

Conceptos de Paralelismo

Cómputo de Alto Desempeño

Paralelismo

- Los sistemas de multitareas trabajan en arquitecturas de dos o más unidades de procesamiento (arquitecturas paralelas).
- En las unidades de procesamiento se ejecutan de forma simultánea las tareas concurrentes, que de acuerdo a su tamaño se clasifican en:

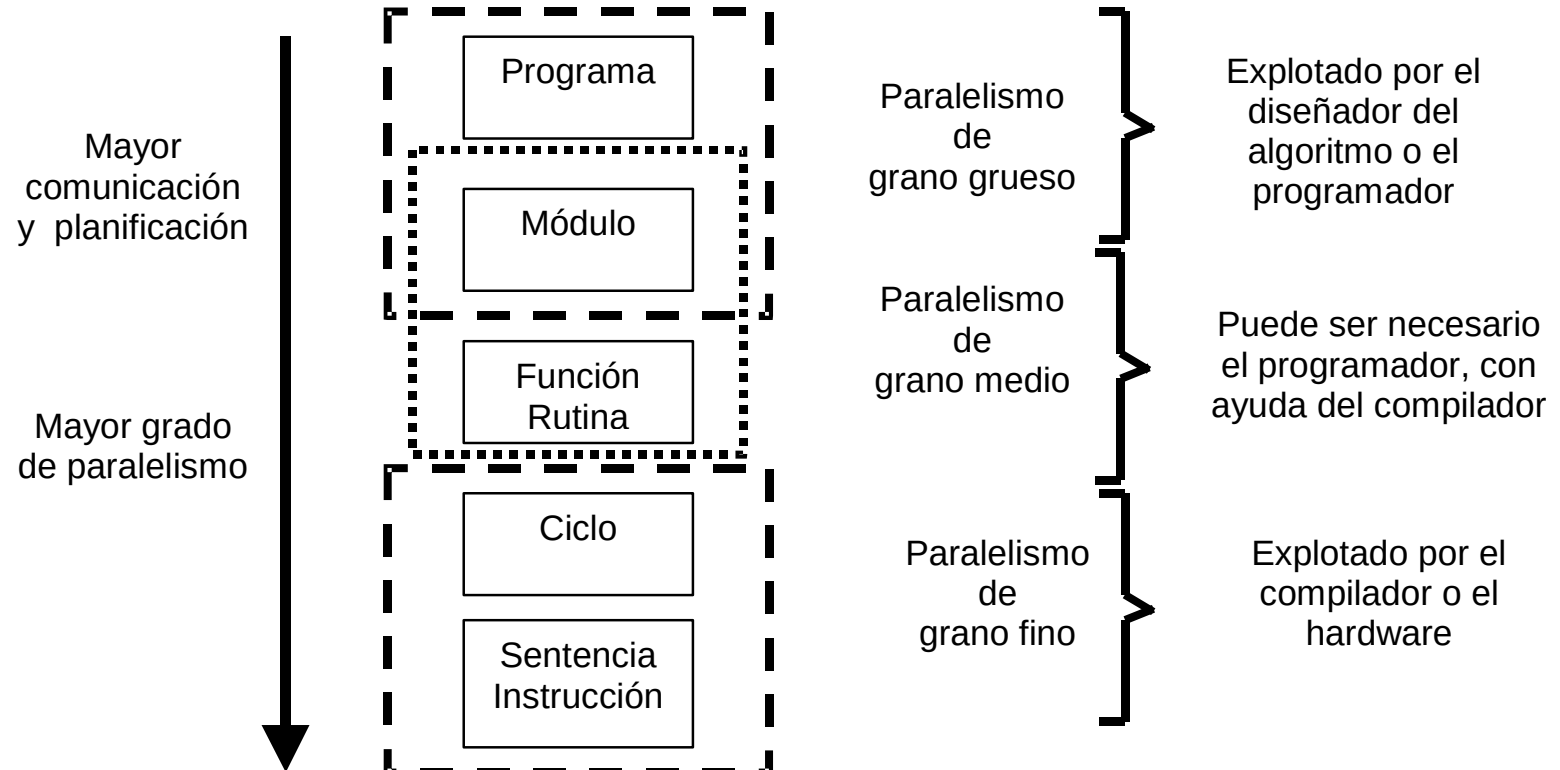
Paralelismo

- Existen dos visiones del paralelismo:
 - Paralelismo Hardware: definido por la arquitectura de la máquina
 - Paralelismo Software: Definido por la estructura del programa. Se manifiesta en las instrucciones que no tienen dependencia

Niveles de paralelismo

- **Paralelismo a nivel de tarea o trabajo**
- **Paralelismo a nivel de función o procedimiento**
- **Paralelismo a nivel de instrucción**
- **Paralelismo a nivel de bit**

Granularidad



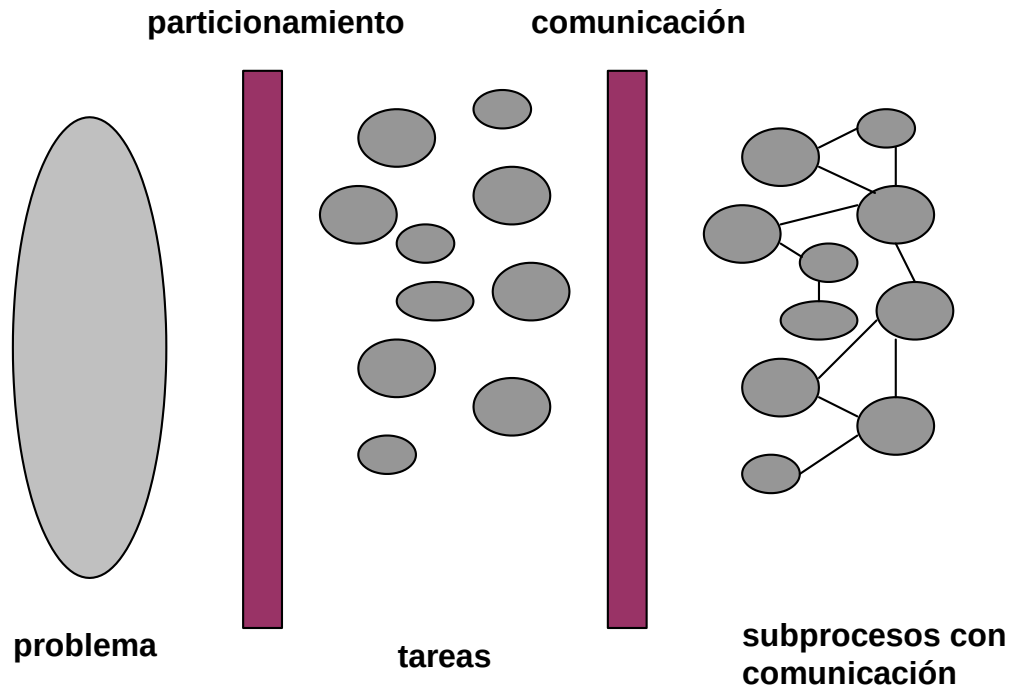
Diseño de un programa paralelo (Ian Foster)

Aplicar las etapas en el diseño de programas paralelos :

- Particionamiento
- Comunicación
- Aglomeración
- Mapeo

- Particionamiento
- Aglomeración
- Comunicación
- Mapeo

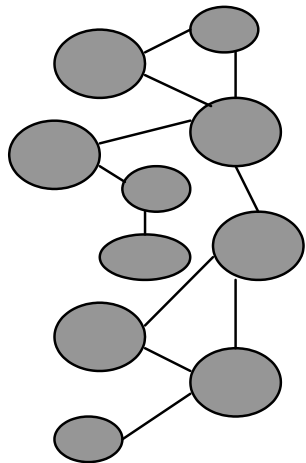
Diseño de un programa paralelo (Ian Foster)



