

PROGRAMACIÓN en PARALELO

FUNDAMENTOS

COMPUTO en PARALELO

tiene

DISEÑO de PROGRAMAS: Ideas clave

paso

1

Particionamiento

si se necesita o no

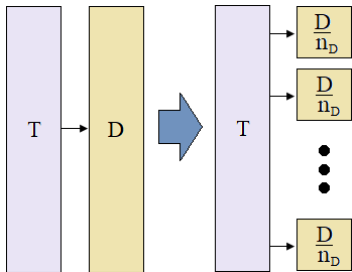
2

Coordinación

tipo

Paralelismo de **DATOS**

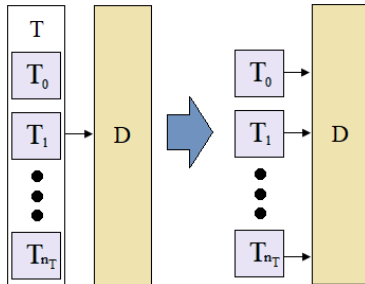
Descomposición del DOMINIO



tipo

Paralelismo de **TAREAS**

Descomposición del FUNCIONAL



en

Comunicación

es

Transferencia de
Datos entre Tareas

en

Sincronización

causa

Necesidad de
Cooperación

hay

Dependencia
entre tareas

en

Balance de Carga

es

Cada Procesador
debe tener la misma
cantidad de trabajo

COMPUTO en PARALELO

tiene

DISEÑO de PROGRAMAS: Método de Foster

paso

1

Particionamiento

2

Comunicación

3

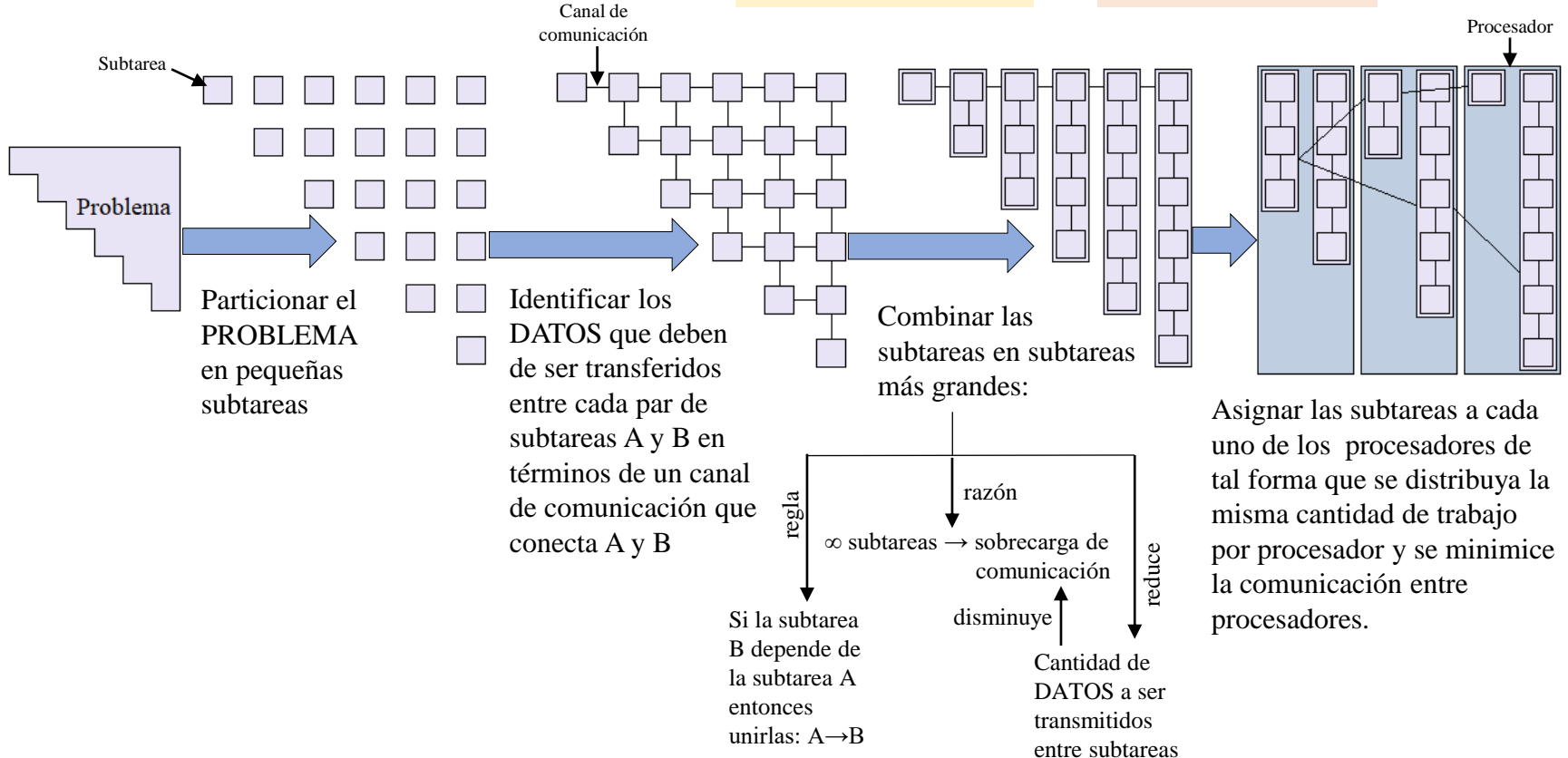
Unión

[Sincronización]

