

Arquitectura de Computadores. Tema 1

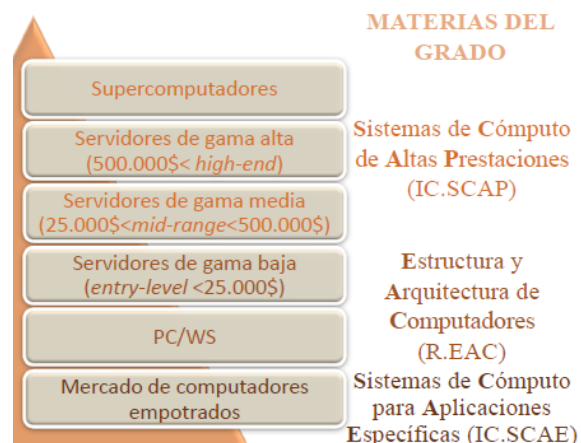
Arquitecturas paralelas: clasificación y prestaciones

Lección2: Clasificación de arquitecturas paralela

- Computación paralela: estudia los aspectos hardware y software relacionados con el desarrollo y ejecución de aplicaciones en un sistema de cómputo compuesto por múltiples cores/procesadores/computadores que es visto externamente como unidad autónoma (multicores, multiprocesadores, cluster)
- Computación distribuida: estudia los aspectos hardware y software relacionados con el desarrollo y ejecución de aplicaciones en un sistema distribuido; es decir, una colección de recursos autónomos (PC, servidores –de datos, aplicaciones,...-, supercomputadores...) situados en distintas localizaciones físicas
- Computación distribuida baja escala: estudia los aspectos relacionados con el desarrollo y ejecución de aplicaciones en una colección de recursos autónomos de un dominio administrativo situados en distintas localizaciones físicas conectados a través de infraestructura de red local
- Computación grid: estudia los aspectos relacionados con el desarrollo y ejecución de aplicaciones en una colección de recursos autónomos de múltiples dominios administrativos geográficamente distribuidos conectados como infraestructura de telecomunicaciones
- Computación cloud: comprende los aspectos relacionados con el desarrollo y ejecución de aplicaciones en un sistema cloud
- Sistema cloud: ofrece servicios de infraestructura, plataforma y/o software, por los que se paga cuando ose necesitan (pay-per-use) y a los que se accede típicamente a través de una interfaz (web) de auto-servicio. Consta de recursos virtuales que:
 - Son una abstracción de los recursos físicos
 - Parecen ilimitados en número y capacidad y son reclutados/liberados de forma inmediata sin interacción con el proveedor
 - Soportan el acceso de múltiples clientes (multi-tenant)
 - Están conectados con métodos estándar independientes de la plataforma de acceso

Criterios de clasificación de computadores

- Segmento del mercado



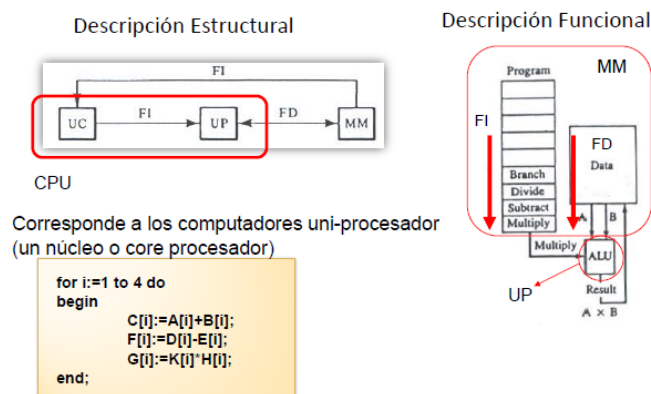
Clasificación de computadores según segmento