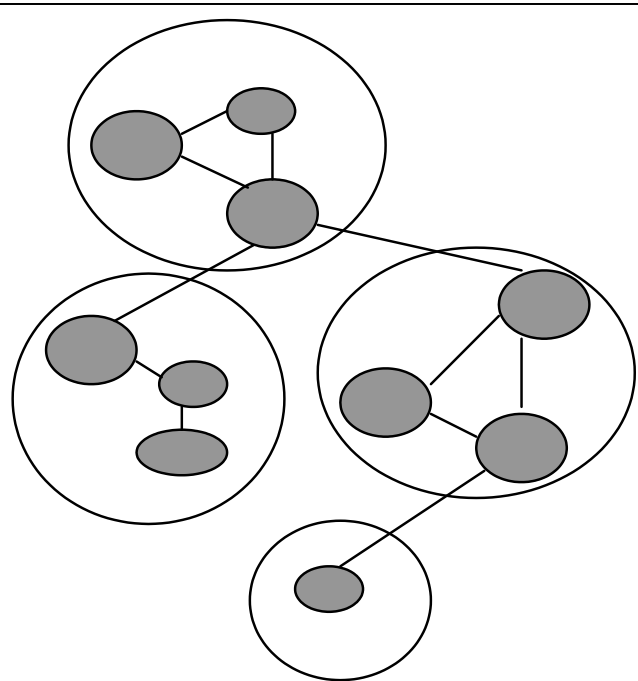
Diseño de un programa paralelo (Ian Foster)

aglomeración

mapeo

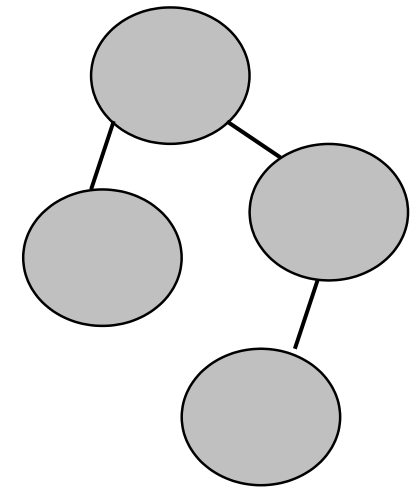


subprocesos con
comunicación



programas
paralelos

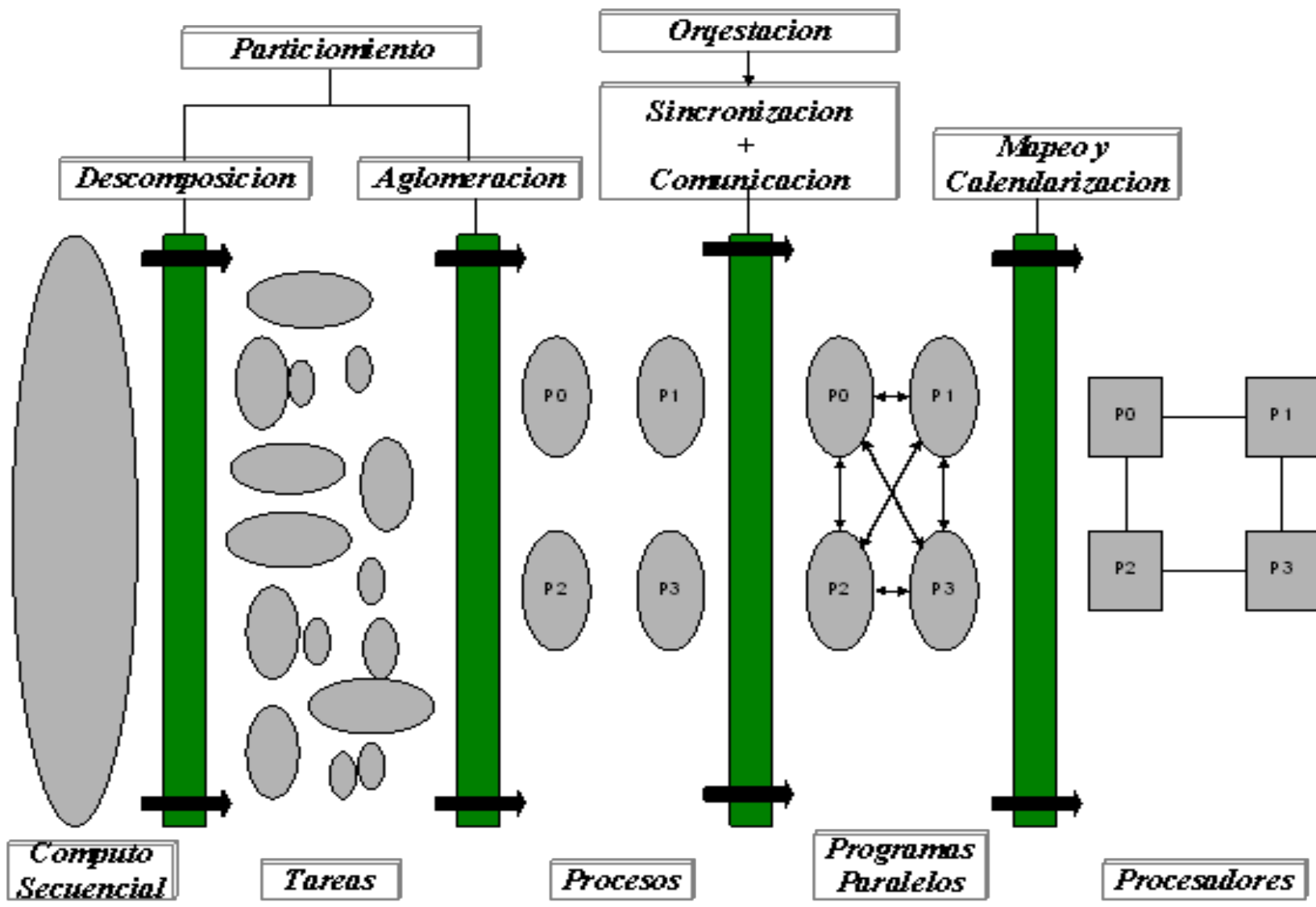
Paradigmas de programación
paralela



procesadores

Planificación de
procesos

Diseño V2



Referencia para diseño de programas paralelos

- Seguir etapas de diseño de un programa paralelo (Ian-Foster)

<http://www.mcs.anl.gov/~itf/dbpp/>

- [The Multicore Association](#) en su *Multicore Programming Practices* ofrece un método para escribir software en arquitecturas multicore utilizando C y C++
 - Algunos integrantes: CriticalBlue, Freescale, Intel, PolyCore Software, Texas Instruments, Wind River's Virtutech subsidiary.

Una forma de empezar

- Comienza con una aplicación secuencial que resuelve el problema original.
- Descompone la aplicación en tareas o conjuntos de datos.
- Analiza la dependencia entre tareas.

Consideraciones que afectan en el diseño

- Eficiencia
- Simplicidad
- Portabilidad
- Escalabilidad

El balanceo de las consideraciones es crítico para obtener el mejor código paralelo

Particionamiento

Descomposición del problema

- Descomposición de dominio o datos o geométrica.
- Descomposición funcional o tareas.

Dominio o Datos

- Primero se trabaja en la división de los datos y después es el cómputo asociado
- Es beneficiosa si el cómputo intensivo surge de un gran volumen de datos.
- o si se aplican operaciones similares a diferentes partes de datos.

Dominio o Datos

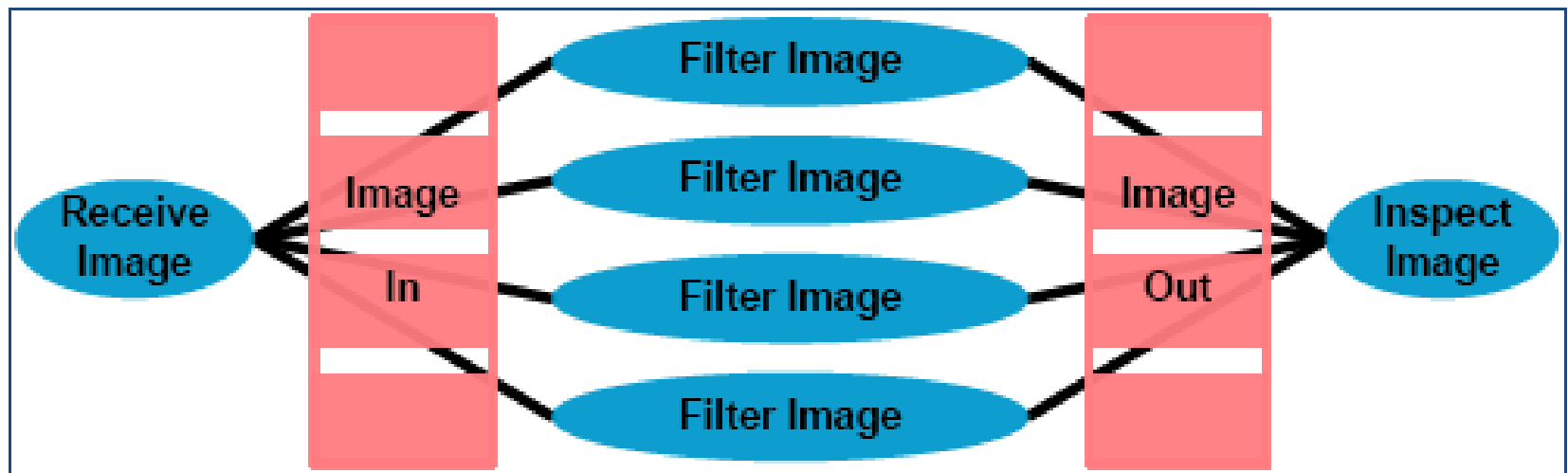
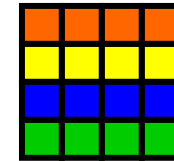


