Análisis del rendimiento de los algoritmos de procesamiento en paralelo en plataformas CPU - GPU

Esther Milagros Bautista Peralta[1]

[1]Universidad Nacional del Altiplano de Puno
[2]Escuela Profesional de Ingeniería Estadística e Informática

Resumen

Uno de los aspectos más relevantes a consideraral momento de analizar la eficiencia de un algoritmo son los tiempos de respuesta, tomando esto en cuenta, este artículo pretende mostrar mediante ejemplos, el comportamiento que presentan los algoritmos paralelos y la eficiencia obtenida.

Se demuestra las diferencias en los tiempos de respuesta entre un enfoque paralelo y un enfoque secuencial utilizados en los estudios realizados entre el 2018 al 2022. Incentivar la programación paralela, puesto que en la actualidad, a pesar de contar con dichas tecnologías, al momento de programar se sigue utilizando una implementación secuencial.

Se efectuó un estudio bibliográfico de revisión sistemática sobre el análisis del rendimiento de los algoritmos de procesamiento en paralelo en plataformas CPU - GPU en los últimos 5 años. Se consideró la búsqueda en la base de datos de la Biblioteca Virtual CONCYTEC y IIIE Xplore. Se utilizó la técnica de la observación para extraer la información y se registraron los indicadores en una ficha de evaluación (Algoritmos, Análisis de datos, Modelo y tiempo de ejecución en CPU y GPU).

La mayoría de los estudios han demostrado que el procesamiento de algoritmos paralelos en la GPU es más eficiente que el procesamiento secuencial en la CPU.

Con base en los resultados mostrados en cada estudio realizado, se puede apreciar el gran potencial que tiene la programación concurrente en comparación con el enfoque secuencial, se presentaron 7 estudios potenciales para la sistematización en los cuales los resultados fueron muy significativos.

Palabras claves: =1programación concurrente, algoritmos paralelos, paralelismo, optimización de algoritmos, enfoque paralelo, CPU, GPU.

Abstract

One of the most relevant aspects to consider when analyzing the efficiency of an algorithm are the response times, taking this into account, this article aims to show by means of examples, the behavior presented by parallel algorithms and the efficiency obtained.

It demonstrates the differences in response times between a parallel approach and a sequential approach used in studies conducted between 2018 to 2022. Incentivize parallel programming, since currently, despite having such technologies, a sequential implementation is still used at the time of programming.

A bibliographic study of systematic review was carried out on the analysis of the performance of parallel processing algorithms on CPU - GPU platforms in the last 5 years. A search in the database of the Virtual Library CONCYTEC and IIIE Xplore was considered. The observation technique was used to extract the information and the indicators were recorded in an evaluation sheet (Algorithms, Data analysis, Model and execution time on CPU and GPU).

Most studies have shown that parallel algorithm processing on GPU is more efficient than sequential processing on CPU.

Based on the results shown in each study performed, it can be appreciated the great potential that concurrent programming has in comparison with the sequential approach, 10 potential studies were presented for systematization in which the results were very significant.

Palabras claves: =1concurrent programming, parallel algorithms, parallelism, algorithm optimization, parallel approach, CPU, GPU.

Introducción

Los sistemas de computación actuales están integrados por cada vez más dispositivos dedicados. Por ejemplo, en

cualquier ordenador personal como los que usamos diariamente tenemos la CPU, encargada de las operaciones principales de gestión de recursos y calculo; una GPU, encargada del procesamiento gráfico; una tarjeta de red, cuya tarea es el envió y recepción de datos a través de diferentes redes de comunicaciones; una tarjeta de sonido, encargada de procesar la entrada y salida de audio; y existen muchos otros. Estos dispositivos nos permiten mejorar el rendimiento de los sistemas sin aumentar en gran medida el coste gracias a su diseño especializado en la tarea para la que están dedicados. La coordinación entre todos los elementos de un sistema, encargando cada tarea al dispositivo específicamente diseñado para ella, nos permite mejorar el rendimiento gracias a un mejor uso de los recursos [1].

