

Generalidades de los Sistemas Paralelos

CC3069 – Programación Paralela y Distribuida

Ciclo 1 - 2020

Computadoras Modernas

- La *Ley de Moore* predecía el crecimiento de los equipos uniprosesador. *Ya no es válida.*
- A partir del 2002 debemos adoptar *estrategias paralelas para seguir mejorando* el desempeño (performance) y rendimiento (throughput)
- *Toda computadora moderna es paralela* (escritorio, server, móvil, embebida)

Paralelismo en múltiples niveles

Procesador

- Microarquitectura, Pipeline, Múltiple emisión, Predicción y Especulación

Multinúcleos

- UP (unidades de procesamiento) separadas, Memoria compartida, Buses integrados

Clusters

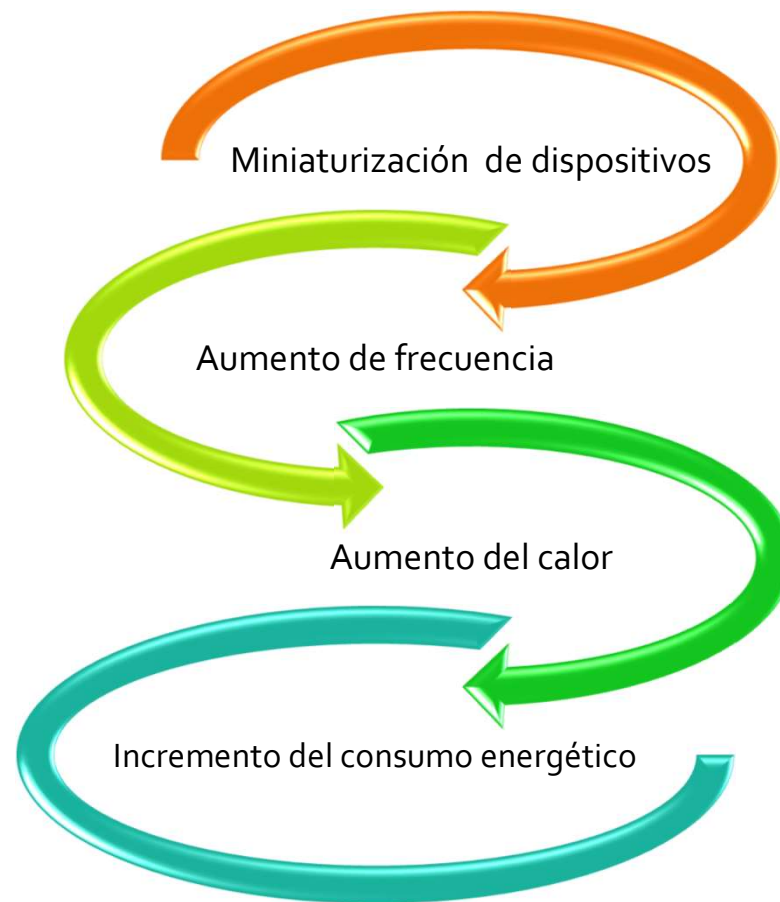
- Equipos multinúcleo con interconexión, Alto desempeño; Entorno Empresarial y científico

