Tarefa 10 - Avaliação de Desempenho do Crivo Paralelo

Ana Carolina Medeiros Gonçalves

Matricula: 591513

Contador	Sequencial	Paralelo(Static)	Paralelo(Guided)
Task-Clock	0.980	1,941	1,97
Stalled-cycles-frontend	-	-	-
Stalled-cycles-backend	-	-	-
Instructions	0,33	0,28	0,35
LLC-load-misses	6,36	5,37	6,27
Time elapsed	4,072	2,77	2,97

Observação: Os contadores Stalled-cycles-frontend e Stalled-cycles-backend, não obtiveram os dados fornecidos no teste.

Speed-up $_{Static}$= 4,072/2,77 = 1,47

Eficiência $_{Static}$= 1,47/ 2 = 0,735 ou 73,5%

Speed-up $_{Guided}$= 4,072/ 2,97 = 1,371

Eficiência $_{Guided}$ = 1,371/2 = 0,685 ou 68,5%

Pelos resultados obtidos nos cálculos de eficiência e Speed-up do código paralelizado static e guided é percebido que os recursos não foram utilizados da melhor forma possível. Como a eficiência de ambas as paralelizações não chegaram perto dos 100%, os recursos não foram aplicados de forma eficiente para um melhor aproveitamento do hardware usado como teste.

A utilização do CPUC foi aproximado a 2, esperado pois foram utilizados apenas 2 núcleos, não chegou no valor total de 2 por causa da fração não paralelizável do código.

A quantidade de instruções caiu por volta de 65% a 68%. A taxa de falta em cache não variou consideravelmente, mas pelo menos apresentou uma melhora se comparado ao código sequencial.

O gargalo é a disputa para verificar um mesmo dado de um respectivo array para marca-lo como true ou false. Isto ocorre porque os valores iniciais possuem mais múltiplos do que o resto do código, assim gastamos mais tempo para processo e assim, provocamos redundância na verificação. Ainda temos uma variável que acumula valores recebidos de todas as Threads(varável sum), que pode ser considerado uma secção criticam causando uma dependência em threads, impactando na queda de instruções por segundo.

Foi notado que os load-misses de ambas as paralelizações não foram responsáveis em evitar a escalabilidade, podemos inferir também que devemos distribuir melhor a carga dos núcleos para evitar o desbalanceamento entre eles. Foi utilizado nos testes um chunk de tamanho 100, que não foi suficiente para satisfazer o melhor aproveitamento do código. Junto a isso, fazer o uso da paralelização sem as condições de disputa(regiões criticas).