WUOLAH

Perfil7
www.wuolah.com/student/Perfil7

413

ApuntesTEMA1AC.pdf

APUNTES con diapositivas y libro

2° Arquitectura de Computadores

Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

- ARQUITECTURA DE COMPUTADORES -

Tema 1

Lección 1

PARALELISMO DE DATOS

Para que dos bloques de código presenten dependencia de datos, se deben dar dos condiciones:

1) Deben hacer referencia a una misma posición de memoria M (variable)
2) El bloque B1 aparece antes que B2 en la secuencia de código

Sabiendo qué es la dependencia de datos, vamos a distinguir sus tipos (B2 respecto a B1):

- **RAW** → Read After Write, o dependencia verdadera

```
a = b+c
d = a+c
```

- **WAW** → Write After Write, o dependencia de salida

```
a = b+c
a= d+e
```

- **WAR** → Write Afer Read, o antidependencia

```
b = a+1
a = d+e
```

Nemotécnico:
Ej.: RAW
Primero W escribes
Después R lees
(Leer al revés)

CRITERIOS DE CALIFICACIÓN DE PARALELISMO IMPLÍCITO

Existen 3 tipos de criterios de clasificación del paralelismo implícito de una aplicación:

- **Niveles de paralelismo implícito en una aplicación**
- **Granularidad**

Niveles	Ejemplos			Granularidad
Programas	Programa 1	Programa 2		Grano Grueso
Funciones	Func1()	Func2()	Func3()	Grano Medio
Bucle	For()	While()		Grano Medio-Fino
Operaciones	*	+	/	Grano Fino

- **Paralelismo de tareas / Paralelismo de datos**

  Paralelismo de tareas / Task level parallelism (TLP) → Cada procesador ejecuta una tarea

  Lo cual se encuentra extrayendo la estructura lógica de funciones de una aplicación

  Relacionado con el paralelismo a nivel de FUNCIÓN

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Primera academia especializada en estudios de la Facultad de Informática UCM

academia@mathsinformática.com

C/Andrés Mellado, 88 duplicado

www.mathsinformatica.com

academia.maths

91 399 45 49

615 29 80 22

PROFESORES TITULADOS

MÁS DE 30 AÑOS DE EXPERIENCIA

Máximo 16 alumnos por grupo

Plazas limitadas

Maths informática

Paralelismo de datos / Data level parallelism (DLP) → Cada procesador ejecuta la misma tarea sobre diferentes conjuntos o subregiones de datos

Se encuentra de forma implícita en las operaciones con estructuras de datos (vectores y matrices). Ej.: V1 = V2+V3 → engloba múltiples sumas de escalares

Se puede extraer de la representación matemática de la aplicación

Está relacionado principalmente con el paralelismo a nivel de BUCLE

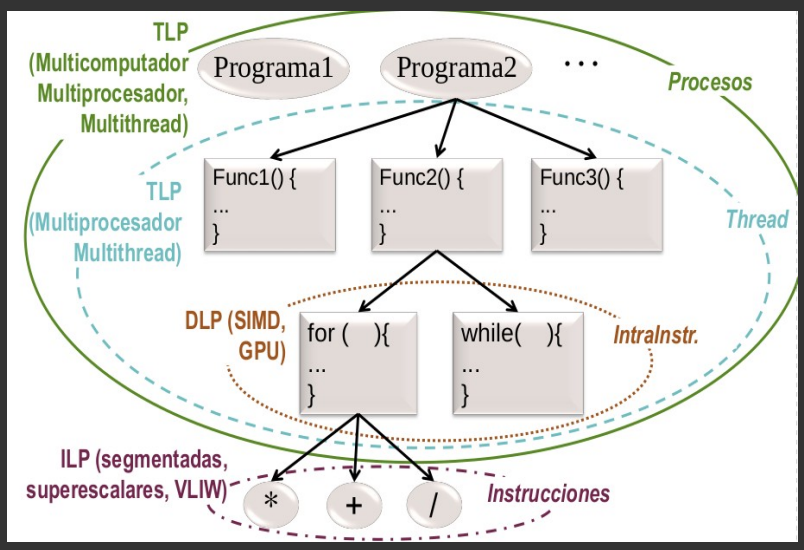

Ilustración 1: Página 11 - T1L1 - Transparencias Mancia Anguita

UNIDADES DE EJECUCIÓN DE UN COMPUTADOR

- **Instrucciones**
  El elemento ejecutado por la unidad de procesamiento y gestionado por la unidad de control
- **Thread o light process (concepto del SO)**
  La menor unidad de ejecución que gestiona el SO
  Por tanto, también es la menor secuencia de instrucciones que se pueden ejecutar de forma concurrente o en paralelo
- **Proceso o process (concepto del SO)**
  Mayor unidad de ejecución que gestiona el SO
  Está formado por uno o más threads