4

Mapeo

[Balance de Carga]

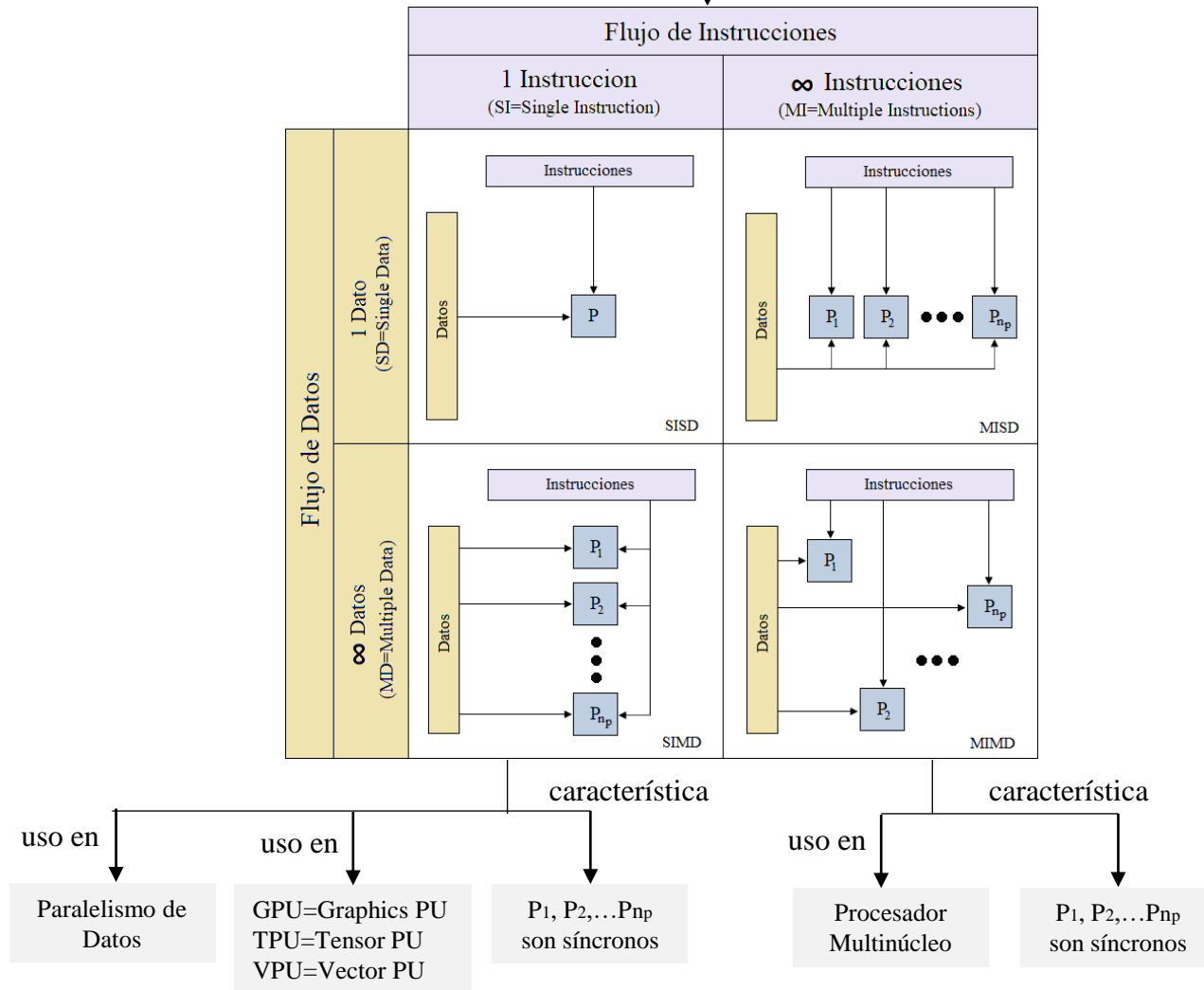


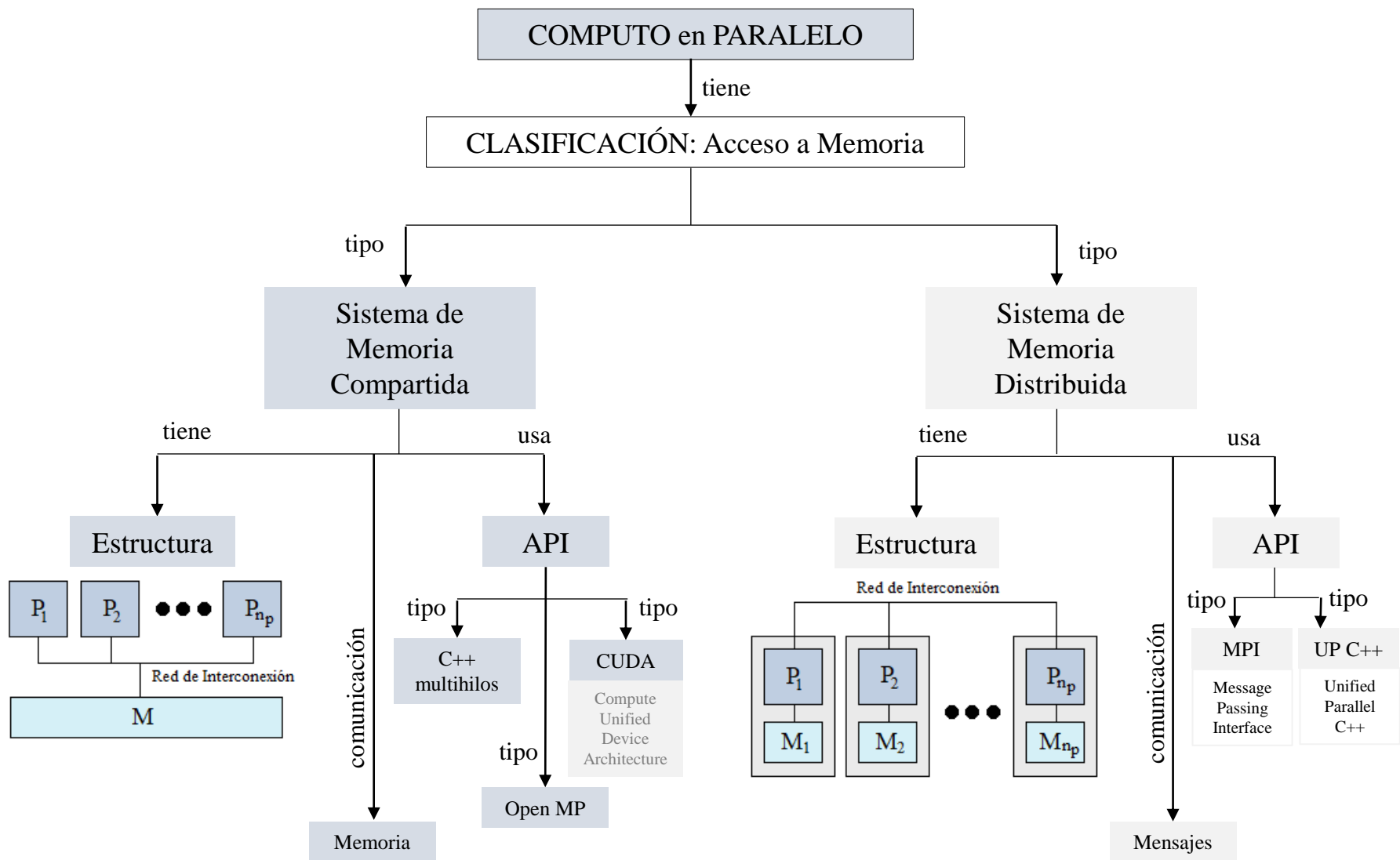
COMPUTO en PARALELO

tiene

CLASIFICACIÓN: Procesamiento (Taxonomía de Flynn)

es





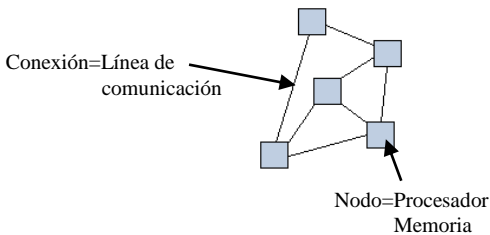
NOTA: P_i = i -esima Unidad de Procesamiento
 M_i = i -esima Memoria
 np =Numero de Procesadores

RED de INTERCONEXIÓN

es

GRAFO

Conjunto de conexiones entre 2 o más elementos



Determina el nivel de

Paralelización de la comunicación

es

Transferencia simultanea de varios mensajes entre diferentes pares de nodos

definición

Ancho de Banda

Razón a la cual el destino recibe los Datos después de haber recibido el primer byte

Ancho de Banda

$$B = \left\lfloor \frac{n_b}{\Delta t} \right\rfloor$$

Número de bytes en Δt

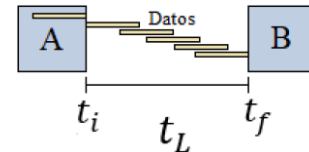
Intervalo de tiempo (s)

definición

t_L =Tiempo de Latencia

es

Tiempo que transcurre entre la transmisión del primer byte de los Datos en A al tiempo t_i hasta la recepción del primer byte de los Datos en B al tiempo t_f

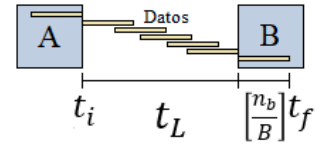


definición

t_T =Tiempo de Transmisión de n_b bytes

es

Tiempo que transcurre entre la transmisión del primer byte de los Datos en A al tiempo t_i hasta la recepción del ultimo byte de los Datos en B al tiempo t_f



