DESARROLLO DE SOFTWARE EN ARQUITECTURAS PARALELAS

1. Motivación y aspectos de la programación paralela.
2. Tipos de sistemas paralelos. Paradigmas de programación paralela.
3. Conceptos básicos y medidas de paralelismo.
4. Diseño de programas paralelos.
5. La interface de paso de mensaje: el estándar MPI.
6. Paralelización de algoritmos: ejemplos y aplicaciones.

DSAP

Conceptos básicos y medidas de paralelismo

- Se llama **grado de paralelismo** de un algoritmo numérico al numero de operaciones en el algoritmo que pueden ser realizadas en paralelo.
- Ejemplo: Consideremos el problema de sumar dos vectores ***a*** y ***b*** de longitud n.
  - El algoritmo que podemos usar sería:

```
Para i=1, 2, ..., n
        ci = ai + bi
```

  - Las n sumas son independientes y pueden realizarse en paralelo.
  - El grado de paralelismo es, entonces, n.
  - Notar que el grado de paralelismo es independiente del número de procesadores. Si n=1000 y p=1000, el algoritmo se realiza en un paso. Si n=1000 y p=10, necesitaríamos 100 pasos.

Conceptos básicos y medidas de paralelismo

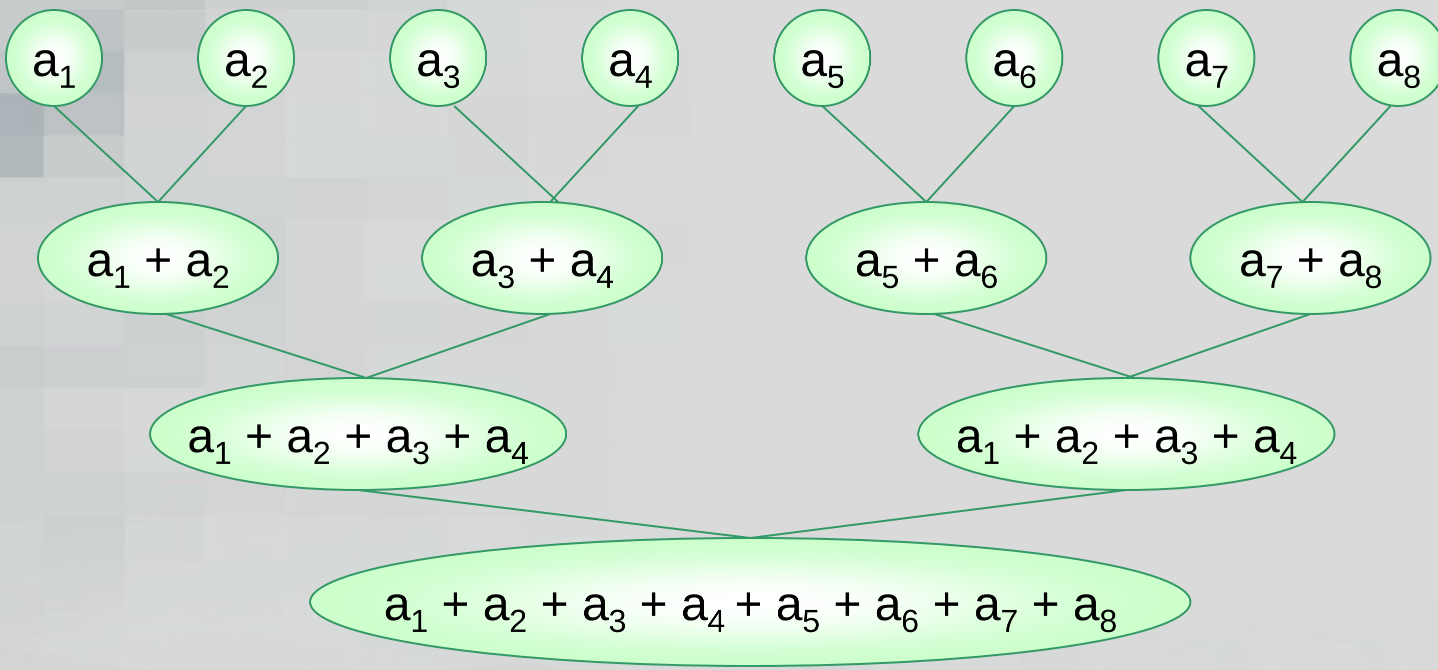
- Ejemplo: Consideremos el problema de sumar n números a_1, a_2, …, a_n.
- El algoritmo secuencial usual sería:

```
s=a1
Para i=2, 3, ..., n
    s = s + ai
```

- Este algoritmo no es posible para un cálculo paralelo.
- Sin embargo, es posible redefinir el algoritmo de otra forma.
- Veamos un ejemplo para n=8.

DSAP

Conceptos básicos y medidas de paralelismo



Etapa	Op. Par.
Etapa 1	4
Etapa 2	2
Etapa 3	1

- En general, para $n = 2^q$, este algoritmo requiere:
  - q=log(n) etapas.
  - Con n/2, n/4, n/8, … operaciones en cada etapa. Las operaciones de cada etapa son independientes.

