Tema 3. Leyes sobre el aumento de prestaciones

Organización de Computadores

LUIS ENRIQUE MORENO LORENTE
RAÚL PÉRULA MARTÍNEZ
ALBERTO BRUNETE GONZALEZ
DOMINGO MIGUEL GUINEA GARCIA ALEGRE
CESAR AUGUSTO ARISMENDI GUTIERREZ
JOSÉ CARLOS CASTILLO MONTOYA

Departamento de Ingeniería de Sistemas y Automática

LEYES

- Ley de Amdahl
- Ley de Gustafson
- Modelo de Sun-Ni

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 tienen que ejecutarse en trozos secuencialmente.

n=número de procesadores homogéneos

m=paralelismo máximo

Idealmente, n>>m

k=procesadores disponibles

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

DOP = i => hay i procesadores ocupados

EJEMPLO: PERFIL DE PARALELISMO

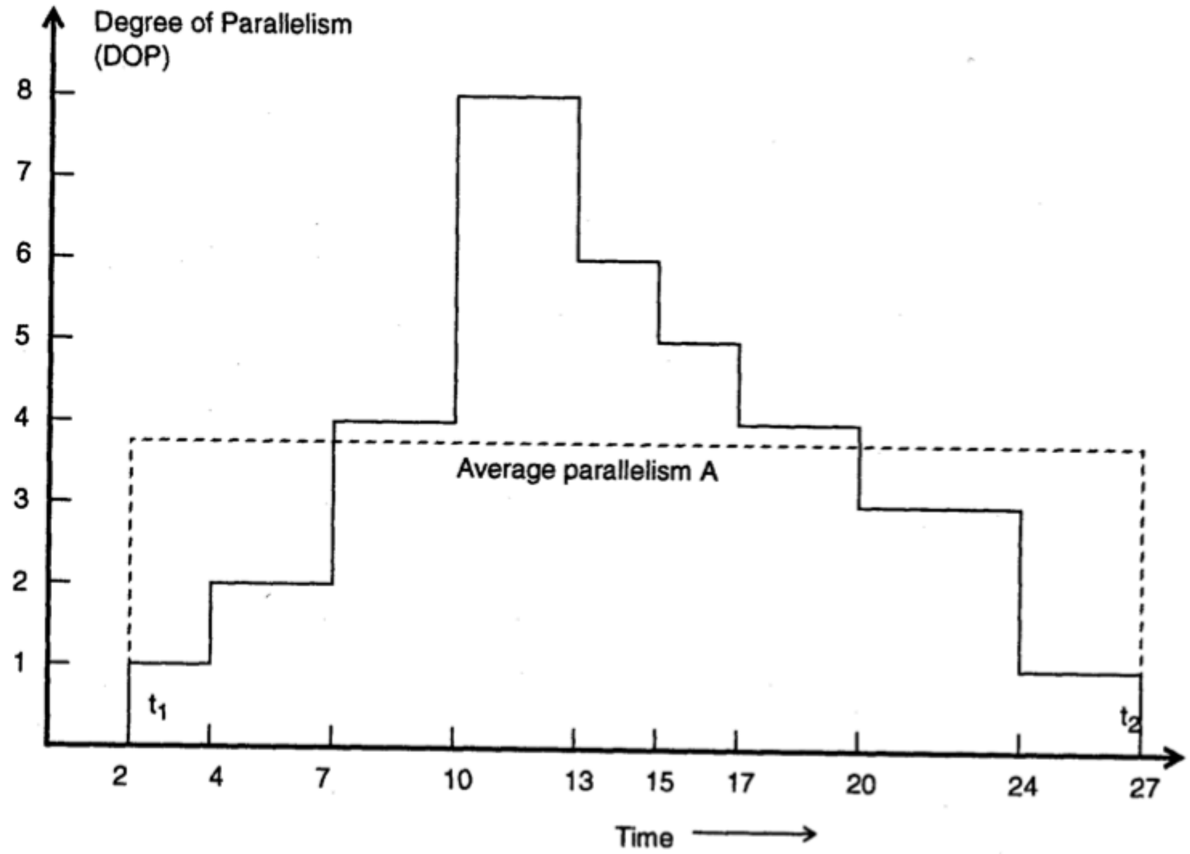


Imagen de Hwang, 1993

PARALELISMO MEDIO

Cantidad de trabajo (instrucciones):

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

Paralelismo medio:

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

EJEMPLO:

Consideremos una arquitectura con 3 procesadores donde los accesos a memoria necesitan para ejecutarse 2 ciclos y las operaciones en punto flotante 5.