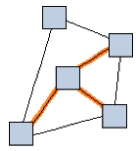
$$t_T = t_L + \left\lfloor \frac{n_b}{B} \right\rfloor$$

definición

N_G =Grado

es

Número máximo de nodos vecinos

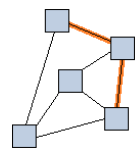


definición

N_D =Diámetro

es

La máxima distancia de entre todas las distancias mínimas entre pares de nodos

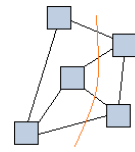


definición

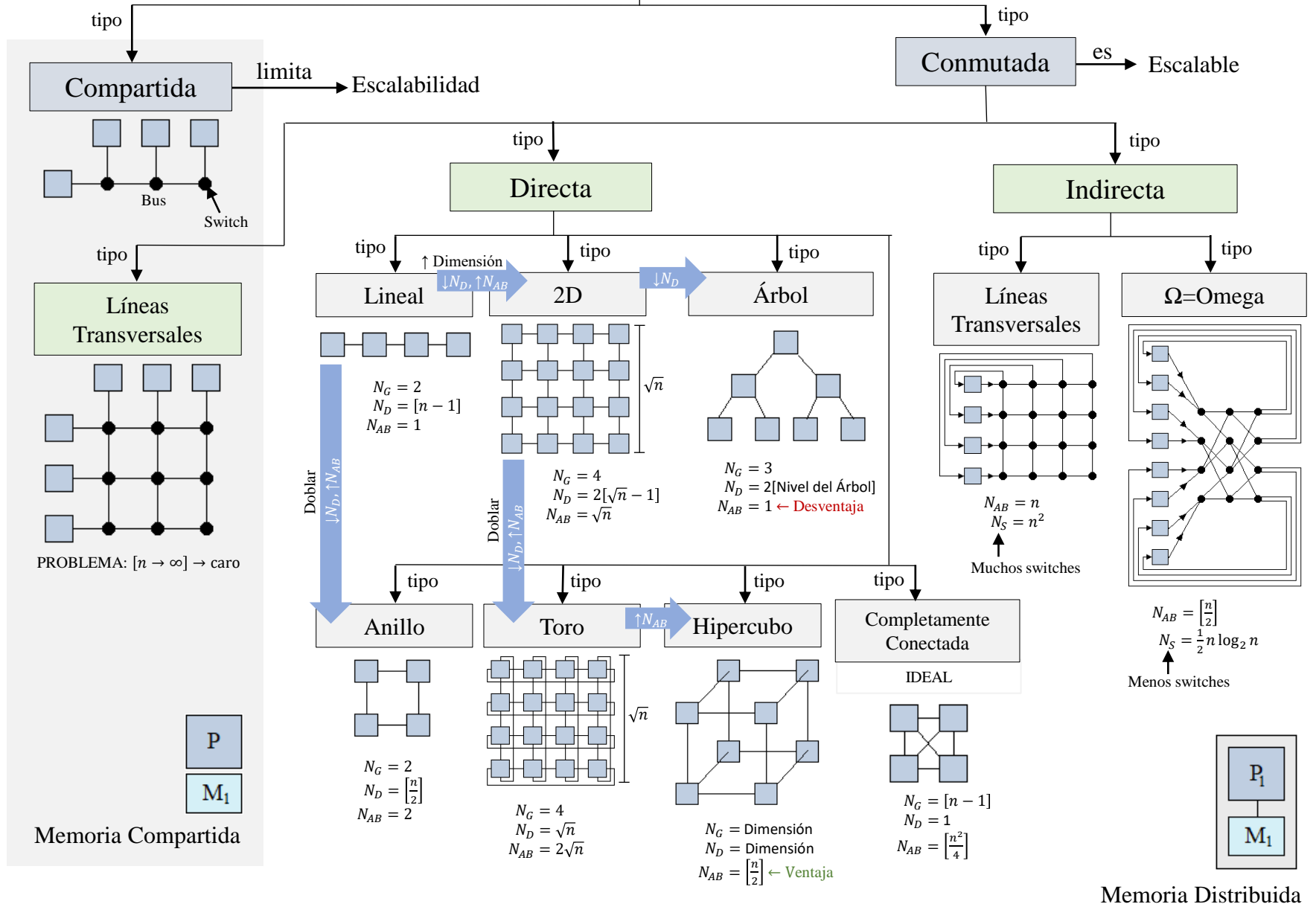
N_{AB} =Ancho de Bisección

es

Número mínimo de aristas a remover para dividir la red en 2 partes iguales



RED de INTERCONEXIÓN



COMPUTO en PARALELO

tiene

PROGRAMAS: Desempeño

definición

$t(n_p, n)$ = Tiempo de Ejecución del Programa

No. de Procesadores

Tamaño del Problema

definición

S = Speedup

t_s = Tiempo en serie

$$S(n_p, n) = \left[\frac{t(1, n)}{t(n_p, n)} \right]$$

t_p = Tiempo en paralelo

definición

Escalabilidad

tipo

Fuerte

es

$E(n_p, n) = \text{cte}$
cuando: $n_p \rightarrow \infty$
 $n = \text{cte}$

tipo

Débil

es

$E(n_p, n) = \text{cte}$
cuando: $n_p \rightarrow \infty$
 $n \rightarrow n_p n$

definición

E = Eficiencia

$$E(n_p, n) = \left[\frac{S}{n_p} \right]$$

definición

C = Costo

$$C(n_p, n) = n_p t(n_p, n)$$

