“AÑO DE LA UNIVERSALIZACIÓN DE LA SALUD”.

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

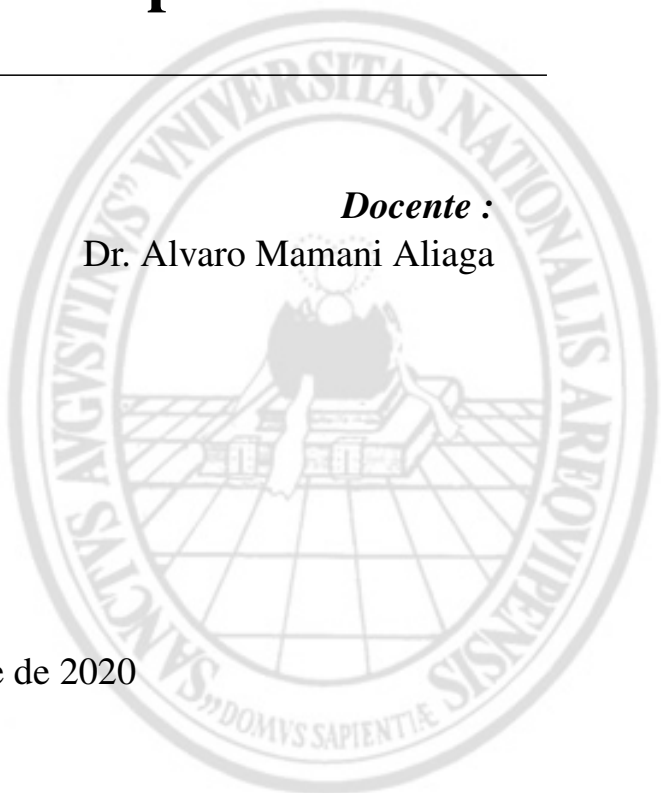
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

COMPUTACIÓN PARALELA

Leyes sobre el aumento de prestaciones

Alumnos:
Miguel Alexander, Herrera Cooper

Docente :
Dr. Alvaro Mamani Aliaga



11 de noviembre de 2020

Índice

1. GRADO DE PARALELISMO (DOP)

DOP: Es el número de procesos paralelos en los que se puede dividir un programa en un instante dado.

Suposición: un sólo programa en ejecución.

El DOP puede exceder el número de procesadores disponibles ->algunas bifurcaciones 3enen que ejecutarse en trozos secuencialmente.

- n=número de procesadores homogéneos.
- m=paralelismo máximo

- Idealmente, nm

  k=procesadores disponibles

  =Capacidad de computación de un procesador, en MIPS, MFLOPS, ...

  DOP = i =>hay i procesadores ocupados

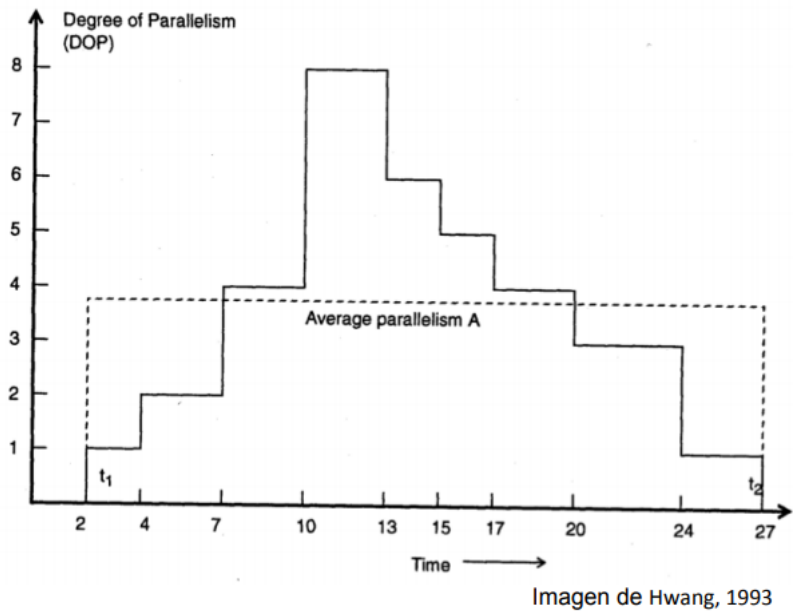


Figura 1: Ejemplo de Perfil de Paralelismo

1.1. PARALELISMO MEDIO

Cantidad de trabajo (instrucciones):

$$W = \Delta \cdot \int_{t_1}^{t_2} DOP(t)dt \qquad W = \Delta \sum_{i=1}^{m} i \cdot t_i$$

Figura 2: Ejemplo de Perfil de Paralelismo

Cantidad de trabajo (instrucciones):

$$A = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} DOP(t)dt \quad A = \frac{\sum_{i=1}^{m} i \cdot t_i}{\sum_{i=1}^{m} t_i}$$

Figura 3: Ejemplo de Perfil de Paralelismo

2. LEY DE AMDAHL

Evalúa como cambia el rendimiento al mejorar una parte de la computadora.