NIVELES DE PARALELISMO EXPLÍCITO

El paralelismo explícito se basa en procesos y threads

Cuando hablamos de PROCESO hacemos referencia a todo el código del programa y todos los datos y recursos necesarios para su ejecución:

-Datos en pila → variables globales y estáticas
-Datos en heap → memoria dinámica
-Contenido de los registros
-Tabla de páginas
-Tabla de ficheros abiertos

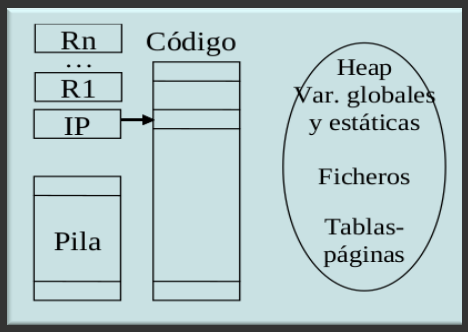

Ilustración 2: Página 13 - T1L1 - Transparencias Mancia Anguita

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Cúelale un troyano al suspenso - Maths Informática

Un proceso puede contener varios flujos de control, los THREADS o procesos ligeros. Estos al igual que los procesos constan de su propio apartado en memoria:

-Su propia pila

-Contenido de los registros, concretamente el de los IP (instruction pointer) y el puntero de pila (stack pointer)

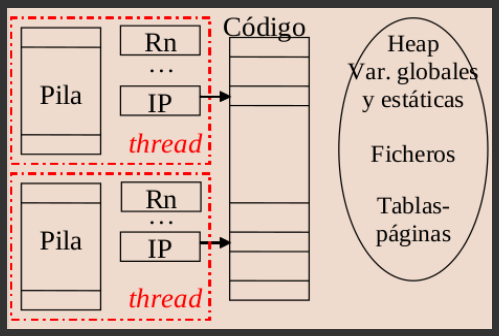

Ilustración 3: Página 13 - T1L1 - Transparencias Mancia Anguita

Una de las diferencias más relevantes entre ambos a tener en cuenta para la eficiencia, es que mientras que la comunicación entre procesos se establece mediante llamadas al sistema, con los threads de un proceso se comunican mediante la propia memoria que comparten