definición

$R_{C \rightarrow C}$ = Razón de Computo a Comunicación

$$R_{C \rightarrow C} = \left[\frac{t_{\text{cálculo}}}{t_{\text{comunicación}}} \right]$$

COMPUTO en PARALELO

tiene

PROGRAMAS: Desempeño

definición

Tiempo de Ejecución
Ideal en Paralelo

$$t_p^{\text{Ideal}} = \left\lfloor \frac{t_s}{n_p} \right\rfloor$$

definición

Tiempo de Sobrecarga

$$t_{sc} = [t_p - t_p^{\text{Ideal}}]$$

en

Memoria
Compartida

es

Secciones Críticas

$$\uparrow [\text{secciones críticas}] \rightarrow \uparrow [t_{sc}]$$

en

Memoria
Distribuida

es

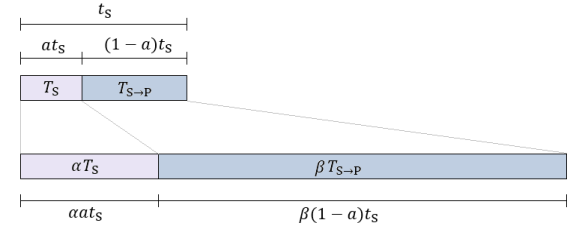
Comunicación

$$\uparrow [\text{Datos}] \rightarrow \uparrow [t_{sc}]$$

definición

Ley General

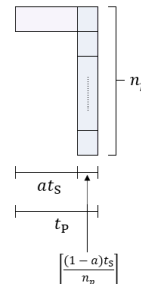
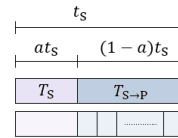
$$S = \left[\frac{a + \frac{\beta}{\alpha} [1 - a]}{a + \frac{\beta}{\alpha} \left\lfloor \frac{1 - a}{n_p} \right\rfloor} \right]$$



$$\left\lfloor \frac{\beta}{\alpha} \right\rfloor = 1$$

Ley de Amdahl

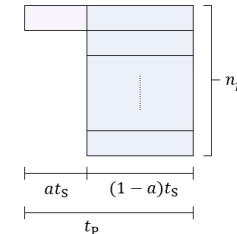
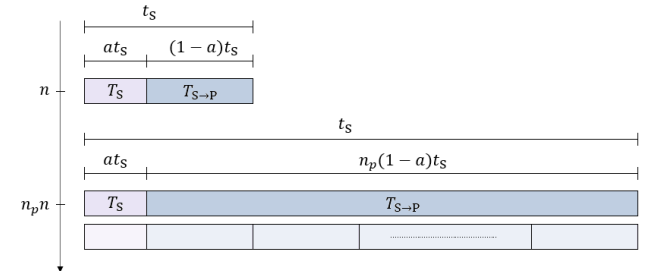
$$S = \left[\frac{1}{a + \left\lfloor \frac{1 - a}{n_p} \right\rfloor} \right]$$



