VOLUME 21, NUMBER 1 JANUARY/FEBRUARY 2019

Special Issue

Theme Articles

Recommended by the Editors

Feature Article

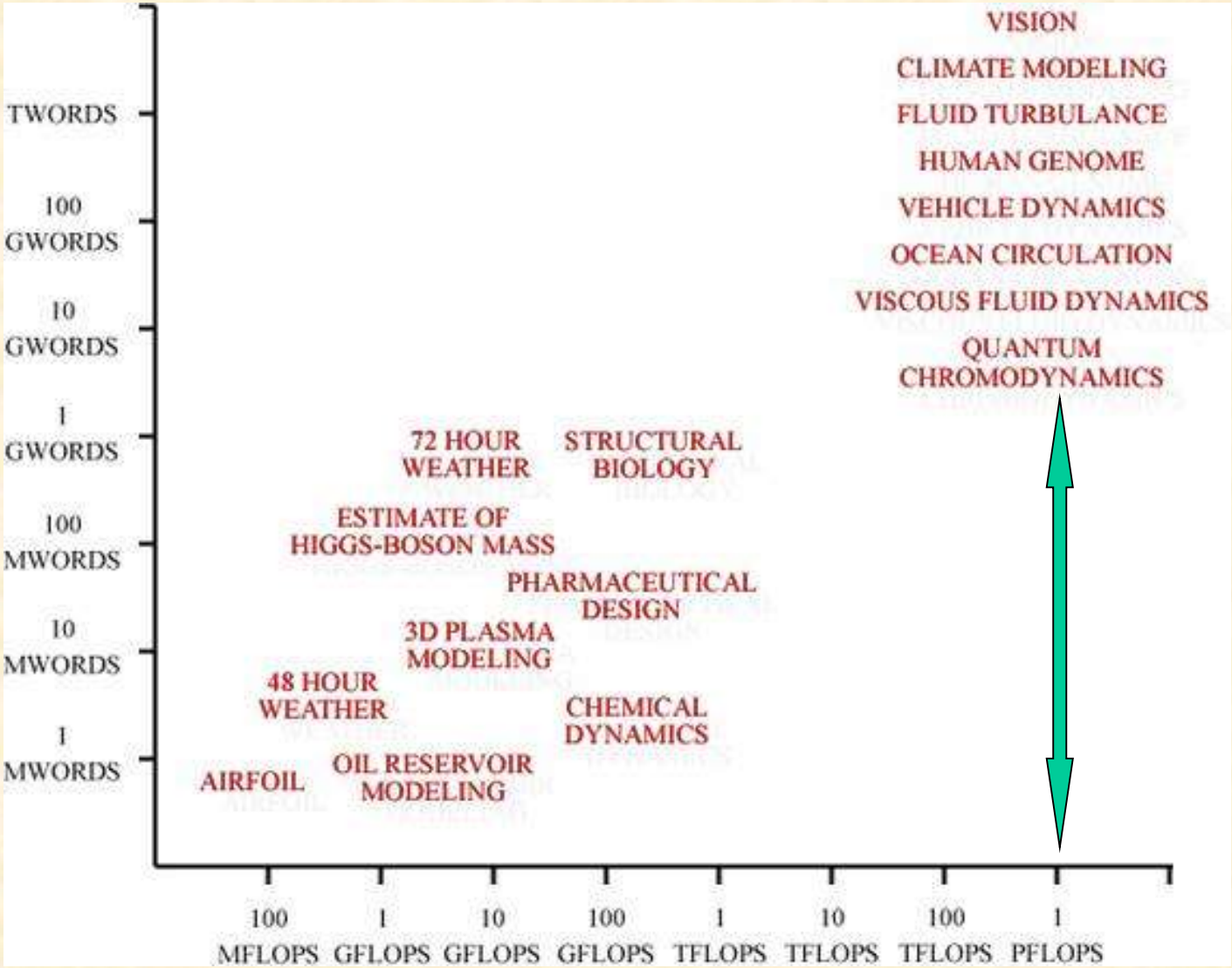
China y Japón ➔ 2020
EEUU y UE ➔ 2023

Computing in SCIENCE & ENGINEERING

Race to Exascale

- Introduction to Terascale Code Development (Sep/2004)

www.psc.edu/training/TCD_Sep04/index.html



¡ MÁS DE 60 AÑOS TRABAJÁNDOSE EN ESTE CAMPO !

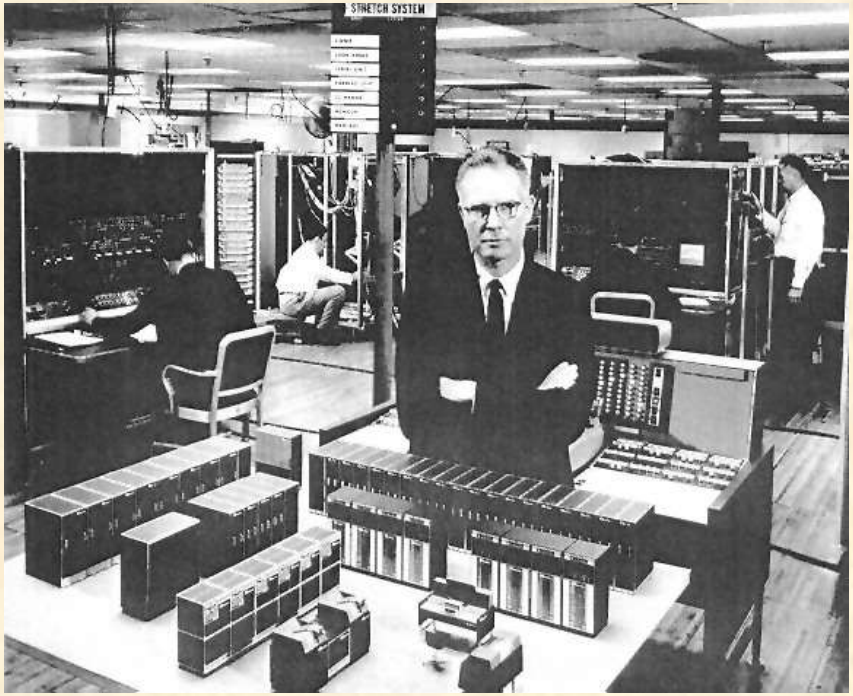
ei.cs.vt.edu/~history/Parallel.html

1955: IBM704 (FPU) Gene Amdahl
1956: IBM STRETCH (* 100 pero 1961 * 50)

1962: Burroughs D825 (1 a 4 CPU's)
1965: Dijkstra (R.C.) Cooley & Tukey (FFT)
1966: Taxonomia de Flynn
1968: Dijkstra (Semáforos)
1969: MULTICS (con 8 CPU's)

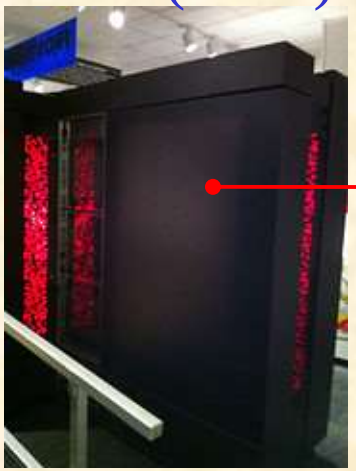
1976: Cray I (Más potente hasta 1985 => Cray II)

IBM STRETCH (1961)



Cray I (1976)

CM5 (1993)



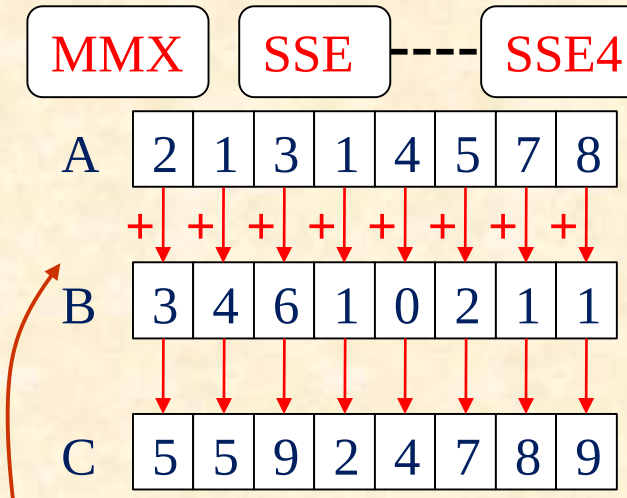
3 días

1 seg

¿1 hora?

IBM Sequoia (2012)

- Combina Flujo de Datos y Flujo de Instrucciones con (Único) *Single* y Múltiple, dando 4 combinaciones:



Flujo de Intrucc. \ Flujo de Datos	Único	Múltiple
Único	SISD	SIMD
Múltiple	MISD	MIMD

SISD Una Instrucción Un Dato — Von Neumann

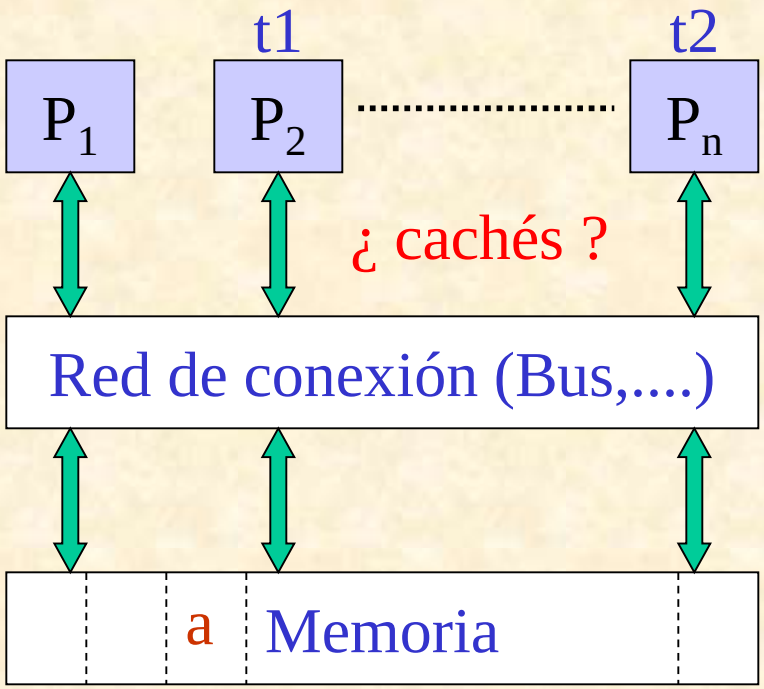
SIMD Una Instrucción Muchos Datos — Vectoriales / Sistólicos

MISD Muchas Instrucciones Un Dato — Pipeline?

