

Atividade 02 – Computação Paralela

Carlos Alberto Ferreira Junior – 385154

Questão 1

Considerando o código:

- 1.1. Quando apenas o laço for externo é paralelizado a quantidade de operações realizadas pelo laço for mais interno (quantidade de produtos escalares), em tese, seria 125000, entretanto como a divisão do $n = 100$ por 8 threads é um valor fracionário 4 núcleos executam 130000 e outros 4 núcleos executam 120000, somando assim as 1000000 operações. Ou seja, faz-se uma divisão desbalanceada.
- 1.2. Com a aplicação da diretiva '*collapse(2)*' no laço mais externo, dividem-se a soma das interações do laço for mais externo e o segundo mais externo pelo número de threads, portanto temos 10000 interações divididas sequencialmente por 8, possuindo um resultado exato. Nesse caso, nota-se uma divisão balanceada do trabalho, fazendo com que cada um dos 8 threads realize 125000 operações.
- 1.3. Ativando as regiões paralelas aninhadas e com '*OMP_NUM_THREADS=2,4*' temos uma situação semelhante ao item 1.2, pois também temos uma divisão balanceada com cada thread executando 125000 operações, entretanto se diferencia em como as divisões são feitas. Nesse caso, cada thread recebe alguns pequenos lotes (não sequenciais) que somados equivalem as 125000 operações no laço for mais interno.

Questão 2

Caso seja utilizada a diretiva '*static,num_chunk*', a quantidade de chunks de tamanho '*num_chunk*' é alocada de forma estática para cada thread, no qual cada thread tem de executar chunks predefinidos.

2.1. '*static,1*'

Thread 3	4	5	7	10	14	18	
Thread 2	3				13	17	
Thread 1	2			9	12	16	20
Thread 0	1		6	8	11	15	19
	t0	t1	t2	t3	t4	t5	t6

'*static,2*'

Thread 3	7	8	15	16					
Thread 2	5	6	13	14					
Thread 1	3				4	11	12	19	20
Thread 0	1		2			9	10	17	18
	t0	t1	t2	t3	t4	t5	t6	t7	t8

'static,3'

Thread 3	10	11	12									
Thread 2	7	8	9	19	20							
Thread 1	4	5	6	16	17	18						
Thread 0	1		2			3				13	14	15
	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11

Caso seja utilizada a diretiva '*dynamic,num_chunk*', a quantidade de chunks de tamanho '*num_chunk*' é alocada de forma dinâmica para a quantidade de threads de forma sequencial, na qual quando uma thread fica ociosa, ela aloca o próximo chunk da fila, até que não existam mais chunks para serem computados.

2.2. '*dynamic,1*'

Thread 3	4	5	7	10	14	18	
Thread 2	3				13	17	
Thread 1	2			9	12	16	20
Thread 0	1	6	8	11	15	19	
	t0	t1	t2	t3	t4	t5	t6

'*dynamic,2*'

Thread 3	7	8	11	12	15	16	
Thread 2	5	6	9	10	13	14	
Thread 1	3				4	19	20
Thread 0	1	2			17	18	
	t0	t1	t2	t3	t4	t5	t6

'*dynamic,3*'

Thread 3	10	11	12	19	20				
Thread 2	7	8	9	16	17	18			
Thread 1	4	5	6	13	14	15			
Thread 0	1		2			3			
	t0	t1	t2	t3	t4	t5	t6	t7	t8