

Ingeniería de Servidores



Los Del DGIIM, losdeldgiim.github.io

Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0).

Eres libre de compartir y redistribuir el contenido de esta obra en cualquier medio o formato, siempre y cuando des el crédito adecuado a los autores originales y no persigas fines comerciales.

Ingeniería de Servidores

Los Del DGIIM, losdeldgiim.github.io

Arturo Olivares Martos

Granada, 2025

Índice general

1. Relaciones de Ejercicios	5
1.1. Introducción a la Ingeniería de Servidores	5
1.3. Monitorización	18

1. Relaciones de Ejercicios

1.1. Introducción a la Ingeniería de Servidores

Ejercicio 1.1.1. Un programa para la simulación de sistemas hidráulicos se ejecuta en 122 segundos. Si las operaciones de división con números reales consumen el 73 % de este tiempo, ¿en cuánto se tendría que mejorar la velocidad de estas operaciones si queremos conseguir que dicho programa se ejecute seis veces más rápidamente? ¿Cuál es la ganancia en velocidad máxima que podríamos conseguir si pudiésemos mejorar dichas operaciones tanto como quisiéramos?

Tenemos que, representando por k el número de veces que se mejora la velocidad de las operaciones de división con números reales, la ganancia en velocidad S es:

$$S = \frac{T_o}{T_m} = \frac{T_o}{(1 - 0,73)T_o + \frac{0,73T_o}{k}} = \frac{1}{0,27 + \frac{0,73}{k}} \leq \frac{1}{0,27} = 3,7$$

Por tanto, la ganancia máxima que vamos a poder conseguir mejorando tan solo las operaciones de división con números reales es de 3,7. Si quisiésemos que el programa se ejecutase seis veces más rápidamente, tendríamos que mejorar otros aspectos del sistema.

Ejercicio 1.1.2. Una mejora en un sitio web ha permitido rebajar de 17 a 9 segundos el tiempo medio de descarga de sus páginas. Si la mejora ha consistido en hacer 3 veces más rápido el subsistema de discos que almacena las páginas del servidor, ¿cuánto tiempo se dedicaba a acceder a los discos antes de realizar la mejora?

Sea T_o el tiempo medio de descarga de las páginas antes de la mejora, y T_m el tiempo medio de descarga de las páginas después de la mejora. Sea además f la fracción del tiempo de descarga de las páginas que se dedica a acceder a los discos. Entonces, se tiene:

$$T_m = (1 - f)T_o + \frac{fT_o}{3} \implies 3 \cdot \frac{T_o - T_m}{2T_o} = f = 3 \cdot \frac{17 - 9}{2 \cdot 17} = \frac{12}{17}$$

Por tanto, antes de la mejora, el tiempo medio de descarga de las páginas que se dedicaba a acceder a los discos era de:

$$fT_o = \frac{12}{17} \cdot 17 = 12 \text{ s}$$

Ejercicio 1.1.3. Un computador tarda 100 segundos en ejecutar un programa de simulación de una red de interconexión para multicomputadores. El programa dedica

el 20 % en hacer operaciones de aritmética entera (AE), el 30 % en hacer operaciones de aritmética en coma flotante (CF), mientras que el resto se emplea en operaciones de entrada/salida (E/S). Calcule la ganancia en velocidad y el tiempo de ejecución si las operaciones aritméticas enteras y reales se mejoran de manera simultánea 2 y 3 veces, respectivamente.

Sea T_o el tiempo de ejecución del programa antes de la mejora, y T_m el tiempo de ejecución del programa después de la mejora. Sea además f_{AE} y f_{CF} las fracciones del tiempo de ejecución del programa que se dedican a las operaciones de aritmética entera y aritmética en coma flotante, respectivamente. Entonces, se tiene:

$$\begin{aligned} T_m &= (1 - f_{AE} - f_{CF})T_o + \frac{f_{AE}T_o}{2} + \frac{f_{CF}T_o}{3} = \\ &= T_o \left[(1 - 0,2 - 0,3) + \frac{0,2}{2} + \frac{0,3}{3} \right] = 0,7T_o = \\ &= 70 \text{ s} \end{aligned}$$

Por tanto, la ganancia en velocidad es:

$$S = \frac{T_o}{T_m} = \frac{T_o}{0,7T_o} = \frac{1}{0,7} \approx 1,43$$

Ejercicio 1.1.4. Una aplicación informática se ejecuta en un computador durante un total de 70 segundos. Mediante el uso de un monitor de actividad se ha podido saber que el 85 % del tiempo se utiliza la tarjeta de red, mientras que el resto del tiempo se hace uso del procesador. Se pide, considerando el sistema original como punto de partida:

1. Calcular el incremento de prestaciones si se mejora en 8 veces la velocidad de la tarjeta de red.

Sea T_o el tiempo de ejecución del programa antes de la mejora, y T_m el tiempo de ejecución del programa después de la mejora. Sea además f la fracción del tiempo de ejecución del programa que se dedica a la tarjeta de red. Entonces, se tiene:

$$\begin{aligned} T_m &= (1 - f)T_o + \frac{fT_o}{8} = \\ &= T_o \left[(1 - 0,85) + \frac{0,85}{8} \right] = 0,25625 T_o = \\ &= 17,9375 \text{ s} \end{aligned}$$

Por tanto, la ganancia en velocidad es:

$$S = \frac{T_o}{T_m} = \frac{T_o}{0,25625T_o} = \frac{1}{0,25625} \approx 3,9$$

2. Determinar en cuánto hay que mejorar el rendimiento del procesador si se quiere ejecutar la aplicación en 25 segundos.

El tiempo que se emplea la tarjeta de red (sin hacer mejoras de esta) es de:

$$fT_o = 0,85 \cdot 70 = 59,5 \text{ s}$$

Por tanto, mejorando únicamente el rendimiento del procesador, no es posible ejecutar la aplicación en 25 segundos.

Ejercicio 1.1.5. De acuerdo con la ley de Amdahl, deduzca una expresión para la fracción de tiempo f en función de S (el speedup) y k (el número de veces mejorado).

La ley de Amdahl establece que la ganancia en velocidad S es:

$$\begin{aligned} S = \frac{T_o}{T_m} &= \frac{T_o}{(1-f)T_o + \frac{fT_o}{k}} = \frac{1}{1-f + \frac{f}{k}} \Rightarrow \frac{1}{S} - 1 = f \left(-1 + \frac{1}{k} \right) \Rightarrow \\ &\Rightarrow f = \frac{k(1-S)}{S(1-k)} = \frac{k(S-1)}{S(k-1)} \quad k \neq 1 \end{aligned}$$

Ejercicio 1.1.6. El administrador de un sistema informático pretende aumentar el rendimiento para evitar que el director del centro lo cese en sus funciones (ha habido más de quince quejas de usuarios en el último mes por el excesivo tiempo de ejecución de los programas). Indíquese, teniendo en cuenta la relación entre prestaciones y coste, qué opción de actualización de un sistema informático, de las dos que se enumeran, resultará más ventajosa:

1. Cambio del procesador (250 €). Esta modificación permite que el 75 % de los programas se ejecuten dos veces más rápidamente.