Calcular el perfil de paralelismo y el DOP medio del siguiente programa:

```
Ld r1, A
Ld r2, B
Ld r7, C
Add r4, r1, r1
Mul r8, r7, r7
Addi r3, r2, 1
Sto D, r4
Sub r5, r4, r8
Ld r6, E
Addi r6, r6, 3
Add r6, r6, r5
Sto F, r6
```

SOLUCIÓN:

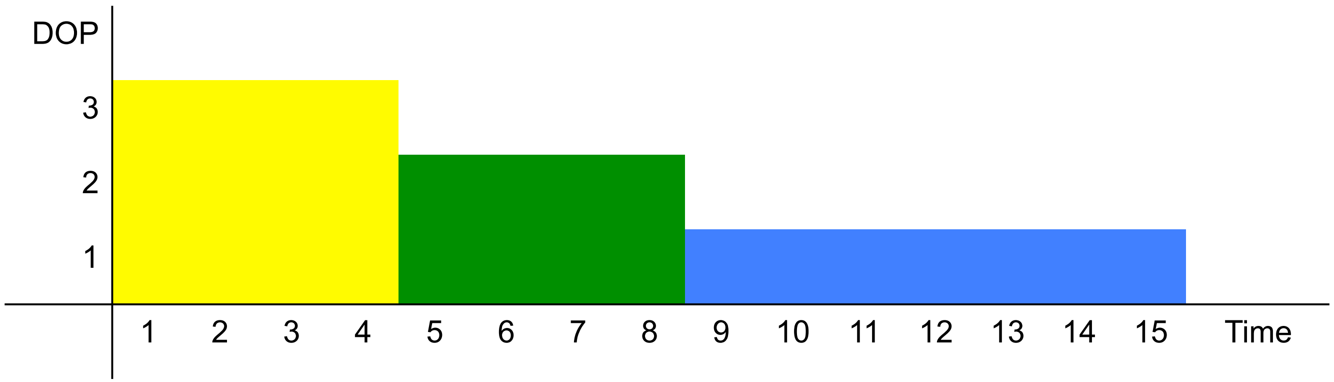
Una posible ejecución del programa sería la siguiente:

```
Ld r1, A
Ld r2, B
Ld r7, C
Add r4, r1, r1
Mul r8, r7, r7
Addi r3, r2, 1
Sto D, r4
Sub r5, r4, r8
Ld r6, E
Addi r6, r6, 3
Add r6, r6, r5
Sto F, r6
```

# ciclo	P1	P2	P3
1	Ld r1, A	Ld r2, B	Ld r7, C
2	"	"	"
3	Add r4, r1, r1	Addi r3, r2, 1	Mul r8, r7, r7
4	Sto D, r4	Ld r6,E	"
5		"	"
6		Addi r6, r6, 3	"
7		Sto G, r6	"
8		"	Ld r4, D
9			"
10			Sub r5, r4, r8
11			Ld r6, G
12			"
13			Add r6, r6, r5
14			Sto F, r6
15			"

SOLUCIÓN:

