

UNIVERSIDADE DE SÃO PAULO
Bacharelado em Ciências de Computação

Guilherme Targon Marques Barcellos - 10724181
Mateus Ferreira Gomes - 10734773
Igor Ambo Takeo de Melo - 10830054
Rodrigo Cesar Arboleda - 10416722
Victor Graciano de Souza Correia - 10431201

Exercício de fixação - Maior palavra

Computação de Alto Desempenho(Turma B/Grupo 03)

São Carlos
Outubro, 2020

Metodologia PCAM

Particionamento - Nesse exercício, após a leitura do string, as tarefas foram divididas com o particionamento do vetor em partes iguais. Porém, caso alguma das tarefas acabe no meio de uma palavra da string, ela se estende até o final, fazendo assim, com que cada tarefa possa (mas não necessariamente) ser responsável por mais de uma palavra. Após tal divisão, cada uma dessas tarefas conta quantos caracteres há nas palavras que estão em seu interior.

O mais comum a ser realizado nesse tipo de exercício seria ler a string toda, em seguida usar `strtok()` para entregar uma palavra a cada tarefa e, assim, cada tarefa contaria o número de caracteres de sua palavra. Todavia, realizar o exercício dessa forma, não traria um ganho de desempenho, uma vez que, para dividir o vetor em palavras, o `strtok()` já teria de percorrer a string toda com uma eficiência $O(n)$, contudo, percorrendo a string toda já seria possível identificar a maior palavra.

Comunicação - A comunicação no exercício proposto é feita através do compartilhamento do texto recebido como parâmetro, o tamanho de caracteres desse texto, o número de threads e o tamanho da maior palavra presente na string processada até o momento. Assim, cada tarefa, ao realizar a contagem de caracteres presentes no bloco pelo qual é responsável por meio de uma variável presente unicamente em cada tarefa, compara com o tamanho da maior palavra encontrada até o presente instante.

Aglomerção - Após a leitura do texto de entrada, teremos todo o texto em um vetor de char de tamanho N. Vamos dividir nosso vetor em blocos(sub-vetores), teremos um total de P blocos, onde P corresponde ao número de elementos de processamento(ou núcleos) disponíveis para uso. Esta aglomeração permitirá aumentar a granularidade de cada processo e diminuir a comunicação necessária para atingir o objetivo final(maior palavra do texto).

Desta forma, a aglomeração das tarefas será feita em uma dimensão apenas, onde cada thread vai processar apenas um bloco de dados contendo $N(\text{tamanho do texto})/P(\text{threads disponíveis})$ caracteres. Por exemplo, um vetor com 160 caracteres ao ser processado por um cluster com 8 threads disponíveis vai ser dividido em 8 blocos, cada bloco com 20 caracteres. Cada thread vai contar o tamanho das palavras que comecam no seu bloco, achando o máximo local para posteriormente acharmos o máximo global entre todas as threads. Por exemplo, vamos processar o texto “Prova de alto desempenho semana que vem” em 2 threads. Nosso vetor contém 39 caracteres, e será dividido em 2 blocos(2 threads disponíveis), a thread 0 vai processar um bloco com 19 caracteres(divisão inteira) e a thread 1 vai processar um bloco com 20 caracteres. Sendo assim, segue quais palavras cada thread vai calcular o tamanho:

- Thread 0: “Prova”, “de”, “alto ”, “desempenho”. Máximo local = 10.
- Thread 1: “semana”, “que”, “vem”. Máximo local = 6.
- No final, os máximos locais são comparados, e então achamos o máximo global(10 para esse exemplo).

OBS: Note que a palavra “desempenho” começa na posição 14(primeiro bloco) e termina na posição 23(segundo bloco), porém é processada pela primeira thread, pois o que importa é a posição inicial da palavra

Mapeamento - O mapeamento ocorre por meio de uma fila circular se considerarmos que os nós do cluster possuem desempenho homogêneo, com cada elemento de processamento recebendo exatamente 1 processo caso P (processos) seja igual a $PROC$. Caso os nós do cluster tiverem desempenho diferente, então o mapeamento tem que considerar a diferença de desempenho antes da execução e em tempo de execução considerando a atribuição ao nó com a menor carga de trabalho em andamento, ou seja, a diferença do desempenho estático e dinâmico.