

# Trabajo de Investigación sobre Ley de Amdahl y Gustafson

Patrick Xavier Márquez Choque  
Univesidad Católica San Pablo  
Arequipa, Perú  
Email: `patrick.marquez@ucsp.edu.pe`

Abril 2021

## Abstract

Este trabajo tiene como objetivo investigar sobre las leyes de Amdahl y Gustafson para realizar un cuadro comparativo de ambas leyes en conjunto con un ejemplo para mostrar las limitaciones y características de cada uno.

## 1 Ley de Amdahl

Es una ley de ciencia de la computación formulada por **Gene Amdahl** (1) utilizada para descubrir la máxima mejora que se puede obtener en el rendimiento de un sistema resultante de la modificación de alguno de sus componentes.

Según el Libro de Peter Pacheco (1) se menciona que en el año 1960, Gene Amdahl dice que a menos que prácticamente cada programa y tarea esté en paralelo entonces la mejora en los tiempos van a estar muy limitados; independientemente del número de núcleos que se tiene disponible.

Hay un ejemplo bastante entendible donde te menciona que:

- EL tiempo serial de ejecución de un programa son de 20 segundos.
- El tiempo ejecución de la parte paralelizada es un 90% de todo el programa.

- Entonces el tiempo de ejecución paralelizada será de  $0.9 * \text{Tiempo}_{\text{serial}} / p = 18$  segundos.
- $P$  es el número de núcleos que vamos a utilizar.
- **Entonces** el tiempo total como mejora en la ejecución paralela serán de 2 segundos.

Entonces la fórmula original de la ley de amdahl (2) es que

$$T_{\text{máx}} = T_{\text{act}} * ((1 - P_{\text{mej}}) + \frac{P_{\text{mej}}}{F_{\text{nuc}}})$$

siendo:

- $T_{\text{máx}}$ : Tiempo máximo de mejora de ejecución.
- $T_{\text{act}}$ : Tiempo actual del programa antes de la mejora.
- $P_{\text{mej}}$ : Porcentaje de la mejorar que el sistema esta utilizando.
- $P_{\text{nuc}}$ : Factor de mejora especifica del sistema.

## 2 Ley de Gustafson

La llamada ley de Gustafson-Barsis es una ley de ciencia de la computación que establece un concepto similar en cuanto a la mejora del tiempo pero especifica que prácticamente cualquier programa puede ser eficientemente paralelo.

Según el Libro de Peter Pacheco (1) se menciona que hay miles de programas utilizados por científicos e ingenieros que obtienen grandes aceleraciones en sistemas de memoria distribuida pero la obtención de aceleración es siempre adecuado, especialmente si el esfuerzo involucrado en el desarrollo del programa paralelo no es muy grande.

Pero es aquí donde se puede apreciar el objetivo de esta ley ya que es obtener mejora en el rendimiento del tiempo de ejecución considerando específicamente la cantidad de unidades de cómputo o núcleos que se está utilizando de la siguiente manera:

$$S(P) = P - \alpha * (P - 1)$$

- $S(P)$  es la mejora obtenida a partir de la cantidad de núcleos o procesadores.
- $P$  es el número de procesadores del computador utilizado.
- $\alpha$  es la parte no paralelizable del proceso del programa que se quiere paralelizar.

### 3 Cuadro Comparativo de Ambas Leyes (2)

Características	Ley de Amdahl	Ley de Gustafson
Objetivo	Obtener el tiempo máximo de mejora como límite de la mejora en el tiempo de ejecución.	Obtener la mejora del tiempo de ejecución y Establecer que cualquier programa puede ser eficientemente paralelizado.
Consideraciones	Considera un porcentaje de mejora del programa paralelizado.	Específicamente considera la cantidad de procesadores a la disposición.
Fórmula	Matemáticamente considera el tiempo de ejecución antiguo y el mejorado.	Matemáticamente considera los procesadores y la parte no paralelizable
Limitaciones	No considera el la cantidad de procesadores para obtener un resultado.	Limitaciones en el hardware y la cantidad de procesadores.
Obtención	Tiempo de mejora máximo.	Tiempo de mejora en base a procesadores.
Fundador	Gene Amdahl	Gustafson-Barsis
Define	El límite de mejora del tiempo de ejecución máxima que puede obtener un programa.	La eficiencia como una necesidad para minimizar la parte secuencial del programa

Figure 1: Cuadro comparativo de Ambas leyes

## 4 Ejemplo de ambas leyes

Vamos a dar el siguiente ejemplo:

En el caso que yo como un jefe de una empresa de cómputo que quiera calcular el tiempo de mejora en el tiempo de ejecución para uno de mis proyectos bastantes importante y con una cantidad considerable de líneas de código y por lo tanto es que necesito de un buen plan para poder paralelizar y obtener una mejora en el rendimiento de mi programa.

Supongamos que el tiempo actual que tengo para mi programa es de unos **30 segundos**, es un proceso que utiliza un montón de tareas y por lo tanto son necesarios una cantidad considerable de tiempos para obtener resultados. Estoy utilizando un procesador que tiene 16 núcleos disponibles para su uso y que debido a la gran cantidad de ejecuciones y procesos que tengo solo tengo un 50% del programa como paralelizable entonces según ambas leyes tenemos que:

- Según la ley de Amdahl tenemos que  $P_{nuc}$  es igual al  $T_{act}$  multiplicado al  $P_{mej}$  dividido entre el número de núcleos ya que es nuestro único metodo para obtener una mejora en nuestros tiempos de ejecución entonces seria:  $0.5 * 30 / 16 + 0.1 * 30 = 3.9375$  entonces el tiempo máximo de mejora será de: 7.619047 como máximo de tiempo de mejora en tiempo de ejecución.
- Según la ley de Gustafson tenemos que nuestra mejora como speed up será de  $16 - 0.94446 * (16 - 1)$  ya que según que tenemos que es paralelizable 50% de nuestro programa entonces tenemos una mejora de 1.83 segundos en tiempo de ejecución al hacer todo esto un porcentaje paralelizable.

Entonces podemos concluir que según la mejora de la ley de amdahl es mayor ya que sería la máxima mejora al realizar cambios dentro de nuestro código pero eso dependerá de nuestro porcentaje de programa paralelizable y ya que en la segunda ley nosotros podemos obtener una mejora en el tiempo de 8.5 segundos es como vemos que esta paralelización es mucho menos ya que nuestro porcentaje de procesos no paralelizables pueden podrían pero no disminuyen entonces se pueden obtener menores tiempos que estos.

## References

- [1] P. Pacheco, *An introduction to parallel programming*. Elsevier, 2011.
- [2] “Reevaluating amdahl’s law,” 2007. [Online]. Available: <https://web.archive.org/web/20070927040654/http://www.scl.ameslab.gov/Publications/Gus/AmdahlsLaw/Amdahls.html>