

FUNDAMENTOS DE PROGRAMACIÓN EN GPUS DE PROPÓSITO GENERAL CON CUDA

LABORATORIO DE INTEL PARA LA ACADEMIA
INSTRUCTORES:
ELBA KAREN SÁENZ GARCÍA

TEMARIO

Antecedentes

GPU – CUDA

- GPGPU
- Arquitectura CUDA
- GPU Genérico

Modelo de ejecución CUDA

- Grids ,Bloques, Hilos

Modelo de programación CUDA

- Device y Host
- Funciones CUDA

Manejo de arreglos unidimensionales

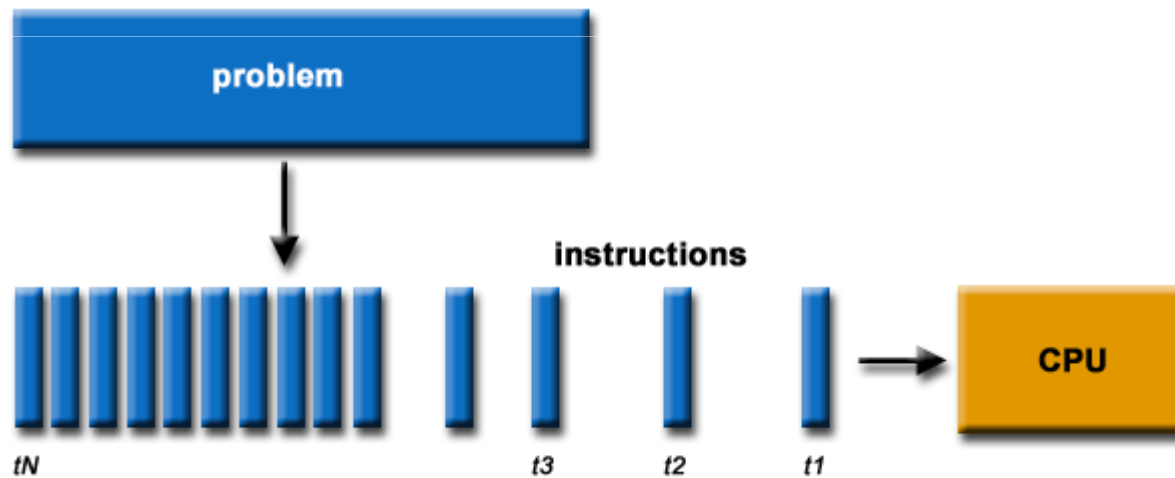
Manejo de arreglos bidimensionales

Memoria Compartida

ANTECEDENTES

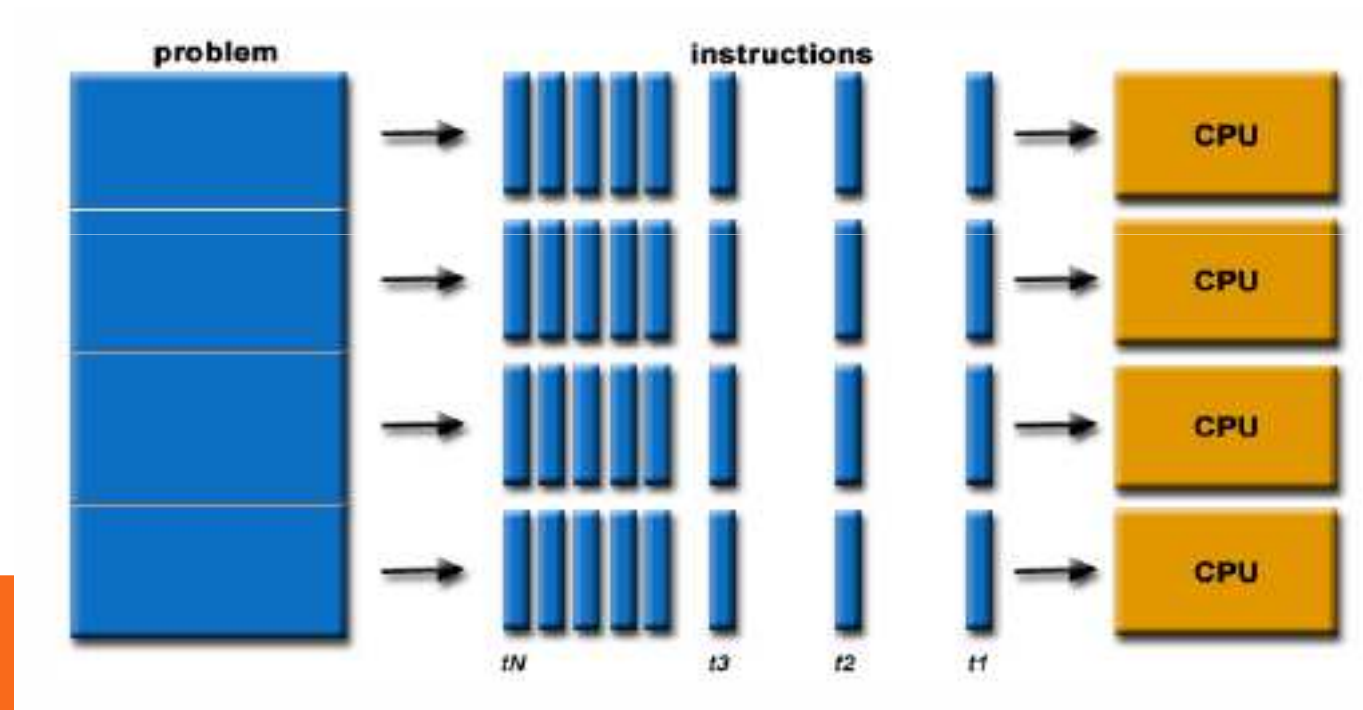
CÓMPUTO SERIAL

Uso de un recurso de cómputo para resolver un problema computacional



CÓMPUTO PARALELO

Uso de varios recursos de cómputo para resolver un problema



CÓMPUTO PARALELO

Recurso Computacional

- Una sola computadora con varios procesadores
- Varias computadoras conectadas en red
- Combinación de ambas