Ilustración 4: Página 14 - T1L1 - Transparencias Mancia Anguita

DETECCIÓN, UTILIZACIÓN, IMPLEMENTACIÓN Y EXTRACCIÓN DEL PARALELISMO

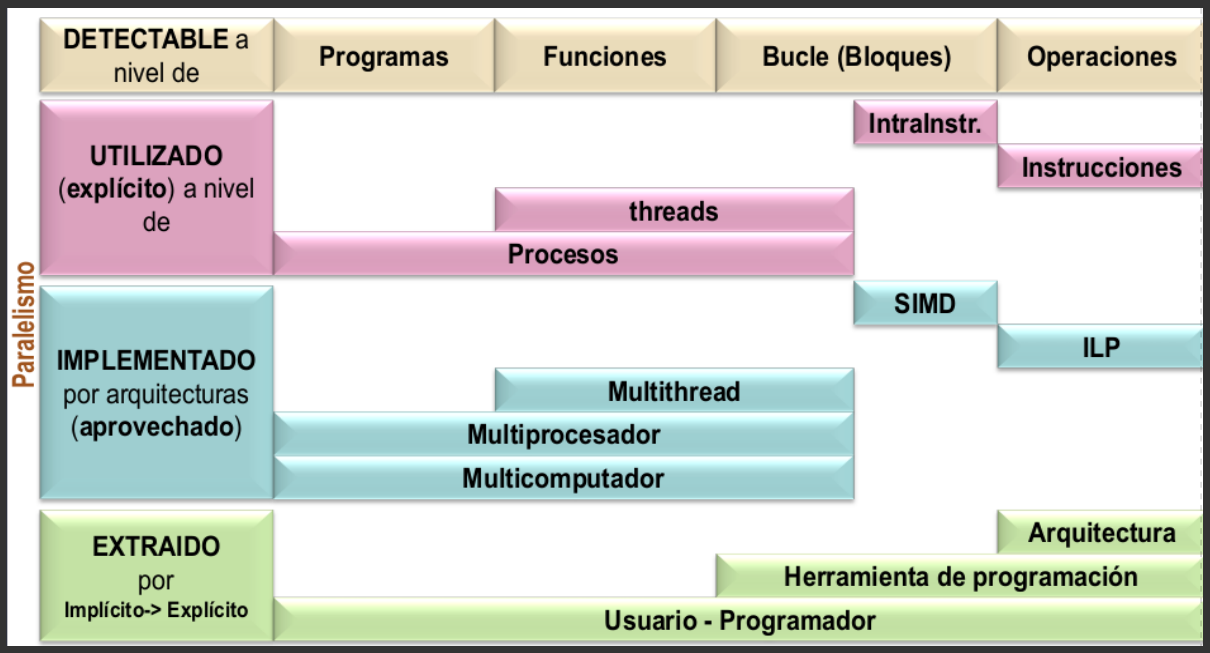

Ilustración 5: Página 15 - T1L1 - Transparencias Mancia Anguita

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Maths Informática, academia especializada en estudios de informática UCM

Maths informática
Profesores titulados | Más de 30 años de experiencia
Máximo 16 alumnos por plaza | Plazas limitadas

www.mathsinformatica.com 615 29 80 22 academia@mathsinformática.com 91 399 45 49 academia.maths C/Andrés Mellado, 88 duplicado

COMPUTACIÓN PARALELA

La computación paralela estudia a nivel de hardware y software las aplicaciones de un sistema de cómputo formado por varios cores/procesadores/computadores que cooperan, de manera que se puede considerar una unidad autónoma.

Auna recursos.

Ej.: multicores, multiprocesadores, multicomputadores, clusters...

No confundir con la **concurrencia**.

Ej.: segmentación

Divide recursos.

COMPUTACIÓN DISTRIBUIDA

La computación distribuida, se encarga de los aspectos hardware y software de un sistema distribuido, es decir, una colección de recursos autónomos(PCs, servidores, supercomputadores, aplicaciones, etc) situados en distintas localizaciones físicas.

Podemos distinguir la computación distribuida por su escala:

- Computación distribuida a baja escala → Desarrollo y ejecución de aplicaciones a través de un conjunto de recursos autónomos de un dominio administrativo, situados en distintas localizaciones físicas e interconectados por una infraestructura red local.
- Computación grid → Desarrollo y ejecución de aplicaciones a través de recursos autónomos de múltiples dominios administrativos geográficamente distribuidos y conectados mediante una infraestructura de telecomunicaciones.
- Computación cloud → Los sistemas cloud ofrecen servicios, infraestructuras o software a través de una interfaz web de autoservicio. Para ello abstraen los recursos físicos, simulan recursos ilimitados, de forma que se crean y liberan de forma inmediata, sin mediación con el proveedor, soportan multiples clientes de forma simultánea(multi-tenant) y están conectados con métodos estándar independientes de la plataforma de acceso.

CRITERIOS DE CLASIFICACIÓN DE COMPUTADORES

Por **segmento del mercado**, podemos hacer la siguiente distinción:

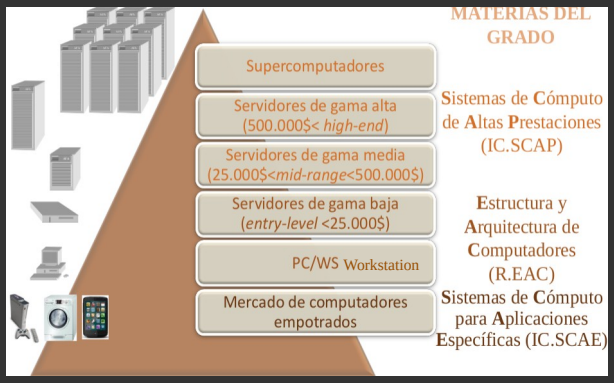

Ilustración 6: Página 27 - T1L2 - Transparencias Mancia Anguita

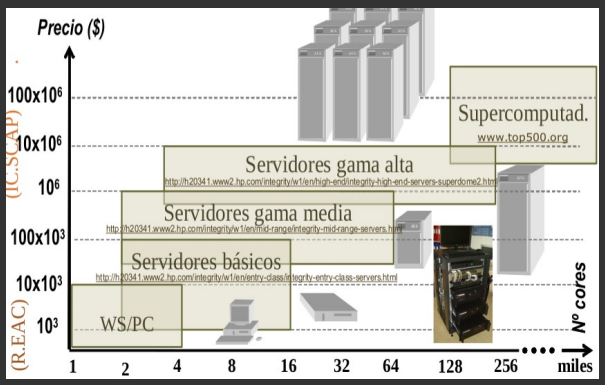

Ilustración 7: Página 29 - T1L2 - Transparencias Mancia Anguita

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad

Siguiendo en segmento de mercado:

-Externo → Como podrían ser los PCs, portátiles, clusters, etc. Estos se emplean para todo tipo de aplicaciones: entretenimiento, oficina, servicios, científico, animación, etc.

