Díaz Hernández Marcos Bryan
Tarea N°6
Grupo: 09

Reporte: Is parallelism for you?

El artículo es una guía practica de como utilizar el paralelismo o aplicarlo en un proyecto-programa, a través de varios puntos muy importantes y los cuales vimos en clase.

Para esto de identificar si un algoritmo es paralelizable se deben considerar varios aspectos importantes y comenzando con el primer momento de analis de lo que se va a realizar en el programa ya que en el paralelismo no todo es posible paralizarse y tampoco todo es conveniente.

Los mas importantes son tres aspectos que se mencionan: tiempo de ejecución, la solución actual y sus necesidades y por último la frecuencia de uso. Estas métricas aunque son sencillas dan la mayor pauta para decidir si el paralelismo puede mejorar algunos de estos aspectos ya que este mejora en bajos niveles algunos algoritmos. Por la razón anterior es necesario conocer las caracteristicas del algoritmo que deseamos paralelizar.

Decidir si paralelizar o no se enfoca al conocimiento del algoritmo, ya que el manejo de la información y las dependencias entre esta nos da la señal de que paralelizar es difícil ya que se nos indican 4 diferentes tipos de paralelización, las cuales se van caracterizando en forma o de acuerdo al manejo de la información, ya que algunos segmentos pueden ser totalmente independientes y otros totalmente dependientes.

Analizar el código permite saber cuando la paralelización es totalmente lineal y no habra comunicación entre los hilos o existira comunicación entre los hilos. En clase vimos este tipo de paralelización porque existe la concurrente y puramente paralelo.

De igual forma el algoritmo es una parte de la paralelización importante pero no solo interviene el código sino también el equipo-hardware que se utiliza interviene y determina en muchos casos la eficiencia y ganancia que se obtendra en caso de desear paralelizar.

Díaz Hernández Marcos Bryan
Tarea N°6
Grupo: 09

El uso o aprovechamiento de la arquitectura que se utiliza para poder ejecutar el algoritmo se basa en la clasificación Flynn que igual vimos en la clase, donde las mas importantes son SIMD y su version MIMD las cuales permiten el paralelismo en dos distintas formas de comunicación interna y distribución de la memoria.

Utilizar adecuadamente la arquitectura permitira dar o sacar un mejor aprovechamiento de la paralelizacion, debido a la organizacion de los procesadores, ya que estas arquitecturas determinan si un algoritmo es paralelizable en funcion de la dependencia o independencia del algoritmo. Debido que SIMD permite la paralelización mediante concurrencia y sus derivados ya que se realiza un proceso mediante varias procesadores.

Para el caso de MIMD este permite el proceso: varias instrucciones y varios procesadores, donde es importante el destacar que no necesariamente n procesadores haran la paralelizacion mas veloz o eficiente, debido a que la independencia de procesos puede otorgar la condición de carrera y que algunos procesadores sean capaces de terminar su ejecucion antes que otros. Además la estructura de comunicación entre los procesadores puede hacer que la velocidad aumente o disminuya por eso es muy importante tener en cuenta la arquitectura.

Algunas condiciones dentro de los procesadores o algunos comportamientos como se mencionaba determinan con las metricas la efectividad de la paralelizacion, porque las condiciones de carrera entre los hilos, la dependencia entre los hilos: tanto que un proceso que es ejecutado por un hilo A es necesitado por un hilo B puede causar que los tiempos aumenten o que menos no sea posible con la ejecución del programa.

Por ello es importante conocer el algoritmo base del cual vamos a partir, normalmente se cuenta con la version secuencial y esta permite el correcto funcionamiento en una metrica base para saber si el paralelismo resulta mejor o peor que la version secuencial.

Díaz Hernández Marcos Bryan
Tarea N°6
Grupo: 09

El artículo menciona la relevancia de la versión secuencial como punto base para poder ver las dependencias, la arquitectura y algo relevante: el lenguaje. No se habla mucho del lenguaje, pero este por las bibliotecas puede permitir u obtorgar métodos ó funciones que vuelvan la implementación mucho mas sencilla, pero es necesario considerar en este sentido el paradigma base ya que este puede volver totalmente a nuestro programa secuencial en otro programa y en términos prácticos no compensaría el esfuerzo a menos que la paralelización demuestre valerlo.

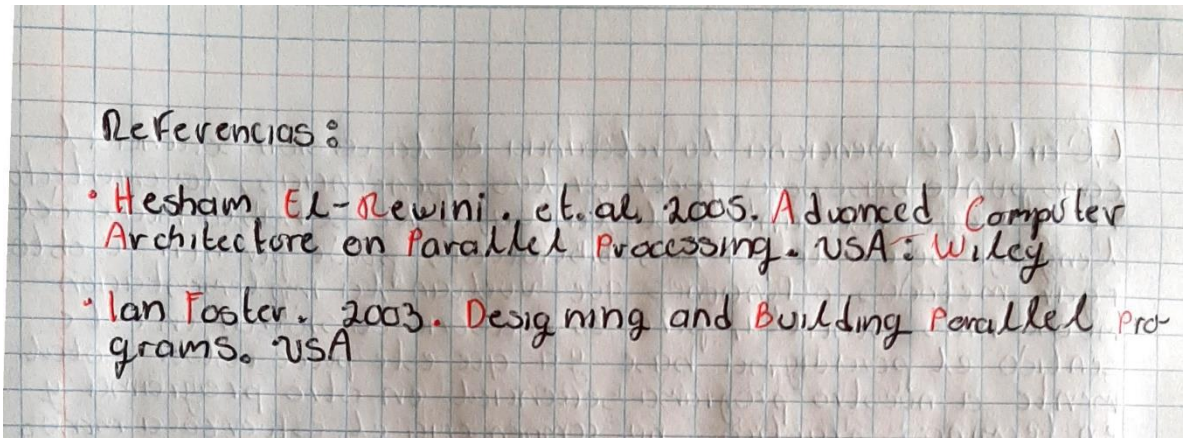
Teniendo dos versiones: secuencial y paralela, la comparación y las expectativas sobre la versión paralela son medibles en base al speedup, la granularidad y por supuesta la medición de las versiones tanto teoricas como practicas ya que en el speedUp teórico hay mucha diferencia entre las versiones, debido a que pueden intervenir todos los factores mencionados.

Una vez se ha decidido que se va a realizar la paralelización en el articulo se nos plantea el analizar la versión secuencial ya que esta permite saber que tanta porción del código se puede paralelizar, en base a esto muchas instrucciones de entrada y salida no son paralelizables en ningún sentido, por esto el identificar más del 95 % de l código que sea paralelizable nos da la pauta para comentar con la tarea de paralelizar sin olvidar todas las condiciones anteriores mencionadas.

- El articulo es vigente y no porque sea muy viejo significa que no esta actualizado, esto es debido a que no se basa en cuestiones avanzadas sino que se enfoca en las instancias necesarias para comentar a paralelizar o damos una idea de este proceso. Por eso es y seguirá siendo vigente hasta que ese proceso de paralelizar sea automatico o una entidad externa lo pueda realizar.

- La aplicación que se le puedar el contenido del / el contenido que nos ofrece el articulo es aplicable al 100 % en la paralelizacion para el proyecto porque es todo el análisis que se debe realizar antes de comenzar a programar y permite obtener la información necesaria para comentar la paralelización.

Díaz Hernández Marcos Bryan
Tarea N°6
Grupo: 09



Referencias:

- Hesham EL-Rewini. et.al, 2005. Advanced Computer Architecture on Parallel Processing. USA: Wiley
- Ian Foster. 2003. Designing and Building Parallel Programs. USA