Tipo de problema computacional

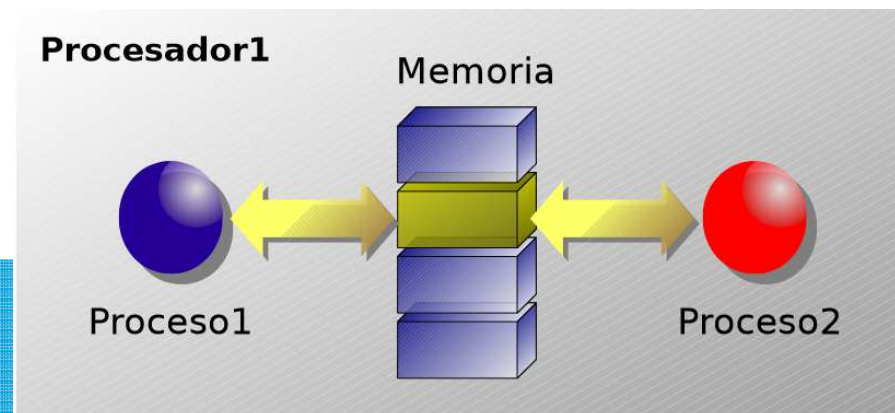
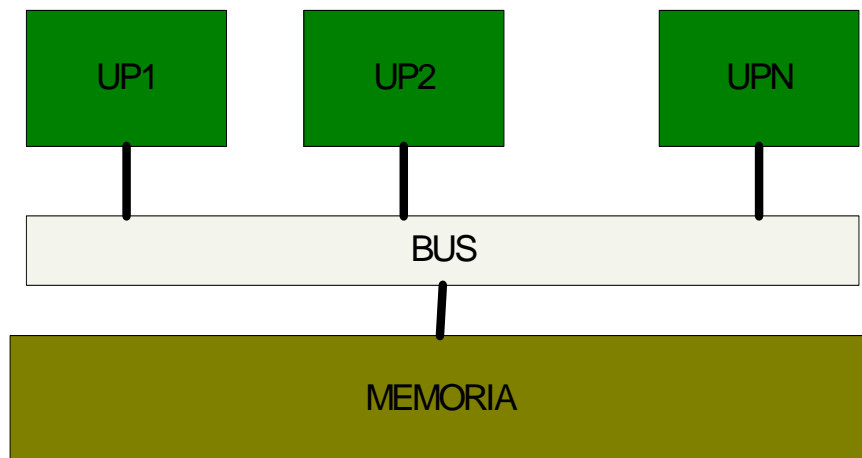
- Se debe poder dividir en partes pequeñas que puedan resolverse de forma simultanea y al mismo tiempo
- Poderlo resolver en menos tiempo con varios recursos computacionales que con solo uno.

MODELOS DE PROGRAMACIÓN PARALELA

De acuerdo al manejo de memoria

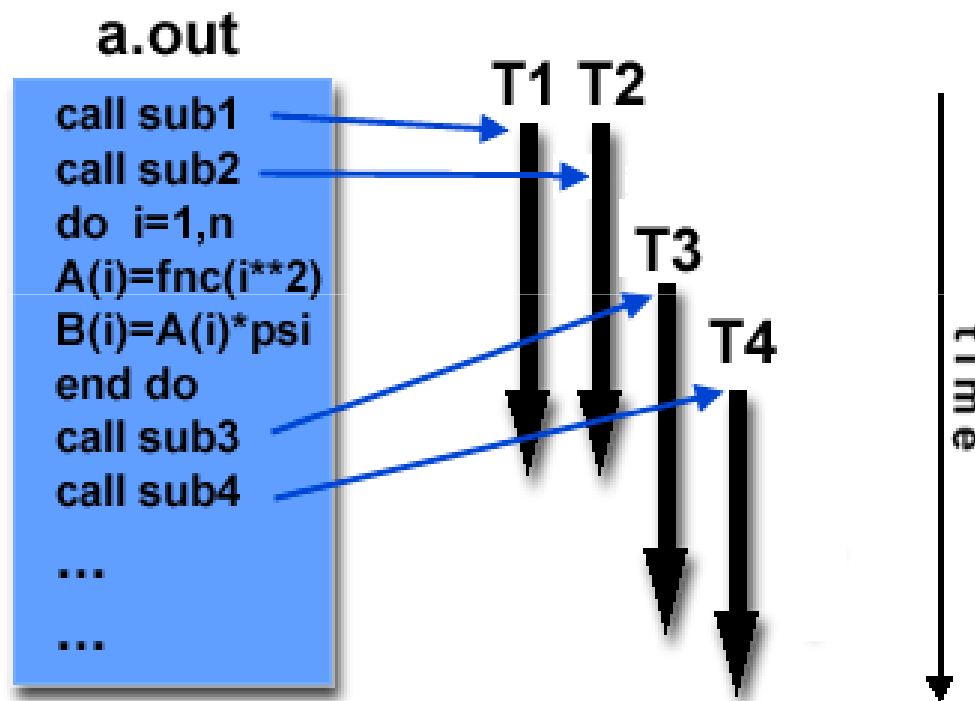
- Memoria compartida (shared memory)
 - Modelo de hilos
- Memoria distribuida (distributed memory)
- Memoria compartida distribuida (distributed shared memory)

MEMORIA COMPARTIDA

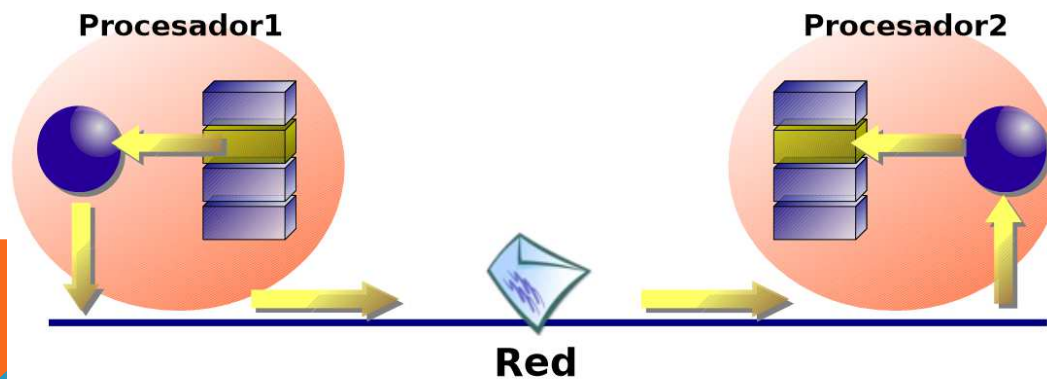
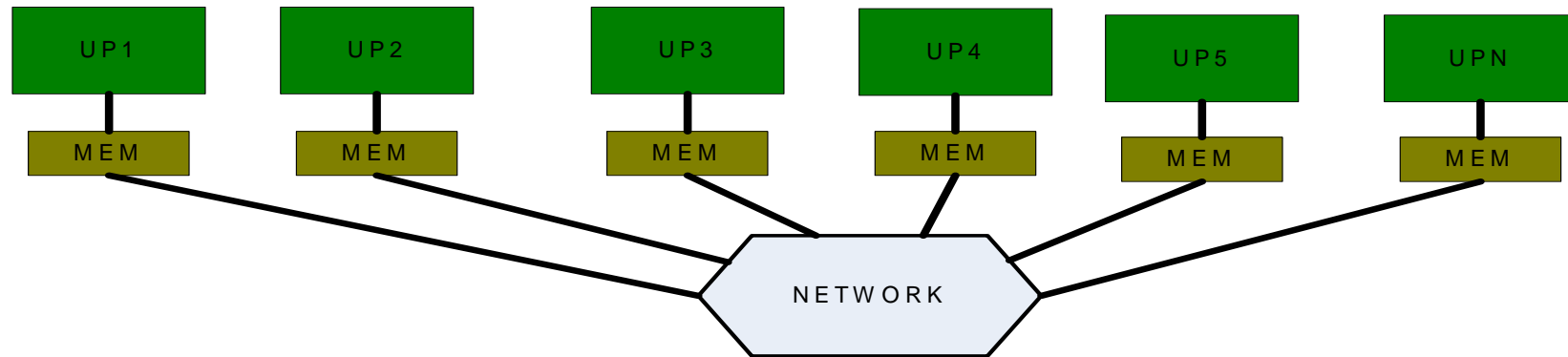