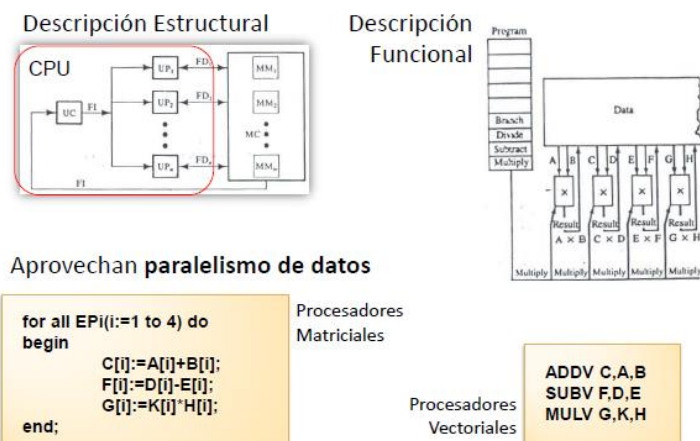
Clasificación de Flynn de arquitecturas, 1972 (Flujo instrucciones/flujo de datos)

Flynn divide el universo de computadores en cuatro clases, según el número de secuencias o flujos de instrucciones y secuencias o flujos de datos que pueden procesarse simultáneamente en el computador:

- Computadores SISD: único flujo de instrucciones (SI, single instruction) procesa operandos y genera resultados, definiendo un único flujo de datos (SD, single data). Un núcleo procesador.

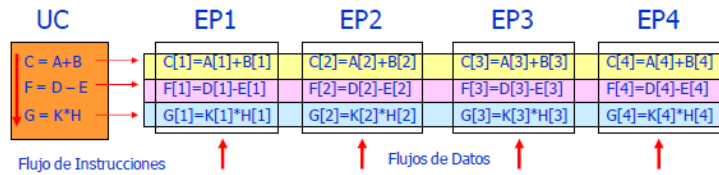


- Computadores SIMD: un único flujo de instrucciones (SI) procesa operando y genera resultados, definiendo varios flujos de datos (MD, multiple data), dado que cada instrucción codifica realmente varias operaciones iguales, cada una actuando sobre operandos distintos.

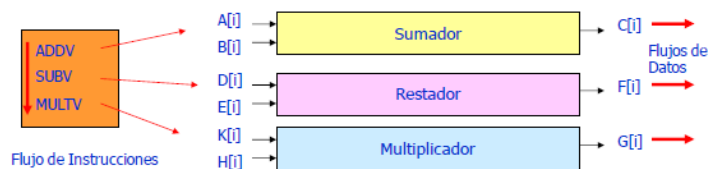


Ejemplo SIMD

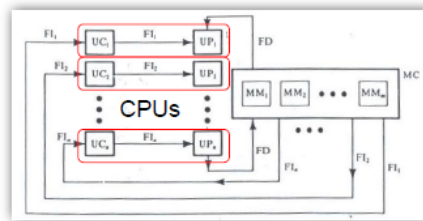
Procesador Matricial



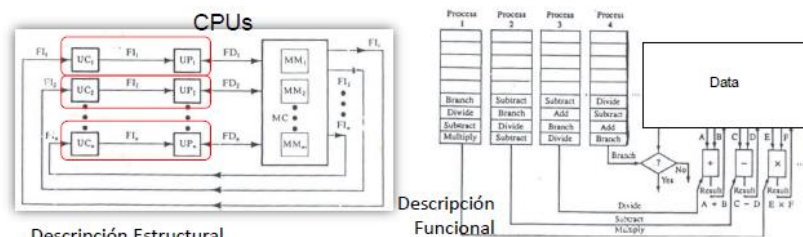
Procesador Vectorial



- Computadores MISD: se ejecutan varios flujos distintos de instrucciones (Multiple instructions, MI) aunque todos actúan sobre el mismo flujo de datos (SD). (gpu, procesadores matriciales)



- No existen computadores que funcionen según este modelo
- Se puede simular en un código este modelo para aplicaciones que procesan una secuencia o flujo de datos
- Computadores MIMD: el computador ejecuta varias secuencias o flujos distintos de instrucciones (MI), y cada uno de ellos procesa operandos y genera resultados definiendo un único flujo de instrucciones, de forma es que existen también varios flujos de datos (MD) uno por cada flujo de instrucciones.



