

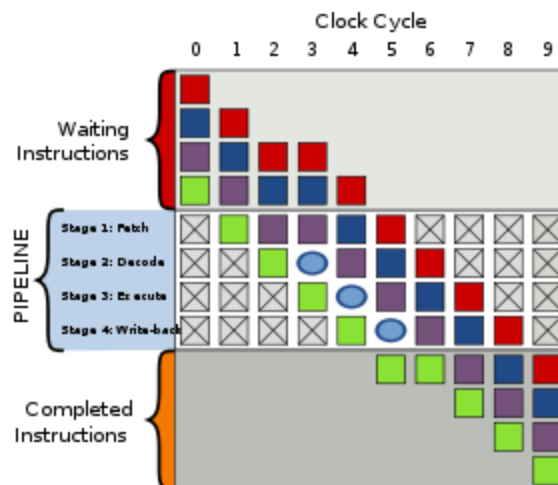
## Procesamiento Pipeline

Se le llama pipeline a una serie de elementos de procesamiento de datos ordenados de tal modo que la salida de cada uno es la entrada del siguiente, como quien dice una cadena de montaje pero en vez de orientada a la manufactura, orientada al procesamiento de datos e instrucciones.

Consiste en ir transformando un flujo de datos en un proceso comprendido por varias fases secuenciales, siendo la entrada de cada una la salida de la anterior.

Esta arquitectura es muy común en el desarrollo de programas para el intérprete de comandos, ya que se pueden concatenar comandos fácilmente con tuberías (pipe).

También es una arquitectura muy natural en el paradigma de programación funcional, ya que equivale a la composición de funciones matemáticas.



## Ley de Amdahl

La Ley de Amdahl, llamada así por el arquitecto de ordenadores Gene Amdahl, se usa para averiguar la mejora máxima de un sistema cuando solo una parte de éste es mejorado. 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.

La fórmula original de la ley de Amdahl es la siguiente:

$$F = F_a \cdot \left( (1 - F_m) + \frac{F_m}{A_m} \right)$$

siendo:

$F_m$  = tiempo de ejecución mejorado y

$F_a$  = tiempo de ejecución antiguo.

Esta fórmula se puede reescribir usando la definición del incremento de la velocidad que viene dado por  $A = F_a/F$ , por lo que la fórmula anterior se puede reescribir como:

$$A = \frac{1}{(1 - F_m) + \frac{F_m}{A_m}}$$

siendo:

$A$  es la aceleración o ganancia en velocidad conseguida en el sistema completo debido a la mejora de uno de sus subsistemas.

$A_m$ , es el factor de mejora que se ha introducido en el subsistema mejorado.

$F_m$ , es la fracción de tiempo que el sistema utiliza el subsistema mejorado.

Ejemplo: Si en un programa de ordenador el tiempo de ejecución de un cierto algoritmo supone un 30% del tiempo de ejecución total del programa, y conseguimos hacer que este algoritmos se ejecute en la mitad de tiempo se tendrá:

$$A_m = 2$$

$$F_m = 0.3$$

$$A \approx 1.18$$

Es decir, se ha mejorado la velocidad de ejecución del programa en un factor de 1,18. La ley de Amdahl se mide en unidades genéricas, es decir los resultados no son porcentajes, ni unidades de tiempo.

La ley de Amdahl se puede interpretar de manera más técnica, pero en términos simples, significa que es el algoritmo el que decide la mejora de velocidad, no el número de procesadores. Finalmente se llega a un momento que no se puede paralelizar más el algoritmo.

## Ley de Gustafson

Establece que cualquier problema suficientemente grande puede ser eficientemente paralelizado. La ley de Gustafson está muy ligada a la Ley de Amdahl, que pone límite a la aceleración que se puede obtener gracias a la paralelización, dado un conjunto de datos de tamaño fijo, ofreciendo así una visión pesimista del procesamiento paralelo. Por el contrario la ley de Gustafson ofrece un nuevo punto de vista y así una visión positiva de las ventajas del procesamiento paralelo. John L. Gustafson enunció por primera vez la ley que lleva su nombre en 1988.

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

donde  $P$  es el número de procesadores,  $S$  es el speedup, y  $\alpha$  la parte no paralelizable del proceso.

La ley de Gustafson aborda las limitaciones de la Ley de Amdahl, la cual no escala la disponibilidad del poder de cómputo a medida que el número de máquinas aumenta. La ley de Gustafson propone que los programadores establezcan el tamaño de los problemas para utilizar el equipamiento disponible en su solución en un tiempo práctico. Por consiguiente, si existe equipamiento más rápido disponible, mayores problemas se pondrán resolver en el mismo tiempo.

La Ley de Amdahl se basa en una carga de trabajo o tamaño de entrada prefijados. Esto implica que la parte secuencial de un programa no cambia con respecto al número de procesadores de la máquina, sin embargo, la parte paralelizable es uniformemente distribuida en el número de procesadores. El impacto de la ley de Gustafson fue el cambio de dirección de los objetivos de investigación hacia la selección o reformulación de problemas a fin de que fuera posible la solución de mayores problemas en el mismo intervalo de tiempo. En particular la ley redefine la eficiencia como una necesidad para minimizar la parte secuencial de un programa, incluso si esto incrementa la cantidad total de cálculos

## Referencias

Type Architectures, Shared Memory, and The Corrolary of Modest Potential, Lawrence Snyder, Ann. Rev. Comput. Sci. 1986. 1:289-317.

Gene Amdahl, "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities", AFIPS Conference Proceedings, (30), pp. 483-485, 1967.