GPUs

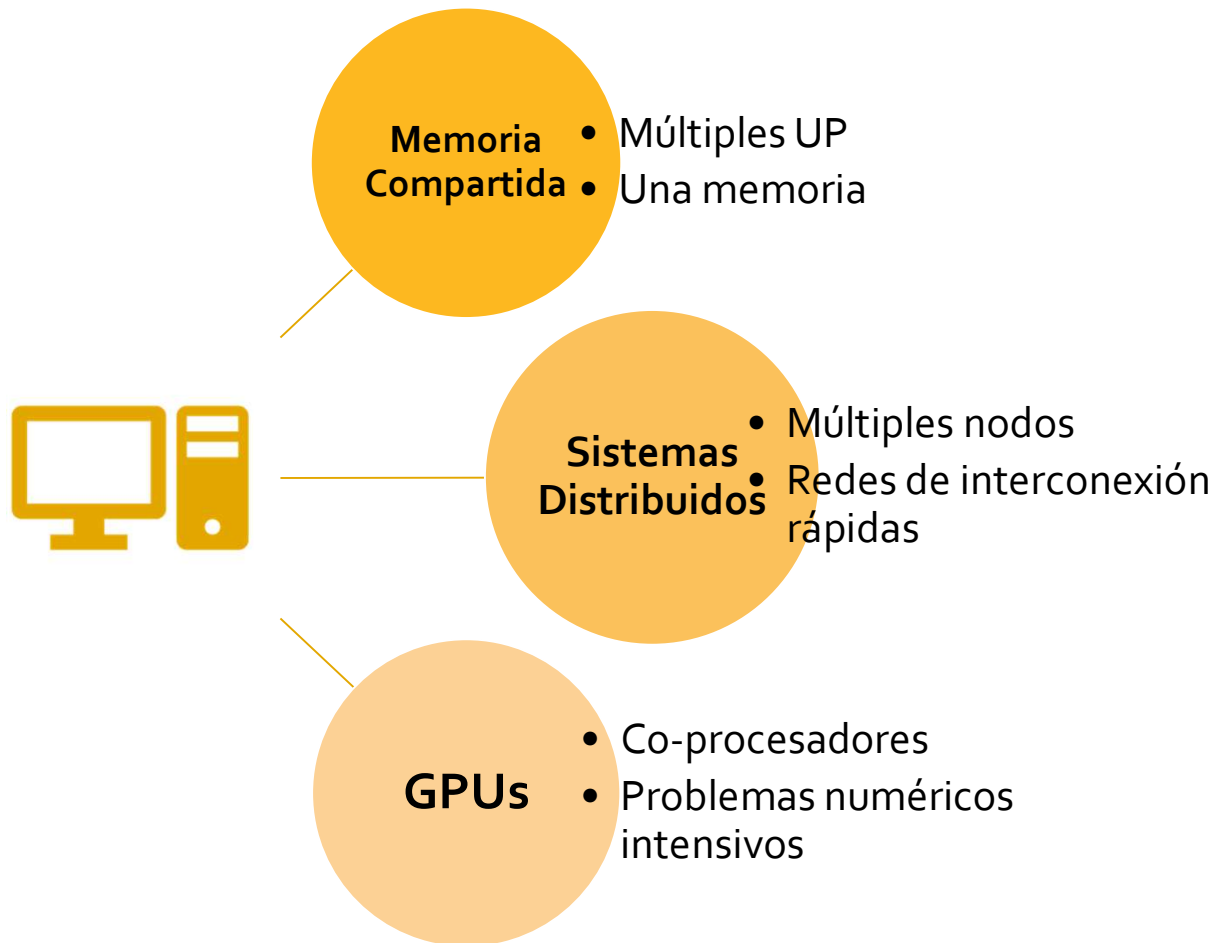
- SIMD, a nivel de equipos de consumo (usuario)