Sea T_o el tiempo total de ejecución de los programas antes de la actualización, y T_A el tiempo total de ejecución de los programas después de la actualización. Sea además f la fracción de los programas cuya velocidad se vería aumentada (75 % en este caso). Entonces, se tiene:

$$\begin{aligned} T_A &= (1-f)T_o + \frac{fT_o}{2} = \\ &= T_o \left[(1-0,75) + \frac{0,75}{2} \right] = 0,625 T_o \end{aligned}$$

2. Ampliación de la memoria principal (150 €). La capacidad extra de memoria mejora tres veces el tiempo de ejecución del 40 % de los programas.

Sea T_o el tiempo total de ejecución de los programas antes de la actualización, y T_B el tiempo total de ejecución de los programas después de la actualización. Sea además f la fracción de los programas cuya velocidad se vería aumentada (40 % en este caso). Entonces, se tiene:

$$\begin{aligned} T_B &= (1-f)T_o + \frac{fT_o}{3} = \\ &= T_o \left[(1-0,4) + \frac{0,4}{3} \right] = \frac{11}{15} \cdot T_o \end{aligned}$$

Aunque vemos que la primera alternativa es, de media, un 17,33 % más rápida que la segunda, hemos de comparar en función de la relación prestaciones/coste.

$$\frac{\text{Prestaciones}/\text{Coste}_A}{\text{Prestaciones}/\text{Coste}_B} = \frac{11/15 \cdot T_o \cdot 150\text{€}}{0,625 \cdot T_o \cdot 250\text{€}} = \frac{88}{125} = 0,704$$

Por tanto, la primera alternativa (cambio del procesador) es 0,704 veces más ventajosa que la segunda. Equivalentemente, y para entendernos, la segunda alternativa es 1,42 veces (un 42 %) más ventajosa que la primera. Por tanto, el administrador debería optar por la segunda alternativa, ampliar la memoria principal.

Ejercicio 1.1.7. Un programa de predicción meteorológica tarda 84 minutos en ejecutarse en un supercomputador diseñado al efecto. Sin embargo, esta cantidad de tiempo origina muchos problemas para los estudios de los meteorólogos. El responsable del equipo informático quiere reducir este tiempo sustituyendo la memoria principal por una más rápida, para lo cual existen dos modelos alternativos:

1. Modelo Lupita (1100 €), que disminuye el tiempo de ejecución hasta los 71 minutos.
2. Modelo Lucho (1300 €), que rebaja este tiempo de ejecución hasta los 63 minutos.

Determine cuál de los dos modelos anteriores representa la mejor opción ateniéndonos a la relación prestaciones/coste. Expresé el resultado como “% de mejora en la relación prestaciones/coste”.

La comparación de la relación prestaciones/coste de los dos modelos es:

$$\frac{\text{Prestaciones}/\text{Coste}_{\text{Lupita}}}{\text{Prestaciones}/\text{Coste}_{\text{Lucho}}} = \frac{63 \cdot 1300}{71 \cdot 1100} = \frac{819}{781} \approx 1,049$$

Por tanto, el modelo Lupita es 1,049 veces (un 4,9 %) más ventajoso que el modelo Lucho.

Ejercicio 1.1.8. El tiempo medio de respuesta de un sitio web es de 15 segundos. Mediante un monitor software se ha podido determinar que el 55 % de este tiempo es utilizado por el subsistema de discos, mientras que el resto se dedica a la ejecución de los scripts en el procesador de 2 GHz de que dispone el servidor. El administrador del sitio, después de soportar estoicamente las quejas de los usuarios, pretende reducir este tiempo por debajo de los 11 segundos. ¿Cuál de las dos opciones planteadas a continuación consigue este objetivo?

1. Adquirir un nuevo procesador que trabaja a 3 GHz.

Sea T_o el tiempo de respuesta del sitio web antes de la actualización, y T_A el tiempo de respuesta del sitio web después de la actualización. Sea además f la fracción del tiempo de respuesta del sitio web que se dedica a la ejecución de los scripts en el procesador. Entonces, se tiene:

$$\begin{aligned} T_A &= (1 - f)T_o + fT_o \cdot \frac{2}{3} = \\ &= T_o \left[(1 - 0,45) + 0,45 \cdot \frac{2}{3} \right] = 0,85 T_o = \\ &= 12,75 \text{ s} \end{aligned}$$

2. Substituir el subsistema de discos por uno de segunda mano 2,5 veces más rápido que el actual.

Sea T_o el tiempo de respuesta del sitio web antes de la actualización, y T_B el tiempo de respuesta del sitio web después de la actualización. Sea además f la fracción del tiempo de respuesta del sitio web que se dedica al subsistema de discos. Entonces, se tiene:

$$\begin{aligned} T_B &= (1 - f)T_o + fT_o \cdot \frac{1}{2,5} = \\ &= T_o \left[(1 - 0,55) + 0,55 \cdot \frac{1}{2,5} \right] = 0,67 T_o = \\ &= 10,05 \text{ s} \end{aligned}$$

Por tanto, la segunda opción es la que consigue el objetivo de reducir el tiempo de respuesta por debajo de los 11 segundos.

Ejercicio 1.1.9. Un programa de simulación de sistemas aerodinámicos de control se ejecuta en 280 segundos. El 70 % del tiempo de ejecución se utiliza el procesador; el resto se dedica a acceder al subsistema de discos. Un incremento del presupuesto aportado por el ministerio ha permitido adquirir un nuevo procesador tres veces más rápido.

1. Determine el tiempo de ejecución del simulador después de actualizar el procesador.

Sea T_o el tiempo de ejecución del programa antes de la actualización, y T_m el tiempo de ejecución del programa después de la actualización. Sea además f la fracción del tiempo de ejecución del programa que se dedica al procesador. Entonces, se tiene:

$$\begin{aligned} T_m &= (1 - f)T_o + fT_o \cdot \frac{1}{3} = \\ &= T_o \left[(1 - 0,7) + 0,7 \cdot \frac{1}{3} \right] = \frac{8}{15} T_o = \\ &= 149,33 \text{ s} \end{aligned}$$

2. Calcule ahora, esto es, después de haber hecho la actualización del procesador, cuál es la fracción del tiempo mejorado de ejecución durante el cual se utiliza el nuevo procesador. Haga un análisis del fenómeno observado.

