



INSTITUTO TECNOLÓGICO DE LAS AMERICAS(ITLA)

Alumno:

Yaher Enrique Hichez García

Matrícula:

202010339

Carrera:

Desarrollo en software

Tema:

Nociones de algoritmos y programación paralela

Asignatura:

Programación Paralela

Luis Bessewell Feliz

Introducción

En este trabajo se plasma un resumen investigativo sobre el capítulo II del modulo llamado "Nociones de algoritmos y programación paralela" en este plasmamos distintos temas relacionados a este tanto el paralelismo de datos el cual es un paradigma de la concurrencia que consiste en subdividir el conjunto de datos de entrada del programa, de modo que cada procesador corresponda a uno de estos datos. Además de esto se plasman sus temas relacionados como Tipos de paralelismo, paseo de mensajes, espacio de direcciones compartidas, paralelismos de control, sistemas computacionales, grafos de dependencia, modelo pipeline, y los algoritmos paralelos.

Tipos de paralelismo

Explícito:–El algoritmo paralelo debe especificar explícitamente qué tan simples son los procesadores de tareas del compilador.

Eso del programador es bastante difícil.

Implícito:–Se necesita un lenguaje de programación secuencial e instrucciones del compilador para ejecutar el programa en paralelo.–El compilador debe analizar y comprender las dependencias para un mapeo efectivo.

Paso de mensajes

El modelo de paso de mensajes utiliza métodos y funciones para realizar el envío de un mensaje desde un proceso de envío a un destinatario. Esto supone un enfoque opuesto al paradigma tradicional en el que los procesos, funciones y subrutinas solo pueden llamarse por su nombre.

Los lenguajes de programación para la memoria compartida o el paso de mensajes son generalmente lenguajes secuenciales complementados con un conjunto de llamadas al sistema especiales. Las llamadas producen primitivas de bajo nivel para pasar mensajes, sincronización de procesos, exclusión mutua, entre otros.

Espacio de direcciones compartido

Los programadores ven el programa como una colección de acceso a un grupo central de variables compartidas. Los lenguajes proporcionan primitivas para resolver estos problemas mutuos.

Paralelismo de datos/control

Paralelismo de datos: Se trata de un paradigma de programación concurrente que consiste en subdividir el conjunto de datos de entrada de un programa, de modo que cada procesador corresponda a un subconjunto de estos datos. Cada procesador realizará la misma secuencia de operaciones que los procesadores en su subconjunto de datos asignado. Idealmente, esta ejecución simultánea de las operaciones da como resultado una aceleración global neta del cálculo. El paralelismo de datos se refiere a escenarios en los que la misma operación se realiza

simultáneamente en partes de una matriz o colección de origen. En las operaciones de datos paralelas, la colección de origen se particiona para que varios subprocesos puedan operar en segmentos simultáneamente.

Paralelismo de control: Ejecución simultánea de diferentes cadenas de instrucciones. Las instrucciones se pueden aplicar a los mismos datos de cadena, aunque generalmente se aplican a diferentes datos de cadena.

Sistemas computacionales

Los sistemas informáticos actuales están formados por múltiples componentes, al igual que los sistemas paralelos, pero estos componentes son de diferentes maneras:

Sistemas con varios núcleos y memoria en común.

Un sistema multinúcleo con una tarjeta gráfica con su memoria, Redes de multinúcleos, cada uno con su propia y posiblemente su tarjeta gráfica, Varias redes conectadas entre sí de forma remota, etc.

Grafos de dependencias

Un grafo de dependencia es un grafo acíclico dirigido, los nodos representan tareas y un borde de uno a un destino representa que para poder ejecutar el destino, el origen debe haberse ejecutado. Los nodos se pueden etiquetar con un valor que el costo de la tarea. Los bordes se pueden etiquetar con un valor que cueste la comunicación.

Modelo Pipeline

El modelo Pipeline es conocido en el mundo de la computación como una serie de elementos de procesamiento de datos dispuestos de tal manera que la salida de cada uno es la entrada del siguiente. Pipeline hace referencia a su nombre, básicamente es como el agua pasa por la tubería, donde el agua es la información o los procesos. La arquitectura Pipeline consiste en transformar un flujo de datos en un proceso compuesto por varias fases secuenciales, la entrada de ser la salida del anterior, con almacenamiento de datos entre los procesos. Comprender cómo funciona una tubería es un paso importante para comprender lo que sucede dentro de un procesador. Este sistema se usa comúnmente en los sistemas operativos porque puede ejecutar una serie de procesos simultáneamente, se ejecutan

secuencialmente a través de un administrador de tareas que tiene diferentes tipos de prioridad y capacidad de procesamiento.

Los algoritmos paralelos

Los algoritmos paralelos son extremadamente importantes para resolver problemas en muchos dominios de aplicación. Esta sección describe los pasos típicos del diseño en paralelo:

1.Particionamiento. Los cálculos se dividen en pequeñas tareas. Es generalmente independiente de la arquitectura o el modelo de programación. Buenas particiones de partición tanto de los cálculos del problema como de los datos sobre los que opera.

2.Comunicación. Las tareas generadas por una partición están diseñadas para ejecutarse simultáneamente, pero generalmente no pueden ejecutarse de forma independiente. Los cálculos en la ejecución de una tarea normalmente requerirán asociados con otras tareas. Los datos deben transferirse entre tareas para que continúen los cálculos.

3.Aglomeración. Las tareas y las estructuras de comunicación definidas en las dos primeras etapas de diseño se evalúan de acuerdo con los requisitos de ejecución y los costos de implementación. Si es necesario, las tareas se agrupan en tareas adicionales para mejorar la ejecución o reducir los costos de comunicación de sincronización.

4.Mapeo. Cada tarea se asigna a un procesador de manera que intenta satisfacer los objetivos de concurrencia en la utilización de la CPU y minimizando el costo de comunicación.

Conclusión

Como vimos en este trabajo se hizo un resumen sobre todos los temas relacionados con los algoritmos lo cuales recordemos que se definen como una secuencia bien definida de pasos finitos resuelve un problema, lo cual nos ayuda a captar que son las nociones de algoritmos, programación paralela, haciendo hincapié en los temas que hacen más énfasis en la base investigativa de estos temas, como los tipos de paralelismo, sistemas computacionales, etc.

Bibliografías

<https://docplayer.es/96306146-Metodologia-de-la-programacion-paralela-modelos-de-programacion-paralela-paradigmas-de-programacion-paralela.html>

<http://dis.um.es/~domingo/apuntes/AlgProPar/1617/sistemasprogramacion.pdf>

<https://www.monografias.com/trabajos106/autooptimizacion-esquemas-paralelos-iterativos/autooptimizacion-esquemas-paralelos-iterativos>

<http://docplayer.es/94900606-Esquemas-algoritmicos-paralelos-particionado-y-paralelismo-de-datos.html>

<http://dis.um.es/~domingo/apuntes/AlgProPar/1516/Grafo+Pipe.pdf>

<http://dis.um.es/~domingo/09/JENUI/presentationWEA.pdf>

<http://dis.um.es/~domingo/09/JENUI/presentationWEA.pdf>