MODELO HILOS

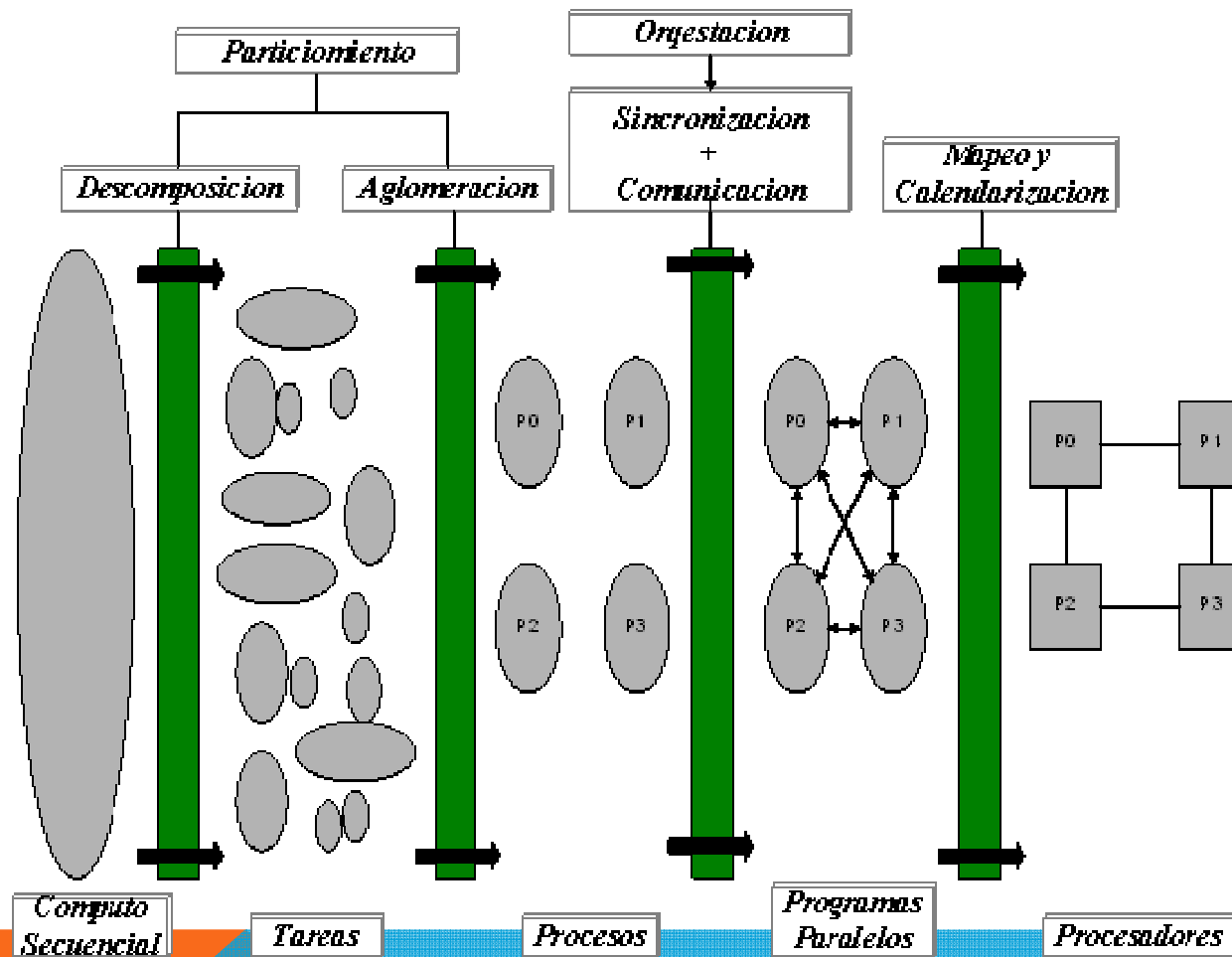
- Tipo de programación en memoria compartida.



MEMORIA DISTRIBUIDA



LA METODOLOGÍA DE DISEÑO DE FOSTER



Ian Foster, *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*, Addison-Wesley, 1995

