Paral·lelisme

- Race condition, starvation, dead lock, live lock.
  - dead lock → solució: ordre de lock predeterminat (ex. alfabètic)

- T_1 = temps lineal
  T_∞ = camí mínim
  Parallelism = T_1 / T_∞
  Parallel slackness = Parallelism / P_{min}

- Efficiency$_P$ = $T_1 / (T_P \cdot P)$
  T_P (temps amb P processadors)
  Speedup$_P$ = T_1 / T_P

[Amdahl's Law]

- ρ - fracció de temps que el programa s'executa en paral·lel

$T_P = (1-\rho)T_1 + \left(\rho \cdot \frac{T_1}{P}\right)$ + overhead

Sp amb $P \to \infty$: $\frac{1}{1-\rho}$

- Strong scaling (canvi temps, canvi P, problema fixat)
  Weak scaling (canvi temps, canvi P, problema canvia però seient per P fixada)

[Data sharing]

- Taccess = ts + m·tw
  - m: bytes

- Pipelining, ILP (instruction level P.)
  TLP (thread level parallelism) (hyperthreading)
  DLP (data level paral.) (SIMD)

- SMP (symmetric multi-processor)
  - tots els processadors tenen accés igual a tota la memòria (UMA)
  - un sol SO
  - bottleneck: bus de memòria
  - coherència de memòria
    - snooping-based coherace protocol
      - write-update (↑ trafic al bus)
      - write-invalidate (dirty = exclusive ownership)
        MSI / MESI
  - "False sharing": dos processadors escriuen freqüentment a la mateixa línia de cache
  - Consistència entre escriptures a diferents variables

- Memory cache
  - Temporal locality (eg. loop)
  - Spatial locality (eg. array)

- NUMA
  - "hub" entre cpu i memòria per assegurar coherència
  - directory-based cache coherency protocol

- Distributed-memory architectures
  - OS level (on page fault)
    user level (message-passing library)
  - network topology: direct / indirect (arbre) etc.
    - latència (timeline / mesh)
    - ample de banda