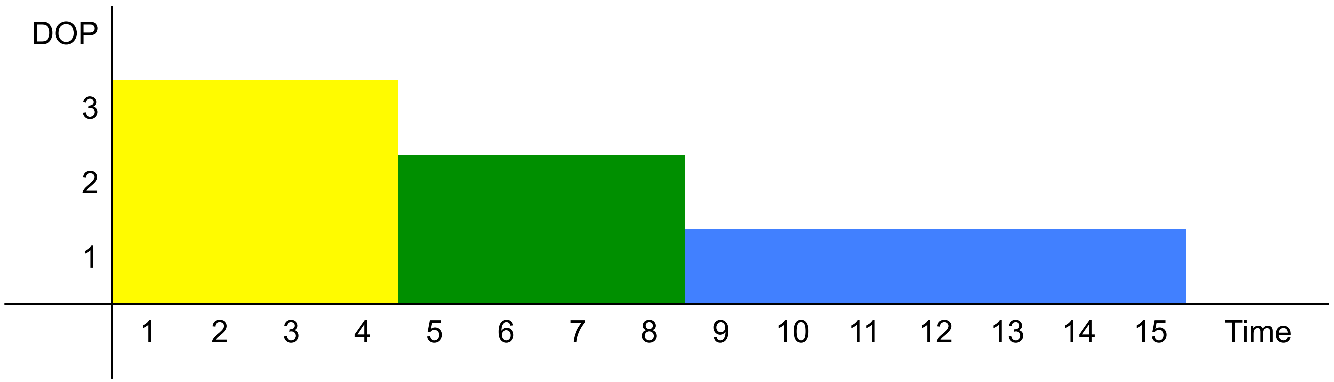
Con lo que se obtiene el siguiente perfil de paralelismo y es posible calcular el paralelismo medio:



$$A = \frac{\sum_{i=1}^{m} i \cdot t_i}{\sum_{i=1}^{m} t_i} \qquad A = \frac{3*4+2*4+7}{15} = \frac{27}{15} = 1.8$$

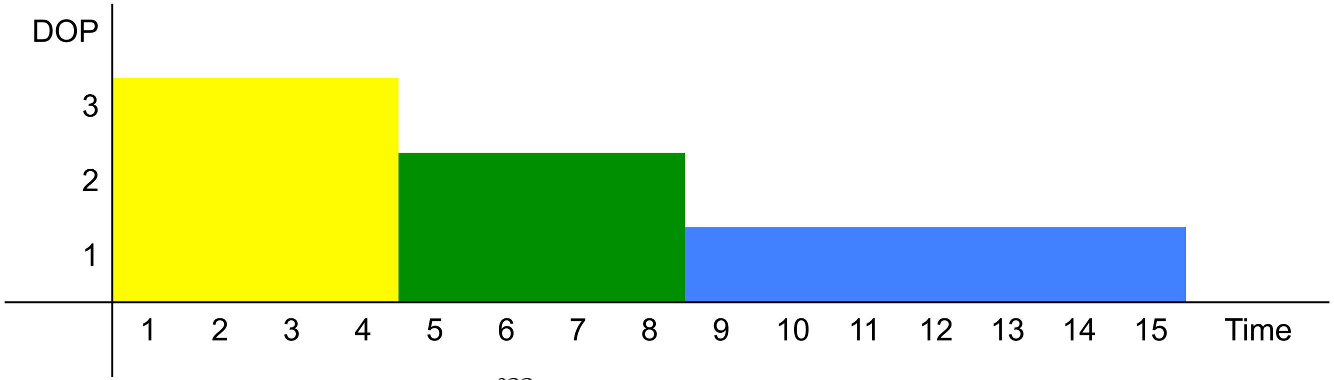
SOLUCIÓN:

Si la máquina anterior tiene una capacidad de cálculo de 8 MFLOPS, ¿cual es la cantidad de trabajo total generada por el programa anterior?



SOLUCIÓN:

Si la máquina anterior tiene una capacidad de cálculo de 8 MFLOPS, ¿cual es la cantidad de trabajo total generada por el programa anterior?



$$W = \Delta \sum_{i=1}^{m} i \cdot t_i$$

$$W = 8 * (3 * 4 + 2 * 4 + 7) = 216\ MFLOPS$$

SPEEDUP ASINTÓTICO

$$W_i = i \cdot \Delta \cdot t_i \qquad\qquad W_i = k \cdot \Delta \cdot t_i$$

Tiempo de ejecución

$$T(n) = \sum_{i=1}^{m} t_i(n)$$

$$T(1) = \sum_{i=1}^{m} t_i(1) = \sum_{i=1}^{m} \frac{W_i}{\Delta}$$

$$T(\infty) = \sum_{i=1}^{m} t_i(\infty) = \sum_{i=1}^{m} \frac{W_i}{i\Delta}$$

Speedup asintótico

$$S_\infty = \frac{T(1)}{T(\infty)} = \frac{\sum_{i=1}^{m} W_i}{\sum_{i=1}^{m} W_i / i}$$

