

# Comparação de *SpeedUp* na Paralelização do Problema Count 3s

Arthur Rodrigues Batista  
ra105422@uem.br

**Index Terms**—IEEE, IEEEtran, journal, L<sup>A</sup>T<sub>E</sub>X, paper, template.

## I. INTRODUÇÃO

No presente trabalho, faremos uma breve comparação entre o desempenho de um programa executado de maneira paralela, contraposto à uma execução sequencial utilizado o problema do *count 3's*. Dado um vetor de  $N$  posições, o problema consiste em contar quantos elementos são o número 3.

A ideal de paralelizar este problema é distribuir porções do vetor para diferentes *threads*. O objetivo é mostrar que com o uso do paralelismo, distribuindo a computação em vários processadores, aumenta-se a performance em relação ao tempo de execução.

A métrica que será utilizada será o *SpeedUP*<sup>1</sup>, sendo a relação do tempo de execução sequencial dividido pelo tempo concorrente. Ao final, concluiremos que o uso do paralelismo garante resultados aprimorados em relação ao tempo, embora possua algumas limitações, como a quantidade de processadores disponíveis em cada máquina.

## II. RESULTADOS E DISCUSSÕES

A princípio, para inferirmos se o paralelismo tende a diminuir o tempo de execução do problema do *count 3's*, faremos testes para instâncias de tamanho  $N = 1800000000$ . A Figura 1 representa uma relação da taxa de *SpeedUp* com a quantidade de *threads*. Por meio dele, é possível constatar que à medida que aumenta-se a quantidade de processos, existe uma tendência para que a taxa de *SpeedUp* diminua, garantido um menor tempo de execução quando paralelizado.

Embora seja constatado uma possível tendência à favor do paralelismo, vale destacar dois pontos. O primeiro sendo: **aumentar a quantidade de threads nem sempre acarreta em um menor tempo de execução**, este fato fica evidente ao passo que a quantidade de tarefas atribuídas para cada fluxo não seja grande o suficiente para que o paralelismo compense. Existe um custo atribuído para cada operação envolvendo a manutenção de programas paralelos, para além disso, com a inclusão dessa técnica, cria-se problemas inexistentes em metodologias sequencias. Podemos exemplificar este tipo de problema utilizando o próprio *Count 3's*. Ao criar vários fluxos, é preciso controlar qual destes escreverá no acumulador de

modo que não haja uma sobrescrita indesejável. Para tal, usa-se a ideia de restringir operações de escrita de acordo com uma fila, ou seja, o primeiro processo indicará aos demais que precisa finalizar seu processamento para que seja liberado para outrem. Portanto, faz-se necessário encontrar um número ideal de *threads* para uma determinada instância, uma vez que os custos extras acabam influenciando na performance geral do programa.

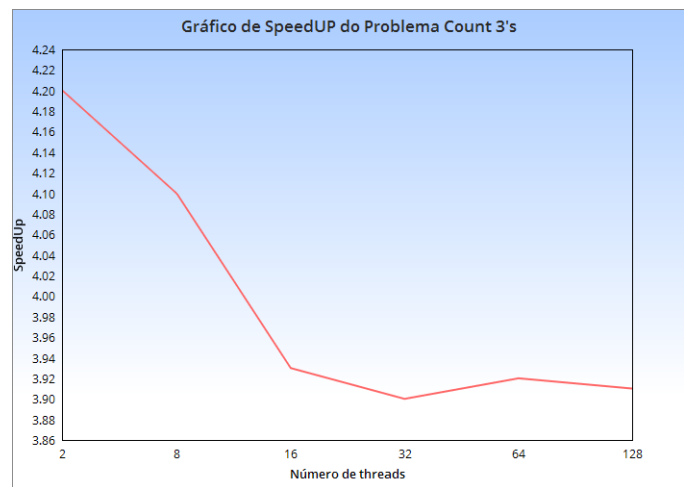


Fig. 1. .

O segundo fator que vale mencionar é a limitação da máquina. Nos testes realizados, nem sempre foi possível aumentar a quantidade de processos pois este está atrelado à quantidade de processamento disponível no momento. Logo, pelo fato das máquinas não possuírem uma capacidade ilimitada de processadores, limita-se superiormente à quantidade de processos que poderão ser criados.

Por fim, conclui-se que mesmo com as limitações apresentadas, a computação paralela deve ser um forte candidato para otimizações de programas, possibilitando à máquina usufruir de todos seus recursos a fim de reduzir o tempo de execução no geral.

<sup>1</sup>Na arquitetura de computador, *SpeedUP* é um número que mede o desempenho relativo de dois sistemas processando o mesmo problema. Mais tecnicamente, é a melhoria na velocidade de execução de uma tarefa executada em duas arquiteturas semelhantes com recursos diferentes.