PCAM

- Daniel Sá Barretto Prado Garcia 103734344
- Felipe Guilermmo Santuche Moleiro 10724010
- Laura Alves de Jesus 10801180
- Tiago Marino Silva 10734748

Particionamento

Para solucionar o problema serão criadas dois tipos de tarefa:

O primeiro tipo se encarrega de dividir a string do problema em substrings separadas por um espaço, ou seja, cada substring conterá uma palavra e gerar tarefas do segundo tipo para serem executadas em paralelo. Esse tipo de tarefa será chamado de tarefa de divisão. O número de substrings/palavras será chamado de M. Será necessário apenas uma tarefa de divisão. O segundo tipo se encarrega de contar quantas letras tem uma palavra passada a ele. Serão criadas M tarefas deste tipo, cada uma recebendo uma respectiva substring/palavra para realizar a contagem. Esse tipo de tarefa será chamado de tarefa de contagem.

Essas tarefas serão feitas paralelamente. A tarefa de divisão cria tarefas de contagem para cada substring encontrada. Assim, estas tarefas criadas são enfileiradas para serem executadas em paralelo às tarefas de ambos os tipos.

Comunicação

A tarefa de divisão, ao gerar uma de contagem, cria uma cópia local da substring encontrada para a tarefa gerada. Assim, a tarefa gerada contém, localmente, a substring a ser processada.

Além disso, há a comunicação entre tarefas de contagem. Estas possuem uma região de memória compartilhada, correspondente à um valor inteiro de 32 bits, que armazena o maior valor encontrado de número de letras. Assim, a tarefa de contagem, após calcular o número de letras de sua substring, entra em uma região crítica para ler e escrever nessa região de memória compartilhada. Nesta região crítica, ela verifica se o valor do inteiro é menor que o número de letras de sua substring, caso seja, ela atualiza o valor do inteiro para o valor calculado. Ao acessar a região crítica, a tarefa impede que outras tarefas entrem nesta mesma região simultâneamente.

Aglomeração

É necessario aglomerar as tarefas em T threads que serão executadas pelo programa. Este T pode ser um número arbitrário, entretanto será utilizado o número de threads baseado no número de processadores lógicos existentes na maquina executada(configuração padrão do omp). Uma thread será responsavel por executar a tarefa de divisão e enfileirar as tarefas de contagem, assim as tarefas de contagem serão atribuídas e escalonadas dinamicamente para outras threads executarem.

Mapeamento

O mapeamento das Threads em cada núcleo de processamento lógico é feito pelo SO e não depende da implementação. Entretanto ao criar um número de threads iguais ao número de núcleos de processamento lógico, espera-se que com sorte o SO escalone cada thread para um núcleo, utilizando toda a capacidade do processador de uma só vez.