Práctica 2.2

CONCEPTOS FUNDAMENTALES DE PARALELISMO

Jordi Blasco Lozano

Razonamiento y representación del conocimiento

Grado en Inteligencia Artificial

## Indice:

[Indice: 2](#_Toc189462408)

[1. Introduccion 3](#_Toc189462409)

[2. Desarrollo 4](#_Toc189462410)

[3. Conclusiones 6](#_Toc189462411)

## Introdución

La computación paralela es un paradigma esencial en la actualidad, ya que permite procesar grandes volúmenes de datos de manera eficiente. A medida que el crecimiento en la velocidad de los procesadores ha comenzado a estancarse, el paralelismo ha surgido como la solución para mejorar el rendimiento y reducir los tiempos de ejecución en diversas aplicaciones. Desde inteligencia artificial hasta simulaciones científicas, la capacidad de dividir una tarea en múltiples subprocesos ha revolucionado el procesamiento de datos y el diseño de software.

En este informe, se explorarán los fundamentos de la computación paralela, sus tipos y aplicaciones, así como las limitaciones y desafíos que enfrenta. Además, se presentará una comparación entre la computación secuencial y la computación paralela, analizando sus ventajas y desventajas.

## Desarrollo

Computación Secuencial vs. Computación Paralela

En la computación secuencial, un solo procesador ejecuta una serie de instrucciones de manera ordenada, sin posibilidad de ejecución simultánea. Este enfoque, aunque más simple, tiene un rendimiento limitado cuando se requiere procesar grandes volúmenes de datos. Por otro lado, la computación paralela permite la ejecución simultánea de múltiples instrucciones, utilizando varias unidades de procesamiento para reducir el tiempo total de ejecución.

## Ejercicio 1 (COMPARACIÓN DE LEY DE AMDAHL Y LEY DE GUSTAFSON)

La aceleración teórica según la Ley de Amdahl se calcula con la fórmula:

Donde:

**- P = 0.7** (fracción paralelizable del código)

**- N = 16** (número de procesadores)

Sustituyendo los valores:

La aceleración según la Ley de Gustafson se calcula con la fórmula:

Donde:

**- = 0.3 (fracción secuencial del código)**

**- N = 16 (número de procesadores)**

Sustituyendo los valores:

**Comparación y conclusiones**

Los resultados muestran que la Ley de Amdahl es más restrictiva, limitando la aceleración máxima debido a la parte secuencial del código. En cambio, la Ley de Gustafson permite mayor escalabilidad cuando el tamaño del problema aumenta. Para aplicaciones donde el problema puede crecer, la Ley de Gustafson es más favorable.

Para mejorar la eficiencia del programa, se debería de reducir la fracción secuencial del código optimizando algoritmos para asi aumentar la aceleracion del problema.

## Ejercicio 2 (OPTIMIZACIÓN DE UN PROCESO INDUSTRIAL MEDIANTE PARALELIZACIÓN)

Identificación de oportunidades de paralelización

\*\*a. Análisis de cada paso del proceso para identificar posibles paralelismos:\*\*

Para identificar oportunidades de paralelización, es fundamental descomponer el proceso en sus componentes individuales y examinar cuáles de estos pueden ejecutarse simultáneamente sin interferir entre sí. Por ejemplo, en una aplicación de procesamiento de imágenes, las etapas de filtrado, detección de bordes y segmentación podrían ejecutarse en paralelo si están diseñadas adecuadamente.

\*\*b. División de tareas dentro de un mismo paso para ejecución simultánea:\*\*

Sí, existen tareas que dentro de un mismo paso pueden dividirse y ejecutarse en paralelo. Por ejemplo, en el cálculo de una función matemática sobre un conjunto grande de datos, cada cálculo individual puede llevarse a cabo en paralelo en diferentes núcleos de procesamiento. Esto justifica la paralelización, ya que reduce el tiempo total de procesamiento al aprovechar múltiples recursos simultáneamente.

### 2. Modelo de paralelismo adecuado

\*\*a. Tipo de paralelismo propuesto:\*\*

Para mejorar el proceso, se podría implementar un \*\*paralelismo a nivel de tareas\*\*. Este enfoque permite que diferentes tareas independientes se ejecuten simultáneamente en distintos procesadores o núcleos, optimizando así el uso de los recursos disponibles y reduciendo el tiempo de ejecución total.

\*\*b. Aplicación en la verificación de calidad:\*\*

Sí, la verificación de calidad puede dividirse entre distintos robots o sensores que trabajen en paralelo. Por ejemplo, múltiples robots podrían inspeccionar diferentes partes de un producto al mismo tiempo, aumentando la eficiencia y reduciendo el tiempo necesario para completar el proceso de verificación.

### 3. Ley de Amdahl o Gustafson

\*\*a. Cálculo de la aceleración teórica usando la Ley de Amdahl:\*\*

La Ley de Amdahl se expresa como:

\[ \text{Aceleración} = \frac{1}{(1 - P) + \frac{P}{N}} \]

Donde:

- \( P = 0.6 \) (60% del proceso puede paralelizarse)

- \( N = 8 \) (número de robots trabajando en paralelo)

Sustituyendo los valores:

\[ \text{Aceleración} = \frac{1}{(1 - 0.6) + \frac{0.6}{8}} = \frac{1}{0.4 + 0.075} = \frac{1}{0.475} \approx 2.105 \]

Por lo tanto, la aceleración teórica es aproximadamente \*\*2.105 veces\*\* más rápida que el proceso secuencial.

\*\*b. Escalabilidad aumentando el número de robots:\*\*

Si el tamaño de la producción se incrementa y se agregan más robots, la parte paralelizable del proceso puede escalarse eficientemente. Sin embargo, según la Ley de Amdahl, existe un límite teórico en la aceleración máxima que se puede lograr debido a la parte secuencial del proceso. A medida que se incrementa \( N \), la influencia de \( P \) se vuelve más significativa, pero nunca se puede superar el límite impuesto por la parte secuencial.

Por ejemplo, si se duplican los robots a \( N = 16 \):

\[ \text{Aceleración} = \frac{1}{0.4 + \frac{0.6}{16}} = \frac{1}{0.4 + 0.0375} = \frac{1}{0.4375} \approx 2.285 \]

Aunque la aceleración aumenta, la mejora es menos significativa debido a la parte secuencial del proceso.

---

Si necesitas más detalles o asistencia con otros ejercicios, no dudes en decírmelo.