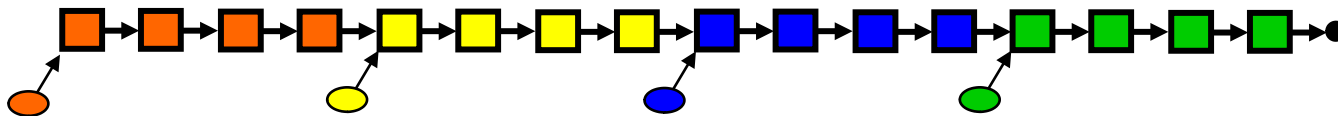
Figure 6. Data parallel image processing example.

Descomposición Dominio (Geométrica)

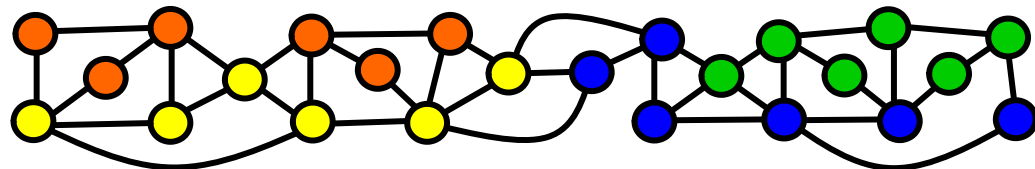
- Arreglos: divide entre una o más dimensiones



- Listas: define sublistas de elementos discretos



- Grafos y Árboles: construye sub-grafos o sub-árboles



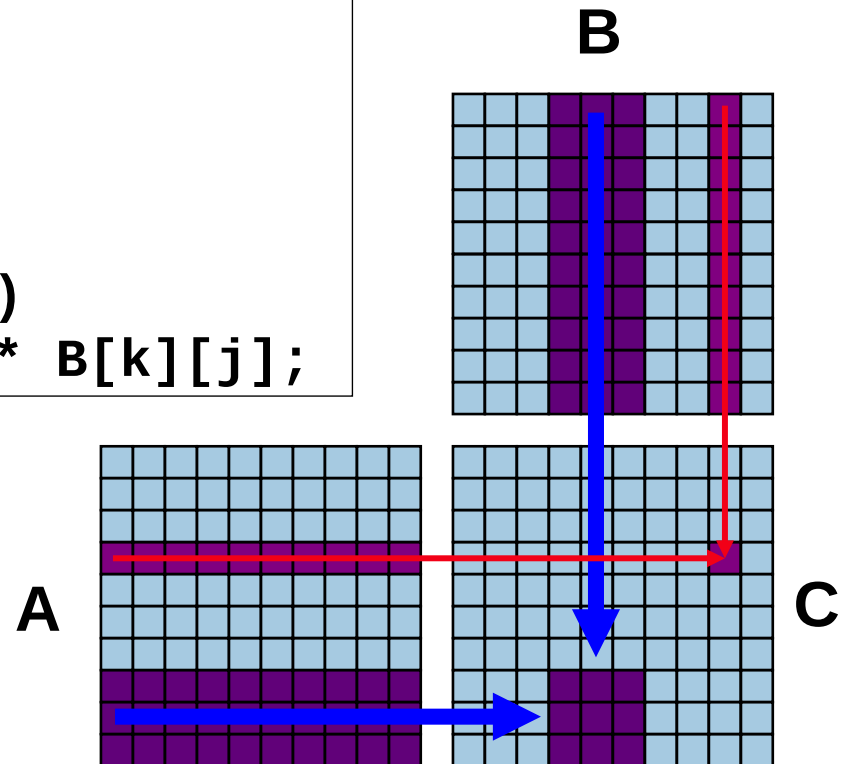
CASO DE ESTUDIO:

Multiplicación de Matrices

```
for (i = 0; i < M; i++)  
  for (j = 0; j < N; j++)  
    C[i][j] = 0.0;  
  
for (i = 0; i < M; i++)  
  for (j = 0; j < N; j++)  
    for (k = 0; k < L; k++)  
      C[i][j] += A[i][k] * B[k][j];
```

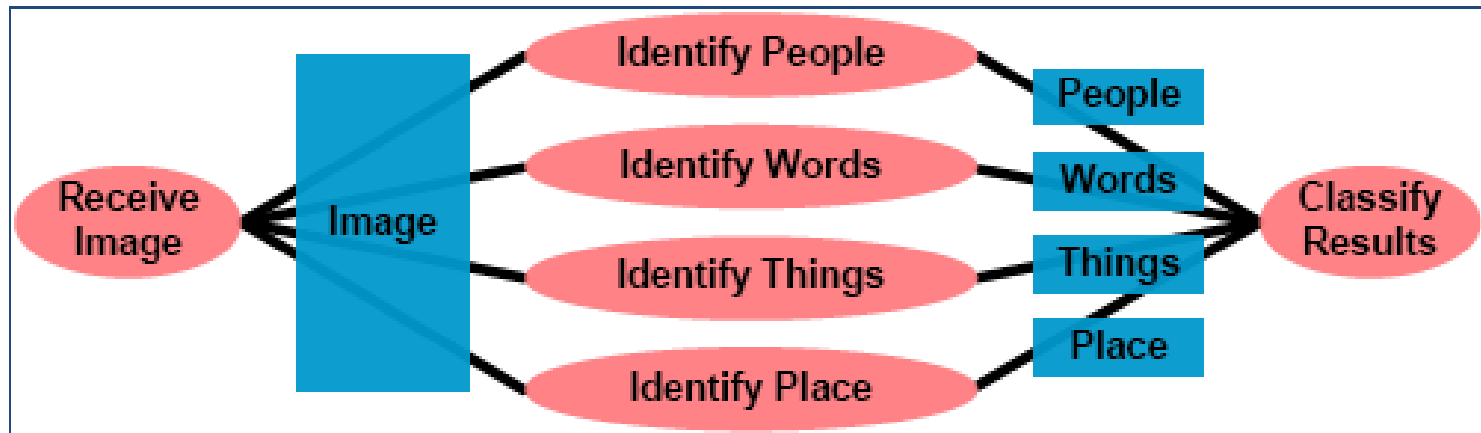
¿Qué se puede calcular de manera independiente?

- Elementos individuales de C
- Renglones de C (usando todo de B)
- Columnas de C (usando todo de A)
- Bloques de C

