1

Comentarios sobre Paralelización

2

- Modelos de programación paralela
- Beneficio de ejeución en paralelo
- Ejecución en paralelo
- SpeedUp versus Eficiencia
- Escalabilidad
- Ley de Amdahl

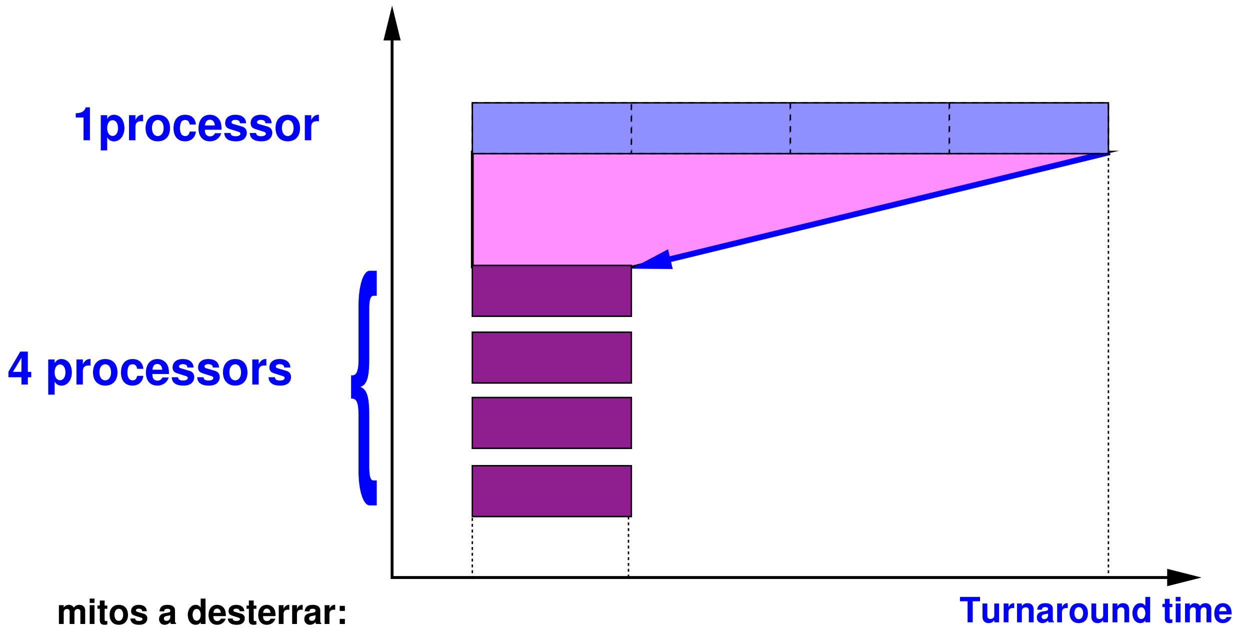
Modelos de programación paralela

3

- automática: PowerF77, PowerC (–pfa, –pca)
- semiautomática: directivas (–mp)
- manual: llamadas a libreria MP
- sproc()'s
- ADA95 tasks
- POSIX threads (pthreads)
- fork(), exec()
- PVM (TCP sockets & SMP version)
- MPI (TCP sockets & SMP version+HiPPi bypass)
- RCP (UDP/TCP)

Beneficio de ejecución paralela

4

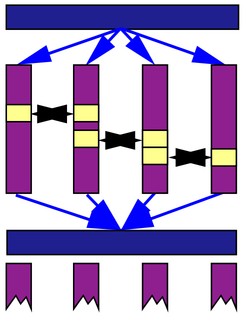


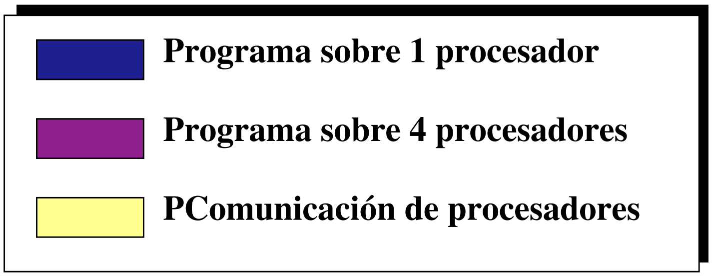
mitos a desterrar:

- con 10 cpu's sólo es 5 veces más rápido !!!
- una máquina XXX me da mejores Speedup's que SGI
- no vale la pena utilizar tantas CPU's porque sólo se gana X y se gasta mucha CPU...