-Empotrado (oculto) → En este caso si existen para un propósito específico como videojuegos, teléfonos, coches, electrodomésticos... . Por la propia limitación de su construcción conllevan unas restricciones típicas que son el consumo de potencia, su precio y su tamaño reducido, además de por necesidad, ser un sistema de tiempo real.

Por la **clasificación Flynn** de arquitecturas de 1972:

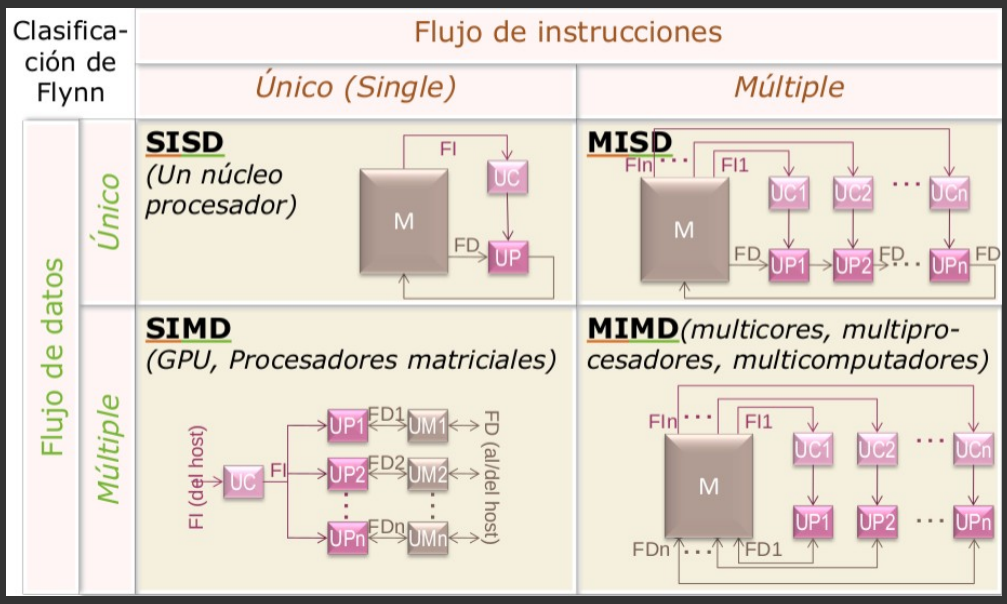

Ilustración 8: Página 31 - T1L2 - Transparencias Mancia Anguita

La taxonomía Flynn divide el universo de los computadores en cuatro clases:

- Computadores SISD → Un único flujo de instrucciones procesa operandos y genera resultados que definen un único flujo de datos

  Computadores Unicore

- Computadores SIMD → Un único flujo de instrucciones procesa operandos y genera resultados, definiendo varios flujos de datos. Esto se puede dar ya que cada instrucción codifica realmente varias operaciones iguales, cada una operando sobre operadores distintos

  Aprovecha el **paralelismo de datos**

- Computadores MIMD → El computador ejecuta varias secuencias o flujos de operaciones, y cada uno de ellos procesa operandos y genera resultados definiendo un único flujo de instrucciones, de forma que existen también varios flujos de datos, uno por cada flujo de instrucciones

  Computadores Paralelos. Multinúcleo, multiprocesador y multicomputador.

  Aprovecha además paralelismo funcional

- Computadores MISD → Se ejecutan varios flujos distintos de instrucciones aunque todos actúan sobre el mismo flujo de datos

  Este modelo, aunque se puede simular un código para aplicaciones que procesan un flujo de datos, no está implementado, no existe computadores que funcionen con este modelo

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Mucho MATHS que una academia

