

Paralelização do problema de contagem de elemento

Douglas Ferreira Delefrati

I. INTRODUÇÃO

Computação paralela pode ser resumido na capacidade dos programas em realizar várias operações simultaneamente. O principal fator por trás da computação paralela é o fato de que grandes problemas podem ser divididos em menores, que podem ser resolvidos em paralelo, ou seja, executados simultaneamente nos recursos de computação disponíveis.

Se usarmos um método dedutivo simplório, pode-se chegar na conclusão de que quanto mais processadores for utilizados na paralelização de um código, melhor será seu desempenho. Entretanto, essa afirmação é falsa, pois existem diversas variáveis (e.g. divisão de processamento igualitário entre os processadores disponíveis) que podem atrapalhar a performance do programa.

Dessa maneira, o presente trabalho propõe-se à analisar o ganho de desempenho de um algoritmo conforme a quantidade de processadores é incrementada. O algoritmo utilizado baseia-se em contar a quantidade de vezes que um determinado elemento aparece em um vetor de números gerados aleatoriamente.

II. METODOLOGIA

O experimento fora executado em uma máquina que possui um processador Intel® Core™ i7-5500U CPU @ 2.40GHz, com quatro CPU's e 4416 KB de memória cache, juntamente com uma GPU GeForce 920M/PCIe/SSE2. Como sistema operacional, utilizou-se o Pop OS 20.04 LTS, com kernel 5.4.0-7642-generic e o compilador GCC 9.3.0-10ubuntu2.

Para executar o código em paralelo, fora utilizado o modelo de execução POSIX Threads (Pthreads), na qual permite que um programa controle vários fluxos de trabalho diferentes que se sobrepõem no tempo. E como linguagem de programação, fora utilizado a linguagem C.

O algoritmo tem alguns parâmetros de entrada, o primeiro é a quantidade de elementos que o vetor possui, no teste fora utilizado um vetor com 999999999 elementos gerados de forma pseudo-aleatória. Dentre todos esses números, 30% das posições do vetor fora sorteados aleatoriamente para conter o elemento específico que o algoritmo deve contar.

III. MÉTRICAS

Para analisar o ganho de desempenho, fora utilizado o *speedup*, no qual pode ser definido como a razão entre o tempo de execução serial de um algoritmo sequencial pelo tempo de execução paralelo do mesmo algoritmo com p processadores.

$$S = \frac{T_s}{T_p}$$

IV. RESULTADOS

O primeiro teste a ser realizado foi a execução do algoritmo de maneira totalmente sequencial e a média do tempo foi 3 minutos e 4 segundos. Em seguida, fora executado o algoritmo de modo paralelo com 5 quantidades de processadores diferentes, iniciando com 2 processadores e aumentando em potência de dois a quantidade de processadores. Observa-se na figura 1 aumento no *speedup* entre 2 à 8 processadores. Entretanto, à partir de 8 processadores, o *speedup* diminuiu consideravelmente, confirmando apenas o acréscimo do número de processadores não é proporcional ao desempenho do algoritmo.

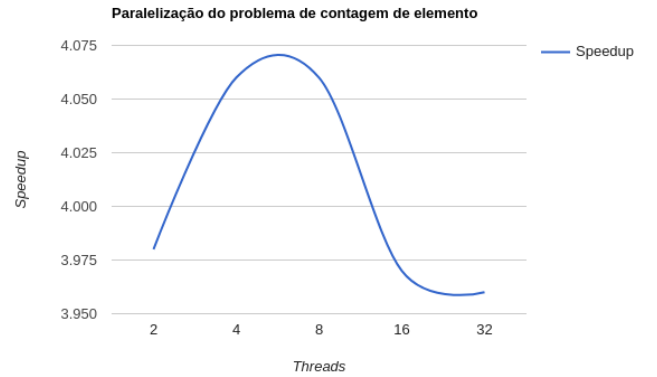


Fig. 1. Speedup do algoritmo de contagem de elemento

De fato, esse comportamento pode ser explicado pela lei de Amdahl's, na qual diz que a velocidade de um programa usando vários processadores em computação paralela é limitada pelo tempo necessário para a fração serial do problema.

V. CONCLUSÃO

A paralelização de um código pode proporcionar um aumento considerável em um programa. Entretanto, em detrimento de obter o melhor resultado essa estratégia carrega consigo algumas complicações que implicam em uma análise detalhada para sua implementação. Essa análise depende de muitos fatores, principalmente relacionados à arquitetura do hardware na qual o programa está sendo executado.

Neste trabalho, fora possível constatar que o número de processadores executando um programa suas limitações. E cada problema algoritmo deve ser analisado individualmente para obter o melhor desempenho em um hardware específico.