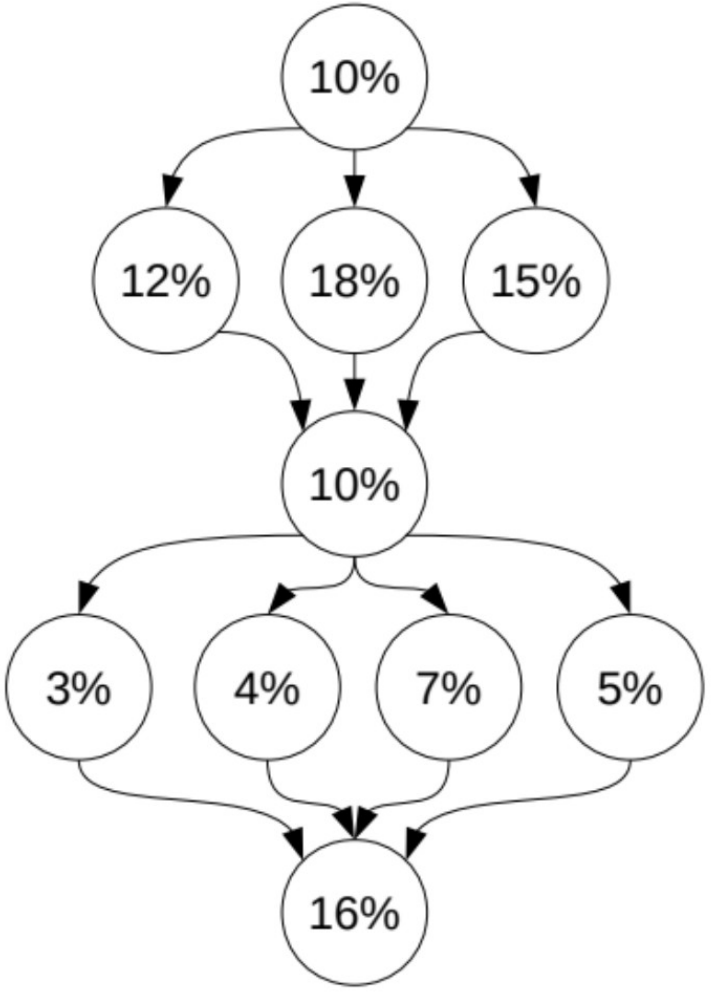
Outline of classes 3 and 10 November:

- Parallel vs. distributed processing
- Types of parallel machines:
  - Memory system: Multicomputer, multiproc → UMA, SMP, communication between instruction flows
- Types of parallelism: functional, data
- Visibility of parallelism: implicit, explicit
- Levels and granularity of parallelism: process/application, function/method, ...
- Parallel programming modes: SPMD, MPMD, mixed
- Communication alternatives: broadcasting, scatering, ...
- **Programing styles**: Message passage, shared variables, ...
- **Parallel program structure**: Master-slave, data parallelism, segmentation, ...
- **Assignment** of tasks
- Example and problems

Esquema clases **15 y 22 de noviembre**:

- Procesamiento **paralelo** vs **distribuido**
- Clasificación máquinas paralelas:
  - Sistema de memoria: **Multicomput**, **multiproc** → UMA, SMP, comunicación entre flujos de instrucciones
- **Tipos** de paralelismo: funcional, de datos
- **Visibilidad** del paralelismo: implícito, explícito
- **Niveles** y **granularidad** del paralelismo: proceso/aplicación, función/método, …
- **Modos** de programación paralela: SPMD, MPMD, mixto
- Alternativas de **comunicación**: difusión, …
- **Estilo** prog. Paralela: Paso de mensaje, variables compartidas, …
- **Estructura** programa paralelo: Master-slave, paralelismo de datos, segmentado, …
- **Asignación de tareas**
- Ejemplo y problemas