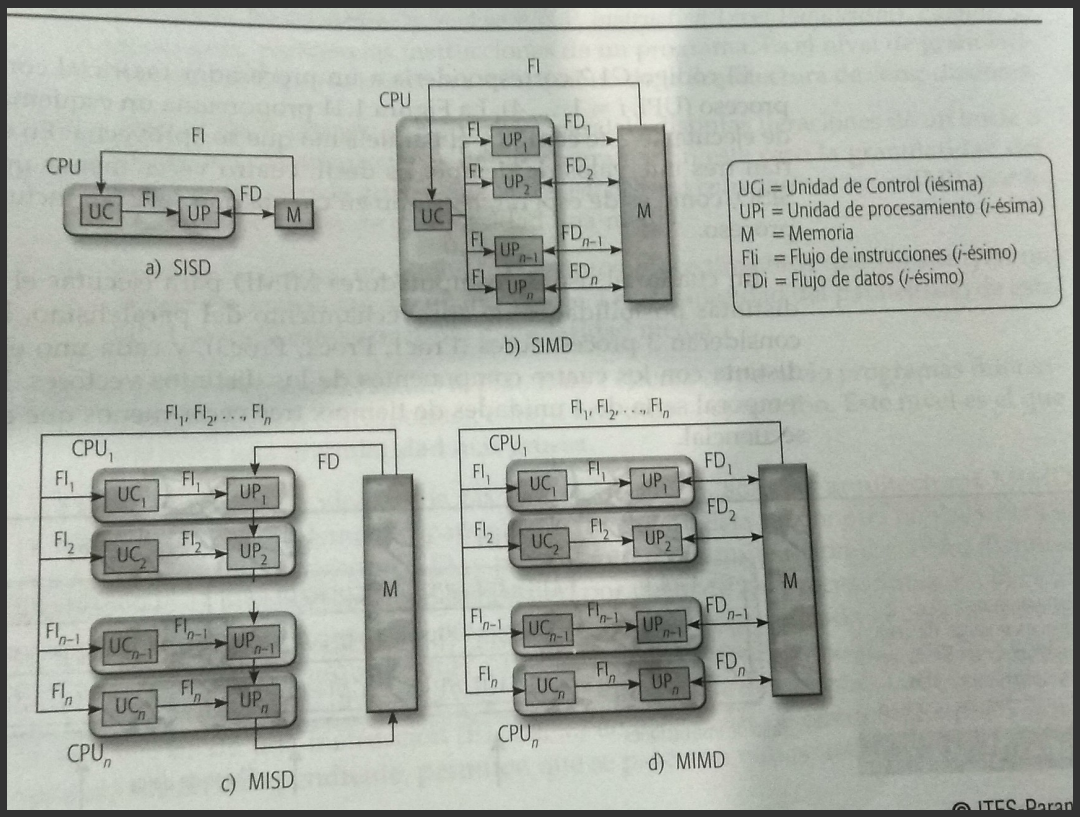

Ilustración 9: Página 27 - Arquitectura de Computadores - Julio Ortega, Mancia Anguita y Alberto Prieto

Al contener MI, se comienza a trabajar con varias Unidades de Control.

Como hemos dicho, los MIMD son los Computadores paralelos, podemos diferenciarlos por el **sistema de memoria**:

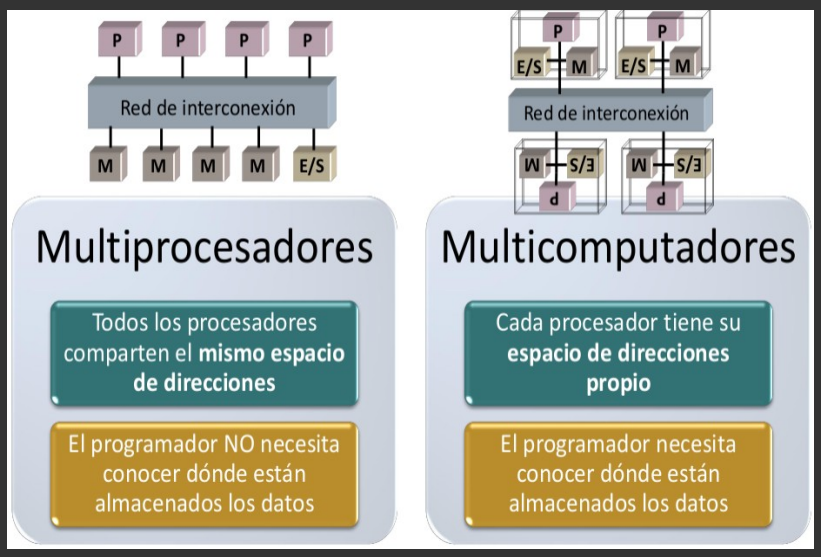

Ilustración 10: Página 38 - T1L2 - Transparencias Mancia Anguita

Los multiprocesadores tienen un sistema de memoria compartida.