PARTICIONAMIENTO O DESCOMPOSICIÓN

Descomposición de dominio o datos

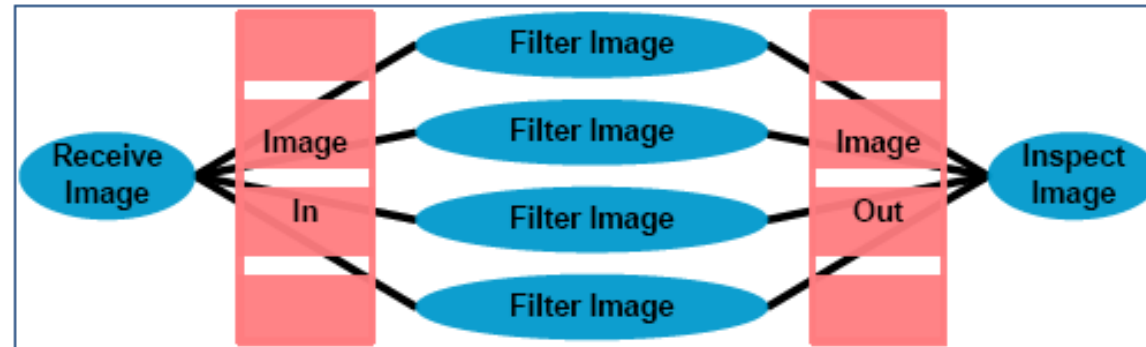
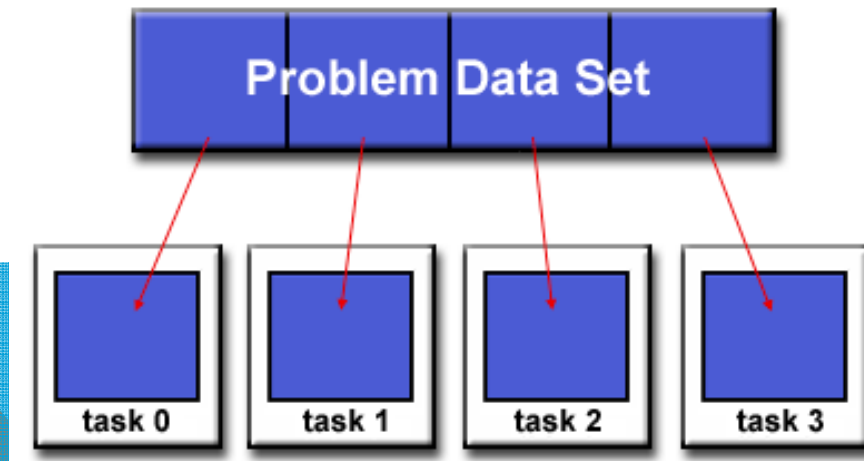
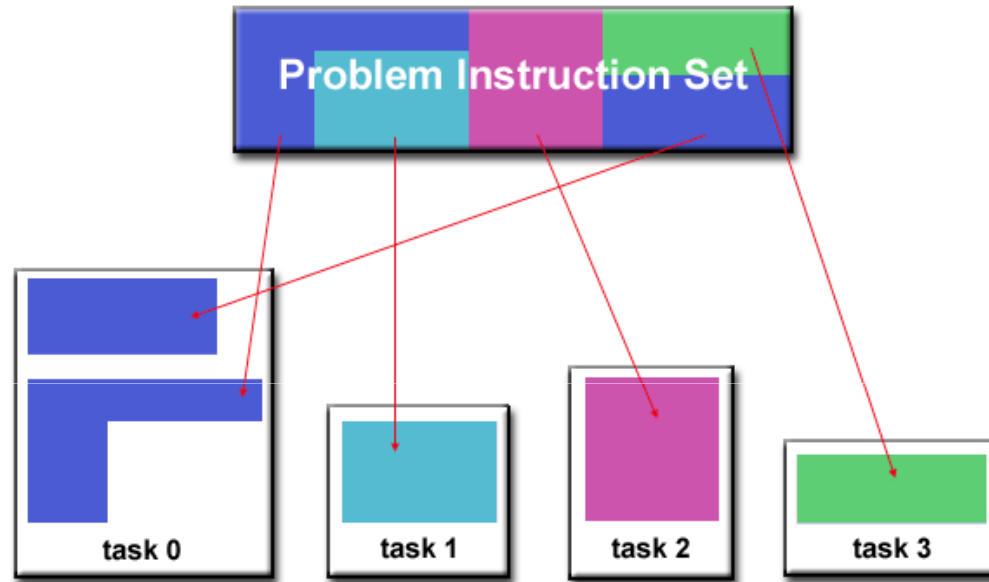


Figure 6. Data parallel image processing example.

