T4: Simulador de Propagação de Vírus com OpenMP

AUTOR: JAİRO FERREİRA GEZ

DİSCİPLİNA: PROGRAMAÇÃO PARALELA

Estratégias de Paralelização

- Solução 1:
- Encontrada em "openmp1", a primeira solução é bem simples e utiliza-se de apenas uma diretiva, aplicada após a etapa de comunicação e antes na etapa de mapeamento, englobando as as etapas de particionamento e aglomeração, a fim de dividir as subtarefas de maneira igual entre as threads.
- Utilizou-se aqui a cláusula de agendamento runtime, com tamanhos de chunk automáticos:

- #pragma omp parallel for schedule(runtime)

Estratégias de Paralelização

- Solução 2:
- Encontrada em "openmp2", essa solução utiliza-se de duas diretivas: a primeira consiste em agendar de forma guiada, na etapa de mapeamento das tarefas. Já a segunda, na etapa de aglomeramento, define uma operação atômica (região crítica), onde apenas uma thread atualiza por vez o percentual de infectados de cada trial, esta e aquela diretivas aparecem abaixo, respectivamente:

- #pragma omp atomic
- #pragma omp parallel for schedule(guided)

Experimentos Realizados

Tipo	População	Repetições	Tempo	Speedup
Sequencial	30	2500	79,768s	
Solução 1(2thr)	30	2500	73,258s	1,08
Solução 2(2thr)	30	2500	69,189s	1,14
Solução 1(4thr)	30	2500	72,615s	1,09
Solução 2(4thr)	30	2500	71,033s	1,11
Sequencial	50	1000	134,477s	
Solução 1(2thr)	50	1000	102,574s	1,31
Solução 1(2thr)	50	1000	95,160s	1,41
Solução 1(4thr)	50	1000	60,981s	2,23
Solução 2(4thr)	50	1000	90,538s	1,48
Sequencial	80	500	240,564s	
Solução 1(2thr)	80	500	160,203s	1,5
Solução 2(2thr)	80	500	158,329s	1,52
Solução 1(4thr)	80	500	143,213s	1,67
Solução 2(4thr)	80	500	144,21s	1,66

Discussão

- A partir dos resultados obtidos, pode-se concluir que o desempenho em termos de tempo de execução melhorou conforme aumentou-se o número de threads em ambas as soluções implementadas.
- Os tempos de execução da solução 2 em comparação com a solução 1 quase sempre foram mais rápidos, salvo em um único caso (população 50, 4 threads), mostrando-se assim mais eficiente.

Material Utilizado

- SO: Debian GNU/Linux 9.11 (stretch)
- Processador: Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz
- Memória: 4GB

Referências

- http://www.inf.ufsc.br/~bosco.sobral/ensino/ine5645/OpenMP_Dynamic_Scheduling.pdf
- http://www.inf.ufrgs.br/~nicolas/pdf/openmp.pdf