La fracción del tiempo de ejecución mejorado de ejecución durante el cual se utiliza el nuevo procesador es:

$$f_m = 1 - \frac{(1 - 0,7)T_o}{T_m} = 1 - \frac{0,3T_o}{8/15T_o} = 0,4375$$

El tiempo dedicado a acceder al subsistema de discos se mantiene constante, pero el tiempo dedicado a la ejecución de los scripts en el procesador se reduce del 70 % al 43.75 %. Por tanto, anteriormente sí era vital mejorar dicho componente, pero ahora, con el nuevo procesador, es más importante mejorar el subsistema de discos.

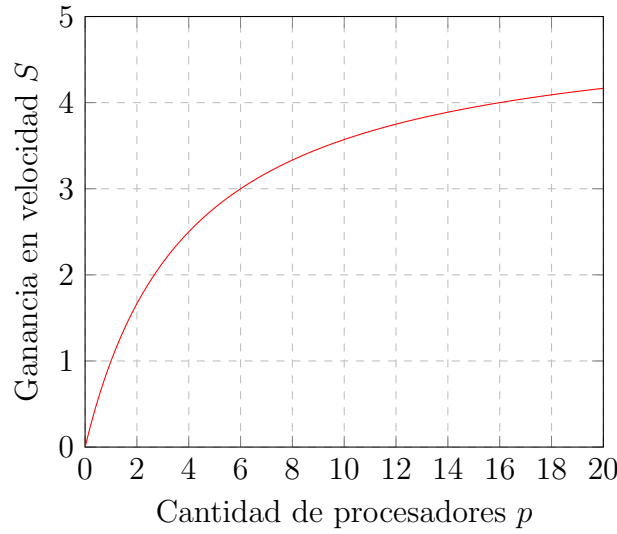


Figura 1.1: Ganancia en velocidad (S) en función del número de procesadores (p).

3. A raíz del resultado obtenido en el apartado anterior, si hubiéramos de mejorar este sistema actualizado, ¿sobre qué componente del mismo deberíamos incidir? Justifique numéricamente la respuesta.

Como hemos indicado, y debido a que siempre hemos de priorizar mejorar el componente que más tiempo de ejecución consume, deberíamos incidir sobre el subsistema de discos.

Ejercicio 1.1.10. Un equipo de biólogos que investiga sobre clonación de células utiliza el multiprocesador ALLIANT para ejecutar un simulador que se puede paralelizar en una fracción f de su tiempo de ejecución. La Figura 1.1 presenta la ganancia en velocidad conseguida por la máquina paralela en la ejecución del simulador para diferentes valores del número de procesadores (p).

1. ¿Cuál es la fracción paralelizable f del programa de simulación?

El tiempo mejorado, T_m , es:

$$T_m = (1 - f)T_o + \frac{fT_o}{p}$$

Por tanto, la ganancia en velocidad S en función del número de procesadores p es:

$$S = \frac{T_o}{T_m} = \frac{T_o}{(1 - f)T_o + \frac{fT_o}{p}} = \frac{1}{1 - f + \frac{f}{p}}$$

Despejando f de forma análoga al Ejercicio 1.1.5, se obtiene:

$$f = \frac{p(S - 1)}{S(p - 1)}$$

Empleando tanto los puntos (6, 3) como (16, 4), se obtiene en ambos casos:

$$f = \frac{4}{5} = 0,8$$

2. Si la parte secuencial (es decir, la no paralelizable) del simulador se ejecuta en 65s, ¿cuánto tiempo han de esperar los biólogos para obtener los resultados de la simulación con una configuración de 6 procesadores?

Como el 20 % del tiempo total de ejecución (en el caso de un único procesador) se dedica a la parte secuencial, el tiempo de ejecución del simulador en dicho único procesador es:

$$0,2T_o = 65 \text{ s} \implies T_o = 325 \text{ s}$$

Por tanto, el tiempo de ejecución del simulador en un sistema con 6 procesadores es:

$$T_m = (1 - f)T_o + \frac{fT_o}{6} = 0,2T_o + \frac{0,8T_o}{6} = T_o \left(0,2 + \frac{0,8}{6} \right) = \frac{T_o}{3} = 108,3 \text{ s}$$

3. Los científicos pretenden obtener resultados del simulador en un tiempo máximo de 70s sin modificar el código del programa. Si el sistema ALLIANT está preparado para ampliar el número de procesadores hasta $p = 30$, ¿podrán conseguir los biólogos su objetivo?

El tiempo de ejecución del simulador en un sistema con 30 procesadores sería:

$$T_m = (1 - f)T_o + \frac{fT_o}{30} = 0,2T_o + \frac{0,8T_o}{30} = T_o \left(0,2 + \frac{0,8}{30} \right) = \frac{17}{75}T_o = 73,667 \text{ s}$$

Por tanto, los biólogos no podrán conseguir su objetivo, puesto que el tiempo de ejecución del simulador en un sistema con 30 procesadores es mayor que su objetivo.

Veamos cuántos procesadores necesitarían para conseguir su objetivo:

$$T_m = (1 - f)T_o + \frac{fT_o}{p} = 0,2T_o + \frac{0,8T_o}{p} \leq 70 \iff 65 + \frac{260}{p} \leq 70 \iff p \geq \frac{260}{5} = 52$$

Por tanto, necesitarían al menos 52 procesadores para conseguir su objetivo.

4. Un informático afirma que el sistema ALLIANT podría conseguir el objetivo anterior con $p = 6$ procesadores si se reduce a la mitad la fracción secuencial (es decir, no paralelizable) del simulador. ¿Es válida esta propuesta?

Supongamos ahora que el tiempo secuencial es tan solo el 10 % del tiempo total de ejecución con un único procesador; es decir, $f = 0,9$. En este caso, el tiempo de ejecución del simulador en un sistema con 6 procesadores sería:

$$T_m = (1 - f)T_o + \frac{fT_o}{6} = 0,1T_o + \frac{0,9T_o}{6} = T_o \left(0,1 + \frac{0,9}{6} \right) = \frac{T_o}{4} = 81,25 \text{ s}$$

Por tanto, la propuesta no es válida, ya que el tiempo de ejecución del simulador en un sistema con 6 procesadores sería mayor que su objetivo.