Establece que: La mejora obtenida en el rendimiento de un sistema debido a la alteración de uno de sus componentes está limitada por la fracción de tiempo que se utiliza dicho componente. El incremento de velocidad de un programa utilizando múltiples procesadores en computación distribuida está limitada por la fracción secuencial del programa.

- Carga computacional fija
- Al incrementar el número de procesadores la carga se distribuye.
- Obje3vo: producir el resultado lo antes posible.
- i>n (número de procesadores limitado)

$$t_i(n) = \frac{W_i}{i \cdot \Delta} \left\lceil \frac{i}{n} \right\rceil \qquad T(n) = \sum_{i=1}^{m} \frac{W_i}{i \cdot \Delta} techo\left(\frac{i}{n}\right)$$

- i<n

$$t_i(n) = t_i(\infty) = \frac{W_i}{i\Delta}$$

2.1. SpeedUp

Define el speedup (aceleración) que se puede alcanzar al usar cierta mejora.

$SpeedUp = \frac{Rendimientoalusarlamejora}{Rendimientosinusarlamejora}$

2.2. Leyes

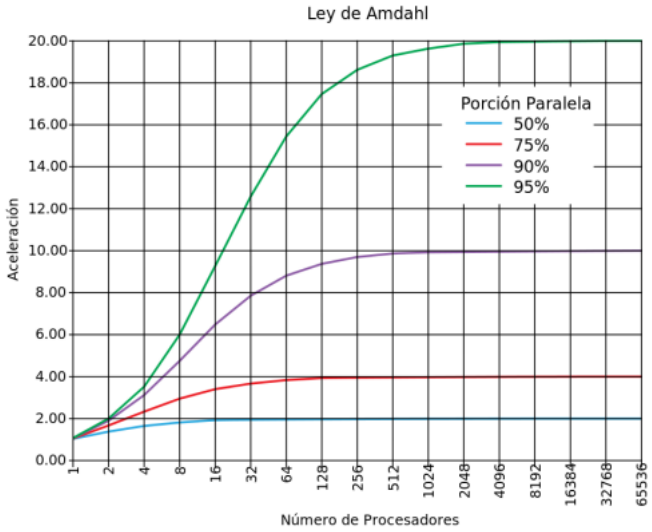
1. El aumento del rendimiento debido a la inclusión de una mejora con un nuevo recurso en el sistema está limitado por el tiempo que se utiliza dicha mejora en la ejecución de la tarea.

2. Cuando se introduce una mejora a un computador previamente mejorado, el incremento del rendimiento es menor que si se introduce la mejora sobre el sistema sin mejorar. Dicho de otra forma, la mejora incremental en la aceleración conseguida con la mejora de una parte se va reduciendo a medida que se van introduciendo nuevas mejoras (Angulo et all, 1995).

2.3. Aplicaciones

El incremento de velocidad de un programa u3lizando múltiples procesadores en computación distribuida está limitada por la fracción secuencial del programa.

Por ejemplo, si la porción 0.5 del programa es secuencial, el incremento de velocidad máximo teórico con computación distribuida será de 2 (1/(0.5+(1-0.5)/n)) cuando n sea muy grande.



Un sistema de discos en un centro de cómputo tiene varios componentes.

Un estudio determina que instalando otra fuente de poder eleva la confiabilidad (tiempo medio entre fallas) de las fuentes de poder en 4,150 veces.

Cuando se presenta una falla, el porcentaje de que sea alguno de los componentes es como sigue:

Componente	Porcentaje
Discos	43%
Controlador SCSI	9%
Fuente de poder	22%
Abanico	22%
Cable SCSI	4%

2.4. Ejemplo

Se desea mejorar el rendimiento de un computador introduciendo un coprocesador matemático que realice las operaciones en la mitad de 3empo. Calcular la ganancia en velocidad del sistema para la ejecución de un programa si el 60 % del mismo se dedica a operaciones aritmé3cas. Si el programa tarda 12 segundos en ejecutarse sin la mejora, ¿cuánto tardará con la mejora?

- Am = 2 y Fm = 0,6

$$A = \frac{1}{(1-0{,}6)+\frac{0{,}6}{2}} = 1{,}42$$

- Con lo que el sistema es un 42% más rápido

$$A = \frac{TiempoEjecuciónSinMejora}{TiempoEjecuciónConMejora} \Rightarrow 1{,}42 = \frac{12}{TiempoEjecuciónConMejora}$$

- Lo que hace que el programe tarde 8,45 segundos

3. Ley de Gustafson

A finales de la década de los 80, John Gustafson y su equipo de investigación en Sandia National Laboratorios se encontraban realizando investigación relacionada con el uso de Procesamiento Masivamente Paralelo y notaron que existía un alto nivel de escepticismo por el uso de estos sistemas paralelos debido a la ley de Amdahl.