LEY DE AMDAHL

- **Carga computacional fija**
- Al incrementar el número de procesadores la carga se distribuye.
- Objetivo: 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

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

LEY DE AMDAHL

El speedup sería:

$$S_n = \frac{T(1)}{T(n)} = \frac{\sum_{i=1}^{m} W_i}{\sum_{i=1}^{m} \frac{W_i}{i} techo[\frac{i}{n}]}$$

$$S_n \leq S_\infty \leq A$$

Si suponemos que el programa trabaja sólo en modo serie o con paralelismo máximo se obtiene la expresión clásica de la Ley de Amdahl.

$$W_i \neq 0 \, si \, i=1 \, ó \, i=n$$

$$\boxed{S_n = \frac{W_1 + W_n}{W_1 + \frac{W_n}{n}}}$$

LEY DE AMDAHL

$$S_n = \frac{W_1 + W_n}{W_1 + \frac{W_n}{n}}$$

- Si lo expresamos en términos de la fracción porcentual del código del programa

$$W_1 = \alpha$$
$$W_n = 1 - \alpha$$

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

Parte no paralela

Parte paralela

$$S_n = \frac{1}{(1-f) + \frac{f}{s}}$$

f=parte paralela

s=speedup de la parte paralela

LEY DE AMDAHL

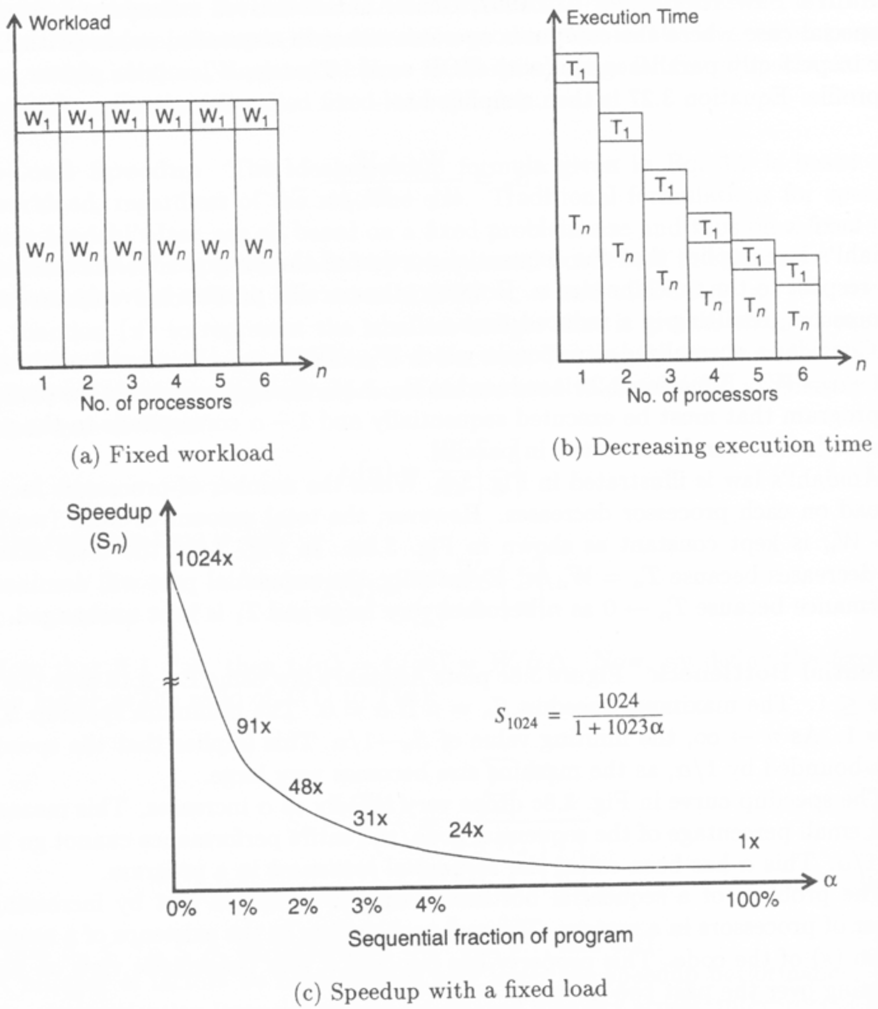


Figure 3.8 Fixed-load speedup model and Amdahl's law.

