|  |  |
| --- | --- |
|  | Computação Paralela  UFSCar - Sorocaba  Bacharelado em Ciência da Computação  Profaª. Yeda Regina Venturini  Profaª. Tiemi Christine Sakata |
| Projeto final da disciplina de Computação Paralela (2º sem. 2013) | |

**Grupo 13**

329550-ADRIANO PAIS RODRIGUES

380075-ARTHUR PESSOA DE SOUZA

379980-JOAO EDUARDO BRANDES LUIZ

**1. Descrição do Problema**

Um grafo completo é um [grafo simples](http://pt.wikipedia.org/wiki/Grafo_simples) em que todo [vértice](http://pt.wikipedia.org/wiki/V%C3%A9rtice) é adjacente a todos os outros vértices. O grafo completo de n vértices é frequentemente denotado por K_n. Pela própria definição, o algoritmo que verifica se um grafo é K_n consiste em verificar em uma matriz de adjacências se todos os vértices são adjacentes a todos os outros vértices.

Isso se torna muito custoso caso o grafo seja muito grande, e como infelizmente não há outra forma de reduzir a complexidade do problema, uma alternativa muito interessante é a redução do problema utilizando técnicas de paralelismo.

**2. Arquitetura do computador onde os testes foram realizados**

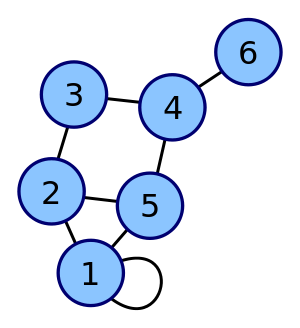
Processador: Intel(R) Core(TM) i5-2300 CPU @ 2.80GHz, 4 Cores, 6mb Intel Smart Cache

Memória RAM disponível: 8gb

Sistema Operacional: Linux Mint

**2. Divisão de dados**

A divisão dos dados consistem em dividir a matriz de adjacências em suas linhas, atribuindo a verificação das linhas por threads (processos, caso seja MPI) distintas, uma vez que para definir se um grafo é completo, basta-se apenas descobrir um vértice (linha) que não possui conectividade com outro vértice. Por exemplo, considerando o grafo a seguir.



Considerando 4 processos paralelos, e sua matriz de adjacência:

P1

A=\begin{bmatrix}
1 & 1 & 0 & 0 & 1 & 0\\
1 & 0 & 1 & 0 & 1 & 0\\
0 & 1 & 0 & 1 & 0 & 0\\
0 & 0 & 1 & 0 & 1 & 1\\
1 & 1 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0\\
\end{bmatrix}

P2

P3

P4

**3. Tempos das Implementações**

Nº de núcleos

**WallClock (s)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Serial** | **Paralelo** | | | | | | |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| MPI |  |  |  |  |  |  |  |  |
| OpenMP |  |  |  |  |  |  |  |  |
| Pthreads |  |  |  |  |  |  |  |  |

**Tempo de CPU (s)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Serial** | **Paralelo** | | | | | | |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| MPI |  |  |  |  |  |  |  |  |
| OpenMP |  |  |  |  |  |  |  |  |
| Pthreads |  |  |  |  |  |  |  |  |

**Consumo de memória (Megabytes)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Serial** | **Paralelo** | | | | | | |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| MPI |  |  |  |  |  |  |  |  |
| OpenMP |  |  |  |  |  |  |  |  |
| Pthreads |  |  |  |  |  |  |  |  |

**Benchmarks**

**Benchmark 1**

Speedup

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Serial** | **Paralelo** | | | | | | |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| MPI |  |  |  |  |  |  |  |  |
| OpenMP |  |  |  |  |  |  |  |  |
| Pthreads |  |  |  |  |  |  |  |  |

Eficiência

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Serial** | **Paralelo** | | | | | | |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| MPI |  |  |  |  |  |  |  |  |
| OpenMP |  |  |  |  |  |  |  |  |
| Pthreads |  |  |  |  |  |  |  |  |

**Análise Parcial**

<<Analise>>

**Benchmark 2**

Speedup

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Serial** | **Paralelo** | | | | | | |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| MPI |  |  |  |  |  |  |  |  |
| OpenMP |  |  |  |  |  |  |  |  |
| Pthreads |  |  |  |  |  |  |  |  |

Eficiência

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Serial** | **Paralelo** | | | | | | |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| MPI |  |  |  |  |  |  |  |  |
| OpenMP |  |  |  |  |  |  |  |  |
| Pthreads |  |  |  |  |  |  |  |  |

**Análise Parcial**

<<Analise>>

...

**Análise Final**

<<Analise>>

**Análise da Escalabilidade**

**Serial**

<<Analise>>

**MPI**

<<Analise>>

**OpenMP**

<<Analise>>

**Pthreads**

<<Analise>>

**Ferramentas Utilizadas**