Según esta ley, aún en problemas con una fracción secuencial de trabajo muy pequeña, la aceleración máxima alcanzable para un número infinito de procesadores era de solamente 1/s. Con los resultados de las investigaciones que realizaron en un sistema con 1024 procesadores demostraron que las suposiciones de la ley de Amdahl no son apropiadas para el caso de paralelismo masivo.

3.1. Implementación de la ley de Gustafson

- Sea n una medida del tamaño del problema.
- El tiempo de ejecución de un programa en un computador paralelo es descompuesto en: a(n) + b(n) = 1
- donde a es la fracción secuencial y b es la fracción paralela.
- En un computador secuencial, el tiempo relativo será igual a a(n) + pb(n) donde p es el número de procesadores para el caso paralelo
- La aceleración es entonces : (a(n) + pb(n))
- Si la función secuencial a(n) disminuye a medida que se incrementa el tamaño del problema n, entonces la aceleración alcanzará p cuando se aproxima a infinito.
- Por lo tanto la ley de Gustafson rescata el procesamiento paralelo que no era favorecido por la ley de Amdahl.
- Durante sus investigaciones Gustafson se dio cuenta que en la práctica, el volumen del problema no es independiente del número de procesadores, ya que con mayor número de procesadores se pueden abordar problemas de mayores dimensiones. Por ello, la ley de Gustafson se refiere al crecimiento del volumen de cálculo necesario para resolver un problema. Como veremos a continuación, en la mayoría de los casos, cuando el volumen del problema crece, lo hace sólo en su parte paralela, no en su parte secuencial. Ello hace que el cuello de botella secuencial tienda a cero cuando el volumen del problema aumenta.

  La razón de que cierto tipo de problemas adquieran gran volumen de cálculo es la disminución del tamaño de la malla de cálculo o también el aumento la extensión espacio-temporal del problema. Esto hace que el número de puntos aumente de forma cúbica respecto al grado de disminución en la malla, si el problema es tridimensional. Hay muchos problemas en que, además de las tres dimensiones del espacio, también interviene el tiempo, por lo que el aumento del volumen de cálculo es todavía mayor. Evidentemente, esto afecta, en general, a la parte paralelizable del problema y no a su parte secuencial, o al menos no en la misma medida. Si suponemos que el número de procesadores crece indefinidamente de la misma forma que las dimensiones del problema tendremos que

$$\lim_{N\to\infty} f = \lim_{N\to\infty} \frac{s}{s+Np} = 0$$

- siendo s y p, respectivamente, las partes secuencial y paralela del problema antes de ser aumen- tado en relación al número del procesadores (por ejemplo, incrementando el número de puntos de la malla).
- Con estas premisas, podremos calcular ahora la ganancia de velocidad para esta nueva situación (la parte paralela del problema ha crecido en la misma proporción que el número de procesadores). Para calcular la ganancia de velocidad supondremos que el tiempo que se tarda- ría en ejecutar el programa (ya incrementado) en un monoprocesador es:
- t(1) = S+Np
- y en un sistema paralelo sería: t(N)=S+P
- Por tanto, la ganancia en velocidad vendrá dada por:
- S = f + N(1 - f) = N - (N - 1)f
- Podría pensarse que hay una aparente contradicción entre las leyes de Amdahl y Gustafson. Esto no es así debido a que las premisas de ambas leyes son distintas: la ley de Amdahl se refiere a procesos con un volumen de cálculo fijo mientras que la ley de Gustafson se refiere a problemas cuyo volumen de cálculo puede aumentar según el número de procesadores (esto se suele denominar escalado del problema). Afortunadamente, esta última situación es más cercana a la realidad.
- La ley de Gustafson argumenta que un aumento cuádruple del poder de cómputo conllevaría a un incremento similar en las capacidades del sistema. Si un minuto de tiempo de inicio de un sistema es aceptable para la mayoría de los usuarios, entonces esto es un punto de inicio desde donde incrementar las funcionalidades y características del sistema. El tiempo de inicio del sistema operativo se mantendrá igual, o sea un minuto, pero el nuevo sistema incluirá mejores características gráficas y funcionalidades para el usuario.

3.2. Ejemplo

Suponemos que un programa consiste de =0.67 partes fraccionales de un código serial, y 0.33 partes fraccionales de código paralelo. Que velocidad se espera para este programa cuando este corre a n=10 procesadores en paralelo?

- Según la ley de Amdhal:

$$S_n = \frac{1}{\alpha + \frac{1-\alpha}{n}}$$

  - S_n=1/(0.67 + (0.33/10)) = 1.42

- Según la ley de Gustafson:

$$S'_n = n - \alpha(n-1)$$

  - S'_n=10-(10-1)(0.67)= 3.97