Imagen de Hwang, 1993

LEY DE AMDAHL

- El incremento de velocidad de un programa utilizando 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.

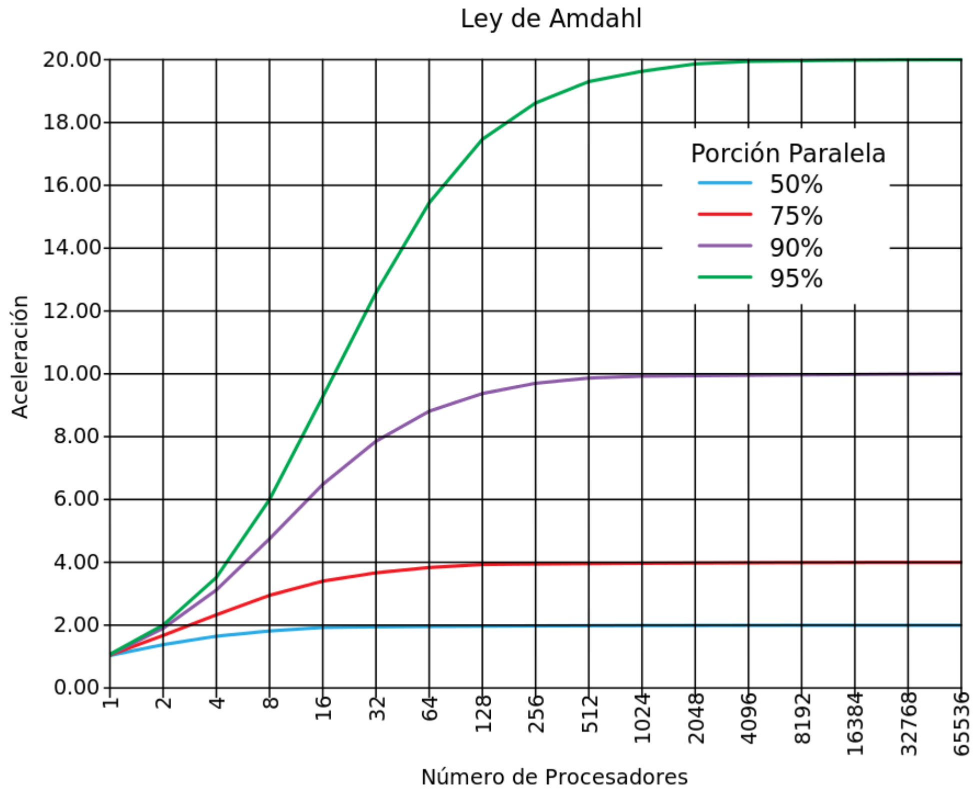


Imagen de Wikipedia, 2014

LEY DE AMDAHL: EJEMPLO

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

LEY DE AMDAHL: EJEMPLO

- Se desea mejorar el rendimiento de un computador introduciendo un coprocesador matemático que realice las operaciones en la mitad de tiempo. Calcular la ganancia en velocidad del sistema para la ejecución de un programa si el 60% del mismo se dedica a operaciones aritméticas. 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

LEY DE GUSTAVSON

- Carga computacional variable
- **Tiempo de cómputo FIJO**
- Al incrementar el número de procesadores la carga se distribuye y se puede calcular más en el mismo tiempo.
- El objetivo: resolver el problema de mayor tamaño posible W' en la máquina paralela (o producir el resultado con la mayor precisión posible) en el mismo tiempo de cómputo que el problema original W en la máquina de referencia.
- $W_i' > W_i$ para $2 \leq i \leq m'$ y $W_1' = W_1$
- Hipótesis de trabajo: T(1) = T'(n)

$$\sum_{i=1}^{m} W_i = \sum_{i=1}^{m'} \frac{W_i'}{i} \left\lceil \frac{i}{n} \right\rceil + Q(n)$$