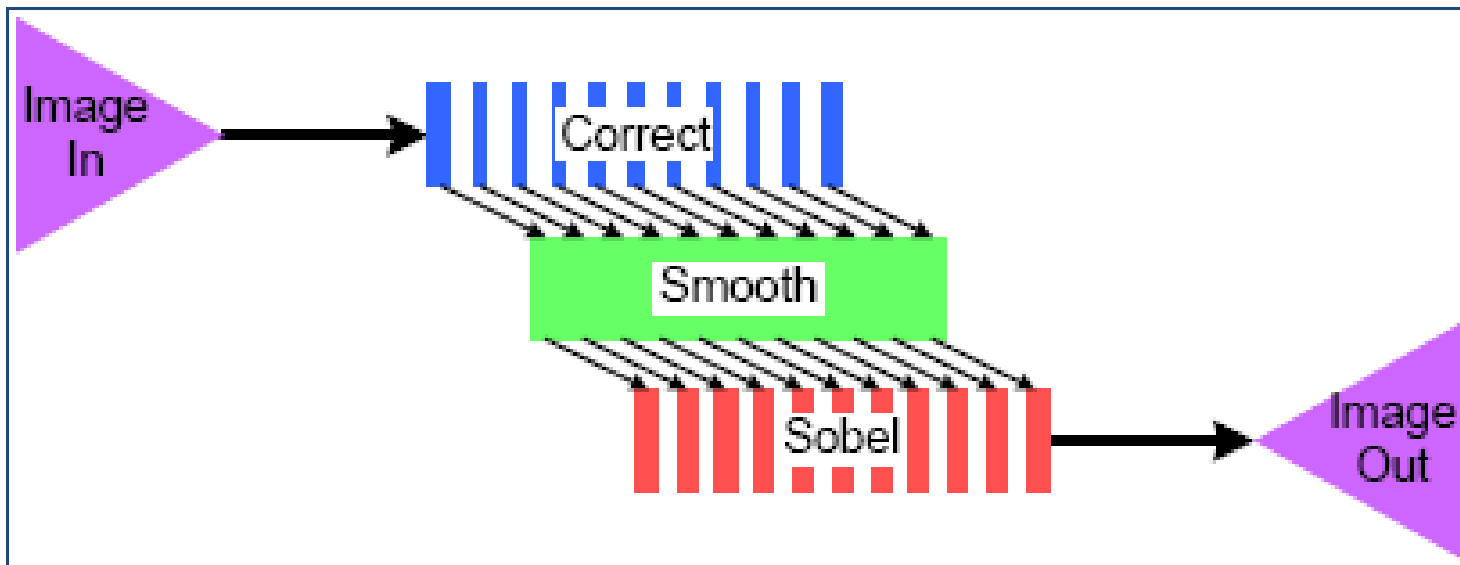


Descomposición Funcional

- Primero se trabaja en la división de tareas o funciones y después en los datos asociados a cada tarea.



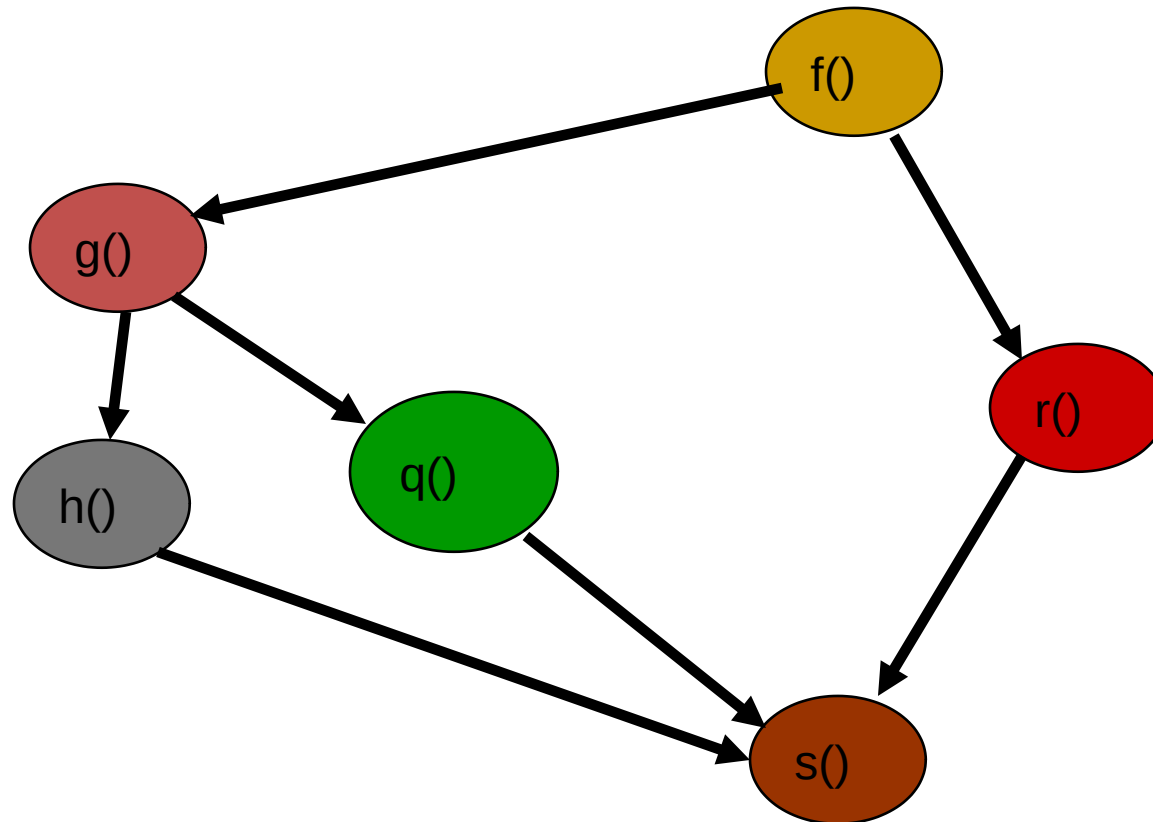
Descomposición pipilined



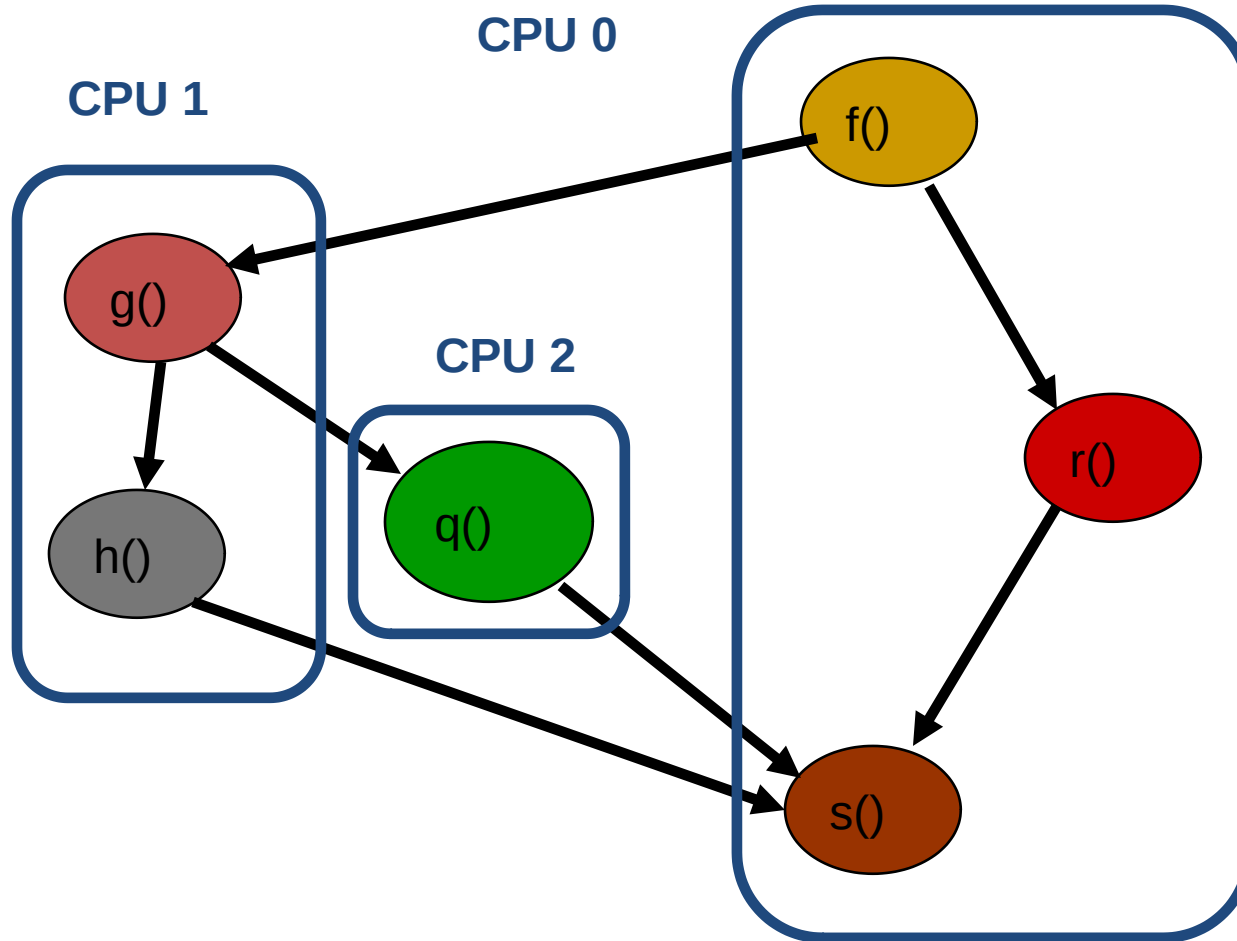
Considerar

- Debe ser al menos un número de tareas como hilos/procesos
 - Preferible que haya más tareas que hilos/procesos
- La cantidad de cómputo dentro de una tarea debe ser lo suficientemente grande como para compensar la sobrecarga de gestión de tareas e hilos