Ejercicio 1.1.11. Ante la necesidad de reducir el tiempo de ejecución de un programa de cálculo de trayectorias espaciales, un equipo de arquitectos de computadores ha diseñado un nuevo procesador que mejora 3 veces la ejecución de las operaciones de coma flotante. El programa, cuando se ejecuta utilizando este nuevo procesador, emplea el 65 % del tiempo en la realización de operaciones de coma flotante.

1. Calcule qué tanto por ciento del tiempo de ejecución necesitaban las operaciones de coma flotante en el sistema con el procesador original.

Sea f la fracción del tiempo de ejecución del programa original que se dedica a las operaciones de coma flotante. Entonces, se tiene:

$$T_m = (1 - f)T_o + \frac{fT_o}{3}$$

$$0,65T_m = \frac{fT_o}{3}$$

Por tanto, tenemos que $fT_o = 1,95T_m$. Por tanto:

$$T_m = T_o - 1,95T_m + 0,65T_m \implies 2,3T_m = T_o$$

Por tanto, tenemos que:

$$0,65 \cdot \frac{T_o}{2,3} = \frac{fT_o}{3} \implies f = 3 \cdot \frac{0,65}{2,3} = \frac{39}{46} \approx 0,848$$

Por tanto, el 84,8 % del tiempo de ejecución necesitaban las operaciones de coma flotante en el sistema con el procesador original.

2. Indique cuál es la ganancia en velocidad global conseguida por el nuevo procesador.

Tenemos que:

$$S = \frac{T_o}{T_m} = \frac{2,3T_m}{T_m} = 2,3$$

Ejercicio 1.1.12. La gráfica de la Figura 1.2 muestra la ganancia en velocidad (speedup), calculada mediante la ley de Amdahl, que se consigue en un computador después de reemplazar la vieja unidad de disco por una nueva, en función de la fracción del tiempo de ejecución en el que se usaba la antigua unidad.

1. Indique cuántas veces es más rápida la nueva unidad de disco respecto de la que se ha retirado del computador.

Tenemos que:

$$S = \frac{1}{1 - f + \frac{f}{k}}$$

Empleando el punto (1, 4), se tiene:

$$4 = \frac{1}{1 - 1 + \frac{1}{k}} \implies k = 4$$

Por tanto, la nueva unidad de disco es 4 veces más rápida que la vieja.

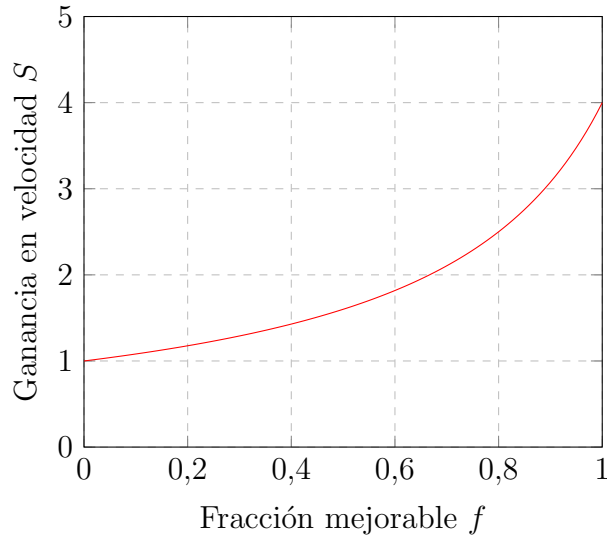


Figura 1.2: Ganancia en velocidad (S) en función de la fracción mejorable (f).

2. El computador, antes de hacer la actualización, tardaba 126 segundos en ejecutar la aplicación. Determine, en el mejor de los casos, cuál sería el tiempo de ejecución en el sistema actualizado. Justifique la respuesta.

Tenemos que:

$$T_m = (1 - f)T_o + \frac{fT_o}{4} = T_o \left(1 - f + \frac{f}{4}\right)$$

El mejor de los casos sería suponer que la fracción del tiempo de ejecución en el que se usaba la antigua unidad era la máxima posible, es decir, $f = 1$. Por tanto, el tiempo de ejecución en el sistema actualizado sería:

$$T_m = \frac{T_o}{4} = 31,5 \text{ s}$$

3. Dibuje sobre la misma gráfica la curva que se obtendría si la nueva unidad de disco fuera 2 veces más rápida que la vieja.

La gráfica obtenida se encuentra en la Figura 1.3, y vendría dada por la ecuación:

$$S = \frac{1}{1 - f + \frac{f}{2}}$$

Ejercicio 1.1.13. Una aplicación informática se ejecuta en un computador durante un total de 70s. Mediante el uso de un monitor de actividad se ha podido saber que durante el 85 % del tiempo de ejecución se utiliza la CPU (CPUo), mientras que el resto del tiempo se hace uso del disco duro (DD). Determine cuántas veces debe ser, como mínimo, más rápido un procesador (CPUm) que cueste el doble que el procesador actual para que hubiese valido la pena comprarlo en lugar de éste ateniéndonos a la relación prestaciones del sistema/coste del procesador.

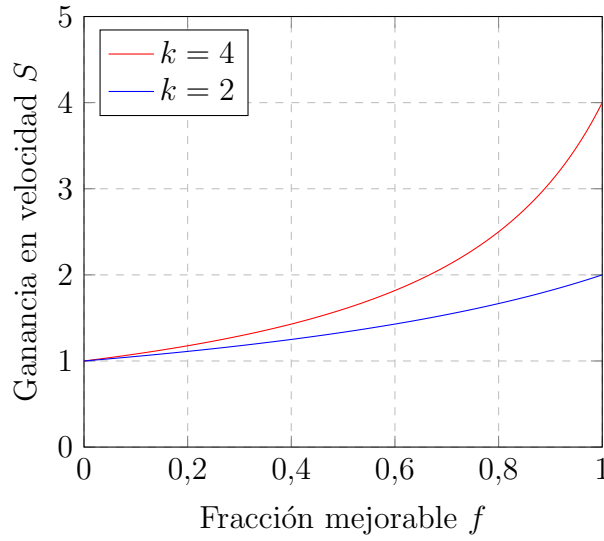


Figura 1.3: Ganancia en velocidad (S) en función de la fracción mejorable (f).

