INSTITUTO TECNOLÓGICO DE CIUDAD MADERO.



UNIDAD 2

PRACTICA 1

ALUMNA: CAVAZOS ARGOT ANA VICTORIA

N° CONTROL: 15071292

PROFESOR: DRA. CLAUDIA GUADALUPE GÓMEZ SANTILLÁN

MATERIA: PROGRAMACIÓN PARALELA

FECHA DE ENTREGA: LUNES 14 DE OCTUBRE DEL 2018

Índice:

[Ejercicio: 3](#_Toc527069927)

[Introducción: 3](#_Toc527069928)

[Marco teórico: 3](#_Toc527069929)

[Grafo: 3](#_Toc527069930)

[Clasificación de los grafos: 3](#_Toc527069931)

[Grafo completo: 3](#_Toc527069932)

[Grafo conexo: 3](#_Toc527069933)

[Grafo euleriano (CAMINO): 4](#_Toc527069934)

[Metodología: 4](#_Toc527069935)

[Conclusiones: 8](#_Toc527069936)

[Bibliografía: 8](#_Toc527069937)

Ejercicio:

Introducción:

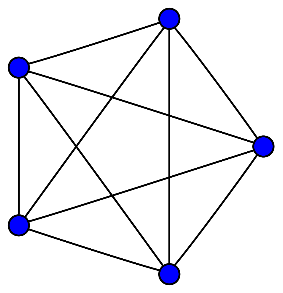
Use el archivo EU2\_variables2.c y resuelva en paralelo si un grafo es completo y/o conexo, además si tiene un ciclo Euleriano. Pruebe con grafos de diferentes tamaños de la librería de TSPLIB. Use las funciones: Omp\_set\_num\_threads(), Omp\_get\_num\_threads(), Omp\_get\_max\_threads(), Omp\_get\_thread\_num() y Omp\_get\_num\_procs().

Compare la versión secuencial con la versión paralela, reporte los tiempos de cada una de las versiones.

Marco teórico:

Grafo:

Un grafo consiste de un conjunto de vértices y otro conjunto de aristas que unen algunos de los vértices.



Clasificación de los grafos:

Grafo completo:

Un grafo es completo si existen aristas uniendo todos los pares posibles de vértices. Es decir, todo par de vértices (a,b) debe tener una arista e que los une.

Grafo conexo:

Un grafo es conexo si cada par de vértices está conectado por un camino; es decir, si para cualquier par de vértices (a, b), existe al menos un camino posible desde a hacia b.

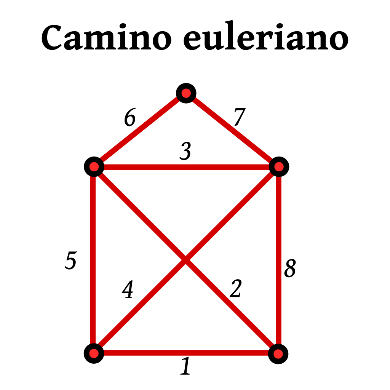
Esto se puede comprobar mediante la siguiente formula:

m = cantidad de aristas

n = cantidad de nodos

si (m>=n-1) entonces es conexo



Grafo euleriano (CAMINO):

Un grafo euleriano es aquel en que pueden recorrerse todas sus las aristas (arcos) de manera consecutiva y sin repetirlas.

Un grafo conexo y no dirigido se dice que es euleriano si cada vértice tiene un grado par o solo 2 vértices tienen grado impar.

Metodología:

Grafo: 1  
**Vectores:** 5

**Coordenadas:**

(1,2),(1,3),(1,4),(2,3),(2,4),(4,3),(4,5),(5,1)

**Tabla de datos:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nodos | Rutas | | | | | Total de rutas por nodo |
| **1** | **2** | **3** | **4** | **5** |
| **1** | 0 | 1 | 1 | 1 | 1 | 4 |
| **2** | 1 | 0 | 1 | 1 | 0 | 3 |
| **3** | 1 | 1 | 0 | 1 | 0 | 3 |
| **4** | 1 | 1 | 1 | 0 | 1 | 4 |
| **5** | 1 | 0 | 0 | 1 | 0 | 2 |

**a) ¿Es completo?**:

Se necesita que cada nodo posea n-1 rutas para el grafo completo: **n-1 rutas = 5 - 1 = 4**

|  |  |  |
| --- | --- | --- |
| Nodos | Total de rutas por nodo | ¿Total es igual que n-1? |
| **1** | 4 | **Si** |
| **2** | 3 | **No** |
| **3** | 3 | **No** |
| **4** | 4 | **Si** |
| **5** | 2 | **No** |

Ya que no todos los nodos cumplen con la condición es grafo **NO ES COMPLETO**.