Los multicomputadores tienen un sistema de memoria distribuida

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Cúelale un troyano al suspenso - Maths Informática

Decimos que un sistema es simétrico porque desde todos los módulos de procesamiento, se accede con la misma velocidad a los módulos de memoria. Por tanto, podemos asegurar, que un sistema multiprocesador donde cada procesador accede a memoria de forma igualitaria, siendo la red de interconexión (puente norte o chipset y buses) la que gestiona estos accesos, se trata de un SMP (symmetric MultiProcessor) o UMA.

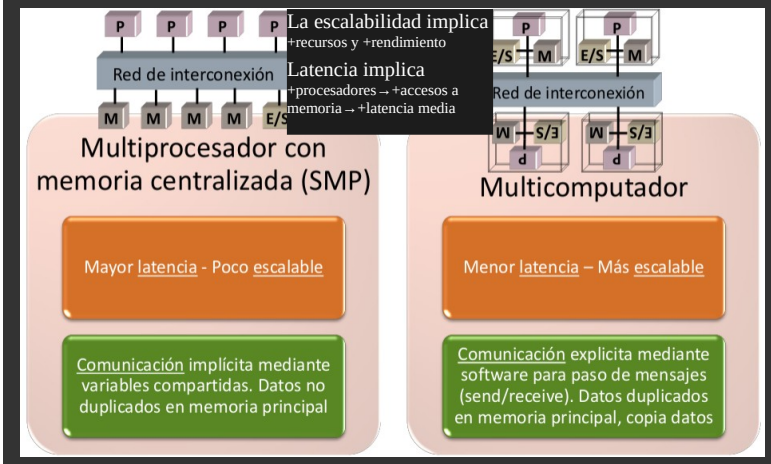

Ilustración 11: Página 39 - T1L2 - Transparencias Mancia Anguita

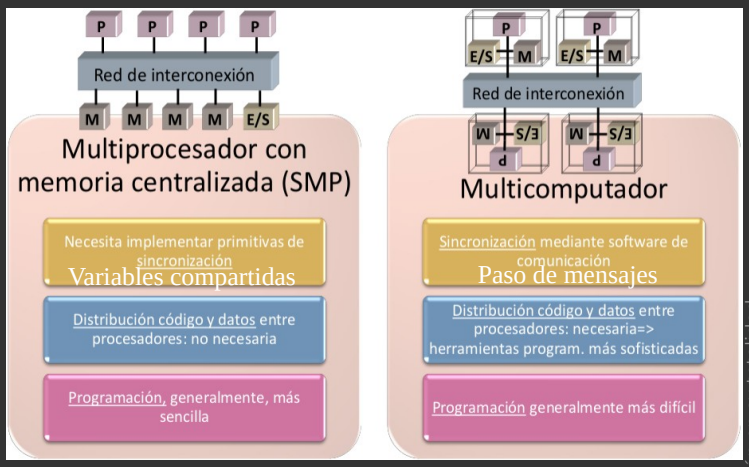

Ilustración 12: Página 40 - T1L2 - Transparencias Mancia Anguita

Sistemas de comunicación uno-a-uno aplicados a multiprocesadores y multicomputadores

MULTIPROCESADOR

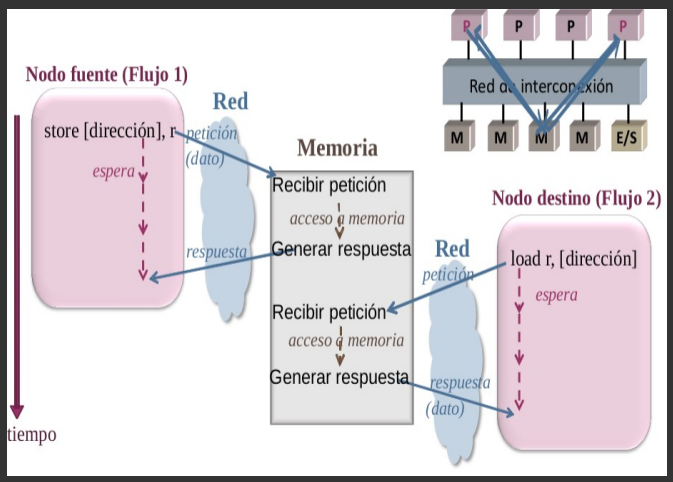

Ilustración 13: Página 41 - T1L2 - Transparencias Mancia Anguita

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

Ilustración 15: Página 42 - T1L2 - Transparencias Mancia Anguita

MULTICOMPUTADOR(Receive Bloqueante)

