

T9 - Francisco Tassinari Fajardo - Trabalho Final

Foi realizado a paralelização de um código onde originalmente eram lido vários arquivos diferentes, então foi adaptado para ler o mesmo arquivo várias vezes (o tempo de abrir e fechar o arquivo contam, pois não foi feito um for dentro dele, então abre e fecha n vezes), pois só se nota vantagens em empregar o paralelismo quando os textos são grandes na memória e assim pode se inserir as palavras na árvore AVL, cada processo com a sua árvore, se fosse muito pequeno o arquivo os tempos demorariam mais na versão implementada em paralelo pois a troca de mensagens consome um bom tempo.

A seguir a tabela com os tempos da execução sequencial do arquivo 1.txt:

NÚMERO DE EXECUÇÕES DO ARQUIVO	TEMPO
1	1.2s
10	13.5s
100	135s
200	271s

A seguir a tabela dos tempos de execução em paralelo:

NÚMERO DE EXECUÇÕES DO ARQUIVO	TEMPO	SPEEDUP
1	1.62s	0.75
10	9.4s	1.44
100	90s	1.5
200	176.2s	1.54

Portanto, como o tempo de processamento (inserção na árvore) é $O(\log n)$ então mesmo para o arquivo testado que tem 15MB observa-se que o ganho abrindo muitas vezes o próprio é baixo, pode ser útil claro, mas o ideal seria buscar uma solução que o tempo de execução diminuísse mais que o obtido, e isso envolve inserções e deleções na árvore que levam em conta o paralelismo.

-SPECS:

Arquitetura: x86_64
Modo(s) operacional da CPU: 32-bit, 64-bit
Ordem dos bytes: Little Endian
CPU(s): 4
Lista de CPU(s) on-line: 0-3
Thread(s) per núcleo: 2
Núcleo(s) por soquete: 2
Soquete(s): 1
Nó(s) de NUMA: 1
ID de fornecedor: GenuineIntel
Família da CPU: 6
Modelo: 61
Nome do modelo: Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz

Step: 4
CPU MHz: 2182.409
CPU MHz máx.: 2700,0000
CPU MHz mín.: 500,0000
BogoMIPS: 4391.85
Virtualização: VT-x
cache de L1d: 32K
cache de L1i: 32K
cache de L2: 256K
cache de L3: 3072K
CPU(s) de nó0 NUMA: 0-3

OBS: na verdade são 2 cores e 2 hiperthreads mas o linux não pega