Sea k el número de veces más rápido que el procesador actual ha de ser el nuevo procesador. Entonces, se tiene:

$$\begin{aligned} T_m &= (1 - f)T_o + \frac{fT_o}{k} = \\ &= T_o \left[(1 - 0,85) + 0,85 \cdot \frac{1}{k} \right] = T_o \left[0,15 + \frac{0,85}{k} \right] \end{aligned}$$

Tenemos por tanto que:

$$\begin{aligned} \frac{\text{Prestaciones}/\text{Coste}_{\text{CPUm}}}{\text{Prestaciones}/\text{Coste}_{\text{CPUo}}} &= \frac{T_o}{T_m} \cdot \frac{\text{Coste}_{\text{CPUo}}}{\text{Coste}_{\text{CPUm}}} = \frac{1}{0,15 + \frac{0,85}{k}} \cdot \frac{1}{2} \geq 1 \iff \\ &\iff 1 - 0,3 \geq \frac{1,7}{k} \iff k \geq \frac{1,7}{0,7} \approx 2,4286 \end{aligned}$$

Por tanto, el nuevo procesador ha de ser al menos 2.43 veces más rápido que el procesador actual para que hubiese valido la pena comprarlo en lugar de éste.

Ejercicio 1.1.14. Se sabe que el tiempo de respuesta de una petición a un servidor de bases de datos es de 23 segundos, y que el 72 % de ese tiempo se emplea en acceder al subsistema de discos, cuyo coste es de 3500 €. Con el objetivo de mejorar las prestaciones del servidor, un ingeniero en informática está estudiando la posibilidad de adquirir, en su lugar, otro subsistema de discos tres veces más rápido pero con un coste de 4800 €.

1. Calcúlese el nuevo tiempo de respuesta del servidor con el subsistema de discos más caro.

El tiempo de respuesta del servidor con el subsistema de discos más caro sería:

$$\begin{aligned} T_m &= (1 - f)T_o + \frac{fT_o}{3} = T_o \left(1 - f + \frac{f}{3} \right) = \\ &= \frac{13}{25} T_o = 11,96 \text{ s} \end{aligned}$$

2. ¿Merece la pena comprar el sub-sistema de discos más caro ateniéndonos exclusivamente a la relación prestaciones/coste?

Tenemos que:

$$\frac{\text{Prestaciones}/\text{Coste}_{\text{caro}}}{\text{Prestaciones}/\text{Coste}_{\text{barato}}} = \frac{T_o}{T_m} \cdot \frac{\text{Coste}_{\text{barato}}}{\text{Coste}_{\text{caro}}} = \frac{T_o}{^{13/25} T_o} \cdot \frac{3500}{4800} \approx 1,4022$$

Por tanto, sí merece la pena comprar el sub-sistema de discos más caro.

3. ¿Cuál es la mejora máxima teórica que se podría alcanzar en el tiempo de respuesta manteniendo el subsistema de discos más barato y mejorando el resto de componentes? Expresa el resultado en “número de segundos más rápido” y en “número de veces más rápido”.

Suponiendo ahora que $f = 1 - 0,72 = 0,28$ y que k es la mejora, se tiene:

$$T_m = (1 - f)T_o + \frac{fT_o}{k} = 0,72T_o + \frac{0,28T_o}{k} \geq 0,72T_o = 16,56 \text{ s}$$

Por tanto, tenemos que:

$$T_o - T_m \leq 23 - 16,56 = 6,44 \text{ s}$$

Es decir, la mejora máxima teórica haría que el tiempo de respuesta fuese 6,44 s más rápido que el original. Por otro lado, se tiene:

$$\frac{T_o}{T_m} \leq \frac{T_o}{0,72T_o} = \frac{25}{18} \approx 1,39$$

Por tanto, la mejora máxima teórica haría que el tiempo de respuesta fuese 1,39 veces más rápido que el original.

Ejercicio 1.1.15. Un computador tarda 1000 segundos en ejecutar un proceso de formateo y conversión de imágenes. De todo ese tiempo, el programa dedica un 30 % en hacer operaciones de aritmética en coma flotante, y 250s en accesos al subsistema de discos.

1. Calcule la ganancia en velocidad que se consigue si añadimos al equipo una GPU de 500€ capaz de ejecutar las operaciones en coma flotante 10 veces más rápido.

Tenemos que:

$$\begin{aligned} T_m &= (1 - f)T_o + \frac{fT_o}{10} = T_o \left(1 - f + \frac{f}{10} \right) = \\ &= 0,73 T_o \end{aligned}$$

Por tanto, la ganancia en velocidad que se consigue es de:

$$S = \frac{T_o}{T_m} = \frac{T_o}{0,73 T_o} = \frac{100}{73} \approx 1,37$$

2. Calcule la ganancia en velocidad que se consigue con respecto al tiempo original si remplazamos el subsistema de discos por otro cuyo precio es de 400€ y consigue que los accesos al mismo sean 5 veces más rápidos.

La fracción del tiempo original que se dedica al subsistema de discos es:

$$f = \frac{250}{1000} = 0,25$$

Por tanto, el tiempo de ejecución del proceso con el nuevo subsistema de discos sería:

$$\begin{aligned} T_m &= (1 - f)T_o + \frac{fT_o}{5} = T_o \left(1 - f + \frac{f}{5} \right) = \\ &= 0,8 T_o \end{aligned}$$

Por tanto, la ganancia en velocidad que se consigue es de:

$$S = \frac{T_o}{T_m} = \frac{T_o}{0,8 T_o} = \frac{5}{4} = 1,25$$

3. Calcule la ganancia en velocidad que se consigue utilizando simultáneamente las dos mejoras de los apartados anteriores.

El tiempo de ejecución del proceso con las dos mejoras sería:

$$\begin{aligned} T_m &= (1 - f_{\text{CPU}} - f_{\text{DD}})T_o + \frac{f_{\text{CPU}}T_o}{10} + \frac{f_{\text{DD}}T_o}{5} = \\ &= T_o \left(1 - f_{\text{CPU}} - f_{\text{DD}} + \frac{f_{\text{CPU}}}{10} + \frac{f_{\text{DD}}}{5} \right) = \\ &= 0,53 T_o \end{aligned}$$

Por tanto, la ganancia en velocidad que se consigue es de:

$$S = \frac{T_o}{T_m} = \frac{T_o}{0,53 T_o} = \frac{100}{53} \approx 1,89$$

4. ¿Qué inversión es la más rentable ateniéndonos únicamente a la relación prestaciones/coste: comprar la GPU, el nuevo sub-sistema de discos o ambos a la vez?