Por lo general, la medida del rendimiento de los algoritmos se realiza utilizando diferentes tamaños de datos. Para grandes conjuntos de datos, los modelos de programación paralela que proporcionan una abstracción sobre características específicas del procesador es una opción como OpenCL, OpenMP , CUDA y MPI al comparar el algoritmo paralelo basado en GPU con el algoritmo secuencial en CPU, los resultados han mostrado un buen efecto de aceleración.

El objetivo principal es demostrar las diferencias en los tiempos de respuesta entre un enfoque paralelo y un enfoque secuencial. Así todas aquellas personas que están familiarizadas con el desarrollo del software, podrán notar las eventuales mejoras en el rendimiento de la aplicación, y de esta manera incentivar la programación paralela, puesto que en la actualidad, a pesar de contar con dichas tecnologías, al momento de programar se sigue utilizando una implementación secuencial [2].

Métodos

Tipo de estudio

Se efectuó un estudio bibliográfico de revisión sistemática sobre análisis global del rendimiento de algoritmos paralelos en los últimos 5 años. Se consideró la búsqueda de información desde los años 2018-2022. Una gran parte de los estudios están publicados en el idioma inglés y la otra parte en el idioma español que se han efectuado en todo el mundo. Las palabras clave utilizadas fueron: programación concurrente, algoritmos paralelos, paralelismo, optimización de algoritmos, enfoque paralelo. Se excluyeron investigaciones que fueron desarrolladas fuera del rango de años considerado, incluyendo hasta diciembre 2022. Se excluyeron los estudios que no habían utilizado ningún tipo análisis de rendimiento en los algoritmos paralelos en CPU - GPU [3] .

Técnicas e instrumentos

Se utilizó la técnica de la observación para sistematizar los artículos originales. Se elaboró una ficha de observación para registrar la información [3]. Los indicadores utilizados fueron, Algoritmos, Análisis de datos, Modelo y tiempo de ejecución en CPU y GPU.

Procedimiento de búsqueda bibliográfica

Se utilizó la base de datos de la biblioteca virtual de CONCYTEC (https://biblioteca.concytec.gob.pe/) cuyo enfoque está basado en la Ciencia y la Tecnología y la base de datos de IEEE Xplore (https://ieeexplore.ieee.org/Xplore/home.jsp).

Este periodo de búsqueda de información tuvo como duración una semana desde el 8 al 20 de noviembre 2022. La mejor información de búsqueda de coincidencia fue: parallelism, parallel algorithms, , algorithm optimization, CPU – GPU.

El proceso de selección de estudios se basó en las sugerencias descritas por Moreno B, Munoz M, Cuellar J, Domancic S y Vil- lanueva J siguiendo las cinco fases del diagrama de flujo donde la figura 2 muestra todo el procesos desarrollado. La figura 1 muestra todo el proceso de elaboración de una revisión sistemática..

En la primera etapa se identificó un total de 76 artículos a nivel internacional y nacional (CONCYTEC y EIII Xplore), los que fueron considerados como posibles estudios potenciales para la sistematización. En la segunda etapa se procedió a eliminar artículos duplicados los cuales fueron removidos quedando un total de 69 artículos posibles. En la tercera etapa se proceded a la lectura del total de los Títulos, resúmenes y la metodología de los estudios donde nos queda 27 artículos . En la cuarta etapa se evaluaron los artículos a criterio donde pasaron a ser elegibles (n = 7). En general, se incluyeron los estudios que tenían que ver con las palabras claves utilizadas en el estudio, los que al final se redujeron a dies estudios.

Figure 1: Proceso de elaboración de una revisión sistemática [4].