LEY DE GUSTAVSON

- Si suponemos que el programa se ejecuta sólo en modo secuencial o paralelo:

$$S'_n = \frac{\sum_{i=1}^{m'} W'_i}{\sum_{i=1}^{m} W_i} = \frac{W_1 + nW_n}{W_1 + W_n}$$

- Que expresado en fracción porcentual toma la expresión:

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

LEY DE GUSTAVSON

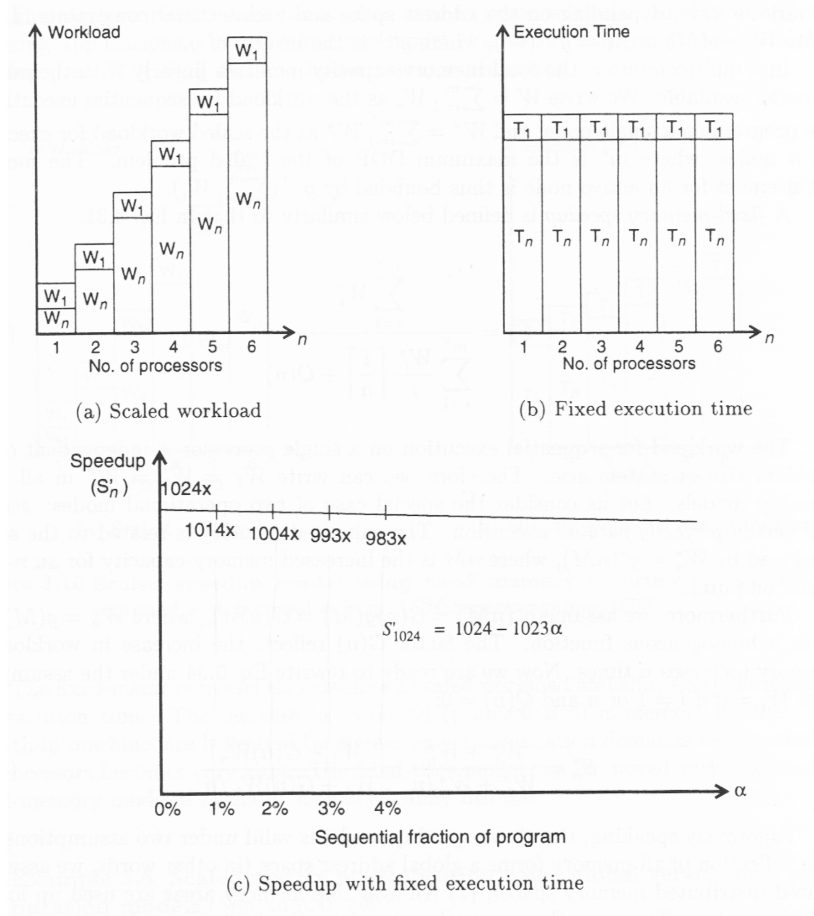


Figure 3.9 Fixed-time speedup model and Gustafson's law.