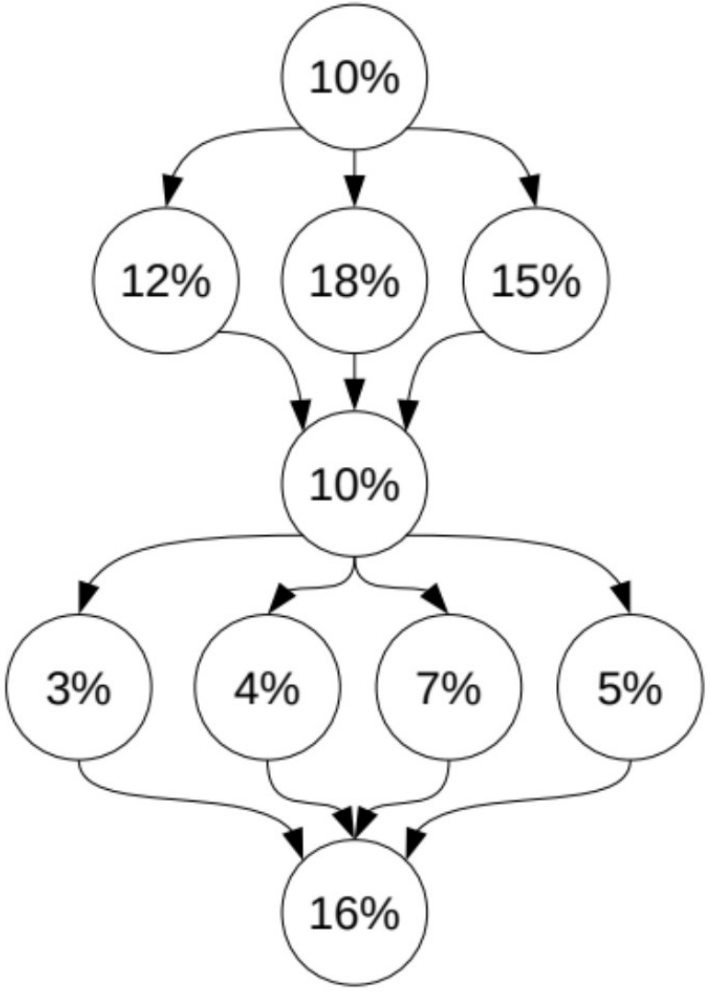


El gráfico muestra el grafo de dependencias entre tareas de una aplicación paralelizada. Suponga $t_{overhead}(p)$ despreciable. Se pide:

a) $S_p(p)$ para p= 2 y 3

b) Tenemos 3 máquinas con n=2,3 y 4, ¿cuál es la más eficiente?

Ojo: Es un problema de asignación de tareas a hebras/procesos.



The graph shows the network of dependencies between tasks of a parallelized application. Suppose $t_{overhead}(p)$ is negligible. Calculate:

a) Sp(p) for p= 2 & 3

b) If we have 3 parallel machines with n=2, 3 and 4, which is the most efficient?

Note: This is a problem of task (threads/processes) assignment.

El **25%** de un programa no se puede paralelizar. **El resto sí**, es decir, podemos distribuir la carga entre un número cualquiera de elementos de procesamiento (*threads*, procesos, cores, nodos, ...).
[English] If only 75% of a program can be parallelized, so we can distribute the load between any number of processing elements (threads, processes, cores, nodes, …), calculate:

1) Calcule la ganancia paralela en velocidad para p unidades de procesamiento. Calculate the speed-up for p processing elements
2) ¿Qué pasa cuando p tiende a + infinito? What if p tends to +infinity?
3) Calcule la eficiencia paralela para el caso 1) y 2). Revisit 1) & 2) calculating the parallel eficiency.
4) ¿A partir de cuántas unidades de ejecución la ganancia paralela de velocidad es superiror a 2? Value of p, from which, the speed-up are >2

Un programa totalmente paralelizable tarda 20s en ejecutarse en un procesador **P1** y 30s en otro procesador **P2**. [English] A fully parallelizable program takes 20s to run on a given processor P1 and 30s on another processor P2.

Despreciando la sobrecarga calcule: Neglecting the overload, calculate:

a) Tiempo de ejecución paralela si distribuimos equitativamente la carga entre P1 y P2. Ganancia paralela en velocidad y eficiencia paralela. Parallel execution time if we distribute the load equally between P1 and P2. Parallel speed-up and efficiency.

b) Distribución de carga que optimiza la ejecución paralela. Calcule en este caso el tiempo de ejecución paralela, la ganancia paralela en velocidad y la eficiencia paralela. Load distribution that optimizes parallel execution. Calculate in this case the parallel time, parallel speed-up and efficiency