Calculamos la relación prestaciones/coste de cada una de las mejoras:

- Para la mejora tan solo de la CPU:

$$\frac{\text{Prestaciones}/\text{Coste}_{\text{CPUo}}}{\text{Prestaciones}/\text{Coste}_{\text{CPUm}}} = \frac{T_o}{T_m} \cdot \frac{\text{Coste}_{\text{CPUo}}}{\text{Coste}_{\text{CPUm}}} = \frac{T_o}{0,73 T_o} \cdot \frac{1}{500} = \frac{1}{365}$$

- Para la mejora tan solo del disco:

$$\frac{\text{Prestaciones}/\text{Coste}_{\text{DDo}}}{\text{Prestaciones}/\text{Coste}_{\text{DDm}}} = \frac{T_o}{T_m} \cdot \frac{\text{Coste}_{\text{DDo}}}{\text{Coste}_{\text{DDm}}} = \frac{T_o}{0,8 T_o} \cdot \frac{1}{400} = \frac{1}{320}$$

- Para la mejora de ambos:

$$\frac{\text{Prestaciones}/\text{Coste}_{\text{CPUo}+\text{DDo}}}{\text{Prestaciones}/\text{Coste}_{\text{CPUm}+\text{DDm}}} = \frac{T_o}{T_m} \cdot \frac{\text{Coste}_{\text{CPUo}} + \text{Coste}_{\text{DDo}}}{\text{Coste}_{\text{CPUm}} + \text{Coste}_{\text{DDm}}} = \frac{T_o}{0,53 T_o} \cdot \frac{1}{900} = \frac{1}{477}$$

Por tanto, la mejora más rentable sería la del disco, con una ganancia en prestaciones por euro invertido de $1/320 = 3,125 \cdot 10^{-3}$.

Ejercicio 1.1.16. Después de reemplazar el antiguo disco duro del servidor de base de datos de una pequeña compañía granadina por una nueva unidad SSD, se ha constatado experimentalmente que el proceso principal se ejecuta 1,5 veces más rápido que antes. También se ha medido que ahora dicho proceso consume el 50 % de su tiempo accediendo a esa nueva unidad SSD.

1. Calcule la fracción de tiempo que el proceso consumía antes accediendo al antiguo disco duro.

Tenemos que:

$$\begin{aligned} 1,5 &= S = \frac{T_o}{T_m} \\ T_m &= (1 - f)T_o + fT_o \cdot \frac{1}{k} \\ 0,5T_m &= fT_o \cdot \frac{1}{k} \end{aligned}$$

Por tanto:

$$T_m = (1 - f)T_o + 0,5T_m \implies T_m = \frac{(1 - f)T_o}{0,5}$$

Sustituyendo en la expresión de la ganancia en velocidad:

$$1,5 = S = \frac{T_o}{\frac{(1 - f)T_o}{0,5}} \implies 1,5 = \frac{0,5}{1 - f} \implies f = 1 - \frac{0,5}{1,5} = \frac{2}{3} \approx 0,6667$$

Por tanto, la fracción de tiempo que el proceso consumía antes accediendo al antiguo disco duro era del 66,67 %.

2. ¿Cuántas veces es más rápida la nueva unidad SSD que el antiguo disco duro?

Tenemos que:

$$1,5 = S = \frac{1}{1 - f + \frac{f}{k}} \implies \frac{1}{1,5} - 1 + f = \frac{f}{k} \implies k = \frac{f}{1/1,5 - 1 + f} = 2$$

Por tanto, la nueva unidad SSD es 2 veces más rápida que el antiguo disco duro.

1.3. Monitorización

Ejercicio 1.3.1. En un sistema Linux se ha ejecutado la orden `uptime` tres veces en momentos diferentes. El resultado, de forma resumida, se muestra en el siguiente listado:

```
... load average: 6.85, 7.37, 7.83
... load average: 8.50, 10.93, 8.61
... load average: 37.34, 9.47, 3.30
```

Indique si la carga crece, decrece, se mantiene estacionaria o bien no puede decir sobre ello.

No hay una tendencia clara en los valores de las medidas, por lo que no podemos concluir nada realmente. Sería necesario saber cuánto tiempo ha pasado entre cada ejecución para estudiar así en detalle la situación. Tan solo podríamos afirmar que, en el último instante, la carga ha crecido de forma notable respecto a las dos anteriores.

Ejercicio 1.3.2. En un sistema Linux se ha ejecutado la siguiente orden:

```
$ time quicksort
real 0m40.2s
user 0m17.1s
sys 0m3.2s
```

Indique si el sistema está soportando mucha o poca carga. Razone la respuesta.

En este caso, la carga del sistema es considerable, puesto que el tiempo de ejecución del programa `quicksort` es de 20,3s (suma de los tiempos `user` y `sys`) y el tiempo real que ha tardado en ejecutarse es de 40,2s. Esto da a entender dos posibilidades:

- El sistema está soportando una carga considerable, ya que el proceso ha estado esperando mucho tiempo para que el sistema le proporcionara los recursos necesarios para su ejecución.
- El proceso `quicksort` ha estado bloqueado por E/S (`I/O blocked`) durante una parte importante de su ejecución, lo que ha provocado que el tiempo real sea mucho mayor que el tiempo de CPU consumido.

Ejercicio 1.3.3. Se sabe que la sobrecarga (*overhead*) de CPU de un monitor software en un determinado servidor es del 4 %. Si el monitor se activa cada 2 segundos, ¿cuánto tiempo tarda el monitor en ejecutarse por cada activación?

Sea T el tiempo que tarda el monitor en ejecutarse por cada activación. Si la sobrecarga es del 4 %, entonces tenemos que:

$$4 = \frac{T}{2} \cdot 100 \implies T = \frac{4 \cdot 2}{100} = 0,08 \text{ s} = 80 \text{ ms}$$

Ejercicio 1.3.4. A continuación se muestra el resultado obtenido tras ejecutar la orden `top` en un sistema informático que emplea Linux como sistema operativo:

```

2:52pm up 17 days, 3:41, 1 user, load average: 0.15, 0.27, 0.32
54 processes: 51 sleeping, 3 running, 0 zombie, 0 stopped
%Cpu(s): 23.8% user, 14.0% system, 0.0% nice, 17.0% idle, 45.2% wa
Mem: 257124K av, 253052K used, 4072K free, 8960K shrd, 182972K buff
Swap: 261496K av, 21396K used, 240100K free, 26344K cached

```

PID	USER	PRI	NI	VIRT	RSS	SHARE	STAT	LC	%CPU	%MEM	TIME	COMMAND
807	joan	0	0	5708	5708	532	R N	0	23.0	2.2	6:16	p_exec
809	joan	0	0	5708	5708	532	R N	0	14.0	2.2	3:42	p_exec
185	tomi	0	0	824	824	632	R	0	0.5	0.3	0:00	top
201	xp	0	0	1272	1208	644	S	0	0.1	0.4	5:49	xp_stat
1	root	0	0	60	56	36	S	0	0.0	0.0	0:03	init
2	root	0	0	0	0	0	SW	0	0.0	0.0	0:13	kflushd
7	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	nfsiod
194	root	0	0	72	4	4	S	0	0.0	0.0	0:00	migetty
195	root	0	0	68	0	0	SW	0	0.0	0.0	0:00	migetty
179	root	0	0	532	312	236	S	0	0.0	0.1	0:00	sendmail

1. ¿Cuánta memoria física tiene la máquina?