USUARIO

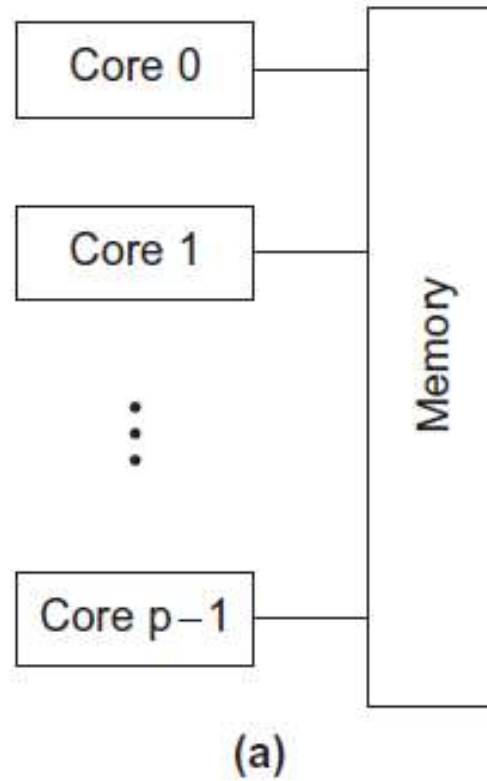
El muro de potencia



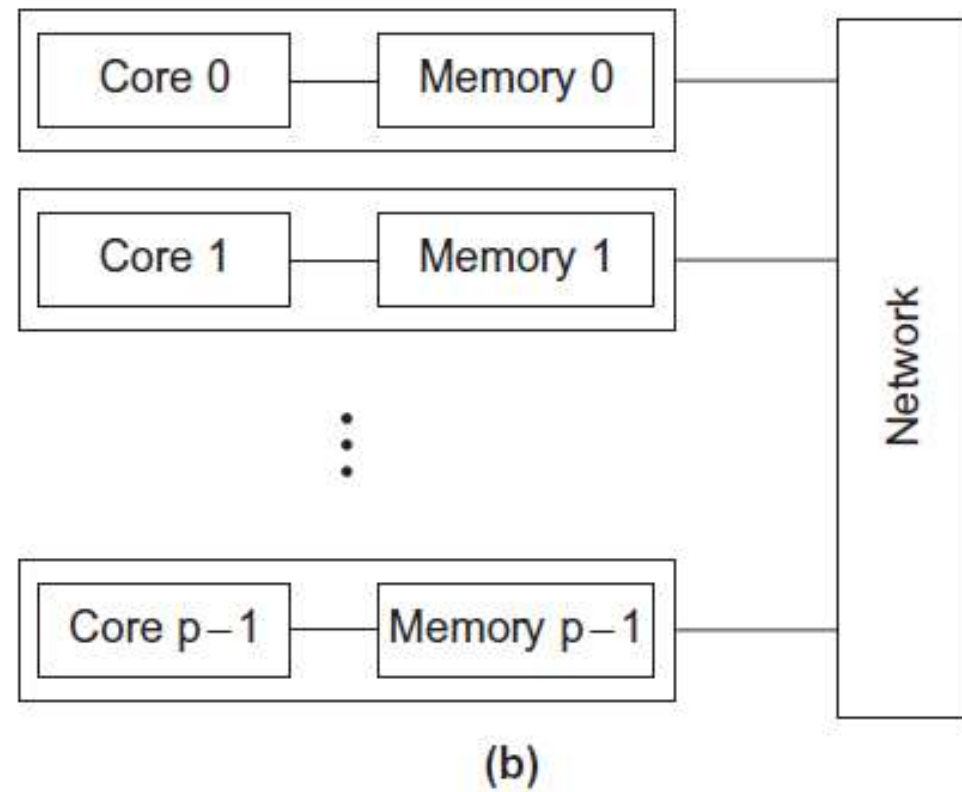
Tipos de Sistemas Paralelos



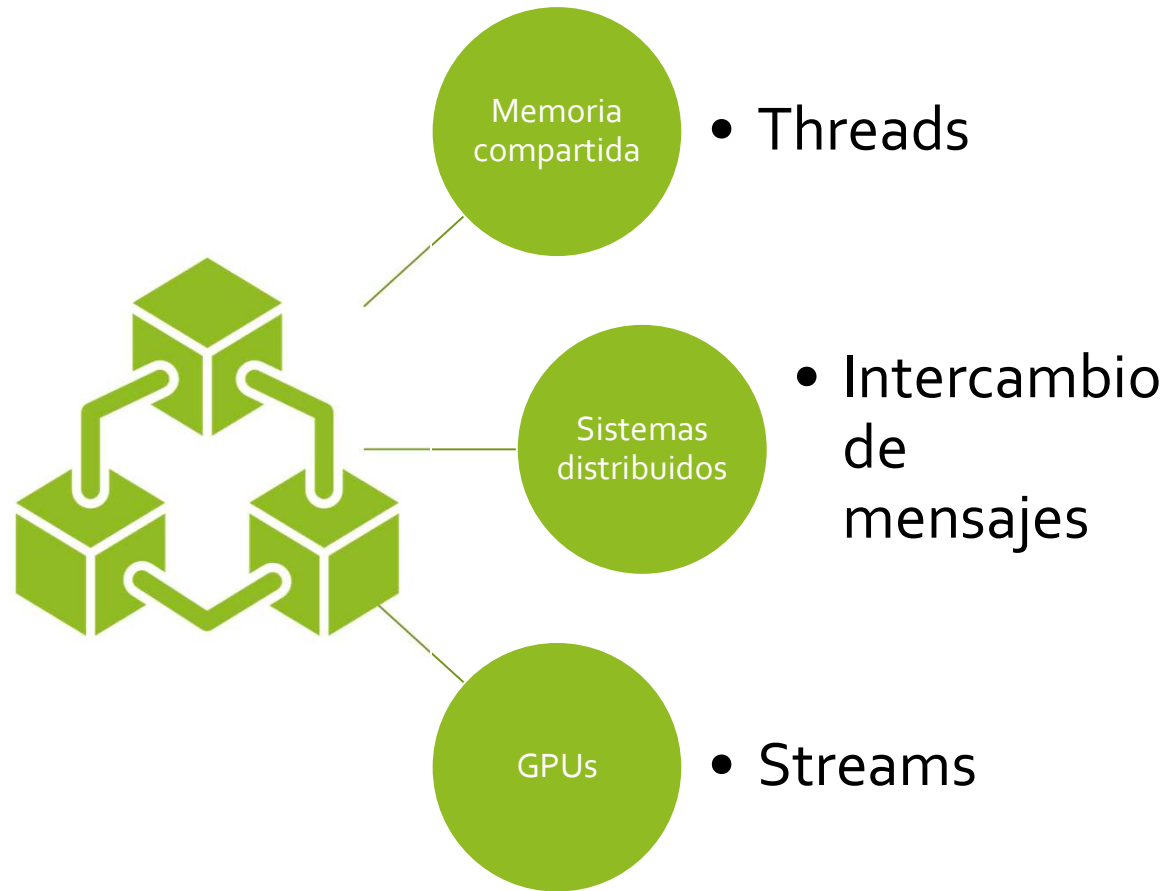
Sistemas de Memoria Compartida



Sistemas de Memoria Distribuida



Estrategias para lograr paralelismo



Estrategias generales de paralelismo en software

Repartición de Datos

- Data-parallel
- División de dominio

Separación de Tareas

- Task-parallel

Rediseño del Algoritmo

- Método de Foster
- Algoritmo Paralelo

Acumulación de valores consecutivos

```
sum = 0;
for (i=0; i<n; i++) {
    x=siguiente_valor(...);
    suma += x;
}
```

- Calcular n valores y sumarlos (serial)
- Podemos dividir el dominio de datos entre diferentes UPs

Acumulación de valores consecutivos

```
mi_sum = 0;
mi_first_i = ...;
mi_last_i = ...;
for (mi_i = mi_first_i;
     mi_i < mi_last_i;
     mi_i++) {
    mi_x=siguiente_valor(...);
    mi_suma += mi_x;
}
```

- Cada UP (núcleo) recibe una porción de los datos
- Variables y sumas parciales locales
- Dividimos los n datos dentro de p UPs

Acumulación de valores consecutivos

- División de dominio:

Core	0	1	2	3	4	5	6	7
my_sum	1,4,3	9,2,8	5,1,1	6,2,7	2,5,0	4,1,8	6,5,1	2,3,9

- Sumas parciales:

Core	0	1	2	3	4	5	6	7
my_sum	8	19	7	15	7	13	12	14

Acumulación de valores consecutivos

```
If(soy_maestro){  
    suma = mi_x;  
    for each core <> maestro {  
        obtener suma_parcial;  
        suma += suma_parcial;  
    }  
} else {  
    send mi_x a maestro;  
}
```

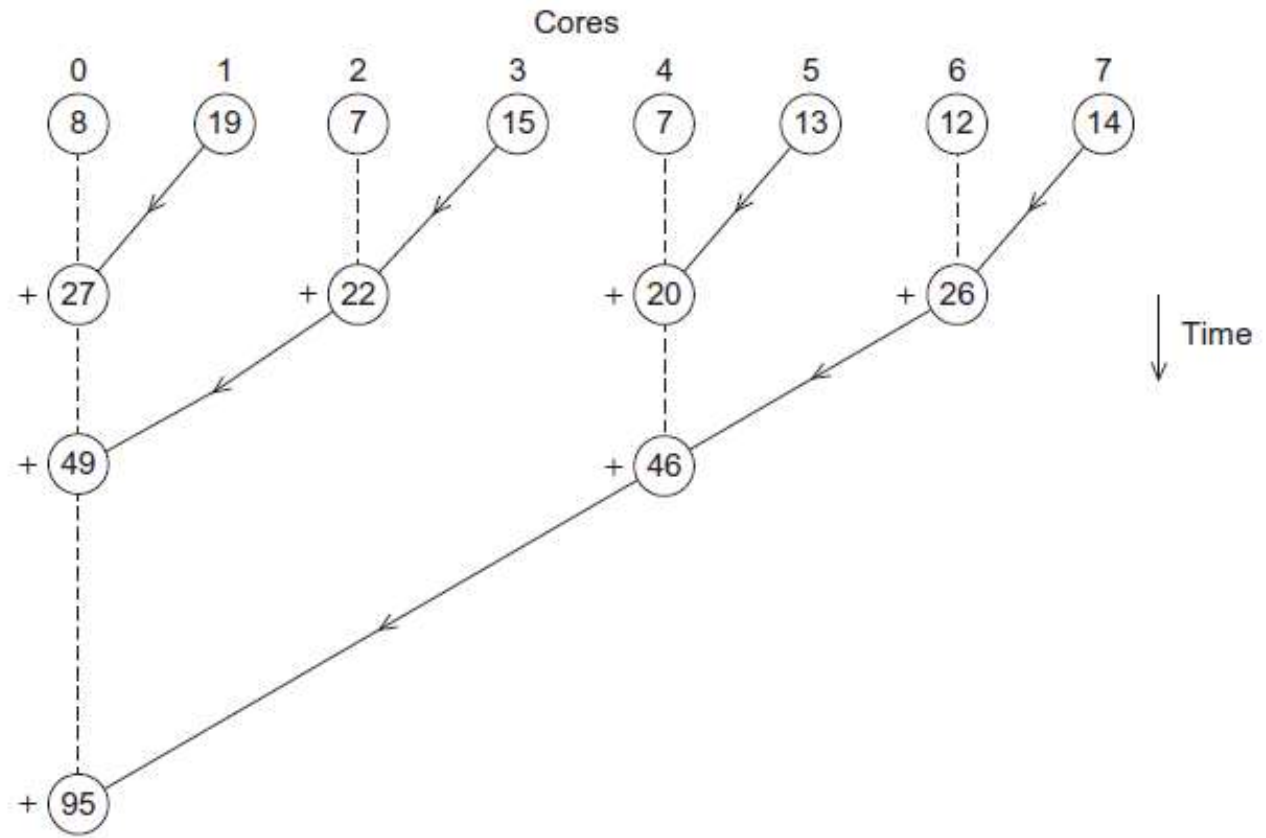
- Diseño 1 UP como maestra para acumular todos las sumar parciales
- Muchas llamadas a 1 sola UP.