MIMD Muchas Instrucciones Muchos Datos — Multiprocesadores / Multicomputadores

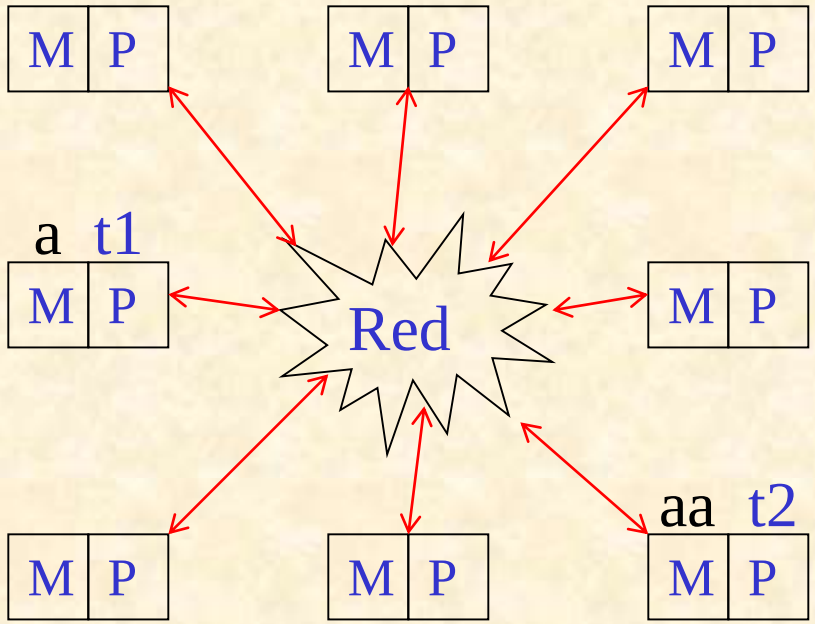
- MIMD: Muchas Instrucciones Muchos Datos

M. Común (Multiprocesador)



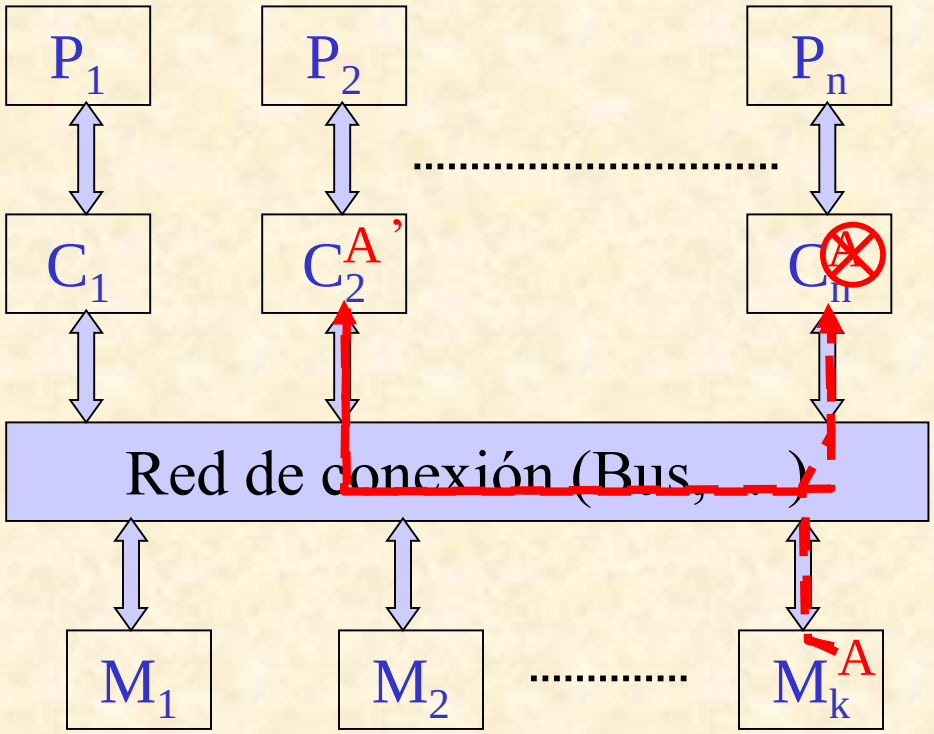
```
varGlobal a: int;
Thread1          Thread2
a = 5;           if (a>0)
```

M. Privada (Multicomputador)



```
var a: int;          | var aa: int;
a = 5;               | rec(t1, &aa)
send(t2, &a)         | if (aa>0)
```

- Problema de tener varias cachés

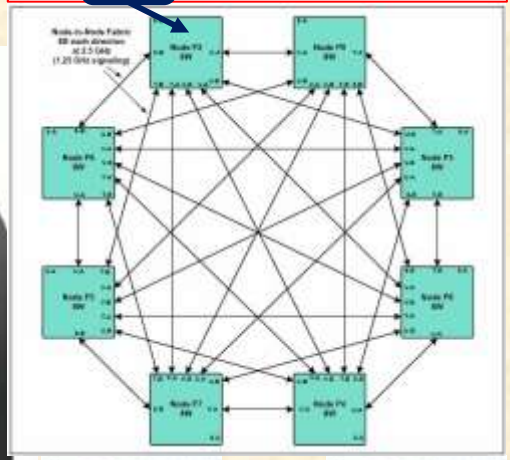


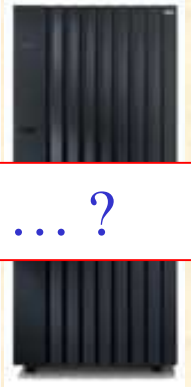
a. P_2.R[A] *
b. P_n.R[A]

c. P_2.W[A'] INV

- En general resuelto por HW
- Sistemas de memoria común con cachés coherentes
- Protocolos de coherencia:
  - Bus Snoop
  - Red Directorios

8x4x8 = 256 núcleos





¿ Intel Core i3, i5, i7 y … ?

Bus ➔ Pocos procesadores

Red

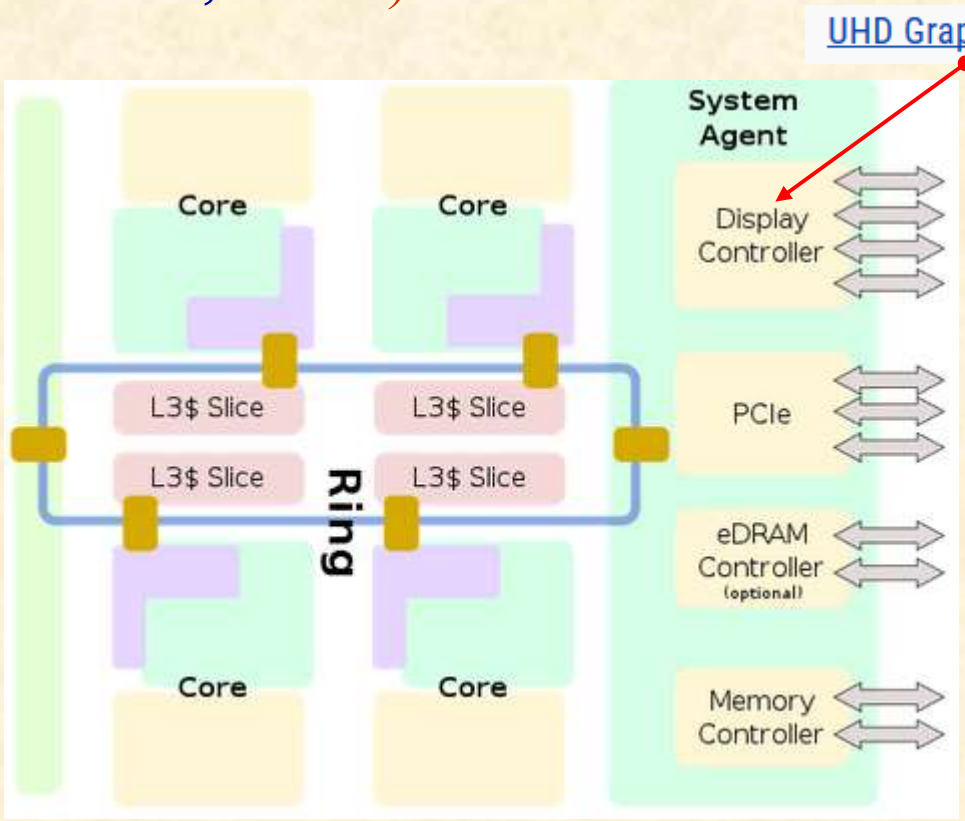
4 Xeon • 2,2GHz • 2MB caché
6GB Mem • 73GB Disco * 4
10.730 € ➔ 2004

24 POWER7 *8cores* y 6TB + 2.463TB

14.276.808 $ con descuento en Ago2010

- Intel Core i3 8100 (4 núcleos a 3,6 GHz) => Mar/2019 603€

14 nm



core

L1I	L1D
32KB	32KB
L2	
256KB	

L1 ➔ 256KB (L1I+L1D)
L2 ➔ 1MB
} privadas
L3 ➔ 6MB (Smart Cache) ⇨ Común LLC
Sin HT ➔ 4 cores | 4 threads

Memoria Privada |

Memoria Distribuída

MPP

ASCI Q (#2 Nov02)

8192 Pi

13,88 TF

175 millones €

COW/NOW

System X (#7 Nov04)

2200 Pi (G5)

10,28 TF

4,5 millones €

Beowulf

12 Pi, 48.000 €

Sep/2008

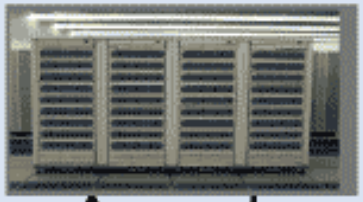
21 servidores Supermicro
Intel Core 2 Q6600
2,4GHz + 4GB