**c) ¿El grafo posee un camino euleriano?**:

Se necesita que de cualquier nodo a se pueda llegar a un nodo b recorriendo todas sus aristas y sin repetirlas.

Si todos los nodos del grafo poseen par rutas este posee un camino euleriano.

Si se da el caso de que solo dos nodos poseen rutas impares también posee un camino euleriano.

|  |  |  |
| --- | --- | --- |
| Nodos | Total de rutas por nodo | Grado |
| **1** | 4 | Par |
| **2** | 3 | Impar |
| **3** | 3 | Impar |
| **4** | 4 | Par |
| **5** | 2 | Par |

Ya que solo dos nodos poseen grado impar el grafo **POSEE UN CAMINO EULERIANO**.

**b) ¿Es conexo?**:

Se necesita que de cualquier nodo a se pueda llegar a un nodo b.

Si un grafo es **completo** significa que también es **conexo**.

En este caso ya que el grafo es completo no se puede aplicar la formula siguiente:

m = cantidad de aristas

n = cantidad de nodos

si (m>=n-1) entonces es conexo

|  |  |  |  |
| --- | --- | --- | --- |
| Nodos | | ¿Existe una  ruta directa? | Ruta  posible |
| a | b |
| 1 | 2 | Si |  |
| 1 | 3 | Si |  |
| 1 | 4 | Si |  |
| 1 | 5 | Si |  |
| 2 | 1 | Si |  |
| 2 | 3 | Si |  |
| 2 | 4 | Si |  |
| 2 | 5 | No | 2,1,5 |
| 3 | 1 | Si |  |
| 3 | 2 | Si |  |
| 3 | 4 | Si |  |
| 3 | 5 | No | 3,1,5 |
| 4 | 1 | Si |  |
| 4 | 2 | Si |  |
| 4 | 3 | Si |  |
| 4 | 5 | Si |  |
| 5 | 1 | Si |  |
| 5 | 2 | No | 5,1,2 |
| 5 | 3 | No | 5,1,3 |
| 5 | 4 | Si |  |

Ya que es posible llegar de cualquier nodo a cualquier otro dentro del grafo, este **ES CONEXO**.

Experimentación y resultados:

Información sobre el equipo:

**Modelo**: Dell OptiPlex 7010

**Procesador**: Intel(R) Core(TM) i5-3550 CPU @ 3.30GHz

**Memoria RAM**: 4.00 GB

**Tipo de sistema**: Sistema operativo de 64 bits

**Sistema operativo utilizado**: Windows 7 Ultimate Service Pack 1

Tabla de resultados:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Experimento | SEED | Hilos | Vértices | Aristas | Resultados | | | | | Tiempo secuencial  (segundos) | Tiempo paralelo OMP  (segundos) |
| **Completo** | **Conexo** | **Euleriano** | **Par** | **Impar** |
| 1 | 45 | 4 | 5 | 3 | NO | NO | NO | 1 | 4 | **0.000** | 0.000 |
| 2 | 45 | 4 | 10 | 8 | NO | NO | NO | 6 | 4 | **0.000** | 0.000 |
| 3 | 45 | 4 | 350 | 350 | NO | SI | NO | 176 | 174 | **0.000** | 0.000 |
| 4 | 45 | 4 | 550 | 550 | NO | SI | NO | 266 | 284 | **0.000** | 0.015 |
| 5 | 45 | 4 | 1,500 | 1,499 | NO | SI | NO | 770 | 730 | **0.015** | 0.31 |
| 6 | 45 | 4 | 3,000 | 2,999 | NO | SI | NO | 1560 | 1440 | **0.062** | 0.62 |
| 7 | 45 | 4 | 10,000 | 9,998 | NO | NO | NO | 5124 | 4876 | **0.624** | 0.577 |
| 8 | 45 | 4 | 20,000 | 20,000 | NO | SI | NO | 10352 | 9648 | **2.684** | 1.856 |

Información sobre el cluster:

**Modelo**:

**CPU**: Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz

**PCI:** Xeon E7 v3/Xeon E5 v3/Core i7 DMI2

Tabla de resultados:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Experimento | SEED | Hilos | Vértices | Aristas | Resultados | | | | | Tiempo secuencial  (segundos) | Tiempo paralelo OMP  (segundos) |
| **Completo** | **Euleriano** | **Conexo** | **Par** | **Impar** |
| 1 | 45 | 2 | 10,000 | 9,998 | NO | NO | NO | 5124 | 4876 | **0.81** | 0.81 |
| 2 | 45 | 4 | 10,000 | 9,998 | NO | NO | NO | 5124 | 4876 | **0.81** | 0.85 |
| 3 | 45 | 6 | 10,000 | 9,998 | NO | NO | NO | 5124 | 4876 | **0.81** | 0.90 |
| 4 | 45 | 2 | 20,000 | 20,000 | NO | NO | SI | 10352 | 9648 | **3.07** | 2.96 |
| 5 | 45 | 4 | 20,000 | 20,000 | NO | NO | SI | 10352 | 9648 | **3.07** | 3.16 |
| 6 | 45 | 6 | 20,000 | 20,000 | NO | NO | SI | 10352 | 9648 | **3.07** | 3.19 |
| 7 | 45 | 2 | 30,000 | 29999 | NO | NO | SI | 15448 | 14552 | **6.89** | 6.61 |
| 8 | 45 | 4 | 30,000 | 29999 | NO | NO | SI | 15448 | 14552 | **6.89** | 6.94 |
| 9 | 45 | 6 | 30,000 | 29999 | NO | NO | SI | 15448 | 14552 | **6.89** | 7.03 |
| 10 | 45 | 2 | 40,000 | 39998 | NO | NO | NO | 23616 | 16384 | **12.23** | 11.83 |
| 11 | 45 | 4 | 40,000 | 39998 | NO | NO | NO | 23616 | 16384 | **12.23** | 12.42 |
| 12 | 45 | 6 | 40,000 | 39998 | NO | NO | NO | 23616 | 16384 | **12.23** | 12.43 |
| 13 | 45 | 2 | 50,000 | 49998 | NO | NO | NO | 33518 | 16482 | **18.95** | 18.31 |
| 14 | 45 | 6 | 50,000 | 49998 | NO | NO | NO | 33518 | 16482 | **18.95** | 18.29 |
| 15 | 45 | 10 | 50,000 | 49998 | NO | NO | NO | 33518 | 16482 | **18.95** | 17.93 |
| 16 | 45 | 12 | 50,000 | 49998 | NO | NO | NO | 33518 | 16482 | **18.95** | 17.95 |
| 17 | 45 | 2 | 100,000 | 99989 | NO | NO | NO | 83682 | 16318 | **76.01** | 72.97 |
| 18 | 45 | 4 | 100,000 | 99989 | NO | NO | NO | 83682 | 16318 | **76.01** | 76.95 |

Conclusiones:

Ya que el propósito de esta Unidad es observar como el paralelismo mejora o varia el tiempo de ejecución del programa se tuvo que eliminar parte del código que genera el conjunto de datos en cada ejecución. Se optó por utilizar conjuntos de datos generados con anterioridad.

Otro cambio que se tuvo que realizar fue utilizar conjuntos de datos más grandes que los registrados en la ejecución secuencial, sin embargo, el máximo de conjuntos que se logró ejecutar fue de tan solo 20,000. La única solución que se encontró fue utilizar el clúster del tecnológico, sin embargo, los resultados en tiempo que este arroja no parecen mejorar con la paralización del programa de una manera muy notable.

Aplicando la paralelizacion del problema apenas se logró una mejora de aproximadamente 3s con respecto al conjunto más grande aplicado de 100,000 vértices. Este conjunto de datos solo fue posible de ejecutar dentro del clúster.

Bibliografía:

*Departamento de Matemática Aplicada.* (s.f.). Obtenido de www.ma.uva.es/~antonio/Industriales/Apuntes\_06-07/LabM/Grafos\_2007-2.pdf

*Universidad Autonoma de Madrid*. (s.f.). Obtenido de Laboratorio de Matemáticas 2010/2011: http://verso.mat.uam.es/~pablo.angulo/doc/laboratorio/b4s2.html

*Universidad de Pamplona.* (s.f.). Obtenido de http://www.unipamplona.edu.co/unipamplona/portalIG/home\_23/recursos/.../grafo3.pdf