Para 32 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
<i>1 thread</i>	2,289(4,8%)	1,908	0,0595
<i>2 threads</i>	2,323(9,8%)	1,821	0,057
<i>4 threads</i>	2,581 (16,5%)	1,717	0,054
<i>8 threads</i>	2,574 (21,2%)	1,65	0,013

- Para 64 processadores

	Tempo na parte sequencial (s)	<i>SpeedUp</i>	Eficiência
<i>1 thread</i>	2,289(4,8%)	1,908	0,0298
<i>2 threads</i>	2,323(9,8%)	1,821	0,0285
<i>4 threads</i>	2,581 (16,5%)	1,717	0,027
<i>8 threads</i>	2,574 (21,2%)	1,65	0,0065