Breve relatório das comparações com threads (com matrizes e modificação nos vetores)

Aluno: Luan de Freitas Uchôa – 21200591

Este relatório tem como objetivo apresentar os resultados da comparação da soma de vetores e multiplicação de matrizes quando realizadas sem o uso de threads e quando realizadas em múltiplas threads. A diferença para a atividade da aula passada consiste no fato de que o método de cálculo da soma foi alterado para propiciar o paralelismo, assim como a inclusão do cálculo de matrizes.

Em relação à soma dos elementos de um vetor, foram realizadas comparações com vetores de 500, 1000, 2000, 5000, 10000 e 50000 posições. Os dados são exibidos na tabela abaixo, com tempo em milissegundos.

Número de threads	Quantidade de valores no vetor					
	500	1000	2000	5000	10000	50000
	Tempo (milissegundos)					
1 (main)	0,002	0,004	0,008	0,021	0,038	0,24
2	0,127	0,1	0,091	0,085	0,098	0,281
4	0,219	0,12	0,108	0,113	0,13	0,247
8	0,324	0,254	0,299	0,249	1,322	0,278
16	0,838	0,434	0,355	1,804	0,48	0,734
32	1,383	1,715	2,442	0,726	1,328	0,991
64	2,920	1,510	2,181	1,843	5,369	3,243

O tempo com threads ainda foi menor que o tempo com threads. No entanto, desta vez a diferença de tempo foi menor, e todos os tempos com threads foram melhores que os seus correspondentes em relação ao exercício anterior. Isso se deve ao fato de que a nova implementação propicia melhor o paralelismo, pois atribui intervalos de soma para as threads e se utiliza de artifícios que eliminam o uso de mutexes, que causariam um aumento perceptível no tempo de execução.

Em relação à multiplicação de matrizes, foi realizada a multiplicação de duas matrizes A e B que possuem tamanho N × N, alocadas dinamicamente e com valores atribuídos de

maneira aleatória. Foram realizados testes para matrizes de ordem 100, 200, 400, 800 e 1000. Os resultados são exibidos na tabela abaixo.

Número de threads	Quantidade de valores no vetor (matriz N × N, valor de N)				
	100	200	400	800	1000
	Tempo (segundos)				
1 (main)	0,008512	0,038946	0,273553	5,837759	11,240763
2	0,006117	0,032884	0,420005	5,587847	11,04467
4	0,004987	0,016225	0,226194	2,853529	5,61183
8	0,005238	0,018082	0,205264	2,552636	5,261298
16	0,004153	0,017441	0,196545	2,520817	5,180391
32	0,004052	0,016875	0,202494	2,501077	5,066895
64	0,004638	0,017276	0,200935	2,584933	5,261223

O tempo com threads se mostrou mais rápido que o tempo sem threads. Em casos com tamanhos de matrizes maiores, o tempo chegou a ser menos da metade. Isso se deve ao fato de que a multiplicação de matrizes é um problema mais propenso para a paralelização, devido à quantidade de operações que devem ser feitas e também por conta do fato de que a implementação ocorreu sem mutexes, então existe um ganho adicional de tempo por não haver a necessidade do controle de concorrência.