Este es el parámetro **Mem: 257124K av**, que indica que la máquina tiene 257124 KiB \approx 251 MiB de memoria física.

2. ¿Qué porcentaje de la memoria física está marcada como usada según el monitor?

Este es el parámetro **Mem: 253052K used**, que indica que 253052 KiB \approx 247 MiB de memoria física está usada. Por lo tanto, el porcentaje de memoria física usada es:

$$\frac{253052}{257124} \cdot 100 \approx 98,41 \%$$

3. ¿Cuál es la utilización media del procesador?

Este parámetro se calcula como la totalidad (100 %) menos el tiempo que el procesador está inactivo (reflejado por los parámetros **idle** y **wa**):

$$100 \% - 17,0 \% - 45,2 \% = 37,8 \%$$

4. ¿Cómo es la evolución de la carga media del sistema, ascendente o descendente?

Esta es descendente, y se puede ver en el parámetro **load average: 0.15, 0.27, 0.32**, que indica que la carga media del sistema ha ido disminuyendo desde 0,32 a 0,15.

5. ¿Cuánta memoria física ocupa el monitor?

La memoria física ocupada por el monitor es la que se muestra en el parámetro **RSS** (*Resident Set Size*) de la línea del proceso **top**, que es de 824 KiB.

Ejercicio 1.3.5. Considere las órdenes siguientes ejecutadas en un sistema Linux:

```
$ time simulador_original
real 0m24.2s
user 0m15.1s
sys 0m1.6s
$ time simulador_mejorado
real 0m32.8s
user 0m10.7s
sys 0m2.1s
```

1. ¿Cuál es el tiempo de ejecución de ambos simuladores?

El tiempo de ejecución del simulador original es de 16,7 s (suma de los tiempos `user` y `sys`) y el del simulador mejorado es de 12,8 s. Notemos que el tiempo real representa el tiempo total que ha tardado en ejecutarse el programa, incluyendo el tiempo de espera por parte del sistema operativo para que le proporcione los recursos necesarios y el tiempo atendiendo interrupciones de otros procesos. Es por esto que el simulador mejorado tarda más tiempo en ejecutarse que el original, puesto que el sistema en ese momento tiene más carga de trabajo.

2. Calcule, si es el caso, la mejora en el tiempo de ejecución del simulador mejorado respecto del original.

La mejora en el tiempo de ejecución del simulador mejorado respecto del original es:

$$\frac{16,7}{12,8} \approx 1,30469$$

Por lo tanto, el simulador mejorado es aproximadamente 1,3 veces más rápido que el original.

Ejercicio 1.3.6. El monitor `sar` (*system activity reporter*) de un computador se activa cada 15 minutos y tarda 750 ms en ejecutarse por cada activación. Se pide:

1. Calcular la sobrecarga que genera este monitor sobre el sistema informático.

La sobrecarga que genera el monitor `sar` se calcula como:

$$\frac{750}{15 \cdot 60 \cdot 10^3} \cdot 100 = \frac{1}{12} \approx 0,08333 \%$$

2. Si la información generada en cada activación ocupa 8192 bytes, ¿la monitorización de cuántos días completos se pueden almacenar en el directorio `/var/log/sysstat` si se dispone únicamente de 200 MiB de capacidad libre?

Suponemos que la totalidad de capacidad disponible se puede emplear para guardar recursos (en realidad, también ha de guardarse información como el nombre de los ficheros, por ejemplo). Calculemos por tanto en primer lugar los registros de cuántas activaciones pueden almacenarse:

$$200 \text{ MiB} \cdot \frac{2^{20} \text{ bytes}}{1 \text{ MiB}} \cdot \frac{1 \text{ registro}}{8192 \text{ bytes}} = 25600 \text{ registros}$$

Calculemos ahora la monitorización de cuántos días completos se pueden almacenar:

$$25600 \text{ registros} \cdot \frac{15 \text{ minutos}}{1 \text{ registro}} \cdot \frac{1 \text{ hora}}{60 \text{ minutos}} \cdot \frac{1 \text{ día}}{24 \text{ horas}} = 266,67 \text{ días}$$

Por tanto, se pueden almacenar la información de 266 días completos.

Ejercicio 1.3.7. El día 8 de octubre se ha ejecutado la siguiente orden en un sistema Linux:

```
% ls /var/log/sysstat
-rw-r--r-- 1 root root 3049952 Oct  6 23:50 sa06
-rw-r--r-- 1 root root 3049952 Oct  7 23:50 sa07
-rw-r--r-- 1 root root 2372184 Oct  8 18:40 sa08
```

Suponiendo que la primera muestra se toma a las 0 : 00 de cada día y que **sadc** se ejecuta con un tiempo de muestreo constante, ¿cada cuánto tiempo se activa el monitor **sar**? ¿Cuál es la anchura de entrada del monitor?

Puesto que las últimas activaciones del día 6 y del día 7 se hacen a las 23 : 50 horas, podemos concluir que el monitor se activa cada 10 minutos. Por tanto, el número de veces que se activa el monitor en un día es:

$$\frac{24 \cdot 60}{10} = 144$$

La anchura de entrada del monitor se calcula como el tamaño del fichero dividido por el número de activaciones:

$$\frac{3049952 \text{ bytes}}{144 \text{ activaciones}} \approx 21180,22 \text{ B/entrada} \approx 20,68 \text{ KiB/entrada}$$

Por tanto, la anchura de entrada del monitor es de aproximadamente 20,68 KiB.

Ejercicio 1.3.8. Indique el resultado que produce la ejecución de las siguientes órdenes sobre un sistema Linux con el monitor **sar** instalado:

1. **sar**

Informa sobre la utilización del procesador durante el día actual (opción por defecto, -u).

2. **sar -A**

Informa sobre toda la información recogida durante el día actual.

3. **sar -u 1 30**

Informa sobre la utilización del procesador en el momento actual, mostrando 30 medidas tomadas con un período de un segundo.

4. **sar -uB -f /var/log/sysstat/08**

Informa sobre la utilización del procesador y paginación de la memoria virtual durante el día 8 del mes.

5. `sar -d -s 12:30:00 -e 18:15:00 -f /var/log/sysstat/08`

Informa sobre las transferencias de disco desde las 12 : 30 hasta las 18 : 15 horas del día 8 del mes.