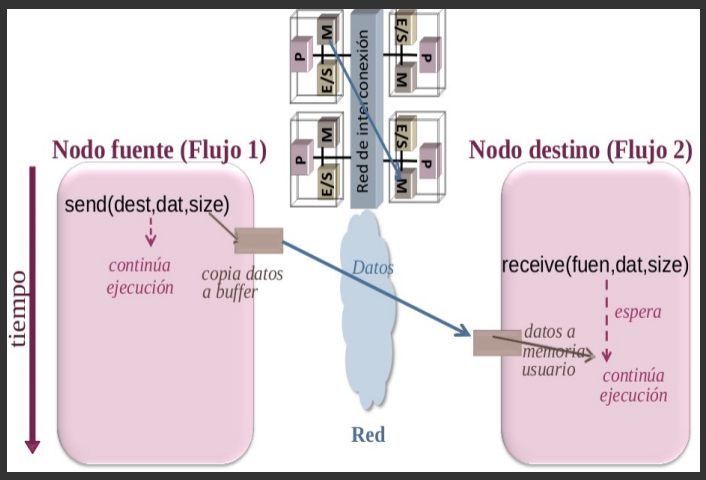

Ilustración 14: Página 43 - T1L2 - Transparencias Mancia Anguita

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

Ilustración 16: Página 43 - T1L2 - Transparencias Mancia Anguita

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Maths informática
Profesores titulados | Más de 30 años de experiencia
Máximo 16 alumnos por plaza | Plazas limitadas

Para poder incrementar la escalabilidad en los multiprocesadores, debemos implementar las siguientes técnicas:

- Añadir caché local a cada procesador o aumentarla si ya la tiene
- Distribuir físicamente los módulos de memoria entre los procesadores (aunque siguen compartiendo espacio de direcciones)
- Usar redes de menor latencia y mayor ancho de banda que un solo bus (mútiples buses, barras cruzadas, multietapa, etc)