84 núcleos + 84GB

20.600€ ➔ 21 servidores
1.000€ ➔ armario

Ejemplos de Jerarquía de Memoria

1 2002



Earth Simulator

- 10 TB Main Memory

Disk Unit

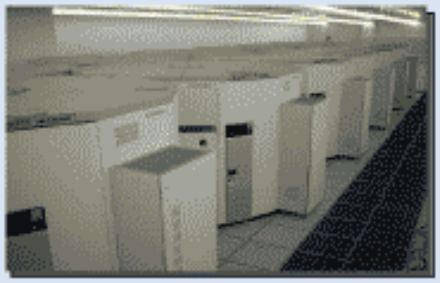
- 230 TB for User Disks
- 460 TB for Work Disk

30

472 I/O Control Station (mover)

Automatic Recall / Migration by the Operation Support Software

- Developed by The Earth Simulator R&D center



ASCI Q	Virginia Tech's X
2003 2 22 TB	4 TB
440 TB	7 2004 176 TB
91	108 2007

1 Jun/2020

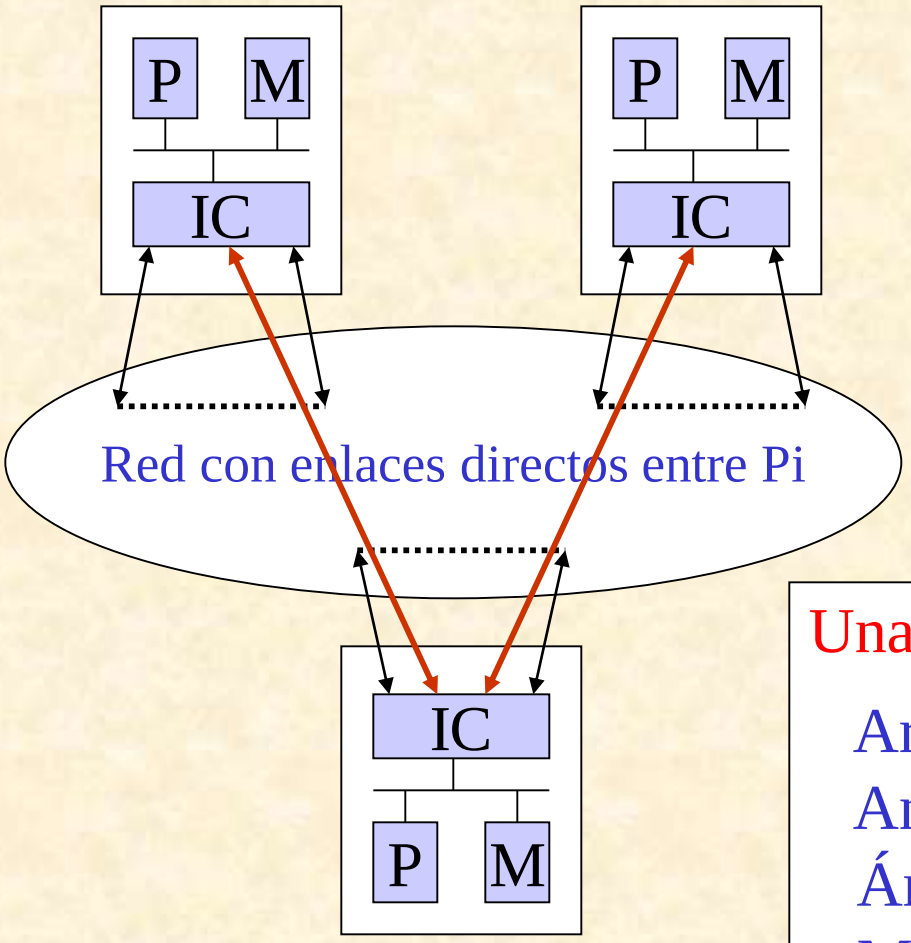
Fugaku
4,9 PB
150 PB

1 Nov/2013

Tianhe-2
1,3 PB
12,4 PB

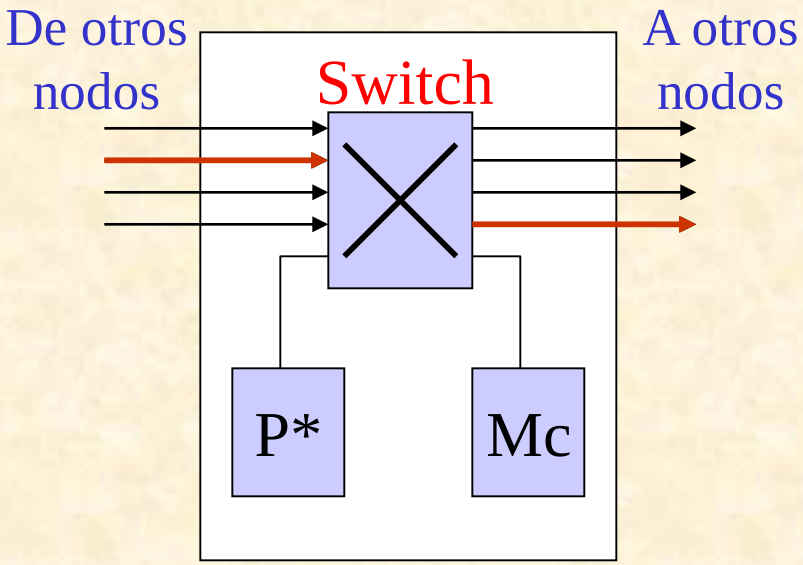
MTBF = 6,16 días*

¡ 3.120.000 núcleos !



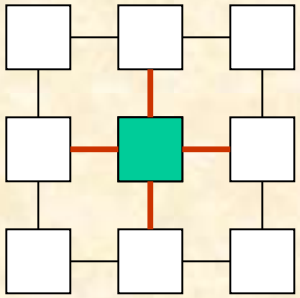
Nodos => PC's o similares

MultiC más integrado



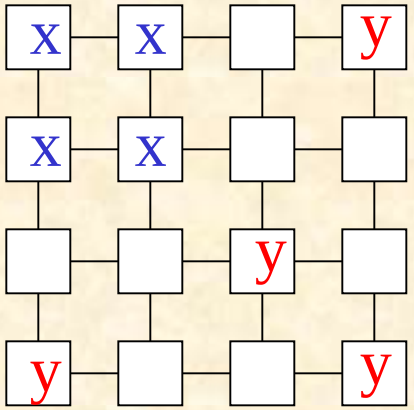
Unas redes directas:

- Array lineal
- Anillo
- Árbol
- Mallas 2D y 3D
- Toros 2D y 3D
- Hipercubo

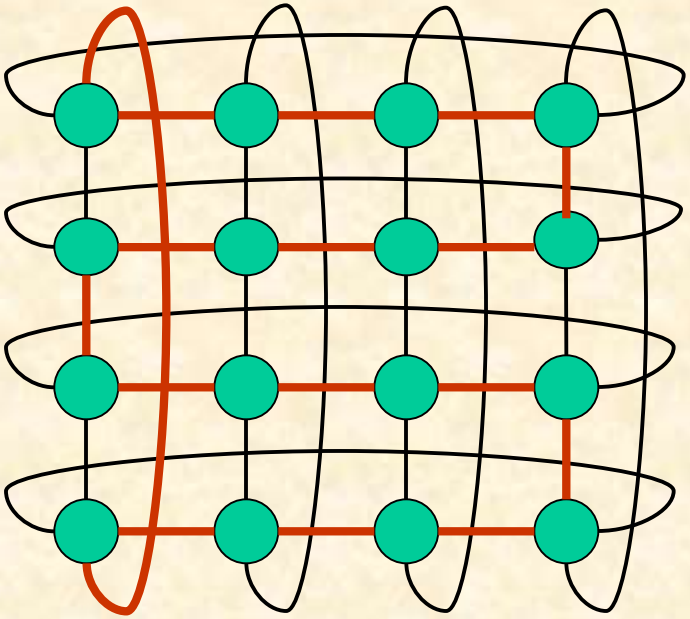


Parámetros de interés:

- Ancho de banda (agregado vs bisección)
- Latencia (Msj[0], Msj[N])
- Diámetro
- Coste (grado: #puertos comunicaciones)

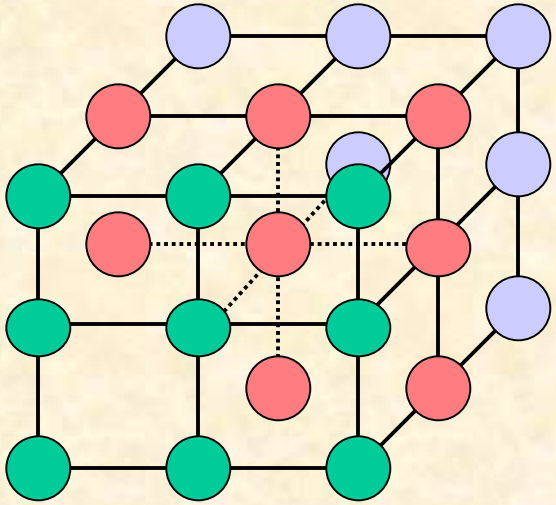


Toro 2D

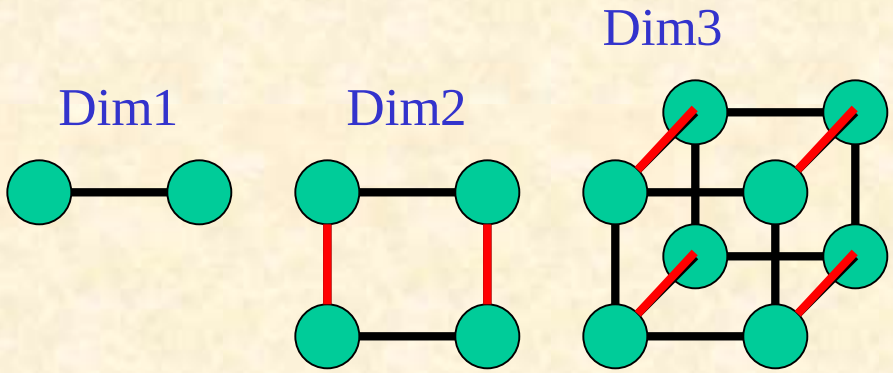


¡ Anillo embebido !

¿ Por qué todo esto ?