Imagen de Hwang, 1993

LEY DE GUSTAVSON

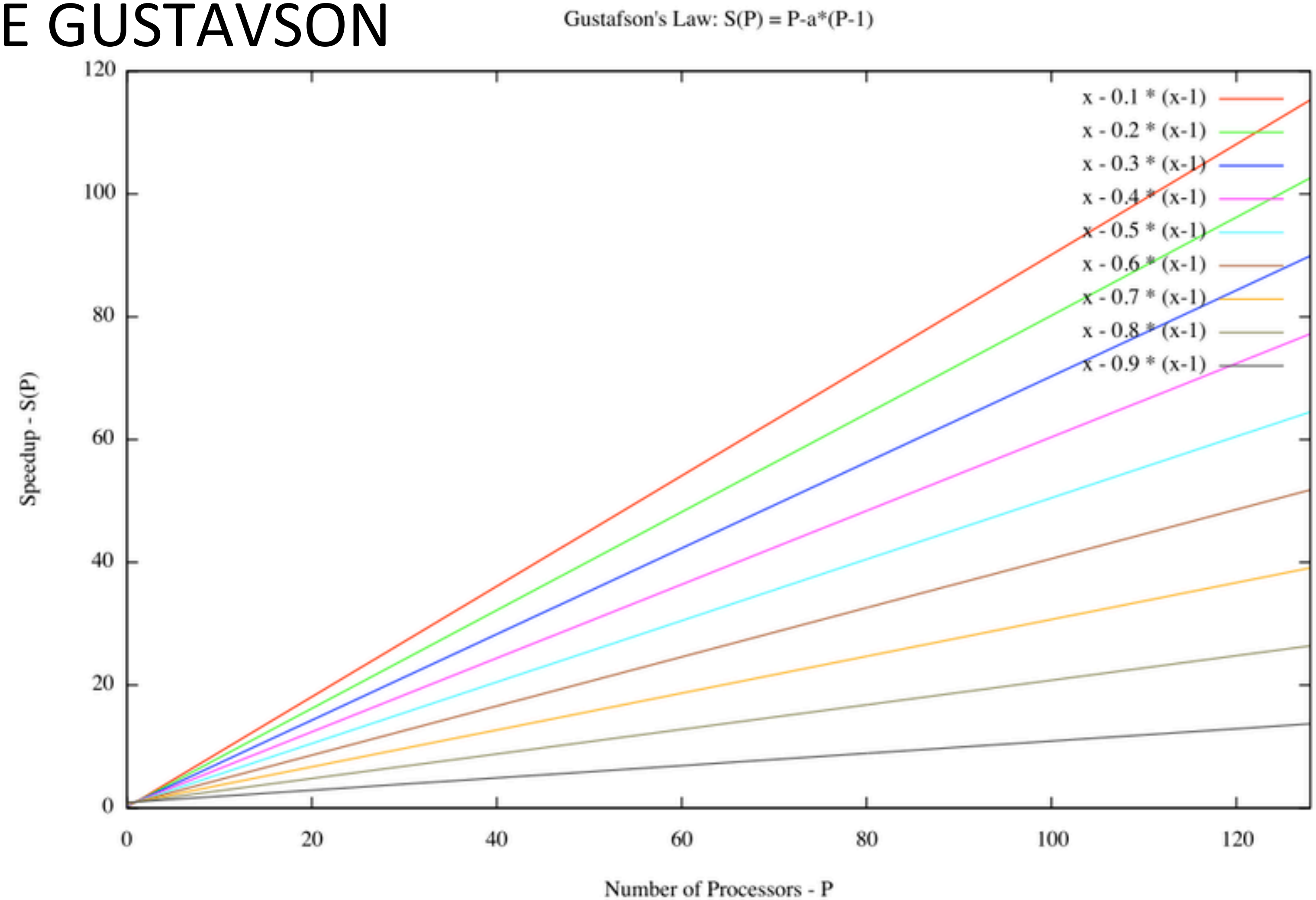


Imagen de Wikipedia, 2014b

LEY DE GUSTAVSON

- Suponemos que un programa consiste de α=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?

´

LEY DE GUSTAVSON

- Suponemos que un programa consiste de α=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

MODELO DE SUN Y NI

- Objetivo: resolver el mayor problema posible que permita la memoria disponible en el sistema.
- **Cantidad de memoria FIJA**
- El modelo de Sun y Ni es válido si la memoria es distribuida y compartida y se emplea toda la memoria en el problema escalado.
- M requisitos de memoria, W = g(M)
- Implica:
  - La carga de trabajo se escala a los recursos disponibles.
  - Se consigue mayor aceleración, mayor precisión y mejor uso de los recursos.
  - La hipótesis subyacente es que el factor que limita el mayor problema a resolver por una máquina es la memoria.