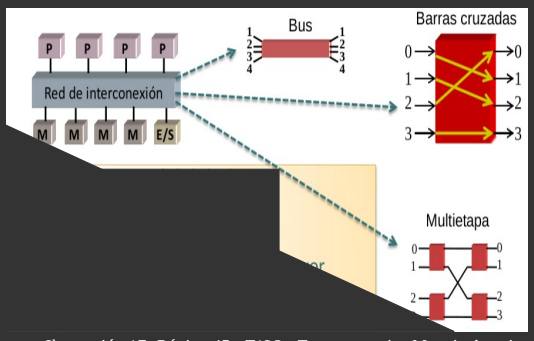

Ilustración 17: Página 45 - T1L2 - Transparencias Mancia Anguita

Redes en sistemas con memoria físicamente distribuida

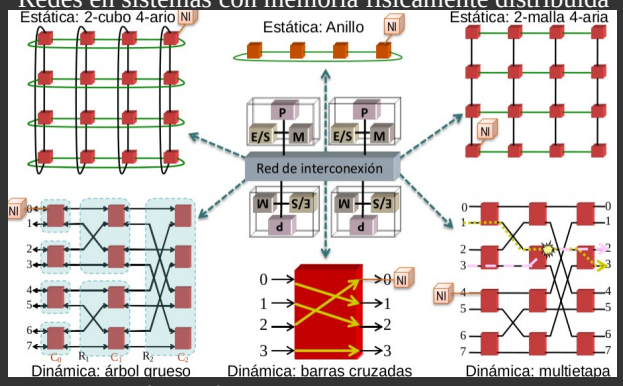

Ilustración 18: Página 47 - T1L2 - Transparencias Mancia Anguita

La red puede ser por ejemplo, un switch

Resumen Clasificación computadores según memoria

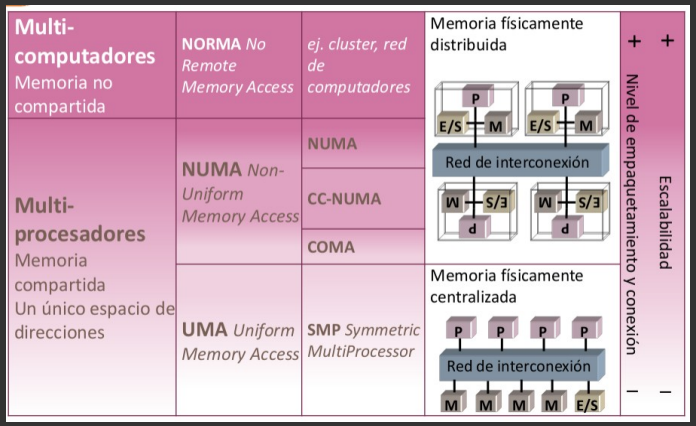

Ilustración 19: Página 46 - T1L2 - Transparencias Mancia Anguita

Clasificación **arquitecturas con múltiples flujos de control o threads**

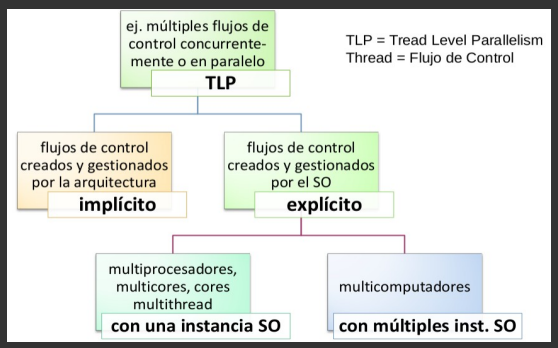

Ilustración 20: Página 51 - T1L2 - Transparencias Mancia Anguita

www.mathsinformatica.com 615 29 80 22 91 399 45 49 academia@mathsinformática.com C/Andrés Mellado, 88 duplicado academia.maths
Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad

Clasificación **paralelismo aprovechado**

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	Tema 4	Temas 3, 5		IC.SCAP
Ejecutan las operaciones de una instrucción **concurr.** o en **paralelo**	Ejecutan múltiples instrucciones **concurr.** o en **paralelo**	Ejecutan múltiples flujos de control **concurr.** o en **paralelo**		Ejec. múltiples flujos de control en **paralelo**
Unidades funcionales vectoriales o SIMD	**Cores** escalares segmentados, superescalares o VLIW/EPIC	**Cores** que modifican la arquit. 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

Ilustración 21: Página 53 - T1L2 - Transparencias Mancia Anguita

Lectura notas históricas

- DLP (*Data Level Parallelism*) — TEMA 5
  - Unidades funcionales (o de ejecución) SIMD (o multimedia)
    - **1989** (Intel i860). **1991** (motorola M88110). **1993** (repertorio MAX en HP PA : PA7100LC). **1995** (repertorio VIS en Sun Sparc: Ultra I). **1997** (repertorio MMX en Intel x86: Pentium MMX). **1999** (repertorio SSE en Intel x86: Pentium III; repertorio Altivec en IBM Power: PowerPC 8000)
- ILP (*Instruction Level Parallelism*) : — TEMA 4
  - Procesadores/cores segmentados
    - ***1961*** *(IBM 7030)*. **1982** (chip Intel i286, Motorola 68020). **1986** (chip MIPS R2000). **1987** (chip AMD Am29000). **1988** (chip Sun Sparc)...
  - Procesadores con múltiples unidades funcionales
    - **1967** (IBM 360/91) ...
  - Procesadores/cores superescalares
    - **1989** (chip Intel 960CA (3)). **1990**: (chip IBM Power1 (4)). **1992**: (chips DEC α21064 (2/4), HP PA 7100 (2/2), Sun SuperSparc (3/5)) ...
  - Procesadores/cores VLIW
    - **1990** (chip DSP Intel i860 (2)).**1997** (chip DSP TMS320C6x (8)). **2001** (chip Intel Itanium)...
- TLP explícito con una instancia de SO: — TEMA 3
  - Multithread grano fino (FGMT)
    - ***1975*** *(Denelcor HEP)*. **2005** (chip Sun UltraSPARC T1) ...
  - Multithread grano grueso (CGMT)
    - ***1990*** *(MIT Alewife)*. **2000** (chip IBM PowerPC RS64 IV (2)). **2006** (chip Intel Itanium Montecito (2)) ...
  - Multithread simultánea (SMT)
    - **2002** (chip Intel Pentium 4/Xeon Hyper-Threading). **2004** (chip IBM Power5) ...
  - Multiprocesadores en un chip (CMP) o multicores
    - **2001** (chip IBM Power4). **2004** (chip Sun UltraSPARC IV). **2006** (chip Intel Core Duo. **2008** (chip Intel Celeron Dual-core) ...
  - Multiprocesadores
    - ***1962*** *(Burroughs D825 - red barras cruzadas)*. ***1966*** *(UNIVAC 1108 - red bus)*. ***1985*** *(IBM RP3 - red multietapa)*. ***1996*** *(SGI Origin 2000 -CC-NUMA, red estática+multietapa)* .**2006** (SGI Altix 4000) ...
- TLP explícito con múltiples instancias del SO (multicomputadores) IC.SCAP
  - ***1985*** *(Intel iPSC1 - i286+red estática con Ethernet)* ... cualquier cluster

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Mucho MATHS que una academia