

Contador	Sequencial	Paralelo (Static)	Paralelo (Guided)
Task-clock	0,988	1,948	1,959
Stalled-cycles-frontend	-	-	-
Stalled-cycles-backend	-	-	-
Instructions	0,38	0,26	0,25
LLC-load-misses	6,29	5,27	6,22
Time elapsed	4,064	2,90	2,95

Legenda:

- Informações não fornecidas no teste

$$\text{SpeedUp}_{\text{Estático}} = 4,064 / 2,90 = 1,4013$$

$$\text{Eficiência}_{\text{Estático}} = 1,4013 / 2 = 70\%$$

$$\text{SpeedUp}_{\text{Dinâmico}} = 4,064 / 2,95 = 1,3776$$

$$\text{Eficiência}_{\text{Dinâmico}} = 1,377 / 2 = 68\%$$

Pelos cálculos do speedup e da eficiência é notório que os recursos não foram utilizados de forma devida. Se sua eficiência estivesse próxima dos 100%, diríamos que os recursos foram aplicados de forma correta e melhor seria o aproveitamento do hardware.

A utilização do CPU foi próxima de 2, o que seria esperado, pois foram utilizados 2 núcleos. Porém, não chegou a esse valor total devido a fração não paralelizável do código. Já a quantidade de instruções caiu por volta de 68%. A taxa de falta em cache não variou muito, o que justifica que não ser o motivo para reduzir a escalabilidade linear, mas apresentou resultados melhores ao sequencial, o que já é satisfatório para a aplicação.

Como gargalo, é apontado a disputa para verificar um mesmo dado do array para marcá-lo como verdadeiro ou falso. Isso ocorre pois os valores iniciais possuem maiores quantidades de múltiplos que os demais, gastando mais tempo para processá-los e, assim, provocando a redundância na verificação. Além disso, apresenta uma variável que acumula valores recebidos de todas as threads (sum), o que pode ser considerado uma seção crítica, que causa uma certa dependência de todas as threads. Isso impacta na queda do número de instruções por segundo.

Observamos que os load-misses não foram responsáveis por evitar a escalabilidade. Com isso, como melhoria, deve-se explorar melhor a distribuição da carga nos núcleos, visto que o problema da aplicação é o desbalanceamento entre eles. Nos testes foi utilizado 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 condições de disputa (regiões críticas).