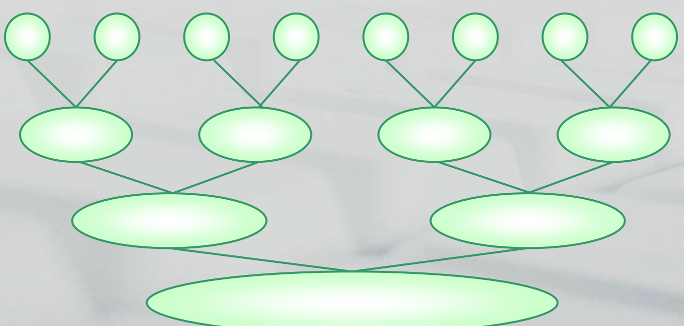
Conceptos básicos y medidas de paralelismo

- Se llama **grado medio de paralelismo** de un algoritmo numérico al numero de operaciones en el algoritmo que pueden ser realizadas en paralelo dividido por el numero de etapas.
- Para sumar $n=2^q$ números el GMP es:

$$\frac{1}{q}\left(\frac{n}{2}+\frac{n}{4}+\ldots+1\right)=\frac{1}{q}\left(\frac{2^q}{2}+\frac{2^q}{2^2}+\ldots+2+1\right)=\frac{1}{q}\left(2^{q-1}+2^{q-2}+\ldots+2+1\right)$$

$$=\frac{1}{q}\left(\frac{1\left(2^q-1\right)}{2-1}\right)=\frac{2^q-1}{q}=O\left(\frac{n}{\log(n)}\right)$$

$$\frac{a_1\left(r\,a_n-1\right)}{r-1}$$



Conceptos básicos y medidas de paralelismo

- Se llama **incremento de velocidad** o **speed-up** a la aceleración experimentada por un programa al hacer uso de *p* unidades de cálculo (CPU) en vez de una única:

$$S_p = \frac{t_{serie}}{t_{paralelo}}$$

  - Obviamente $S_p \leq p$.

Conceptos básicos y medidas de paralelismo

- Se llama **incremento de velocidad** o **speed-up respecto al mejor algoritmo secuencial** a la aceleración experimentada por un programa al hacer uso de *p* unidades de cálculo (CPU) comparado con el mejor algoritmo secuencial:

$$S'_p = \frac{t_{mejor\_serie}}{t_{paralelo}}$$

  - Obviamente $S'_p \leq S_p \leq p$.

Conceptos básicos y medidas de paralelismo

- Se llama **eficiencia** de un algoritmo paralelo, respecto a sí mismo a:

$$E_p = \frac{S_p}{p}$$

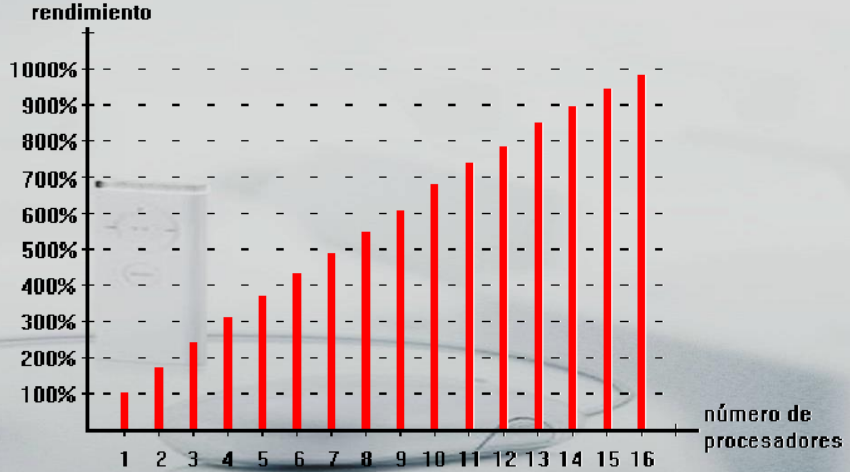
- Siendo la eficiencia respecto al mejor algoritmo secuencial:

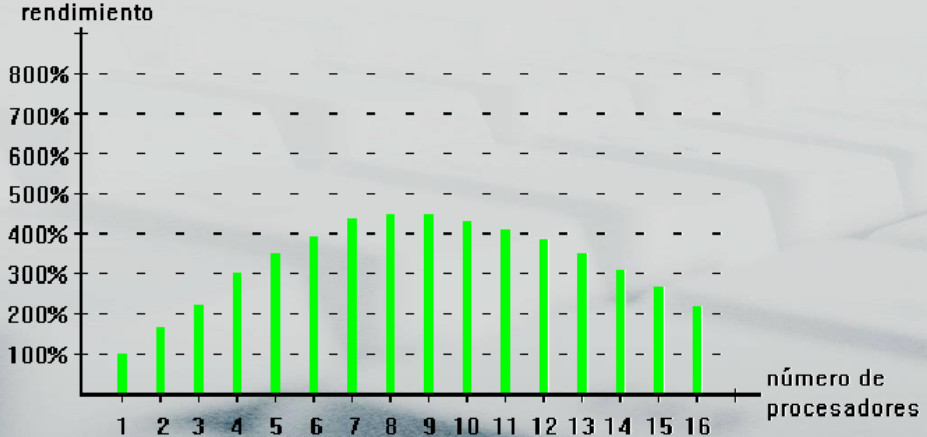
$$E'_p = \frac{S'_p}{p}$$

- Obviamente $E'_p \leq E_p \leq 1$.