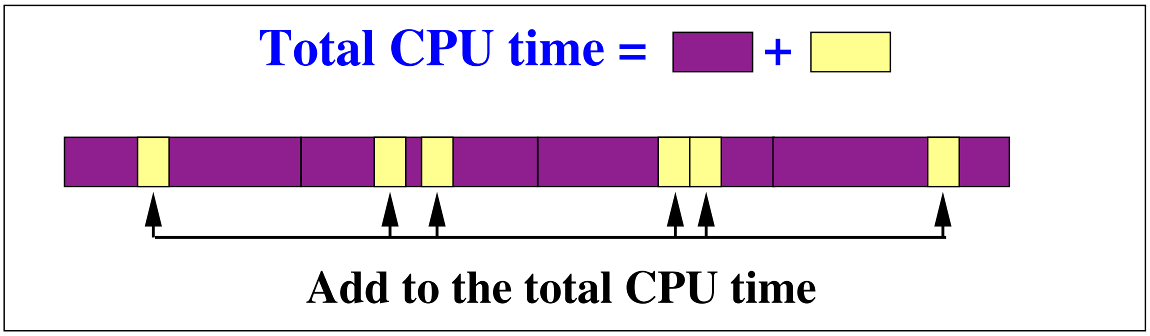
Ejecución en paralelo

5





La paralelización eficiente minimiza la comunicación



Speed–Up vs Efficiency

6

- Both are a measure of how well a program has been parallelized
- Let T(n) be the turnaround time on n CPUs
- The *Speed up* S(p) and *Efficiency* E(p) are then defined as:

  S(n) = T(1)/T(n) (n=1,2,3,....) E(n) = S(n)/n

- For a perfect parallel speed–up we get:

  T(n) = T(1)/n

  S(n) = T(1)/T(n) = T(1)/(T(1)/n) = n

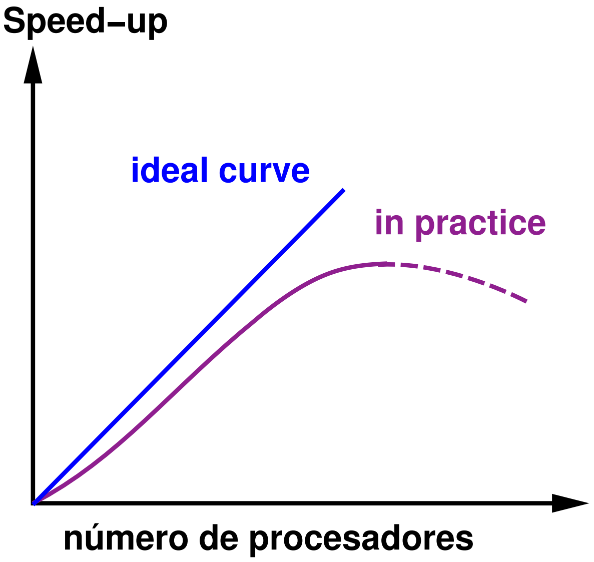
  E(n) = S(n)/n = n/n = 100%

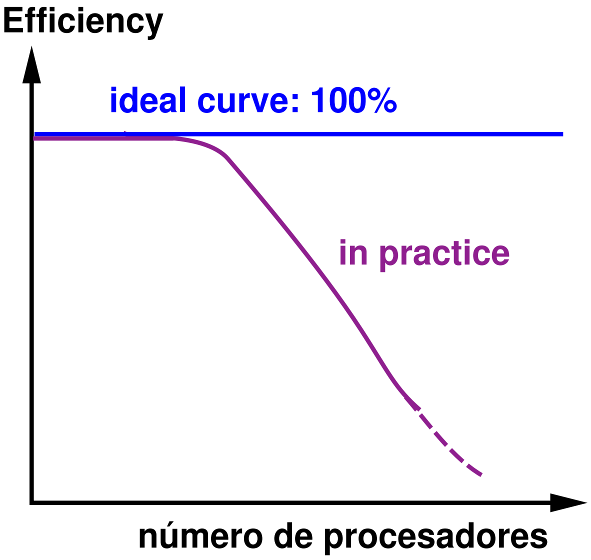
Ejemplo

Time on 1 CPU 100seconds (= T(1))
Time on 15 CPUs 12seconds (= T(15))