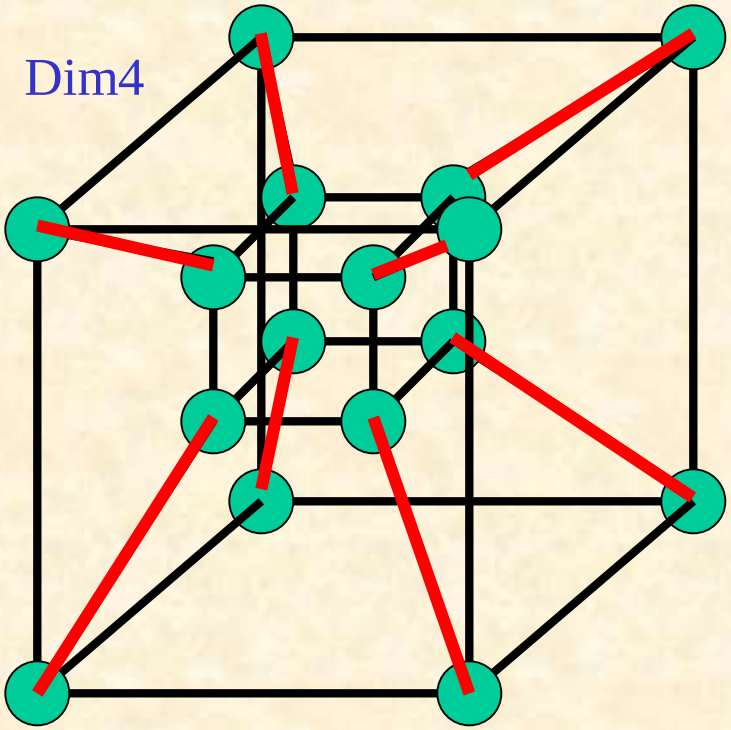
HIPERCUBO

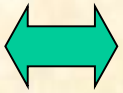


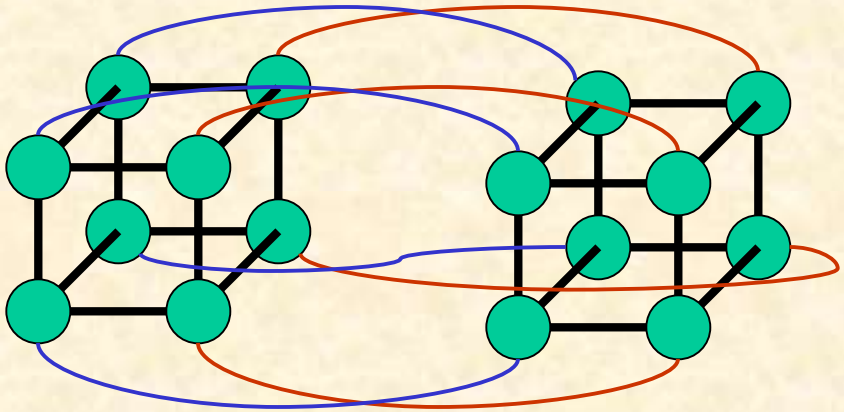
Diámetro = $\log_2 N$

Grado = $\log_2 N$

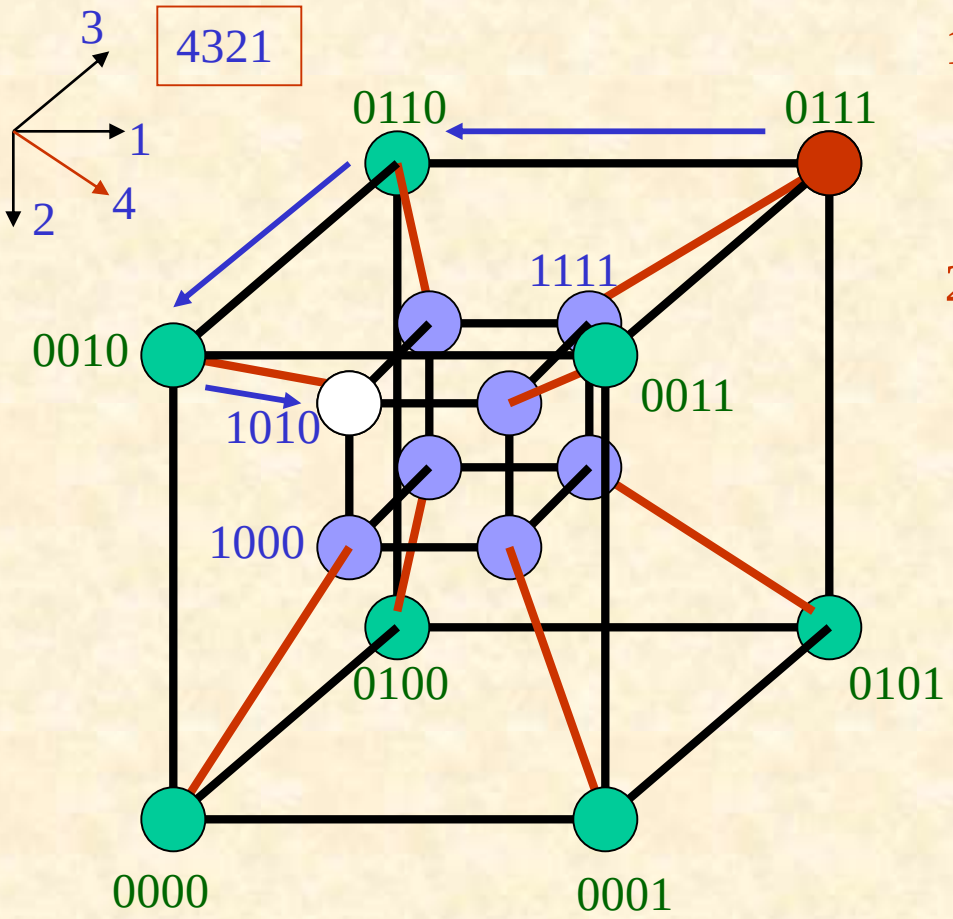
Fácil encaminar







Encaminamiento en HIPERCUBO (Sea N=16)



1. Numerar nodos en binario. Nodos adyacentes difieren en un bit (el asociado a la dirección que les une)
2. Enviar mensaje por el enlace asociado a la menor dirección donde no coinciden bit del nodo actual y bit del nodo destino

¿ Realizar ORX ?

0111 ORX 1010 = 1101

Nodo actual	0111	→ 0110	→ 0010	→ 1010
Nodo destino	1010	1010	1010	1010

Ventajas del entorno PC, o similar, para Sistemas Paralelos

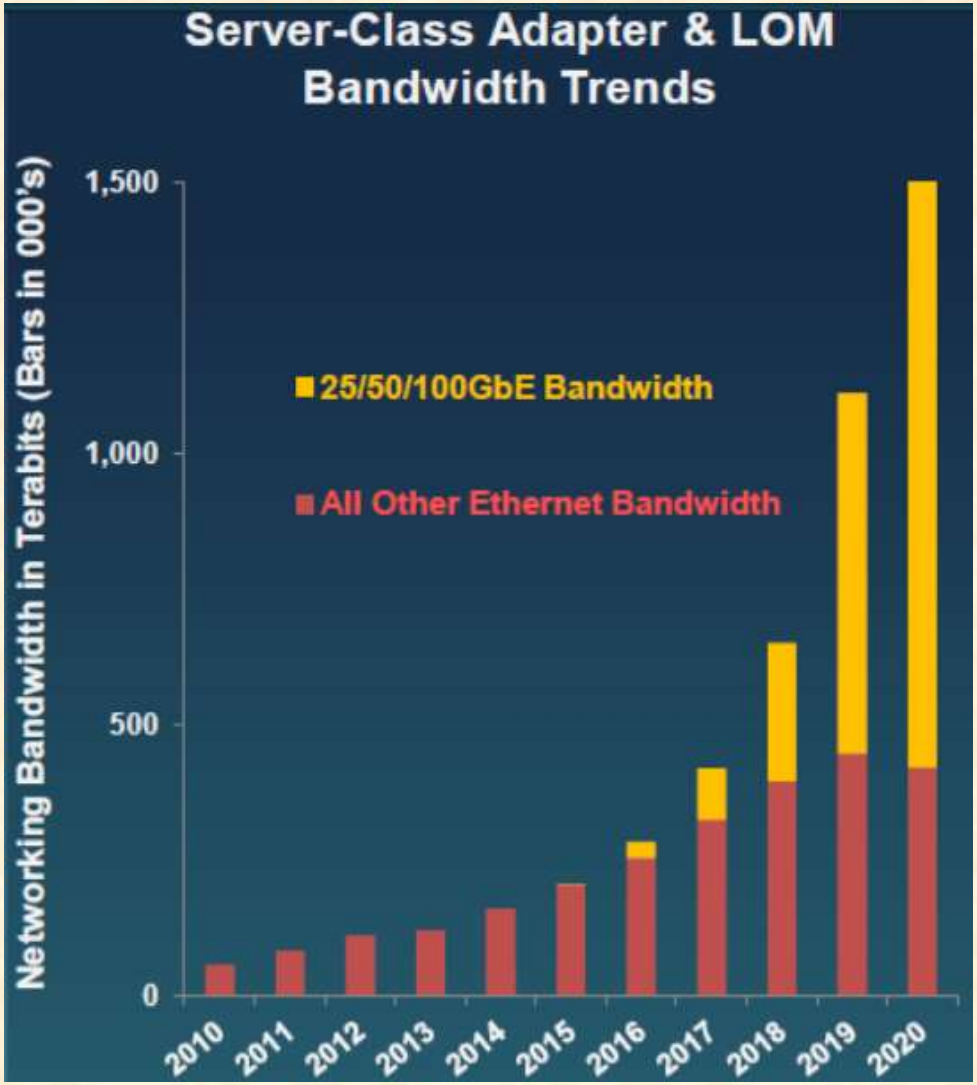
- Hardware rápido y barato (cada año | año y medio más)
  - CPU (Core i7, AMD Zen, IBM PowerPC, …)
  - Memoria P. y Caché (4..16GB y 2MB..8MB)
  - Disco (2..8TB 7200rpm RAID)
- Tecnologías de interconexión
  - Ethernet (Fast, Giga, 10Giga), Infiniband, ...
- Software
  - Sistema Operativo (Linux, Solaris, Windows, ...)
  - Entorno de programación (PVM, MPI, ... ⇔ C, C++, ...)