Conceptos básicos y medidas de paralelismo

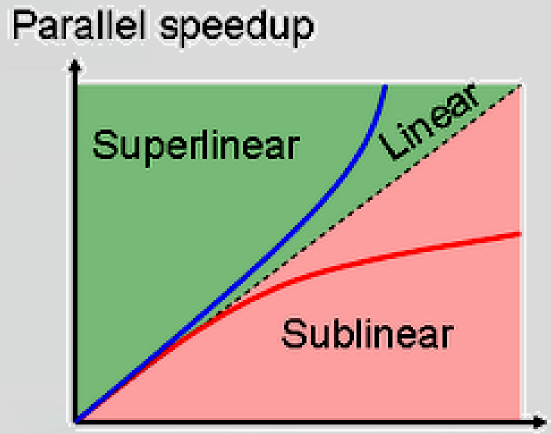
- La **escalabilidad** de un sistema paralelo refiere a la habilidad para proporcionar un incremento proporcional del speed-up con el aumento del número de unidades de cálculo (CPU). Depende de:
  - Hardware: red de interconexión.
  - El algoritmo concreto que se esté usando.
  - Parallel overhead (tiempo de inicialización y finalización de tareas, sincronizaciones, comunicaciones, tiempos adicionales impuestos por el lenguaje, librería, SO, …).





Conceptos básicos y medidas de paralelismo

- Se llama **speed-up superlineal** a la habilidad de un programa de tener una eficiencia paralela mayor que la unidad.



- **Comunicaciones**: acción de envío de datos o instrucciones entre procesos pertenecientes a un mismo programa.
- **Latencia:** el tiempo requerido por la red de comunicaciones en inicializar el envío de datos.
- **Ancho de banda**: cantidad de datos que se pueden enviar en un segundo a través de la red de comunicaciones.

Conceptos básicos y medidas de paralelismo

- **Coste de las comunicaciones** entre dos máquinas cualesquiera. Por simplicidad, suele asumirse que el coste de las comunicaciones punto a punto entre las máquinas de una red se puede caracterizar mediante el modelo lineal clásico:

$$t_{com} = \beta + \tau \cdot \text{Tamaño\_Mensaje}$$

donde:

- β es la latencia de la red, que es independiente del tamaño del mensaje,
- y τ es el tiempo de transferencia por byte, que es la inversa del ancho de banda del enlace entre los procesadores.

Conceptos básicos y medidas de paralelismo

$$t_{com} = \beta + \tau \cdot \text{Tamaño\_Mensaje}$$

- El coste de comunicación entre dos procesadores modelado de esta forma incluye el tiempo que tarda un procesador en enviar datos y el que invierte el segundo procesador en recibirlos.
- En protocolos de comunicación que fragmentan los mensajes en bloques (como el protocolo TCP/IP), el valor del tiempo de transferencia será diferente para mensajes pequeños y grandes.
- La solución es estimar el parámetro T haciendo variar el tamaño de los mensajes.
- Se deben usar dos mensajes (uno de ida y otro de vuelta, usualmente conocido como ping/pong) y dividir el tiempo entre dos.
- Algunas ideas: realizar varias repeticiones, tomar medias, descartar valores atípicos, …

DSAP

Conceptos básicos y medidas de paralelismo

- Límites a la paralelización: Ley de Amdahl.
  - Sea $T_{original}$ el tiempo original que se tarda en ejecutar una determinada aplicación.
  - Sea *f* la fracción de código que es paralelizable.
  - Obviamente:

$$T_{original} = (1\text{-}f) \times T_{original} + f \times T_{original}$$

  - Si *p* es el número de procesadores a usar, el tiempo mejorado $T_{mejorado}$ se obtendrá reduciendo *p* veces la parte paralelizable:

$$T_{mejorado} = (1\text{-}f) \times T_{original} + (f \times T_{original})/p$$

  - Con lo que el speed-up teórico sería:

$$\text{Speed-up teórico} = \frac{T_{original}}{T_{mejorado}} = \frac{T_{original}}{(1-f)\times T_{original} + (f \times T_{original})/p} = \frac{1}{(1-f) + f/p}$$

Conceptos básicos y medidas de paralelismo

- Límites a la paralelización: Ley de Amdahl.

$$\text{Speed-up teórico} = \frac{1}{(1-f)+f/p}$$

  - *f:* fracción de código que es paralelizable.
  - *p:* número de procesadores a usar.
  - La parte no paralelizable **limita la escalabilidad:**