$$\left\lfloor \frac{\beta}{\alpha} \right\rfloor = n_p$$

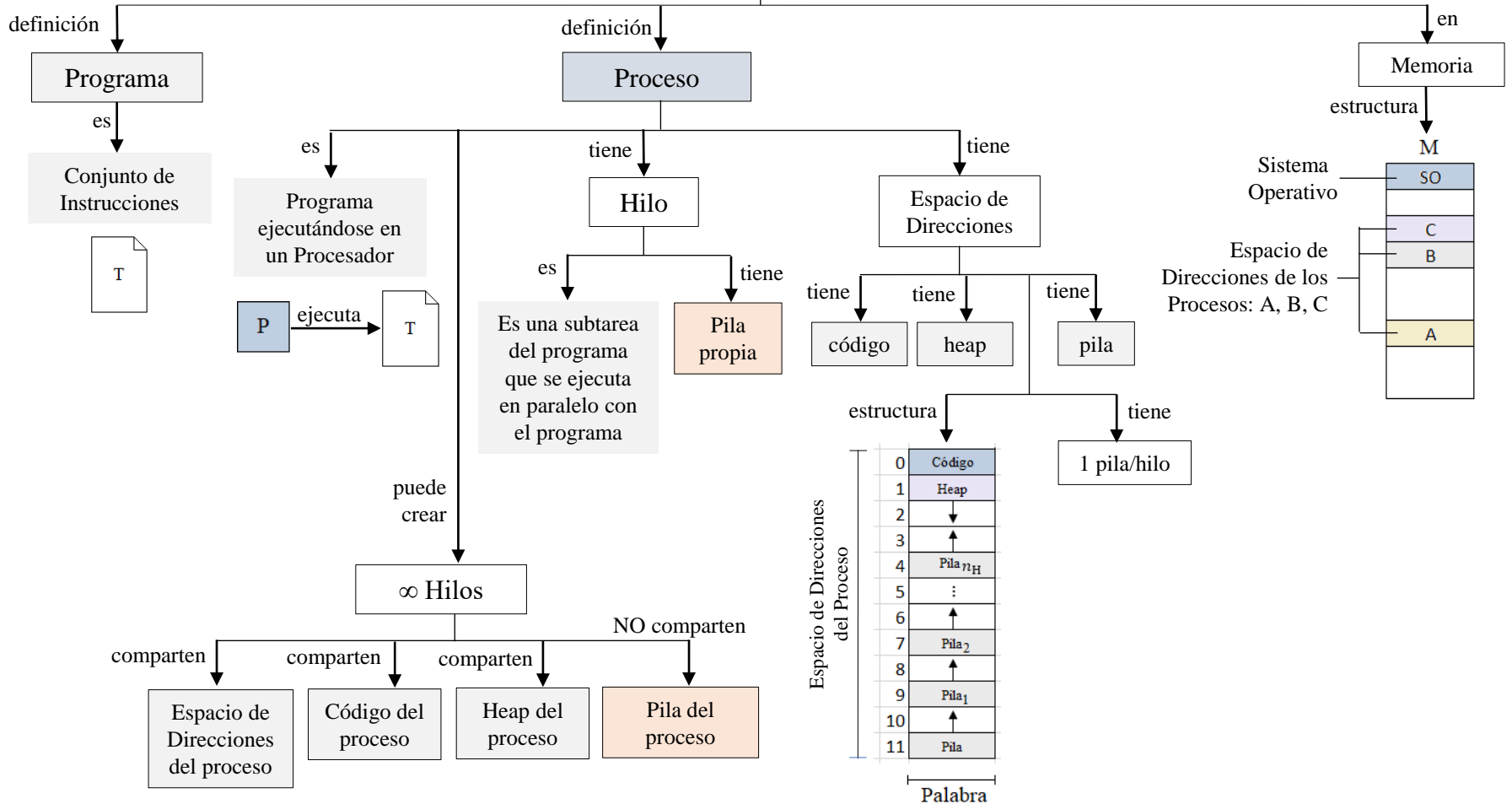
Ley de Gustafson

$$S = [1 - a]n_p + a$$



COMPUTO en PARALELO

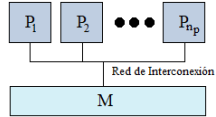
PROGRAMAS: Estructura



COMPUTO en PARALELO

tiene

PROGRAMAS: Estructura



en

Memoria Compartida

crea

1 Proceso
 ∞ Hilos

porque

Todos los procesadores
 P_i ejecutan un solo SO
compartido

comunicación

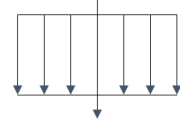
A través de
la Memoria

tiene

Hilos

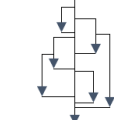
tipo

Estáticos



tipo

Dinámicos



problema

Sección Crítica

es

Pedazo de código

accesa

Recurso
Compartido

da origen

Condición
de Carrera

es

∞ procesadores P_i
modifican la misma
localidad de memoria
a la vez

evita

Inconsistencia

da

Exclusión
Mutua

garantiza

1 solo procesador P_i
accesa a la sección
crítica a la vez.

aumenta

Sección $\rightarrow 0$
Crítica

disminuye

Serialización

en

Memoria Distribuida

crea

∞ Procesos

porque

Cada procesador
 P_i ejecuta un SO
independiente

comunicación

Paso de
Mensajes

se puede
ver como

Sistema de Memoria
Compartida

Si se usa un SO distribuido
entre todos los procesadores
 P_i entonces se puede
crear: 1 Proceso distribuido
 ∞ Hilos distribuidos

