

Universidade Federal do Rio Grande do Norte
Instituto Metrópole Digital
IMD1116 - COMPUTAÇÃO DE ALTO DESEMPENHO - T01 (2025.1)
Tarefa 7: Processamento Paralelo de Lista Encadeada com OpenMP Task

Docente: SAMUEL XAVIER DE SOUZA

Discente: Iago Gabriel Nobre de Macedo (20220037927)

Link do repositório da atividade no Github: https://github.com/IagoGMacedo/IMD1116-Computacao-de-Alto-Desempenho/tree/main/exercicios/processamento_paralelo_lista_encadeada

Implementação

Em primeiro lugar, desenvolvi o código conforme a instrução pede, executando uma bateria de testes para analisar as questões propostas. Após isso, implementei os pontos que solucionam os erros levantados e demonstrei sua eficácia.

Resultados Obtidos

Com o primeiro cenário, obtenho uma saída similar a:

```
Arquivo: file1.txt | Thread: 6  
Arquivo: file2.txt | Thread: 6  
Arquivo: file3.txt | Thread: 6  
Arquivo: file1.txt | Thread: 2  
Arquivo: file1.txt | Thread: 5
```

O resultado varia com a execução, mas sempre obtemos erros relacionados ao não processamento de alguns nós (nesse caso, com o nó que possui o file4.txt); o processamento de um nó repetidas vezes (como ocorre com o nó que possui o file1.txt); e a variação do comportamento obtido (imprevisibilidade), além de alguns casos resultar em um erro de Segmentation fault.

Tais erros ocorrem pois a variável node não é protegida, assim, o seu valor pode ser alterado antes da execução da tarefa, levando a múltiplas tarefas processarem o mesmo nó.

A correção desse problema está no segundo código, que utiliza firstprivate(node) na diretiva da tarefa. Dessa forma, cada tarefa recebe uma cópia independente do ponteiro para o nó correspondente, evitando que diferentes tarefas acessem ou modifiquem o mesmo nó simultaneamente. Além disso, o uso de #pragma omp single garante que apenas uma thread seja responsável pela criação das tarefas, eliminando possíveis condições de corrida durante a geração das tarefas e assegurando que cada nó da lista seja processado exatamente uma vez, de maneira segura e previsível.

Após a correção, obtenho pequenas variações do seguinte resultado:

```
Arquivo: file1.txt | Thread: 11  
Arquivo: file3.txt | Thread: 0  
Arquivo: file2.txt | Thread: 3  
Arquivo: file4.txt | Thread: 9
```