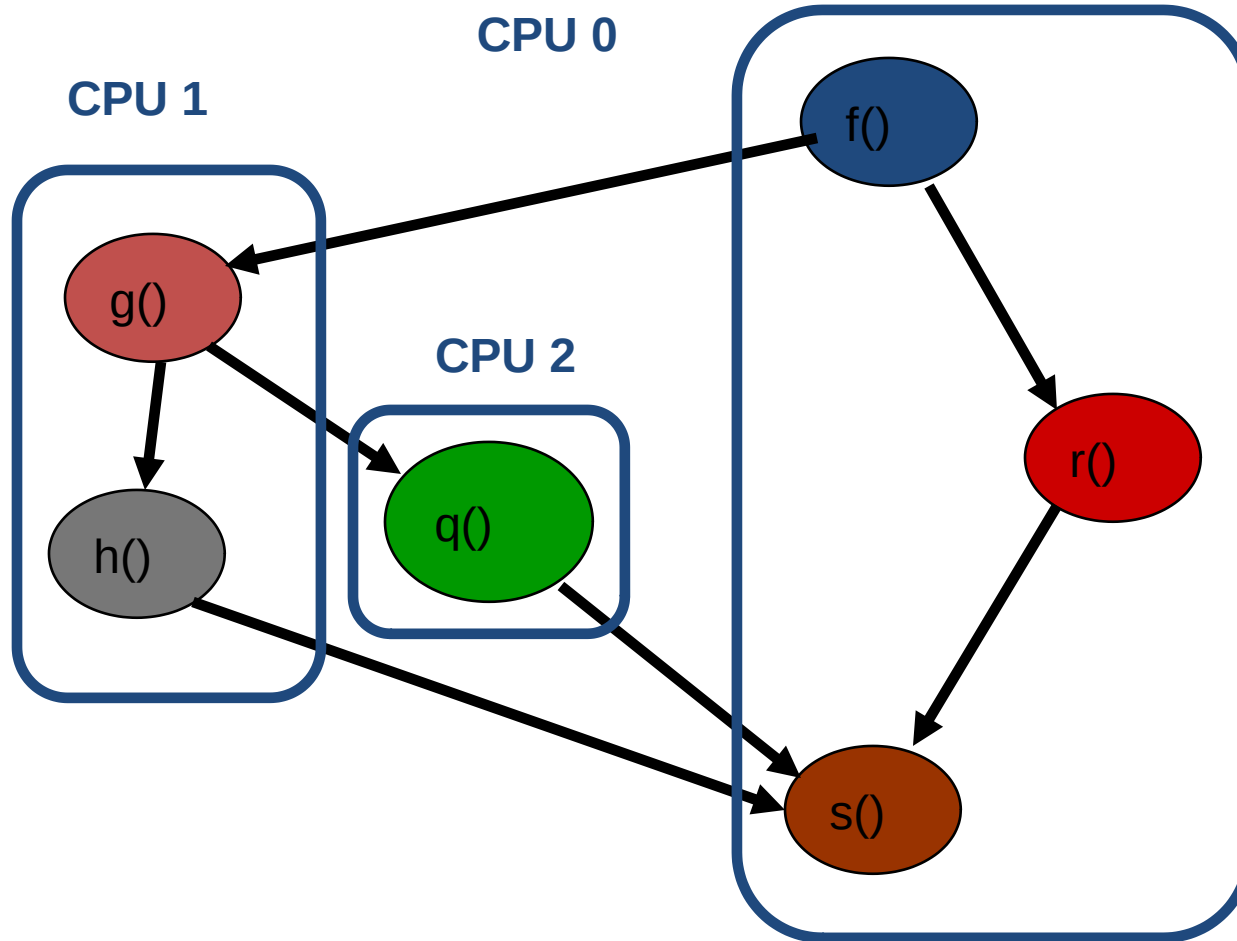
Descomposición Funcional



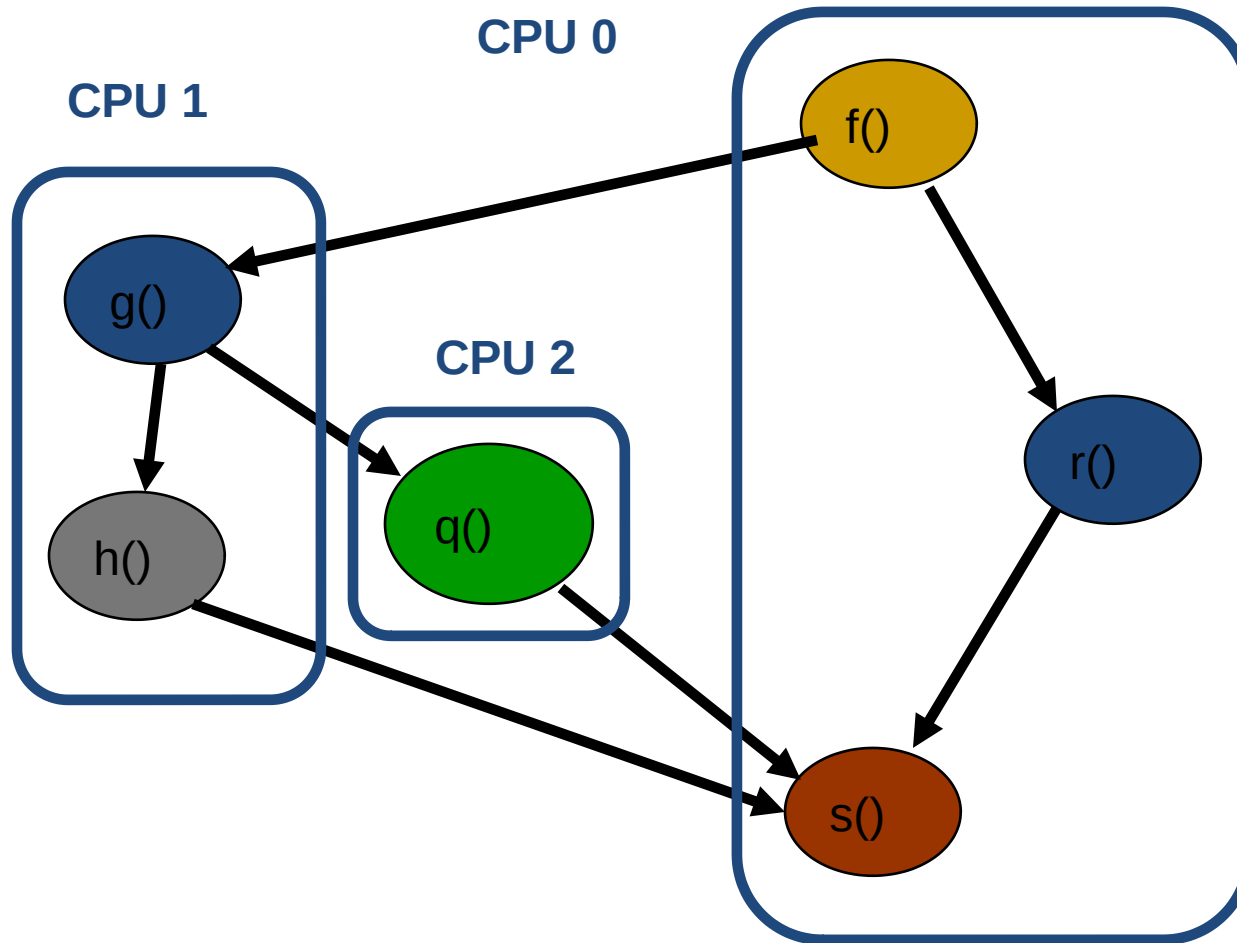
Descomposición Funcional



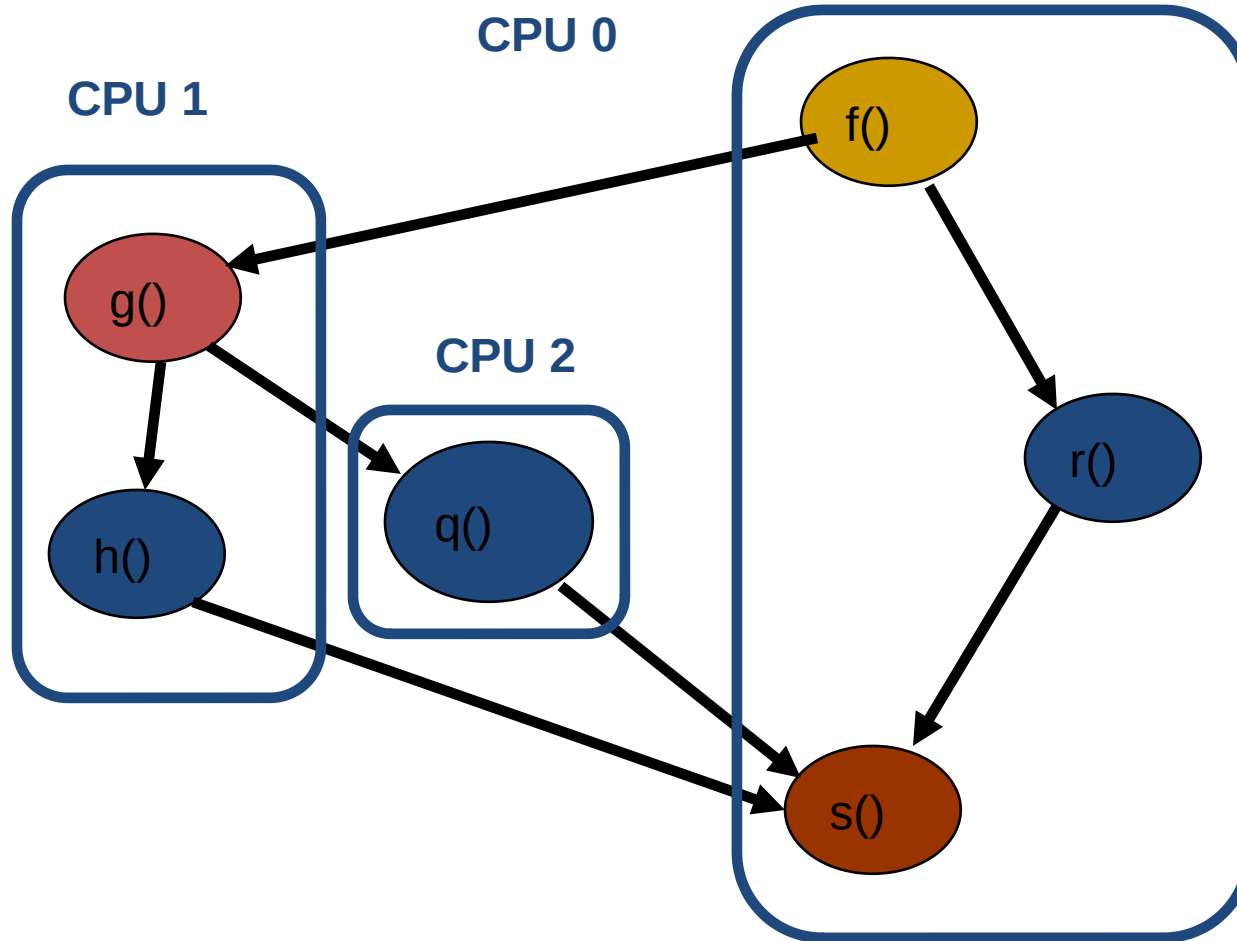
Descomposición Funcional



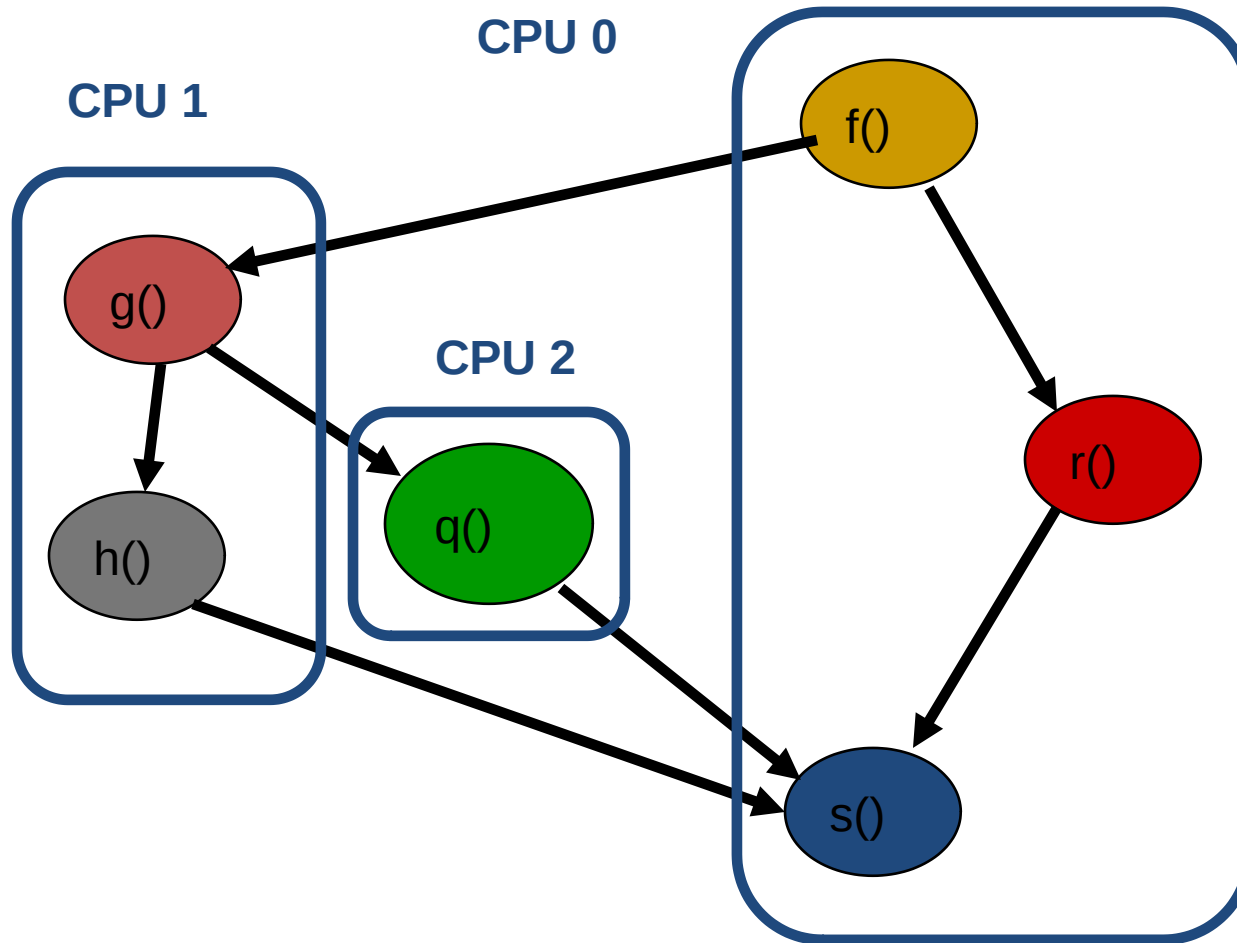
Descomposición Funcional



Descomposición Funcional



Descomposición Funcional



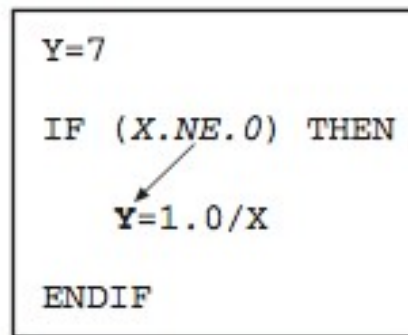
Dependencias

- En los programas de computadora, cuando un evento A debe suceder antes de otro evento B, se dice que B es dependiente de A.
- Para poder ejecutar diferentes segmentos de código en paralelo, tenemos que asegurarnos de que son independientes, es decir, el orden de ejecución no importa (no afecta al resultado).
- El análisis de dependencias es estático, es decir, no se realiza en tiempo de ejecución.