Thomas Sterling, "Beowulf Cluster Computing with Linux" | "Windows" , 2002

25/50 and 100Gb Ethernet Soon to be Most Deployed Ethernet Bandwidth

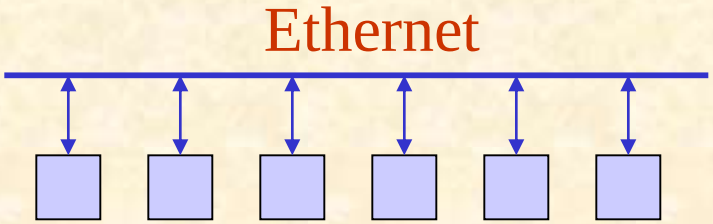
Mellanox Blog

January 30, 2017 Tim Lustig



- Tecnologías de interconexión

Más común (barato) ➔

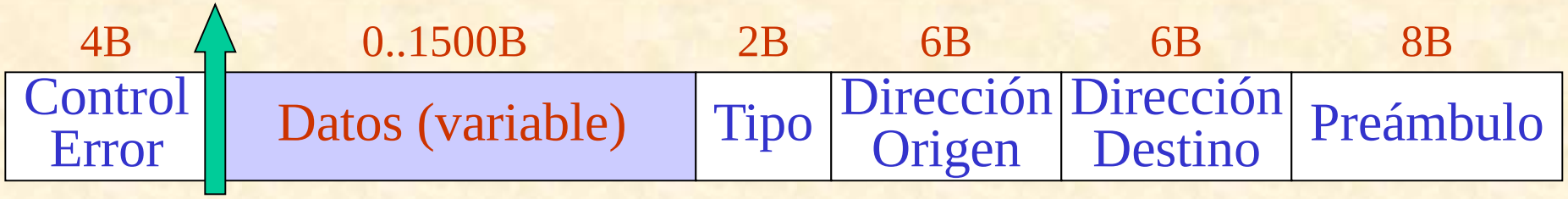


	Gbps	Latencia?	TarjetaRed	Switch (16)
GigaEthernet	1	29..120µs	10€	80€
10GigaEthernet	10	10µs	400€..	1.600€..
Infiniband	10	4µs	500$..	6.000$..

Red ETSISI GigaEthernet

¡ Ojo, todavía menos !

MsjMin 64B



+ COLISIONES

dirección

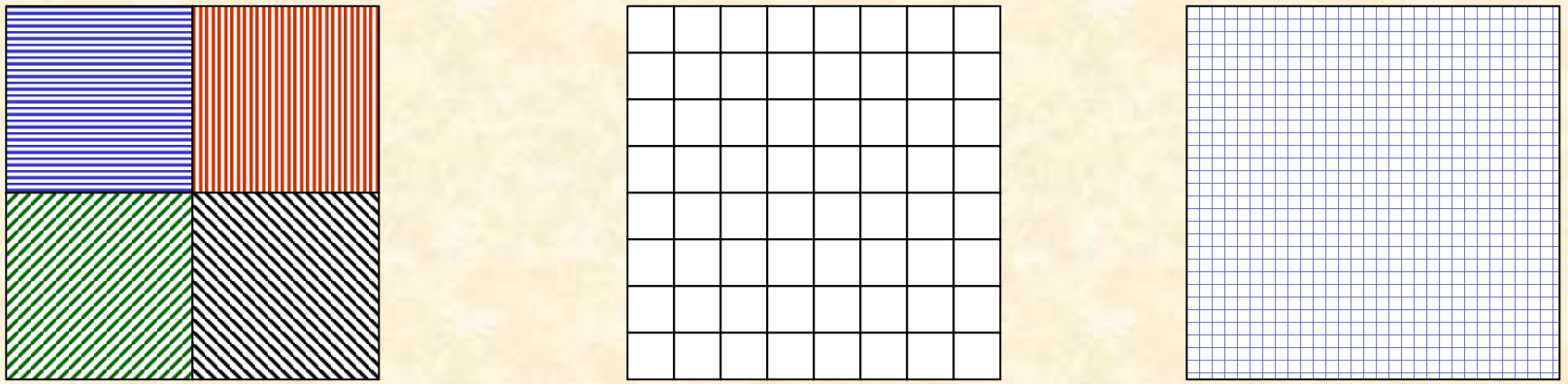
- ALGUNAS MEDIDAS DE PARALELISMO

GRADO: Número de CPU's para las que tengo trabajo simultaneo

GRANO: Tamaño (# instrucciones) asignado a cada CPU

Granularidad gruesa, media, fina

Miles I. Cientos I. Varias I.



+ grado

+ comunicación

Maximizar ratio = Tcómputo / Tcomunicación

- GRADO y GRANO: ¡ Pintar 18 habitaciones !

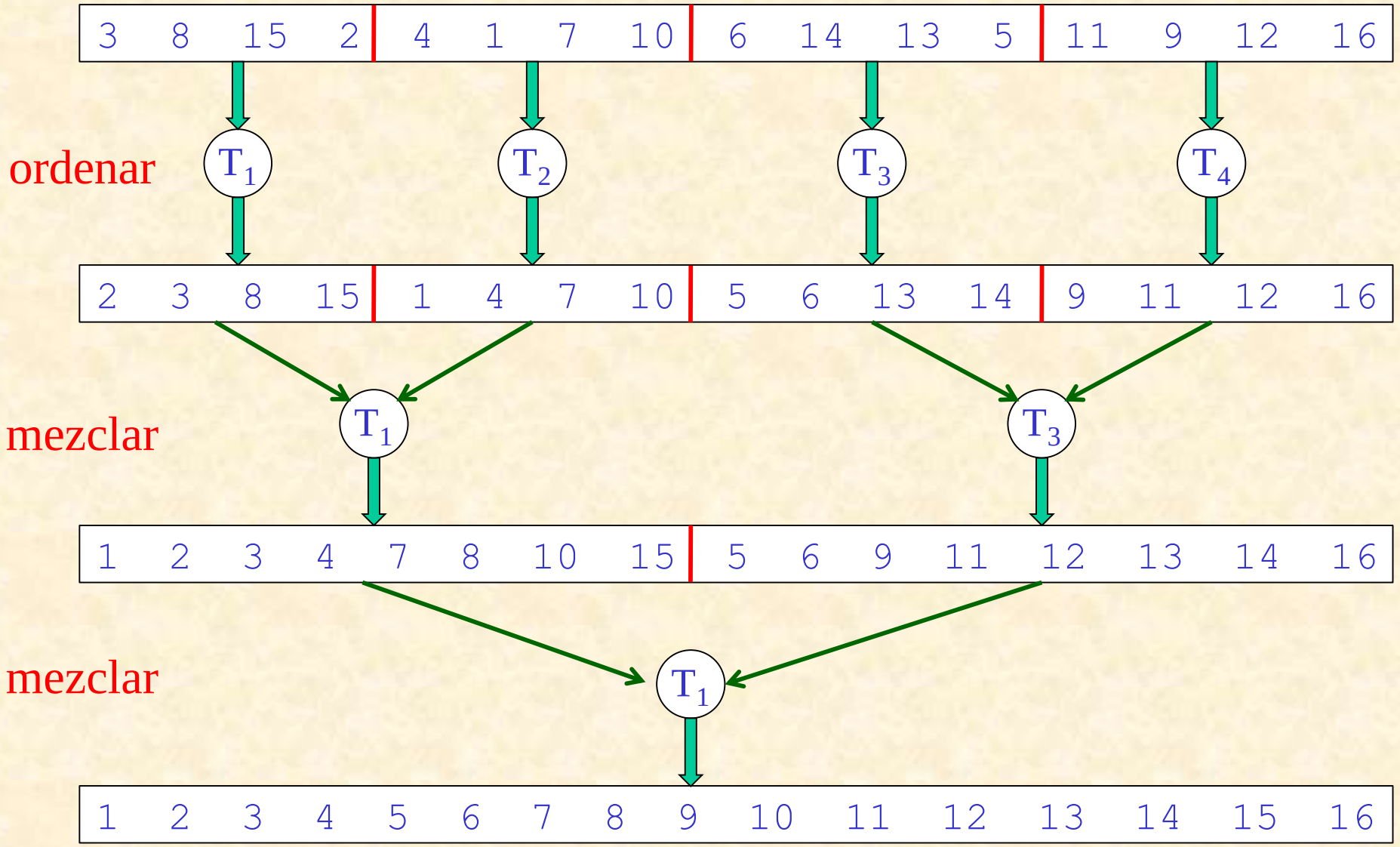
¿18 pintores => Grado = 18
Grano = 1h?

¿Sólo cuento con 3 pintores?

¿Tengo 7 pintores?

¿Me la pega mi marido | mujer?

- ordenarPar: Ordenar un vector en memoria



- ACELERACIÓN: "Speedup" Cuántas veces más rápido al contar con "n" CPU's en vez de una. (Absoluta)

$$Sn = T_{secuencial} / T_{paralelo} = T^1 / T^n$$

SI CON UNA CPU SE ORDENA EN 1 MINUTO,

¿CON 4 CPU's SE ORDENA EN? ===> 15", 20", ...???