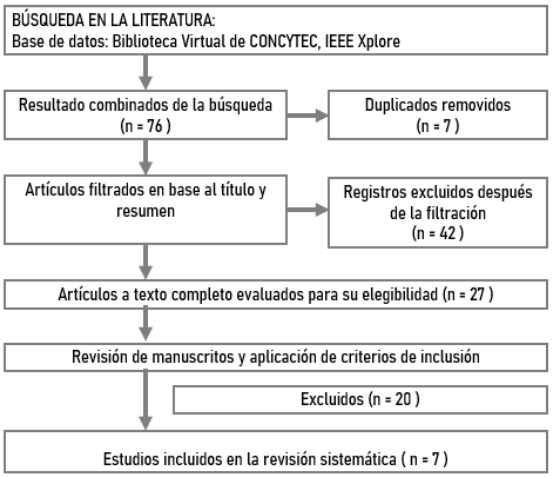

Figure 2: Diagrama de flujo de selección de artículos 2018 - 2022.

Resumen de evaluación

Análisis de estudios

El proceso de organización de los datos de cada uno de los artículos se digitó en tablas en el Excel. A partir de ello, se sistematizó la información por medio de análisis del rendimiento de algoritmos de acuerdo de CPU y GPU. Se describieron los datos en tabla estos indicadores permitieron cuantificar de forma descriptiva los siete estudios según Algoritmos, Análisis de datos, Modelo y tiempo de ejecución en CPU y GPU.

N°	Algoritmos	Análisis de datos	Modelo	Tiempo de Ejecución en milisegundos	
				CPU	GPU
01	multiplicación para matrices cuadradas de 256x256.	50	OpenCLIPER	1.69	$8 \cdot 10^{-2}$
02	algoritmo K-Means	65536	OpenCL	2	3,48
03	Algoritmo de Jacobi: Multiplicación de 2 matrices 8192 × 8192	50 iteraciones		0,986	0,539
04	algoritmos de coincidencia de cadenas BMH (Boyer-Moore-Horspool)	el tamaño de los datos de la cadena es extremadamente grande	*OpenMP*	2.626	2,444
05	Compute Similarity Matrix	200	-	0.0220	0.0331
06	Sparse Eigensolver	200	-	0.3281	0.224
07	K-means Clustering	200	-	0.05154	0.02407

Figure 3: tempo de respuesta de los algoritmos paralelos en CPU y GPU.

Resultados

Finalmente, tras analizar el rendimiento de los algoritmos, se ha visto las fortalezas y debilidades de las diferentes propuestas implementadas.

Discusión

Con base en los resultados mostrados anteriormente, se puede apreciar el gran potencial que tiene la programación concurrente en comparación con el enfoque secuencial. Se presentaron 10 ejemplos en los cuales los resultados fueron muy significativos, no obstante este enfoque puede aplicarse a una gran cantidad de problemas que hoy día solo se han implementado de forma secuencial, sin embargo también hay que aclarar que no en todos los casos los resultados son sobresalientes, existen situaciones en donde los tiempos se mantienen similares con ambos enfoques, esto es porque no todos los algoritmos son susceptibles a la programación multinúcleo. Cada año la innovación en las computadoras va creciendo considerablemente, mejorando sus capacidades de procesamiento, además se cuenta con lenguajes de programación, que nos permiten hacer uso de esas prestaciones ejemplos de ellos son Java y C#, entonces ¿Por qué no aprovechar los recursos? Puede que haya que adaptarse a un entorno algo desconocido, sin embargo las ventajas y el rendimiento proporcionado es lo suficientemente bueno como para aceptar este reto [2].

References

[1] González Díaz J, et al. Desarrollo y optimización de algoritmos paralelos de cálculo general sobre el framework OpenCLIPER. 2022.

[2] Mauricio JALMD, Alpizar R. Análisis de rendimiento de algoritmos paralelos.

[3] Sulla-Torres R, Vidal-Espinoza R, Pacheco-Carrillo J, Apaza-Cruz J, Sulla-Torres J, Luarte-Rocha C, et al. Estimulación eléctrica en niños y adolescentes con parálisis cerebral: Una revisión sistemática. Revista Ecuatoriana de Neurología. 2020;29(1):86-91.

[4] Moreno B, Muñoz M, Cuellar J, Domancic S, Villanueva J. Revisiones Sistemáticas: definición y nociones básicas. Revista clínica de periodoncia, implantología y rehabilitación oral. 2018;11(3):184-6.