Dependencia de control

- Cuando se esta la espera de que ocurra un salto, o la salida de un ciclo o
- El valor de una variable puede depender también del flujo de control del programa

```
Y=7  
IF (X.NE.0) THEN  
    Y=1.0/X  
ENDIF
```



Dependencias

- Cada tipo está presente en cada programa, sólo que en grados distintos.
- El objetivo es eliminar tantas de ellas como sea posible.
- Reacomodar el código, de forma que dos segmentos sean menos dependientes entre sí, expone el paralelismo, o las oportunidades de hacer muchas cosas a la vez.

Graficas de dependencias

- Consideremos dos secuencias de instrucciones S1 y S2, para las
- que existe un camino de ejecución entre ellas (S1 primero, y después S2).

Gráfica de dependencias

Grafica = (nodo, flecha)

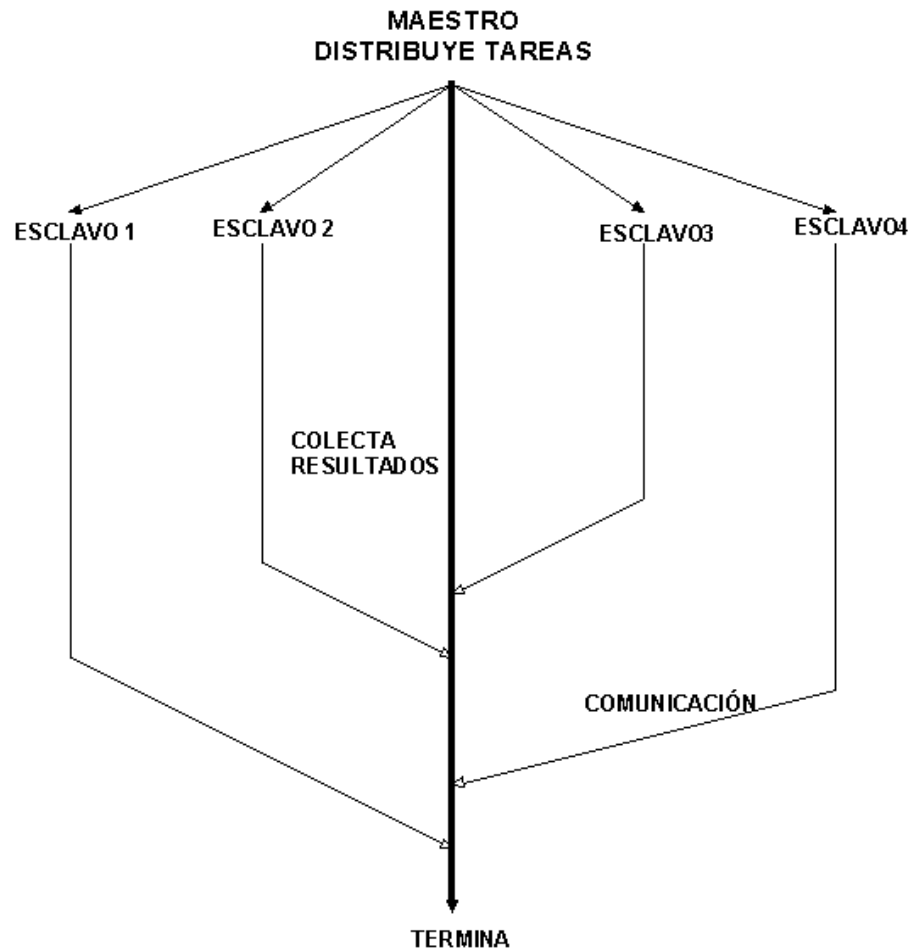
- Nodo por cada
 - variable (excepto variables índice)
 - Constante
 - Operador o función
- Flechas indican uso de variables y constantes

Patrones de Programación paralela

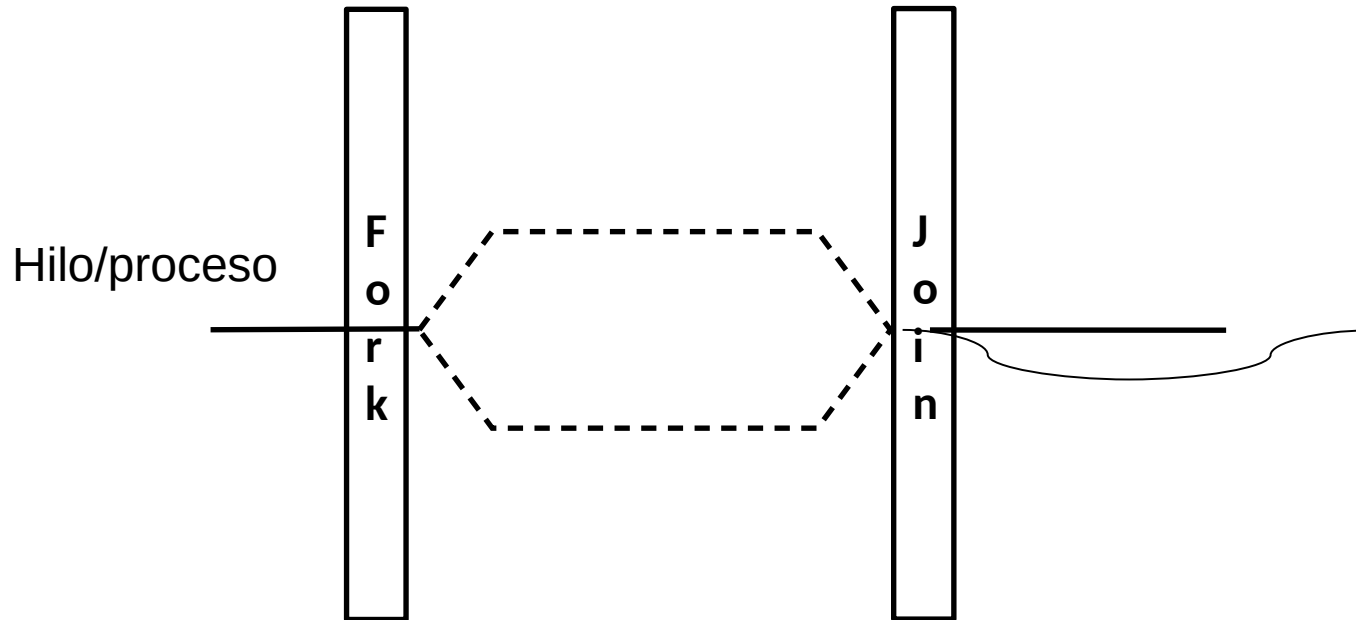
- Pretenden ser de ayuda para agilizar el proceso de desarrollo de programas paralelos proponiendo mecanismos generales de sincronización y comunicaciones.
- Proveen una estructura de coordinación para procesos o hilos en la formación de un programa paralelo.
- A veces llamados paradigmas de programación paralela.

Maestro- Esclavo

ESTRUCTURA ESTÁTICA MAESTRO/ESCLAVO



Fork - Join



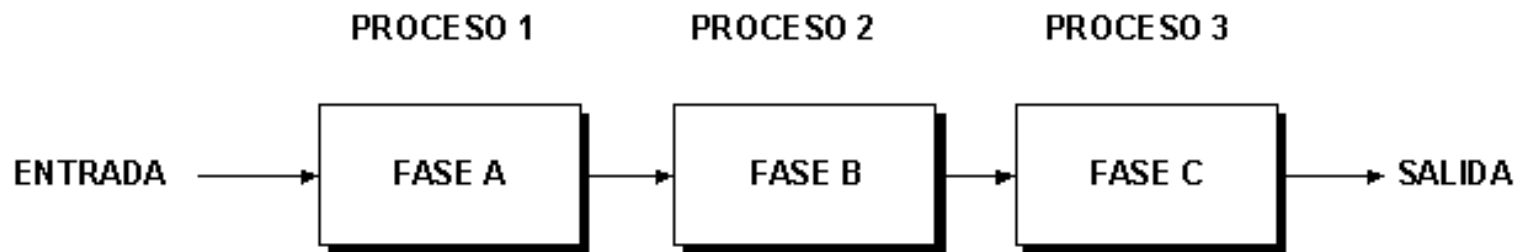
SPMD

- Se usa un numero fijo de hilos o procesos
- Cada proceso ejecuta el mismo código en diferentes partes de los datos
- Los hilos /procesos trabajan de forma asincrona pueden trabajar con diferentes partes del programa