ideal

No. de Hilos = n_p

0 n_p $n_H = \text{No. de Hilos}$

ideal

1 Hilo/Procesador

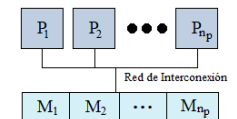
problema

∞ Cambios de Contexto
SO serializa la ejecución de los hilos

se puede
ver como

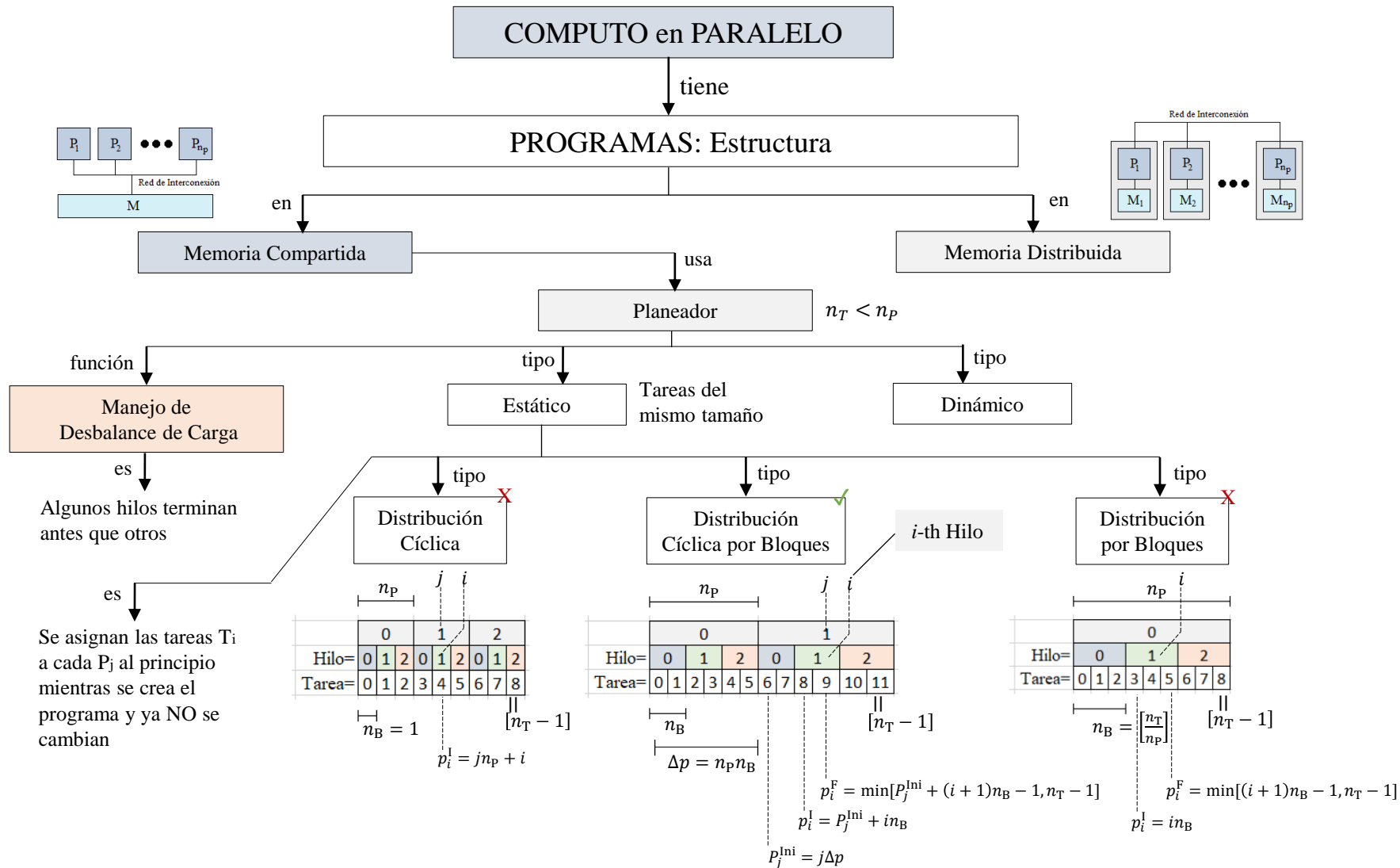
Sistema de Memoria
Distribuida

Si la Memoria se parte y se
asigna a cada procesador P_i



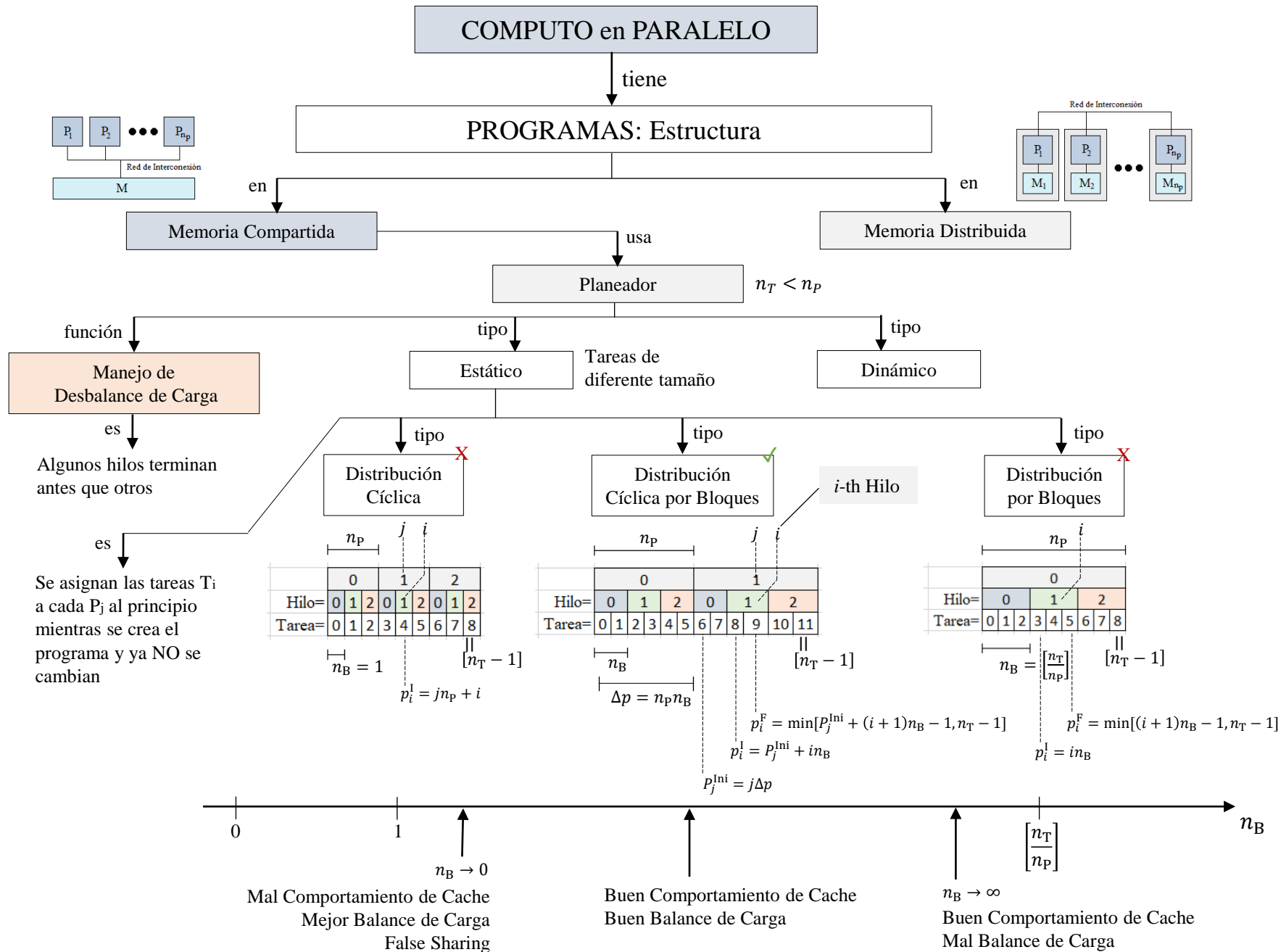
Definiciones:

N_{T_i} = Tamaño de T_i



Definiciones:

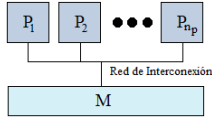
N_{T_i} = Tamaño de T_i



COMPUTO en PARALELO

tiene

PROGRAMAS: Estructura

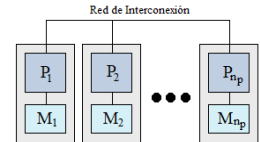


en

Memoria Compartida

usa

Planeador



en

Memoria Distribuida

función

Manejo de Desbalance de Carga

es

Algunos hilos terminan antes que otros

es