Acumulación de valores consecutivos

- ANALISIS
 - La UP maestra recibe 7 llamadas y realiza 8 sumas

Acumulación
de valores
consecutivos

MEJORA



Acumulación de valores consecutivos

MEJORA

- Repartimos el trabajo hacia otras UP
- Trabajan en parejas (par-impar)
 - Core 0 suma valores del core 1
 - Core 2 suma valores del core 3
 - Core 4 suma valores del core 5 ...
- Repetimos en el siguiente nivel
 - Core 0 suma acumulado del core 2
 - Core 4 suma acumulado del core 6
- En el ultimo paso, el core 0 suma la mitad de valores acumulados en el core 4
 - En total realiza 3 operaciones de comunicación (3 niveles)

Acumulación de valores consecutivos

LOS NÚMEROS IMPORTAN

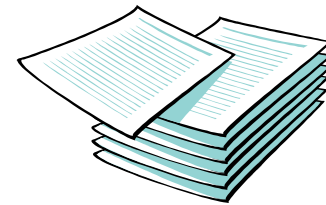
- A mayor número de UPs, mayor ganancia en la reducción de la comunicación inter-UP/proceso.
- Con 1000 Ups:
 - Acumulación en UP maestra = 999 mensajes recibidos
 - Acumulación por parejas = 10 mensajes recibidos máximo en la UP o
- Mejora relativa = $999 / 10 \approx \mathbf{100}$

División de Tareas

30 preguntas
300 exámenes

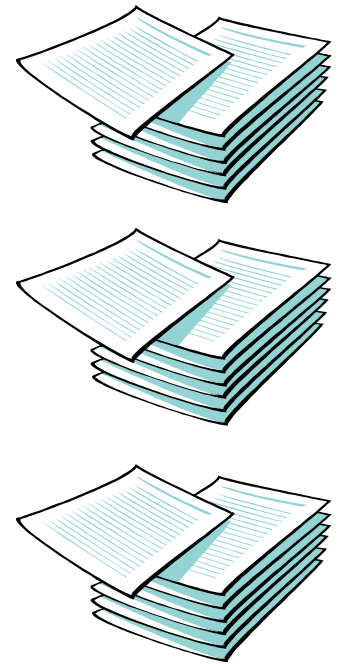


3 auxiliares



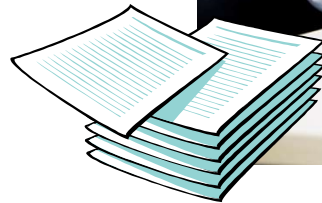
División de Tareas

División de Datos – 100 exámenes cada uno

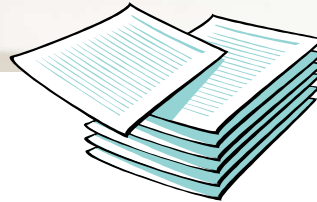


División de Tareas

División de Tareas – 10 preguntas cada uno



Preguntas 1 - 10



Preguntas 11 - 20



Preguntas 20 - 30

Ambas estrategias mezcladas

¿Dónde?

```
sum = 0;
for (i=0; i<n; i++) {
    x=siguiente_valor(...);
    suma += x;
}
```

Data-parallel

```
If(soy_maestro){
    suma = mi_x;
    for each core <> maestro {
        obtener suma_parcial;
        suma += suma_parcial;
    }
} else {
    send mi_x a maestro;
}
```

**Task-parallel
(Recibir,
sumar)**

Coordinación de UPs

Es común que tengamos que mantener coordinación del procesamiento y la carga de cómputo:

COMUNICACIÓN

- Intercambio de resultados parciales

BALANCE DE CARGA

- Aprovechar los recursos de forma equitativa. Ocupar las UPs al mismo tiempo

SINCRONÍA

- No podemos asumir las mismas condiciones en cada UP. El mismo proceso correrá diferente cada vez aún en la misma UP.