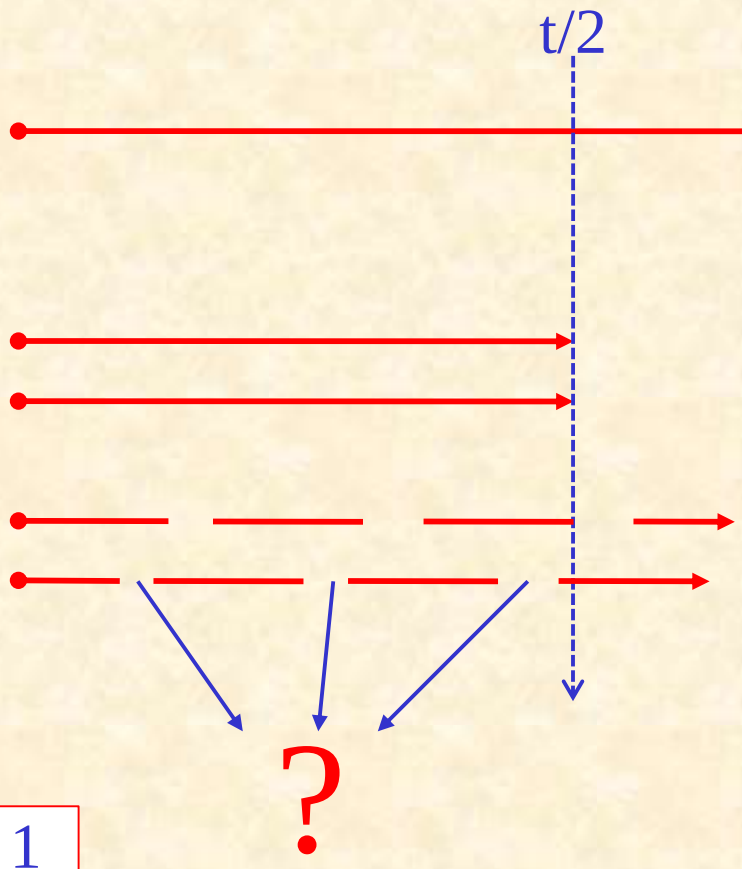
$1 <= Sn <= n$ n = Máximo teórico

- EFICIENCIA: Lo mismo, pero teniendo en cuenta "n". (Relativa)

$$En = S_n / n = T^1 / nT^n$$

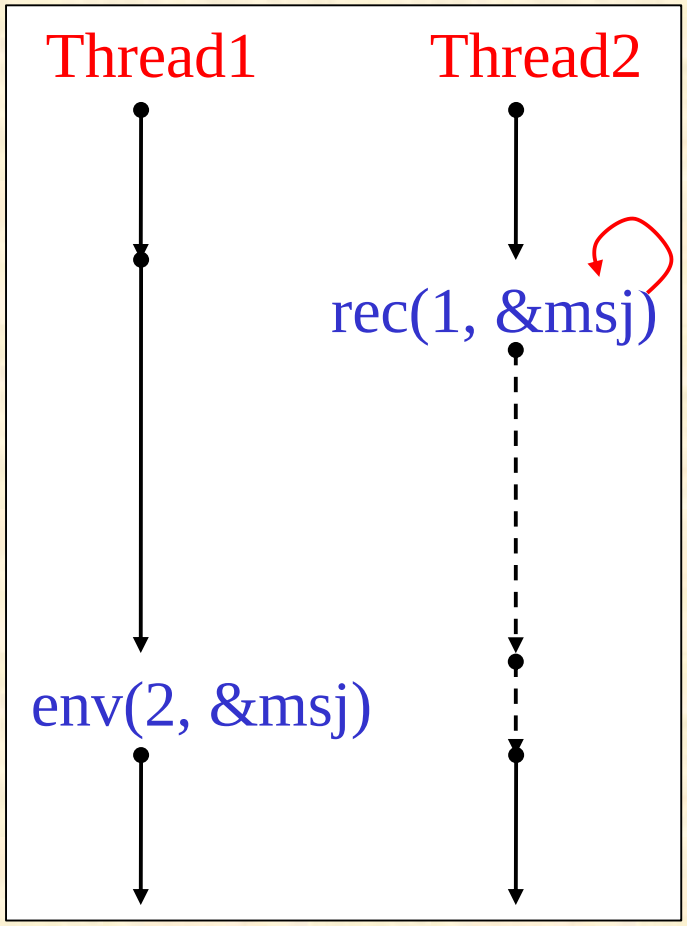
$1/n <= En <= 1$ 1 = Máximo teórico

- ACELERACIÓN: ¿Por qué no tanto?

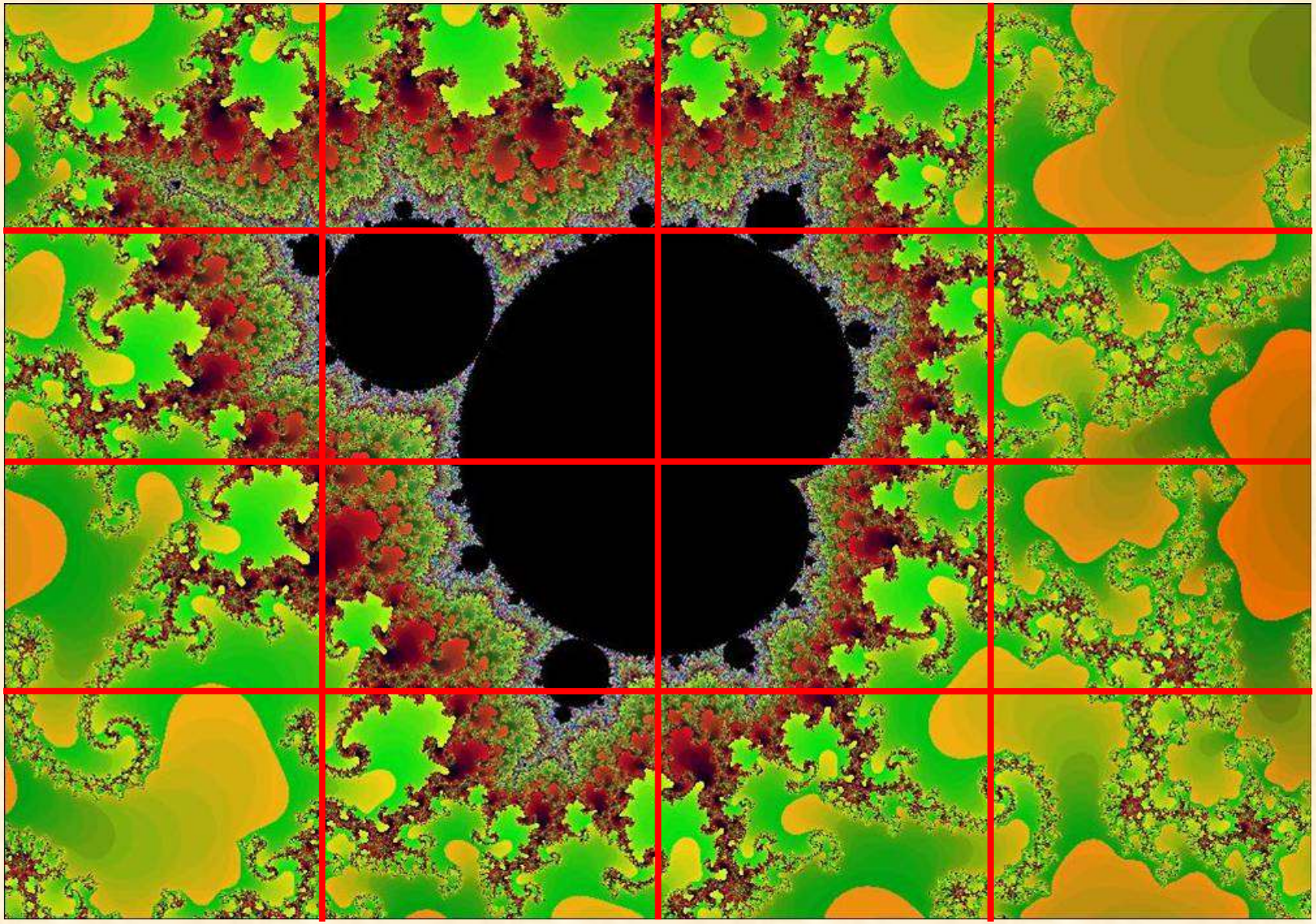


1

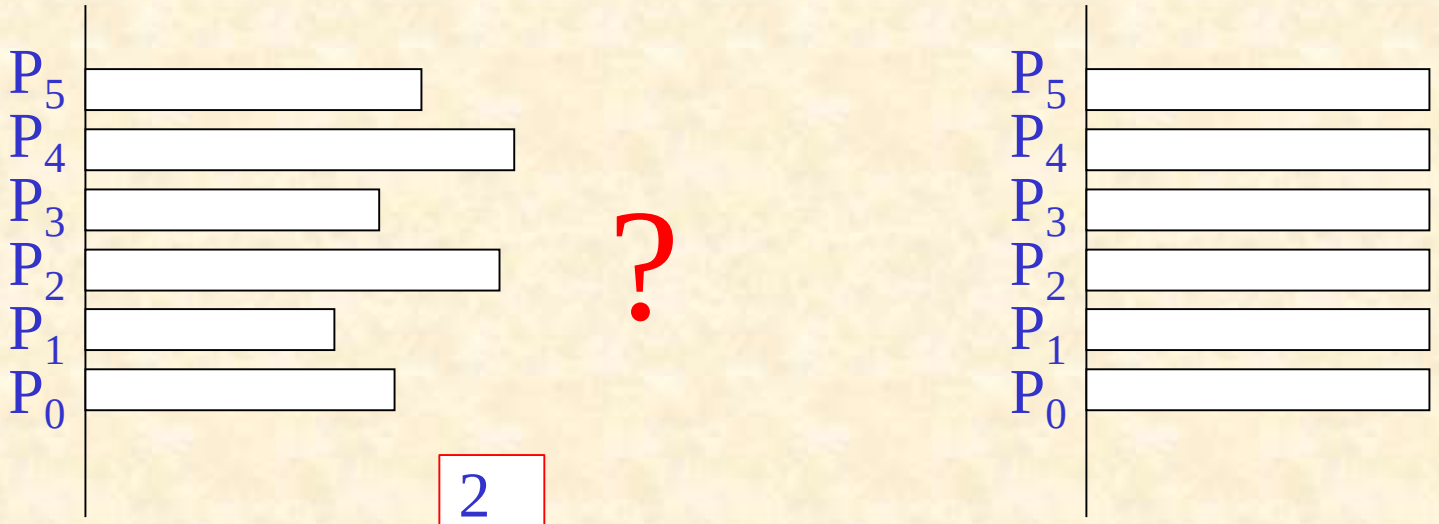
Acceso a datos comunes
Sincronismo
Comunicación



Posibilidades de acelerar ...



- ACELERACIÓN: ¿Por qué no tanto?