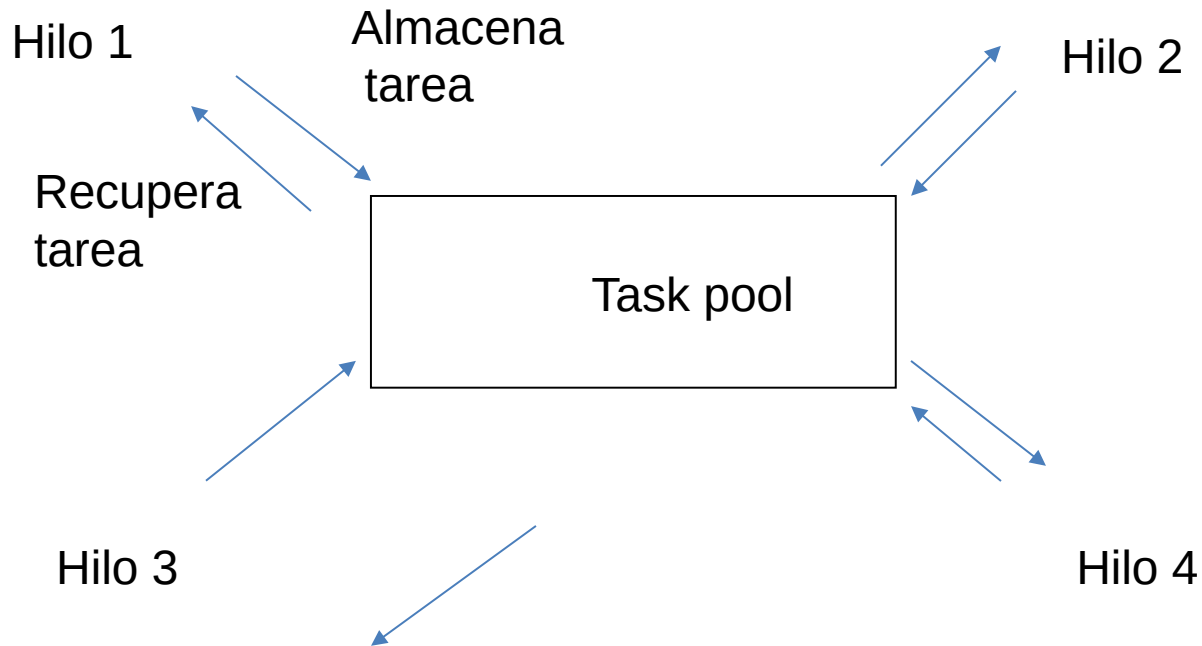
Entubamiento de datos (Data Pipelining)

- Llamado “Paralelismo de flujo de datos”

ESTRUCTURA DATA PIPELING

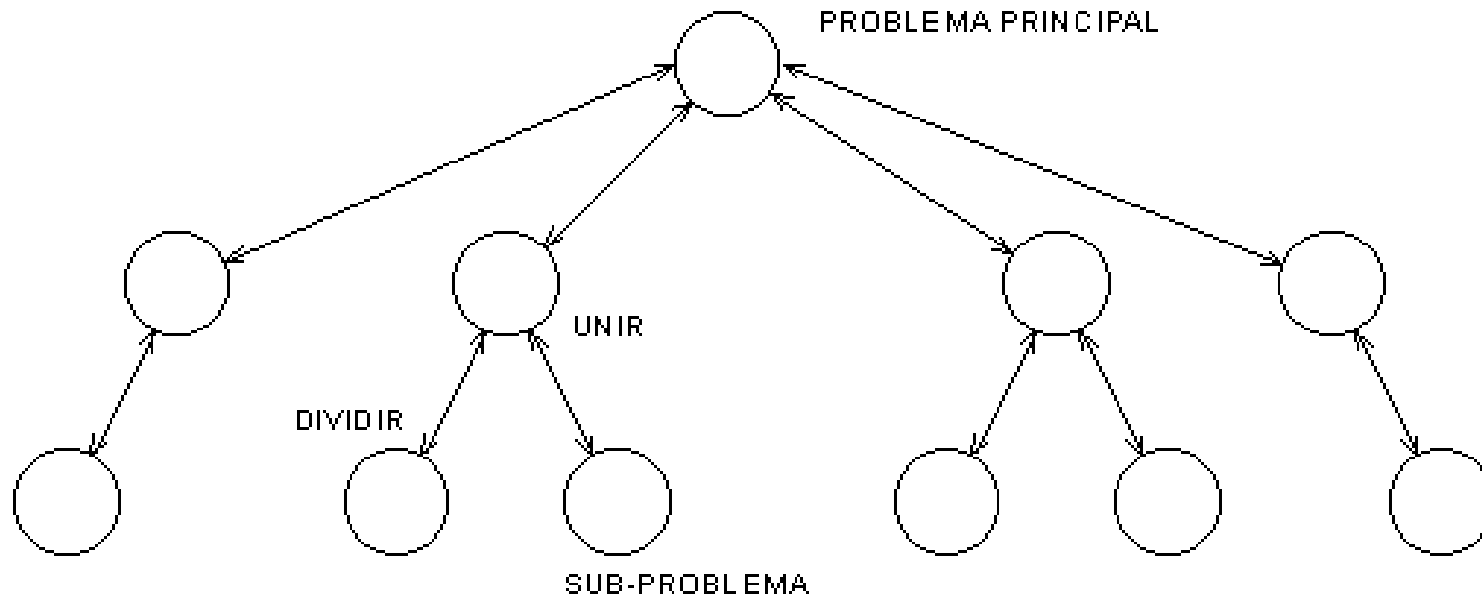


Task Pools



Divide y Conquista

DIVIDE Y CONQUISTA



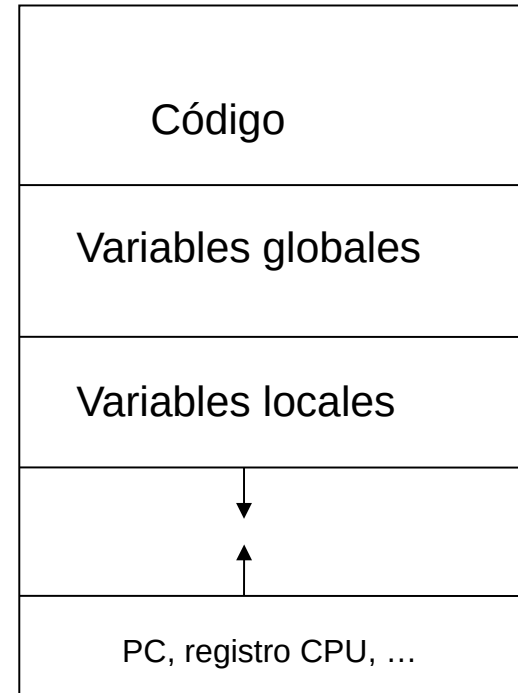
Procesos e hilos

- Un proceso es un hilo principal
- El proceso o hilo principal crea los hilos
- Los hilos siempre pertenecen a un proceso o hilo principal.
- Generalmente si termina el hilo principal, los hilos creados por éste salen de ejecución aunque no hayan terminado.

Ing. Laura Sandoval Montaña, M.I. Elba Karen Sáenz García

Hilos

- Permiten la ejecución concurrente de varias secuencias de instrucciones asociadas a funciones dentro de un mismo proceso (hilo principal).
- Los hilos hermanos entre **sí comparten** la misma imagen de memoria, es decir:
 - Código,
 - Variables de memoria globales, y
 - Los dispositivos y archivos que tuviera abierto el hilo principal.
- Los hilos hermanos **no comparten**:
 - El Contador de Programa: cada hilo podrá ejecutar una sección distinta de código.
 - Los registros de CPU.
 - La pila en la que se crean las variables locales de las funciones a las que se va llamando después de la creación del hilo.
 - El estado: puede haber hilos en ejecución, listos, o bloqueados en espera de un evento.



Referencias

- Introduction to parallel programming, Intel Software College, junio 2007
- Multicore programming practices guide, [The Multicore Association](#)
- *Patterns for Parallel Programming*, Timothy G. Mattson, Beverly A. Sanders, Berna L. Massingill, Addison-Wesley, 2005, ISBN 0321228111
- Designing and Building Parallel Programs, Ian Foster, Addison-Wesley Inc., Argonne National Laboratory, and the NSF Center for Research on Parallel Computation.