



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Departamento de Matemática y Ciencia de la Computación

Algoritmo en Paralelo para Determinar si un Grafo es Fuertemente Conectado Utilizando Memoria Compartida.

Alan Higuera Da Silva
alan.higuera@usach.cl

Algoritmos Distribuidos - 22629
Licenciatura en Ciencia de la Computación

Semestre Primavera 2024

1 Introducción

Un grafo fuertemente conectado es el grafo tal que desde todos sus vértices existe un camino hacia todos los demás vértices. Este informe presentó la implementación de un algoritmo en paralelo que utiliza memoria compartida para identificar este tipo de grafo dada su representación matricial.

2 Ejecución

La entrada consta de un archivo de texto compuesto por los datos de la matriz de adyacencia que definen el grafo a evaluarse. En la primera línea se entregó la dimensión de la matriz, la cual siempre debe de ser cuadrada. Las demás líneas del archivo fueron compuestas por los elementos $a_{i,j}$ de la matriz en análisis. Ya por consola se reciben los siguientes parámetros:

K : cantidad de hebras a ocupar.
 P : -H Partición Horizontal o -V Partición vertical
 T : -S Modo Silencioso o -V Modo Verboso

3 Procedimiento

Algorithm 1 Función de Coordinación de Procesos

Input: Argumentos = id de la hebra, vértice de inicio, modo de ejecución, cantidad de hilos, dimensión de la matriz, lista de nodos visitados

Output: Vacío

```
cantidad_revisados ← 0
while cantidad_revisados < q_columnas and nsimo_vertice < q_columnas
and Bandera = 0 do
    if modo = 1 then
        Print "Se inicia el hilo myid y verificará el nodo nsimo_vertice"
    if Bandera = 0 then
        Alcanzar_nodos(Argumentos)
    cantidad_revisados ← cantidad_revisados + 1
    nsimo_vertice ← nsimo_vertice + hebras
    visitados ← KillAll(visitados)
```

Algorithm 2 Alcanzar_nodos

Input: Argumentos = id de la hebra, vértice de inicio, modo de ejecución, cantidad de hilos, dimensión de la matriz, lista de nodos visitados

Output: Vacío

```
if visitados es nulo then
    Insertar(visitados, nsimo_vertice, 0)
    fila ← nsimo_vertice
if Bandera = 0 then
    while todos_marcados(visitados) = 0 do
        for i from 0 until dim do
            if matriz[fila][i] = 1 and fila ≠ i and i no esta en visitados then
                Insertar(visitados, i, 0)
                marcar(visitados, fila)
                fila ← nsimo_elemento_no_markado(visitados)
Print "Revision de nodos visitados desde el vertice nsimo_vertice"
tamanho_lista = completitud_visitas(visitados, Argumentos)
if tamanho_lista ≠ dim then
    Bandera ← 1
```

Antes de empezar con la coordinación de procesos se definieron bandera y la matriz como variables globales para que todas las hebras pudiesen saber cuando detenerse y además pudieran leer la matriz en la que se busca. Por otra parte, cada hebra llevó consigo su identificador, el vértice de partida, la dimensión de la matriz, la cantidad de hebras que hay, y por último una lista enlazada propia para almacenar los vértices que logra visitar desde un nodo inicial. Enseguida, las hebras ingresaron a la función de coordinación de procesos, la cual se encargó de asignar el nodo inicial a cada hebra y además reasignar cuando la hebra termine su trabajo hasta que se hayan terminado los nodos iniciales a revisar o la bandera se haya levantado.

El proceso de revisión desde un nodo se le encargó a la función Alcanzar_nodo. Se recorre la matriz desde la fila del nodo en cuestión y se registra el número del nodo que se puede alcanzar en la lista local denominada "visitados" y enseguida se visitan tales nodos y se marcan los que ya se visitaron mientras que se agregan nuevos nodos que se puedan alcanzar hasta que todos los elementos de la lista hayan sido marcados. Si al final del procedimiento la lista no posee una cantidad de elementos igual a la dimensión de la matriz, entonces es porque no se ha logrado visitar algún nodo y se alza la bandera y todos los nodos detienen su ejecución.

4 Conclusiones

Se ha presentado la implementación de un algoritmo que decide si un grafo es o no fuertemente conectado de forma paralela y utilizando memoria compartida. Las tareas fundamentales fueron registrar los nodos que se lograron visitar desde un vértice inicial y cómo coordinar la acción de las hebras que actuaron sobre el grafo.