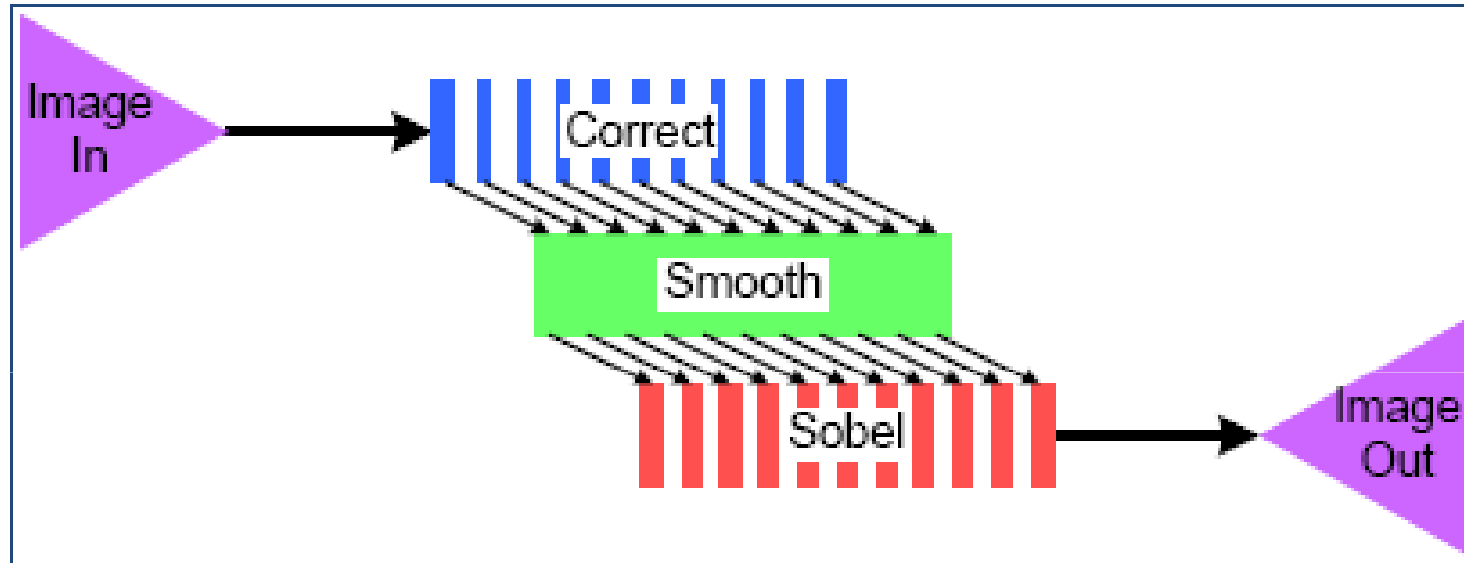


PARTICIONAMIENTO

- Descomposición funcional o tareas

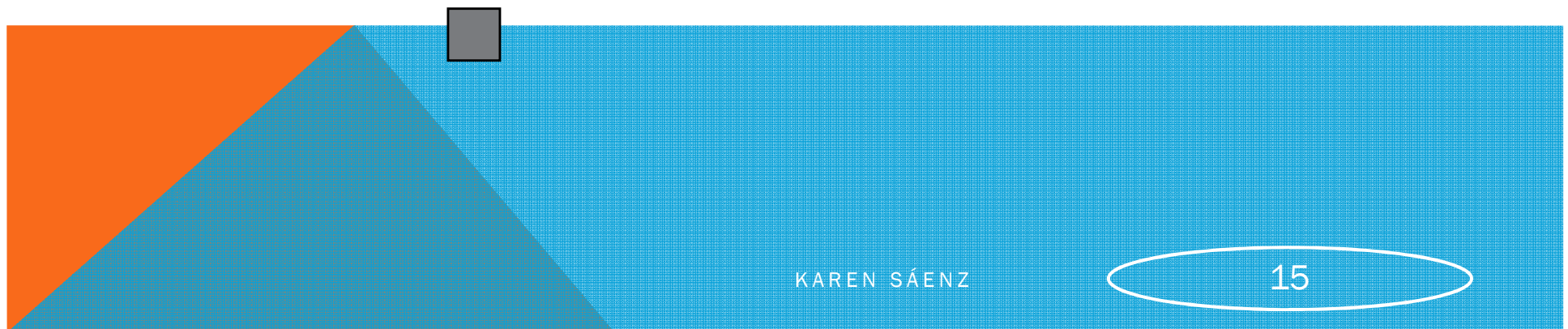
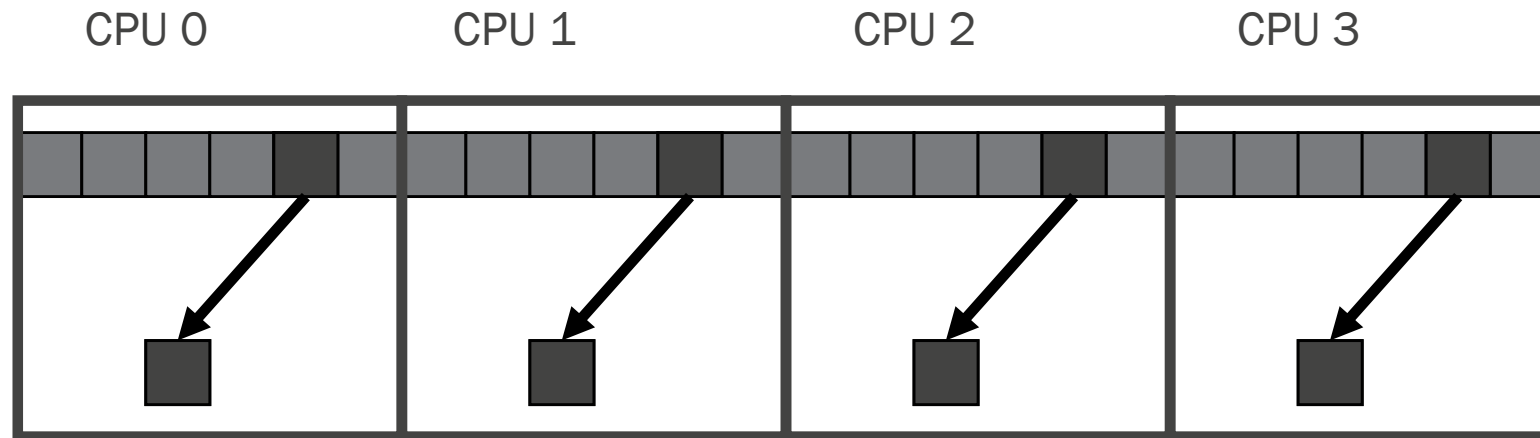