Ejercicio 1.3.9. Después de instrumentar un programa con la herramienta `gprof` el resultado obtenido ha sido el siguiente:

```
Flat profile:
Each sample counts as 0.01 seconds.
%   cumulative   self           self         total
time  seconds    seconds   calls   s/call   s/call   name
59.36    27.72    27.72        3     9.24    14.39  reduce
33.08    43.17    15.45        6     2.57     2.57  invierte
 7.56    46.70     3.53        2     1.76     1.76  calcula
```

El grafo de dependencias muestra que `invierte()` es llamado desde el procedimiento `reduce()`.

1. ¿Cuál es el procedimiento cuyo código propio sería más conveniente optimizar?

Se debe optimizar la función con mayor tiempo propio (`self seconds`), que en este caso es `reduce()`, ya que consume casi el 60 % del tiempo de CPU del programa.

2. Si el código propio de `reduce()` se sustituye por otro tres veces más rápido, ¿cuánto tiempo tardará en ejecutarse el programa?

El tiempo que tardaría sería:

$$27,72 \cdot \frac{1}{3} + 15,45 + 3,53 = 28,22 \text{ s}$$

3. Si el procedimiento `invierte()` se sustituye por una nueva versión cuatro veces más rápida, ¿qué mejora se obtendrá en el tiempo de ejecución?

La mejora sería:

$$S = \frac{T_o}{T_m} = \frac{27,72 + 15,45 + 3,53}{27,72 + \frac{15,45}{4} + 3,53} \approx 1,33$$

4. Calcule cuál es la ganancia en velocidad máxima que se podría conseguir en el tiempo de ejecución mediante la optimización del código del procedimiento `invierte()`.

Sea k el factor de mejora del procedimiento `invierte()`, entonces la ganancia en velocidad máxima sería:

$$\lim_{k \rightarrow \infty} S = \lim_{k \rightarrow \infty} \frac{27,72 + 15,45 + 3,53}{27,72 + \frac{15,45}{k} + 3,53} \approx 1,49$$

Por tanto, la máxima ganancia en velocidad que se podría conseguir es de aproximadamente 1,49.

Ejercicio 1.3.10. Un informático desea evaluar el rendimiento de un computador por medio del benchmark SPEC CPU 2017. Una vez compilados todos los programas del paquete y lanzado su ejecución monitoriza el sistema con la orden `vmstat 1 5`. El resultado de las medidas de este monitor es el siguiente:

procs		-----memory-----				--swap--		----io----		---system---			----cpu----		
r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa
0	0	8	14916	92292	833828	0	0	0	3	0	7	3	1	96	0
1	0	8	14916	92292	833828	0	0	0	0	1022	40	100	0	0	0
3	0	8	14916	92292	833828	2	1	16	3	1016	34	99	1	0	0
1	0	8	14916	92292	833828	0	4	0	8	1035	36	98	2	0	0
2	0	8	14916	92292	833828	1	5	4	28	1035	36	99	1	0	0

Indique si, a la vista de los datos anteriores, los resultados obtenidos en la prueba de evaluación serán correctos o no. Justifique la respuesta.

No aparentan ser correctos, puesto que este benchmark tiene como objetivo medir el rendimiento de la CPU, y no se producen entonces intercambio con el disco. Por tanto, deberá haber otros programas en ejecución que los provoquen, alterando los resultados de la medición.

Ejercicio 1.3.11. La monitorización de un programa de dibujo en tres dimensiones mediante la herramienta `gprof` ha proporcionado la siguiente información (por errores en la transmisión hay valores que no están disponibles):

```
Flat profile:
%      cumulative   self           self       total
time   seconds    seconds   calls   s/call   s/call   name
xxxxx      xxxxx    15.47         3     5.16     5.16  colorear
xxxxx      xxxxx     1.89         5     0.38     0.38  interpola
xxxxx      xxxxx     1.76         1     1.76     3.65  traza
xxxxx      xxxxx     0.46
                                main

Call graph:
index % time   self  children   called    name
[1]   100.0    0.46   19.12      3/3      main [1]
              15.47    0.00      3/3      colorear [2]
              1.76    1.89      1/1      traza [3]

              15.47    0.00      3/3      main [1]
[2]   79.0    15.47    0.00        3      colorear [2]

              1.76    1.89      1/1      main [1]
[3]   18.6     1.76    1.89        1      traza [3]
              1.89    0.00      5/5      interpola [4]

              1.89    0.00      5/5      traza [3]
[4]    9.7     1.89    0.00        5      interpola [4]
```

1. ¿En cuánto tiempo se ejecuta el programa de dibujo?

Hay varias formas de verlo, y evidentemente todas ellas coinciden.

- En el `flat profile`, podemos obtener el tiempo total de ejecución como la suma de los `self seconds`:

$$15,47 + 1,89 + 1,76 + 0,46 = 19,58 \text{ s}$$

- Desde el `call graph`, como el método `main` se ejecuta el 100% del tiempo, podemos calcular este tiempo como la suma de su tiempo propio de ejecución más el de sus hijos:

$$0,46 + 19,12 = 19,58 \text{ s}$$

2. Indique cuánto tiempo tarda en ejecutarse el código propio de `main()`.

El código propio de `main()` se ejecuta en 0,46 s, que es el valor del campo `self seconds` de la línea correspondiente al procedimiento `main()` en el `flat profile`.

3. Establezca la relación de llamadas entre los procedimientos del programa así como el número de veces que se ejecuta cada uno de ellos.

El procedimiento `main()` llama 3 veces al procedimiento `colorea()` y una vez al procedimiento `traza()`; a su vez, el procedimiento `traza()` llama 5 veces al procedimiento `interpola()`.

4. Calcule el nuevo tiempo de ejecución del programa si se elimina el código propio de `main()` y se reduce a la mitad el tiempo de ejecución del código propio del procedimiento `traza()`.

El nuevo tiempo de ejecución del programa sería:

$$15,47 + 1,89 + \frac{1,76}{2} + 0 = 18,24 \text{ s}$$

5. Proponga y justifique numéricamente una acción sobre el programa original que no afecte el procedimiento `colorea()` (ni su código ni el número de veces que es ejecutado) con el fin de conseguir que el programa se ejecute en 10 segundos.

Esto no es posible, puesto que el procedimiento `colorea()` ya consume más de 10 segundos de tiempo de ejecución propio, y no se puede reducir su tiempo de ejecución sin afectar a su código o al número de veces que se ejecuta. Por tanto, cualquier acción que se proponga para reducir el tiempo de ejecución del programa afectará necesariamente al procedimiento `colorea()`.