$$W_1 = W_1^{'} = W_1^* \qquad W_n^* = g^*(nM) = G(n)g(M) = G(n)W_n$$

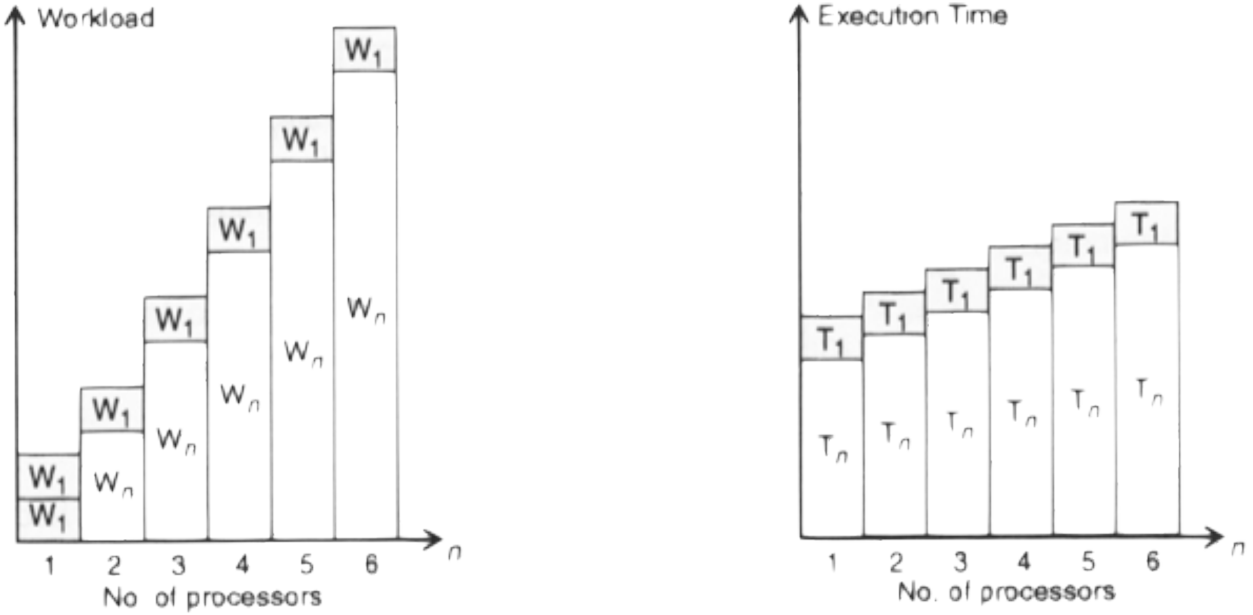
$$S_n^* = \frac{\sum_{i=1}^{m^*} W_i^*}{\sum_{i=1}^{m^*} \frac{W_i^*}{i}\left\lceil \frac{i}{n} \right\rceil + Q(n)} = \frac{W_1 + G(n)W_n}{W_1 + G(n)W_n / n}$$

MODELO DE SUN Y NI

$$S_n^* = \frac{\sum_{i=1}^{m^*} W_i^*}{\sum_{i=1}^{m^*} \frac{W_i^*}{i}\left\lceil \frac{i}{n} \right\rceil + Q(n)} = \frac{W_1 + G(n)W_n}{W_1 + G(n)W_n / n}$$

- Este modelo es general y engloba a las leyes de Amdahl y Gustavson ya que:
  - Si G(n)=1 la carga está fijada, estamos en el modelo de carga fija: Ley de Amdahl.
  - Si G(n)=n la carga aumenta proporcionalmente a la memoria, estamos en el modelo de tiempo fijo: Ley de Gustavson.
  - Si G(n)>n la carga crece más que la memoria.

MODELO DE SUN Y NI



(a) Scaled workload

(b) Slightly increased execution time

Figure 3.10 Scaled speedup model using fixed memory (Courtesy of Sun and Ni; reprinted with permission from *ACM Supercomputing*, 1990)

Imagen de Hwang, 1993

CONCLUSIONES

- El aumento del tamaño de máquina no acarrea de forma directa una mejora en las prestaciones proporcional a dicho tamaño.
- Hay que analizar cada problema.

REFERENCIAS

[Hwang, 1993] K. HWANG. Advanced Computer Architecture. Mc Graw-Hill, 1993.

[Wikipedia, 2014] Ley de Amdahl, http://es.wikipedia.org/wiki/Ley_de_Amdahl

[Wikipedia, 2014b] Ley de Gustafson, http://es.wikipedia.org/wiki/Ley_de_Gustafson