2

Reparto no equilibrado

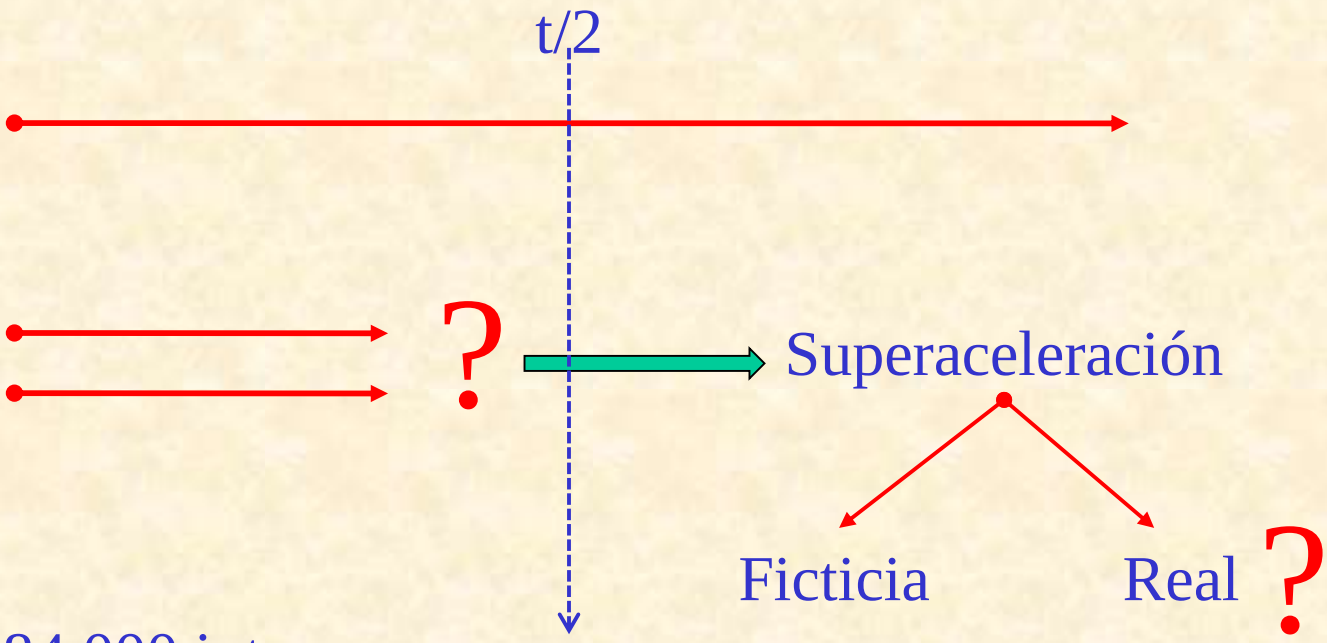
A unos más trabajo que a otros

Unos cores más rápidos que otros

3

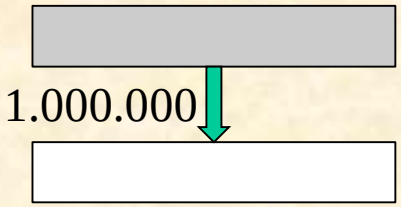
Código secuencial puro

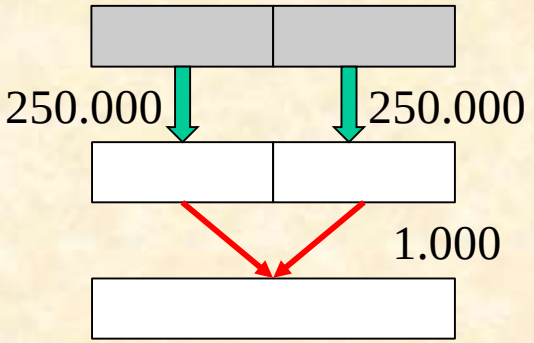
- ACELERACIÓN: ¿Puede que todavía más?

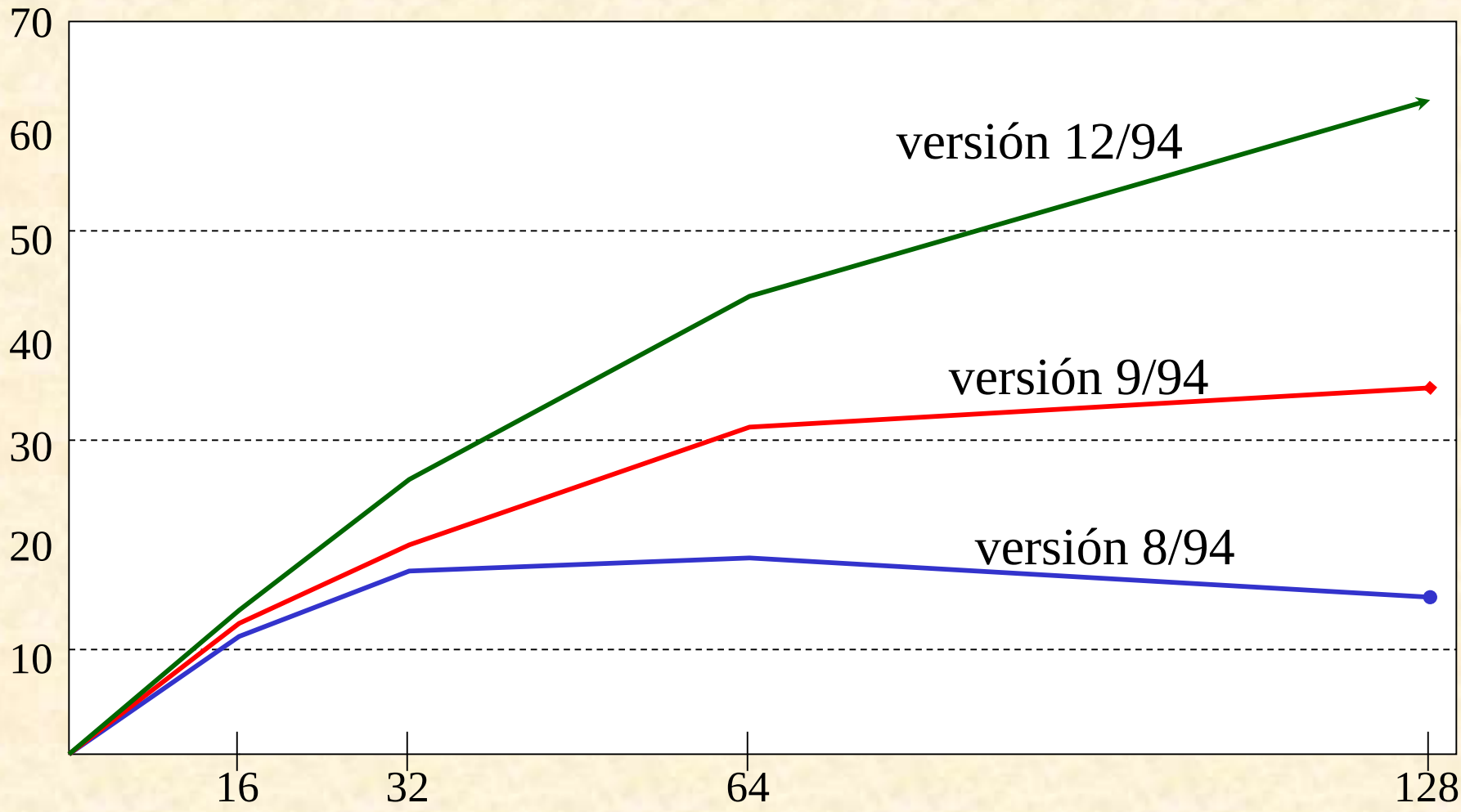


- Ordenar 84.000 int

#nodos	T	S^n
1	29:381	
2	8:693	3,38
4	2:181	13,47





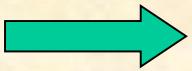


Aceleración en tres versiones de un programa paralelo

Pfeiffer et al. 1995 (AMBER en Intel Paragon 128 μP)

- Sn y En resolviendo Sistemas Lineales de 1000 variables (Jack Dongarra, 2004)

ORDENADOR	#UP	T^1 (S)	T^N (S)	S_N	E_N
Hitachi S-3800/480	4	0,10	0,032	3,21	0,80
NEC SX-3*4R	4	0,13	0,044	2,91	0,73
Cray C90	16	0,74	0,062	11,95	0,75
IBM ES/9000	8	1,58	0,293	5,34	0,67
Convex C4/	2	0,95	0,501	1,89	0,95
Meiko CS2	32	6,89	1,030	6,69	0,21
Fujitsu AP1000	512	160,0	1,100	147,0	0,29
Intel Delta	64	22,00	1,900	11,50	0,18
Intel iPSC/860	128	22,00	2,800	7,68	0,06
Sun Sparc2000	8	26,71	3,370	7,92	0,99



- Sn y En resolviendo Sistemas Lineales de 1000 variables (Jack Dongarra, 2004)

ORDENADOR	#UP	T^1 (S)	T^N (S)	S_N	E_N
Intel Delta	512	22,0	1,5	14,70	0,03
Intel Delta	256	22,0	1,6	13,80	0,05
Intel Delta	128	22,0	1,7	12,90	0,10
Intel Delta	64	22,0	1,9	11,50	0,18
Intel Delta	32	22,0	2,2	10,00	0,31
Intel Delta	16	22,0	2,9	7,59	0,47
Intel Delta	8	22,0	4,1	5,37	0,67
Intel Delta	4	22,0	6,7	3,28	0,82
Intel Delta	2	22,0	11,6	1,90	0,95