Corresponde con Multinúcleos, Multiprocesadores y Multicomputadores: Puede aprovechar, además, **paralelismo funcional**

```
for i:=1 to 4 do
begin
  C[i]:=A[i]+B[i];
end;
```

Proc 1

```
for i:=1 to 4 do
begin
  F[i]:=D[i]-E[i];
end;
```

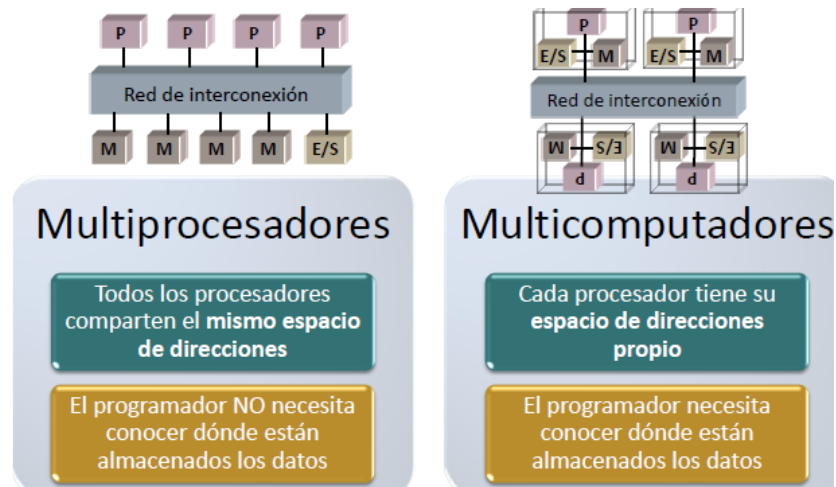
Proc 2

```
for i:=1 to 4 do
begin
  G[i]:=K[i]*H[i];
end;
```

Proc 3

Clasificación de Computadores Paralelos

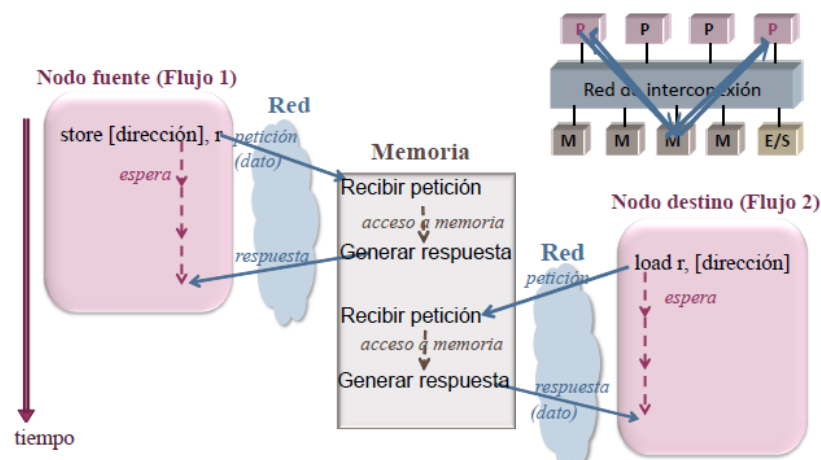
MIMD según el sistema de memoria



Comparativa SMP (Symmetric MultiProcessor) y multicomputadores

- Multiprocesador con memoria centralizada (SMP)
 - Mayor latencia – poco escalable
 - Comunicación implícita mediante variables compartidas. Datos no duplicados en MP
 - Necesita implementar primitivas de sincronización
 - Distribución código y datos entre procesadores: no necesaria
 - Programación más sencilla
- Multicomputador
 - Menor latencia – más escalable
 - Comunicación explícita mediante software para paso de mensajes (send/receive). Datos duplicados en MP, copia datos
 - Sincronización mediante software de comunicación
 - Distribución código y datos entre procesadores: necesaria=> programas más sofisticados
 - Programación más difícil