Se van asignando las tareas T_i a cada P_j según vayan terminando su tarea asignada mientras el programa ejecuta

tipo

Estático

tipo

Dinámico

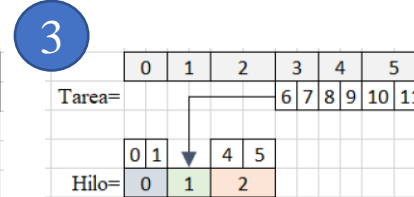
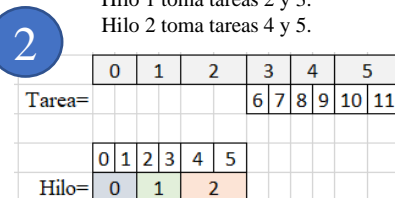
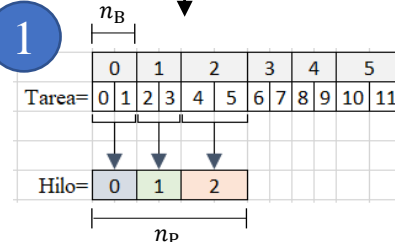
Tareas de diferente tamaño

uso

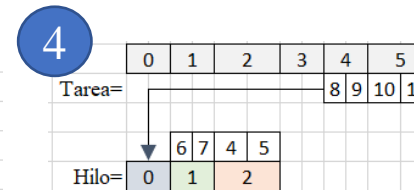
Cuando hay desbalance de carga que NO se conoce el patrón o es muy complejo

$n_B \rightarrow 0$

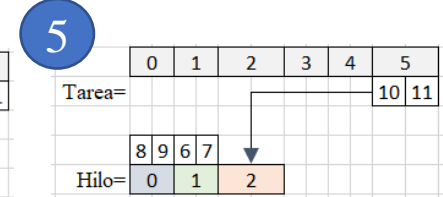
Mejor Balance de Carga



Cuando un hilo termina sus tareas toma las siguientes n_B tareas disponibles en la cola.
Hilo 0 terminó y toma las tareas 6 y 7.
Hilo 1 terminó y toma las tareas 8 y 9.



Hilo 0 terminó y toma las tareas 8 y 9.



Hilo 0 terminó y toma las tareas 10 y 11.