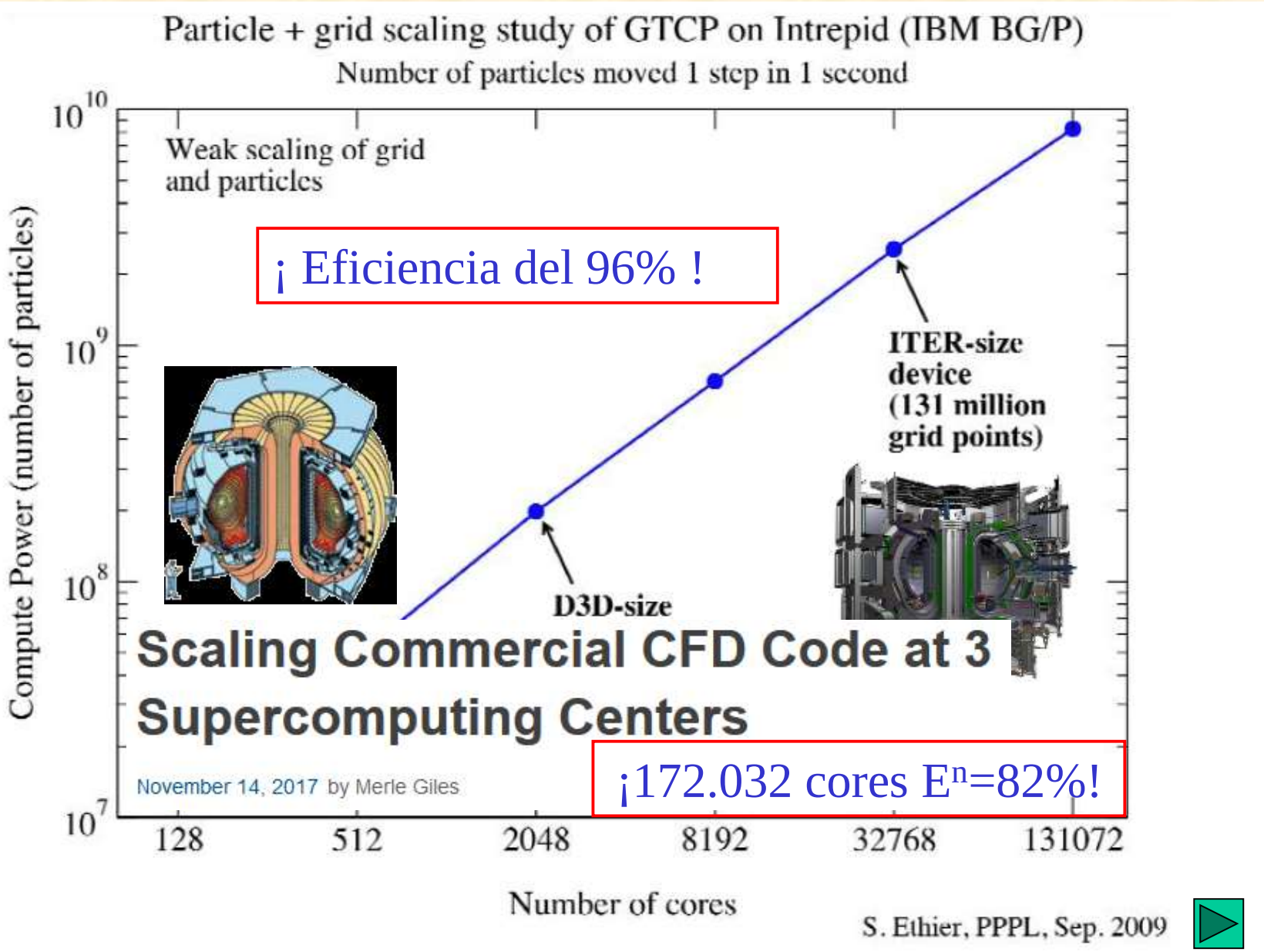
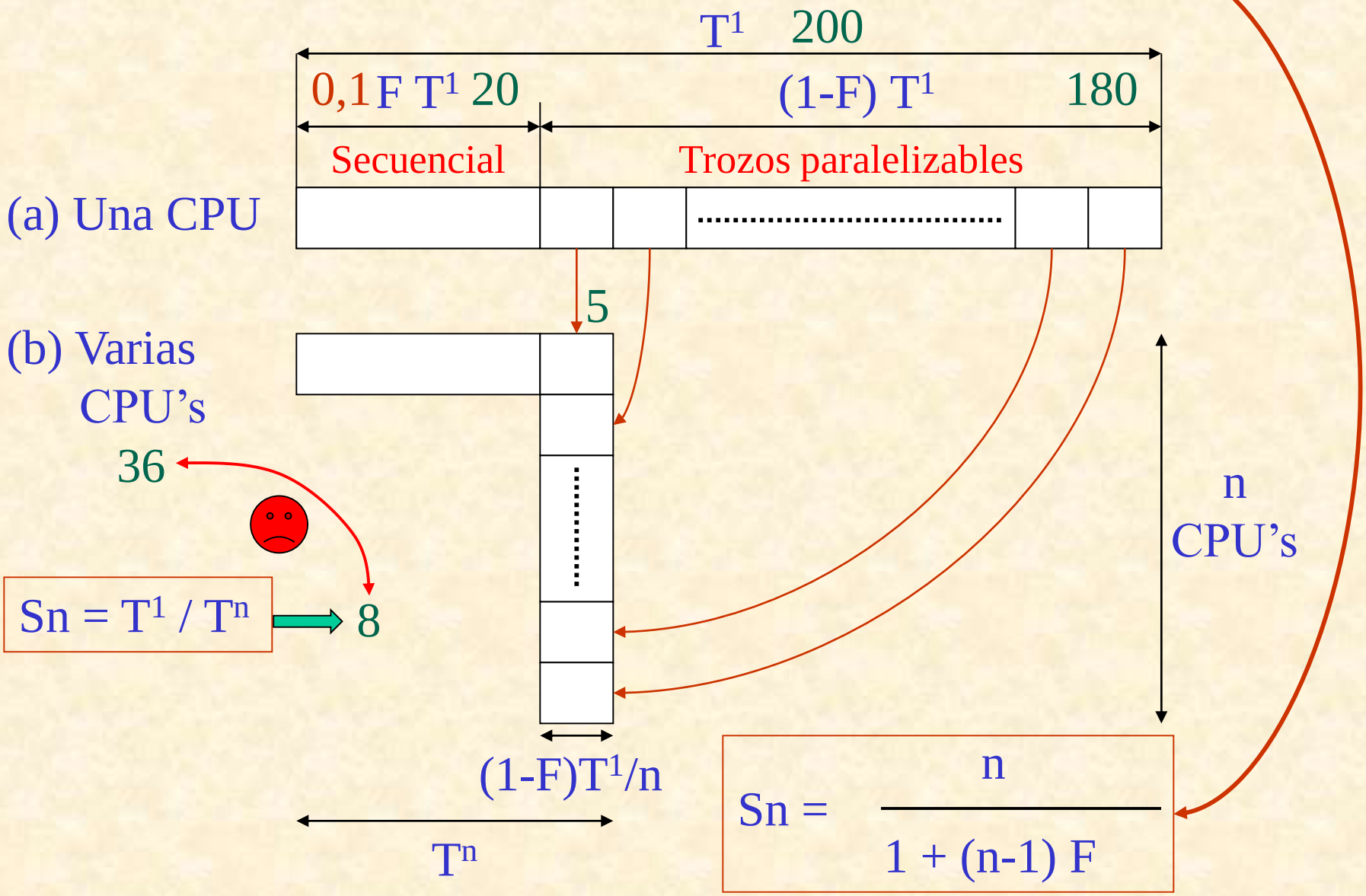


Figure 5: Speedups of the GTC-P code on the quad-core IBM Bkye-Gene-P (courtesy of S. Ethier)

¿Máxima Aceleración? – Ley de Amdahl



Significado

$$Sn = \frac{n}{1 + (n-1)\,F}$$

¡ F=5% => Lim Sn = 20 !

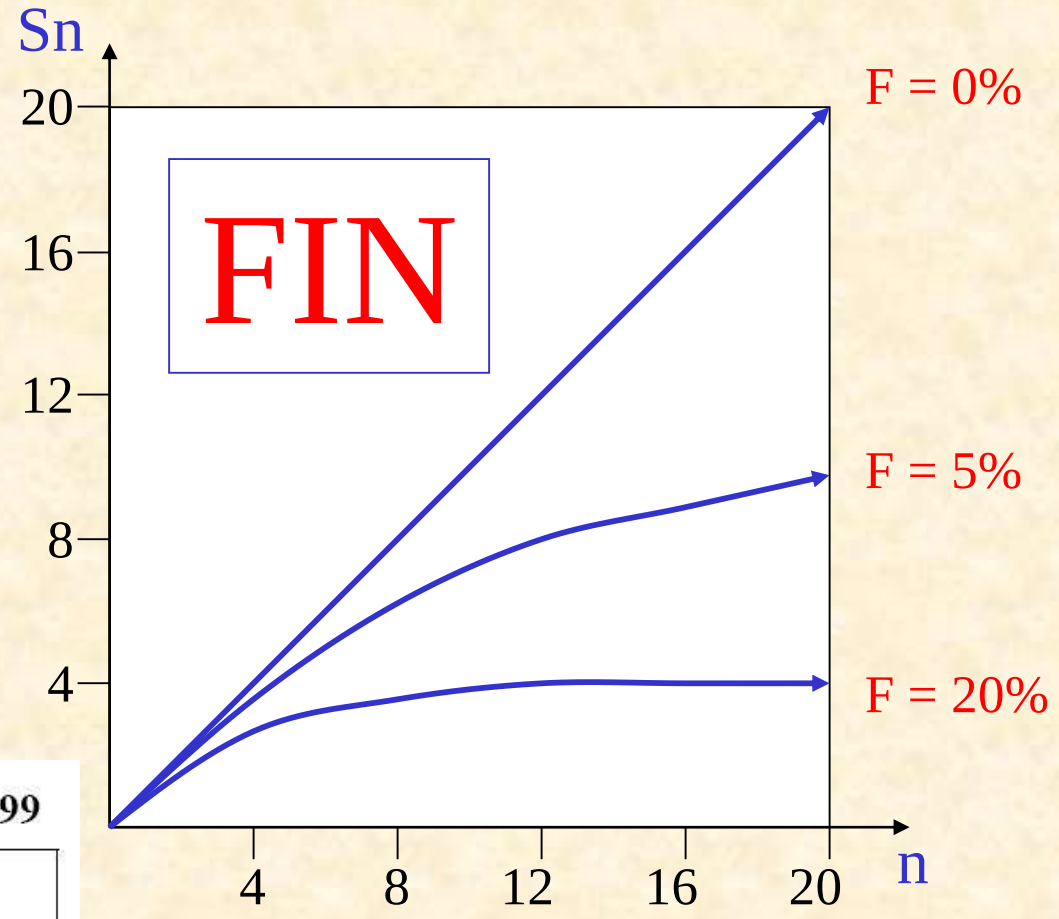


TABLE 1. Upper Bound on Speedup, f=0.99

# Base Core Equivalents	Base Amdahl	Symmetric	Asymmetric	Dynamic
16	14	14	14	< 16
64	39	39	49	< 60
256	72	80	166	< 223
1024	91	161	531	< 782

"Amdahl's Law in the Multicore Era"

Mark D. Hill & Michael R. Marty 2007