Comunicación 1-a-1 en un multiprocesador



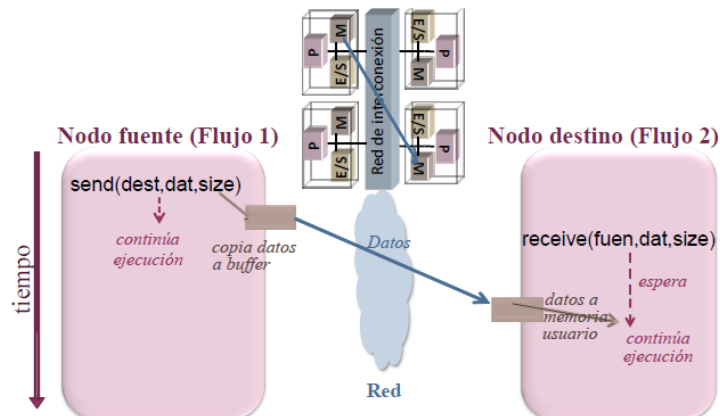
Secuencial	Paralelo	
...	F1	F2
A=valor;
...	A=valor;	copia=A;
copia=A;
...

F1 es el flujo de control productor del dato (envía el dato)
F2 es el flujo de control consumidor del dato (recibe el dato)

- Se debe garantizar que el flujo de control consumidor del dato lea la variable compartida (A) cuando el productor haya escrito en la variable el dato

Paralelo multiproc. (K=0)	
F1	F2
...	...
A=valor;	while (K=0) { };
K=1;	copia=A;
...	...

Computación 1-a-1 en multicomputador (receive bloqueante)



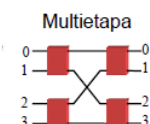
Secuencial	Paralelo	
...	F1	F2
A=valor;
...	A=valor;	copia=A;
copia=A;
...

F1 es el flujo de control productor del dato (envía el dato)
F2 es el flujo de control consumidor del dato (recibe el dato)

Paralelo multicomputador (size = 4 byte)	
F1	F2
...	...
send(F2, valor, 4);	receive(F1,copia,4);
...	...

Incremento de escalabilidad en multiprocesadores y red de interconexión

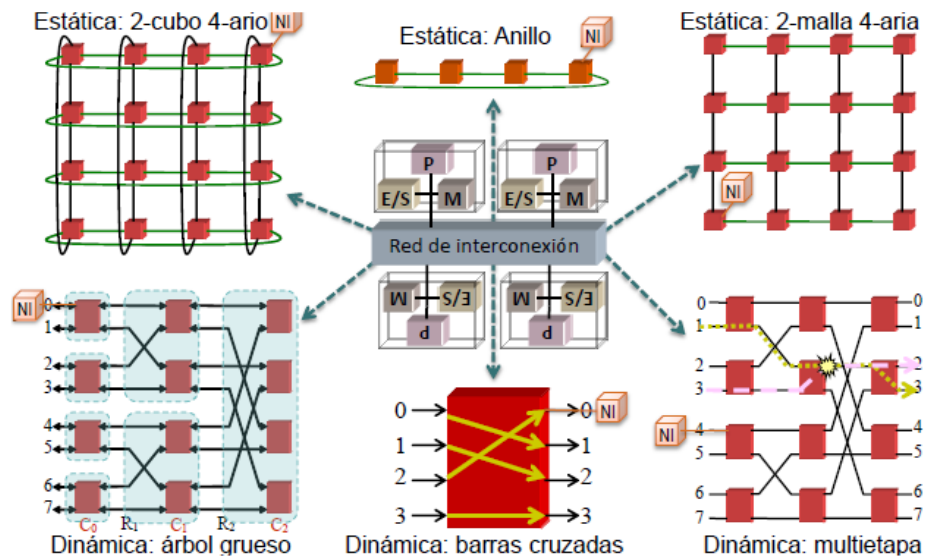
- Incremento escalabilidad multiprocesadores:
- Aumentar caché del procesador
 - Usar redes de menor latencia y mayor ancho de banda que un bus (jerarquía de buses, barras cruzadas, multietapa)
 - Distribuir físicamente los módulos de memoria entre los procesadores (pero se sigue compartiendo espacio de direcciones)