Speed–up = 100/12 = 8.3

Efficiency = 8.3/15 = 56%





Escalabilidad

- Escalability es un término un poco vago...
- Normalmente se refiere a que la E(n) se mantiene constante en un rango muy amplio.
- En general, un sistema con procesadores malos escala mejor.

Escalabilidad

9

- A program is 80% parallel: T(1) = 80 + 20 (seconds)
- The processors of System B are <u>2x faster</u> than the processors of System A
- The additional <u>parallel overhead on</u>, say, 4 CPUs is 5 seconds on *both* systems.

On 4 processors we obtain the following:

<u>System A</u>

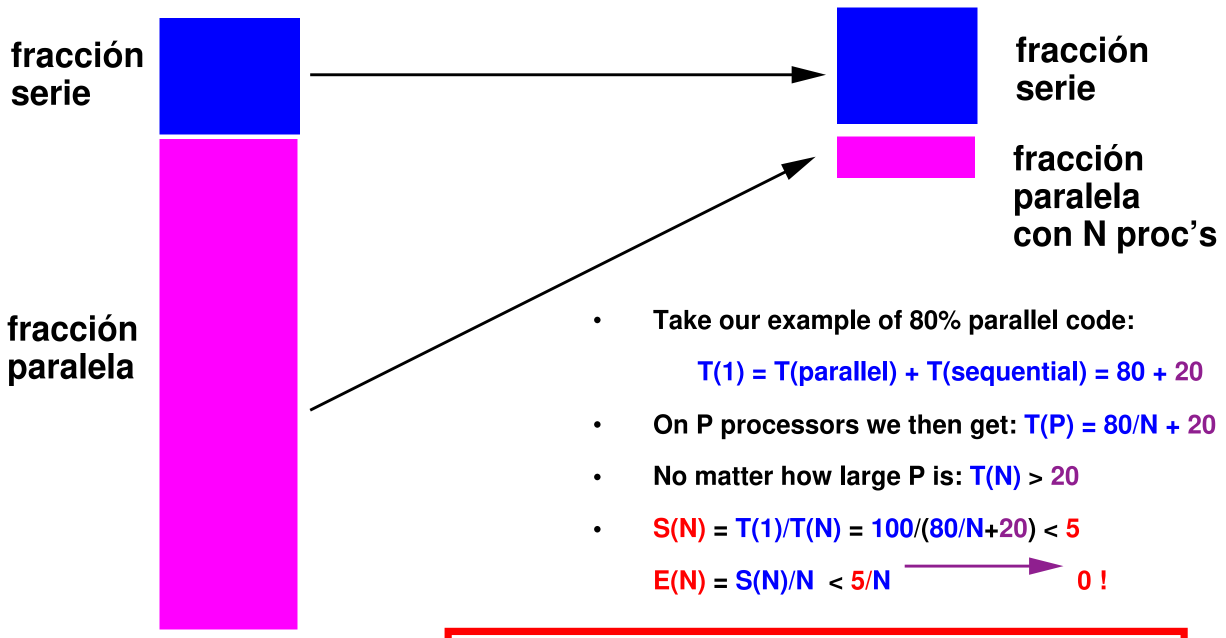
T(1)	=	100	=	80+20	
T(4)	=	80/4+20+5	=	20+20+5=	45
Speed–up	=	100/45	=	2.22	

<u>System B</u>

T(1)	=	50	=	40+10	
T(4)	=	40/4+10+5	=	10+10+5=	25
Speed–up	=	50/25	=	2.00	

Ley de Amdahl

10



- Take our example of 80% parallel code:

  T(1) = T(parallel) + T(sequential) = 80 + 20
- On P processors we then get: T(P) = 80/N + 20
- No matter how large P is: T(N) > 20
- S(N) = T(1)/T(N) = 100/(80/N+20) < 5

  E(N) = S(N)/N < 5/N $\longrightarrow$ 0 !

$$\boxed{S(N) = \frac{T(1)}{T(N)} = \frac{100}{s+\frac{p}{N}} \le \frac{100}{s}}$$

Ley de Amdahl

11

tiempo en una cpu T(1)
tiempo en dos cpu T(2)

con ello se puede despejar la parte serie y la parte paralela:

$$\boxed{p = \frac{N}{N-1} \; \frac{S(N)-1}{S(N)} * 100}$$

T(1) = 100
T(2) = 60

p = (2/1) ((1.67–1)/1.67) *100 = 80 %

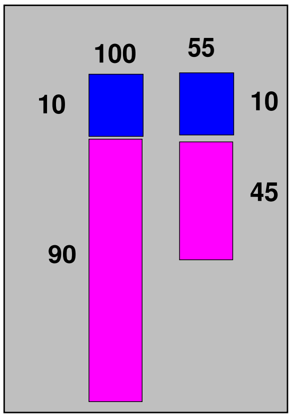
Ley de Amdahl

12

T(1) = 100
T(2) = 55

S = 100 / 55 = 1.81

$$p = \frac{N}{N-1} \; \frac{S(N)-1}{S(N)} * 100 = \frac{2}{1} \; \frac{1.81-1}{1.81} * 100 = 90\%$$



Supongamos que T(10) = 20

El usuario se queja porque:

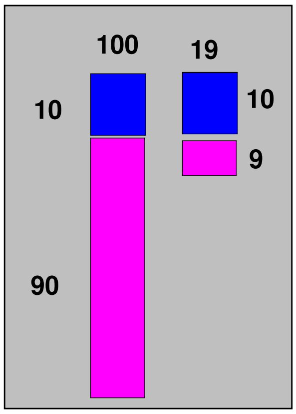
S(10) = T(1) / T (10) = 100/20 = 5 <<< 10

Pero no debería quejarse porque la Ley de Amdhal
le da un máximo de T(10) = 10 + 90/10 = 19

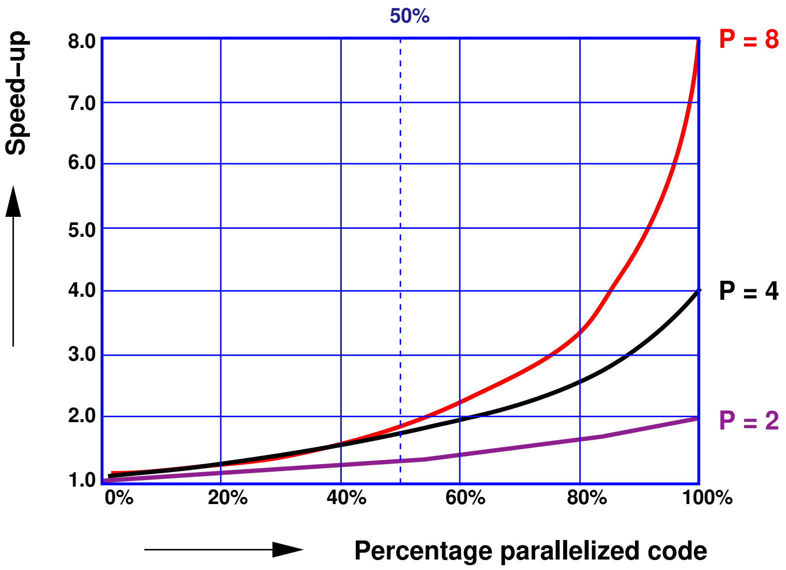
aunque la Eficiencia sea: E(N) = 5/10 = 50 %

la eficiencia respecto a la ley de Amdahl es de:

Eficiencia = 19 / 20 = 95%



Ley de Amdahl



$$p = \frac{N}{N-1} \; \frac{S(N)-1}{S(N)} * 100$$