DESCOMPOSICIÓN PIPILINED



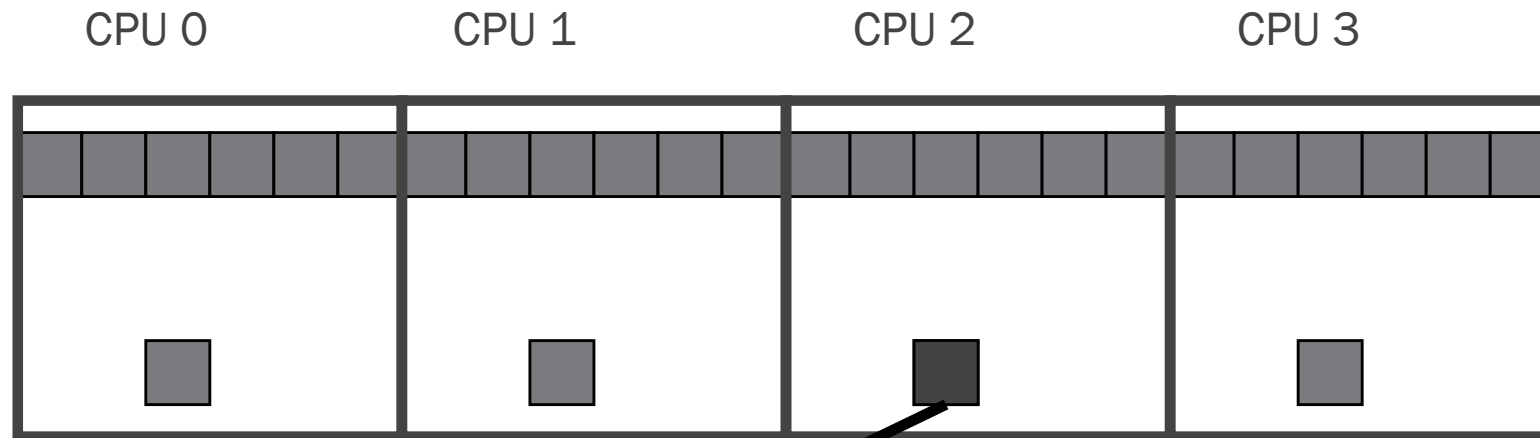
DESCOMPOSICIÓN DE DOMINIO

Encontrar el elemento mayor en un arreglo de enteros



DESCOMPOSICIÓN DE DOMINIO

Encontrar el elemento mayor en un arreglo de enteros



EJEMPLO DESCOMPOSICIÓN DE DOMINIO 0 DATOS

Sumar los renglones de una matriz de 4*28

```
for (i=0; i<4i++)  
{  
    B[i]=0;  
    for (j=0;j<28)  
        B[ i ] = A[ i ][ j ] + B[ i ];  
}
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		406
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		406
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		406