Clasificación completa de computadores según el sistema de memoria

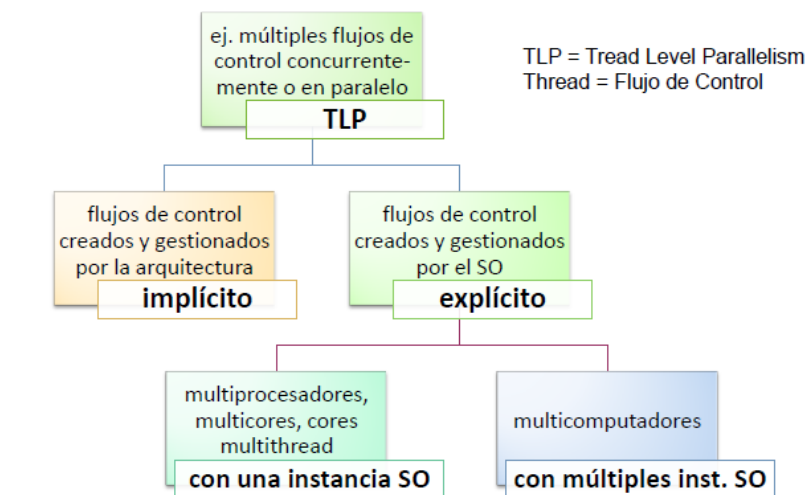
Multi-computadores Memoria no compartida	NORMA <i>No Remote Memory Access</i>	<i>ej. cluster, red de computadores</i>	Memoria físicamente distribuida	+	+
Multi-procesadores Memoria compartida Un único espacio de direcciones	NUMA <i>Non-Uniform Memory Access</i>	NUMA		Escalabilidad	+
		CC-NUMA			
		COMA			
	UMA <i>Uniform Memory Access</i>	SMP <i>Symmetric MultiProcessor</i>	Memoria físicamente centralizada	-	-

Red en sistemas con memoria físicamente distribuida (NI: Network interface)



Ejemplo: Placa CC-NUMA con red estática

Propuesta clasificación arquitecturas con múltiples flujos de control o threads



Arquitecturas con DLP, ILP Y TLP (thread=flujo de control)

Arq. con DLP (Data Level Parallelism)	Arq. con ILP (Instruction Level Parallelism)	Arq. con TLP (Thread Level Parallelism) explícito y una instancia de SO	Arq. con TLP explícito y múltiples instancias SO
Tema 5 Ejecutan las operaciones de una instrucción concurr. o en paralelo	Tema 4 Ejecutan múltiples instrucciones concurr. o en paralelo	Temas 3, 5 Ejecutan múltiples flujos de control concurr. o en paralelo	IC SCAP Ejec. múltiples flujos de control en paralelo
Unidades funcionales vectoriales o SIMD	Cores escalares segmentados, superescalares o VLIW/EPIC	Cores que modifican la archit. escalar segmentada, superescalar o VLIW/EPIC para ejecutar threads concurr. o en paralelo	Multi-procesadores: ejecutan threads en paralelo en un computador con múltiples cores (incluye multicores)
			Multi-computadores: ejecutan threads en paralelo en un sistema con múltiples computadores