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		406

EJEMPLO DESCOMPOSICIÓN DE DOMINIO O DATOS

Opción 1 Dividir el número de Renglones de A entre el número de CPUs disponibles

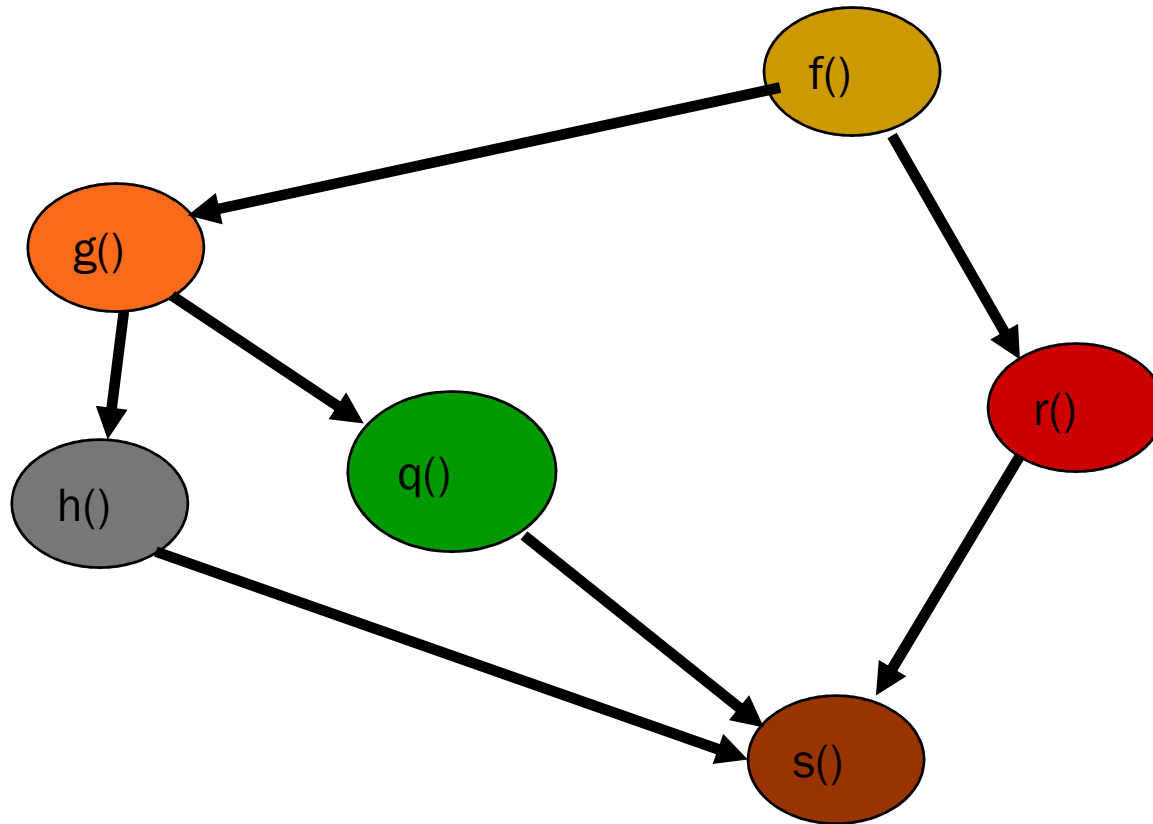
cpu 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		406	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		406	
cpu 2																															406
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		406	

Opción 2 Asignar cada renglón a un CPU se necesita

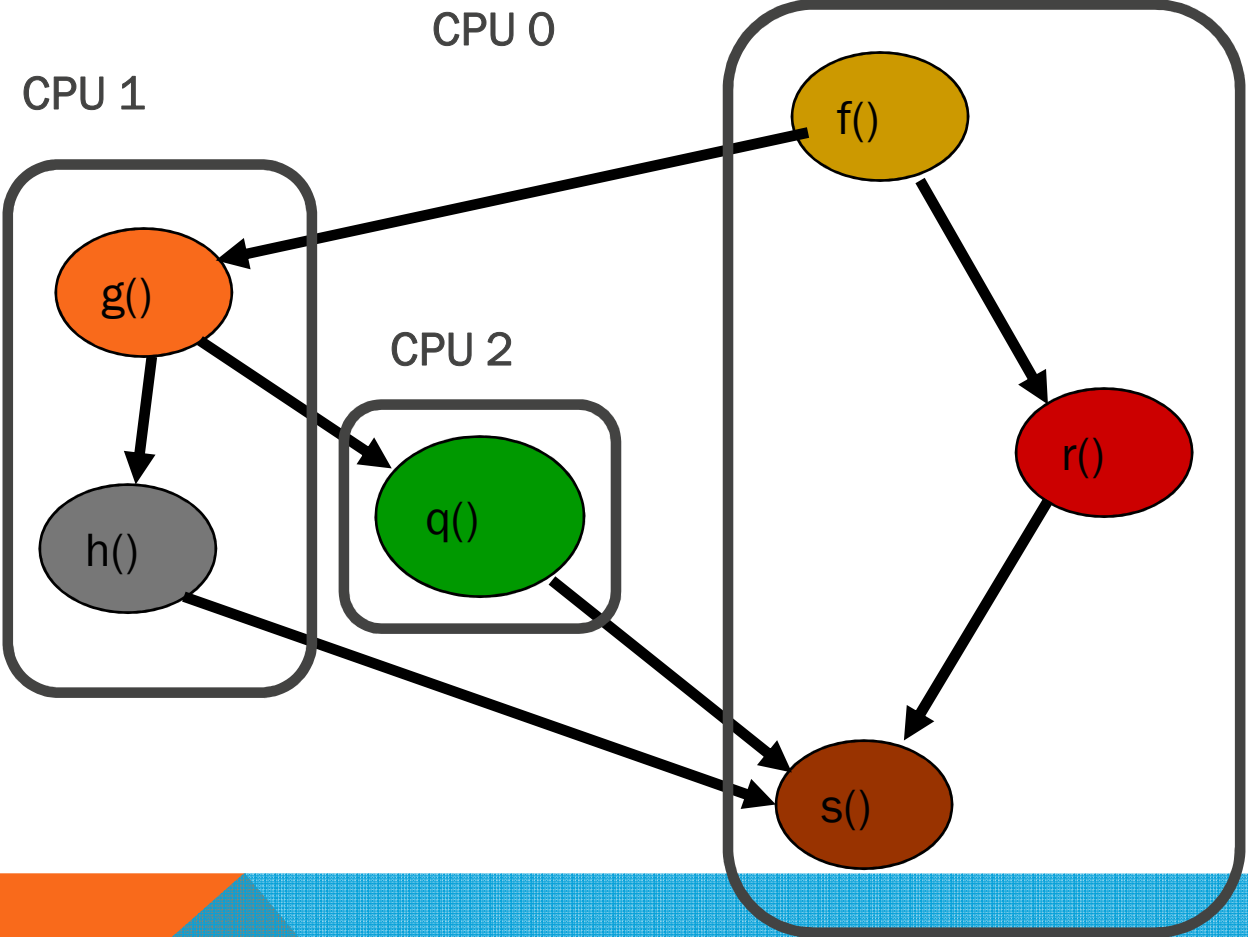
cpu 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		406
cpu 2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		406
cpu 3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		406
cpu 4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		406

¿Que pasaría con si la matriz A fuera de 1000×1000 ?

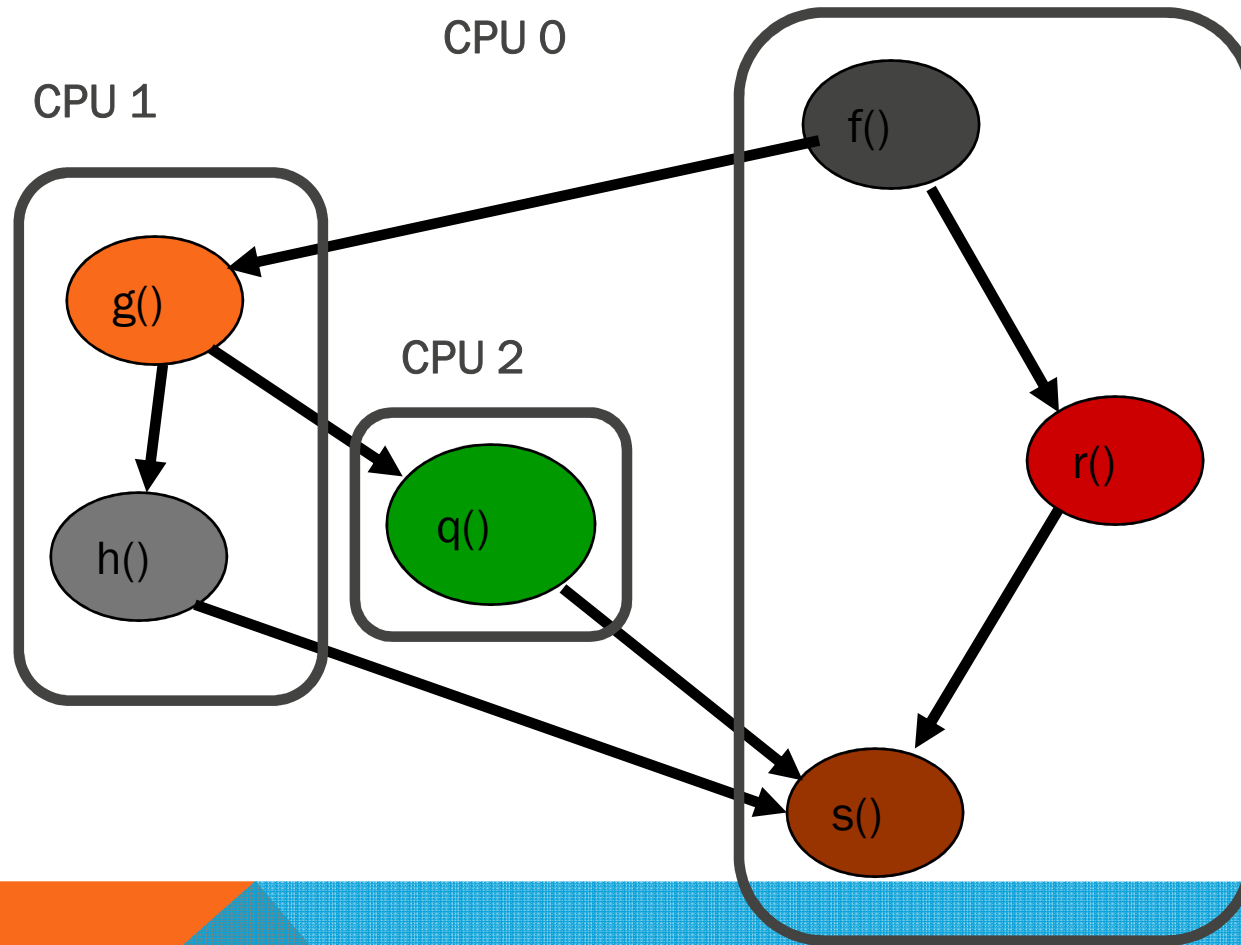
DESCOMPOSICIÓN FUNCIONAL



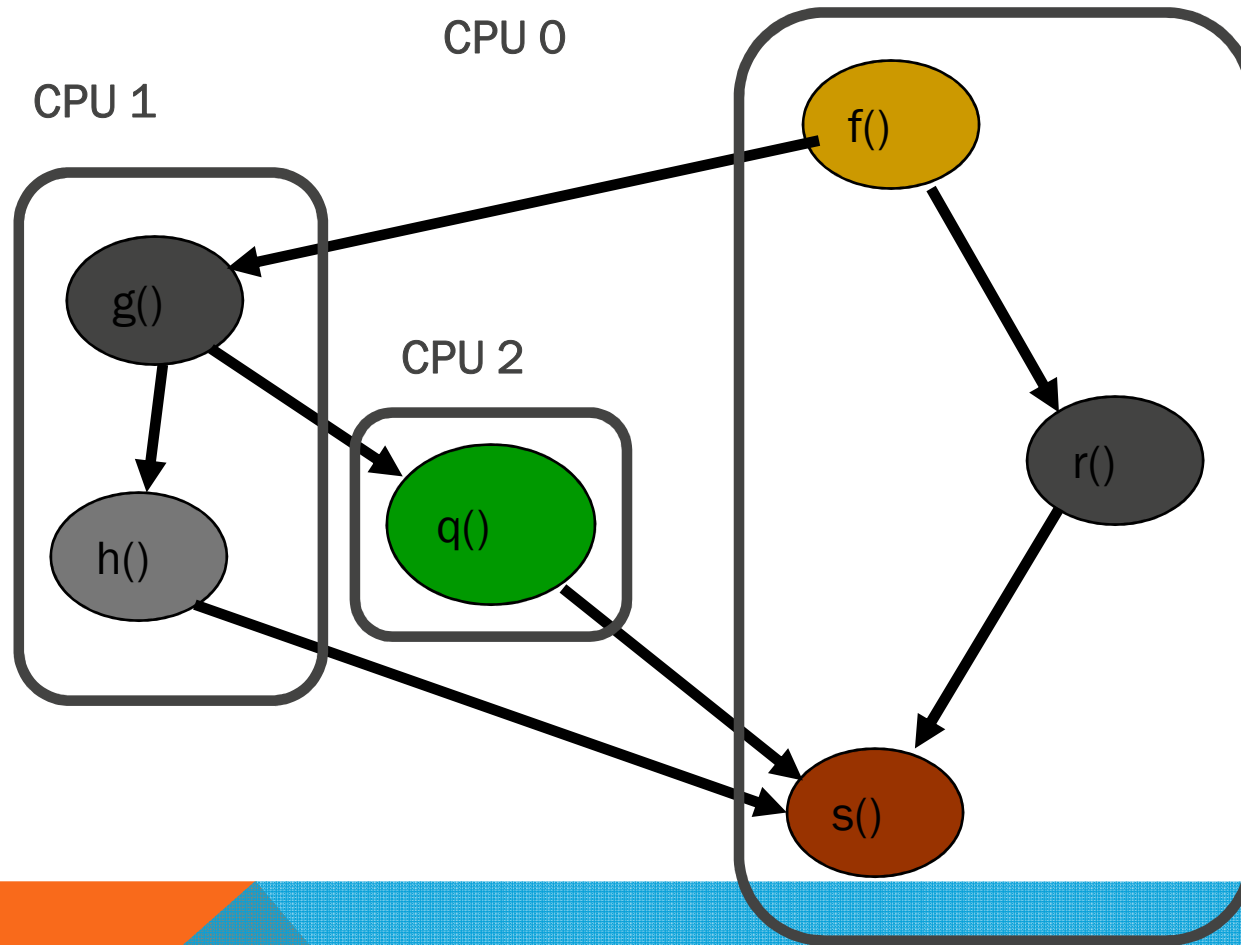
DESCOMPOSICIÓN FUNCIONAL



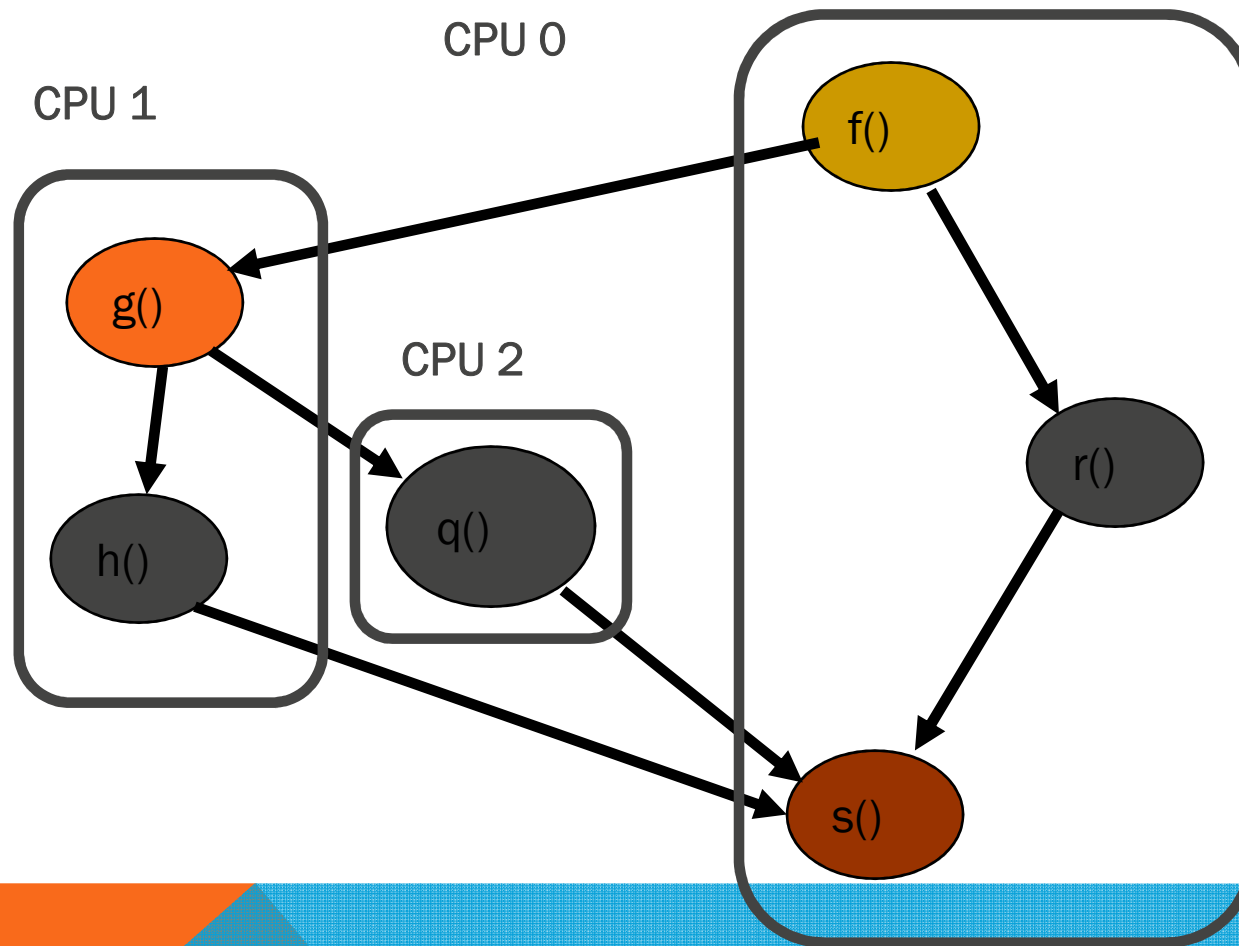
DESCOMPOSICIÓN FUNCIONAL



DESCOMPOSICIÓN FUNCIONAL



DESCOMPOSICIÓN FUNCIONAL



DESCOMPOSICIÓN FUNCIONAL

