

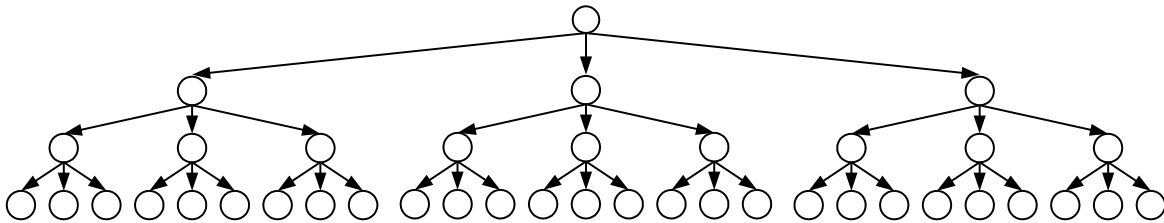
## PROGRAMACIÓN PARALELA Y DISTRIBUIDA

### PRACTICA N° 4: EVALUACIÓN Y RENDIMIENTO DE LAS APLICACIONES PARALELAS

**TEMAS:** Índices de rendimiento. Rendimiento de aplicaciones paralelas.

**OBJETIVOS:** Que el alumno analice, interprete, evalúe y valore el comportamiento paralelo frente al secuencial: ganancias, ventajas, desventajas, relación de compromiso.

**Ejercicio N°1** – En la siguiente figura se presenta un grafo de dependencias. Se asume que la ejecución de cada tarea insume una unidad de tiempo y que la comunicación entre procesos es instantánea (es decir que no consume tiempo).



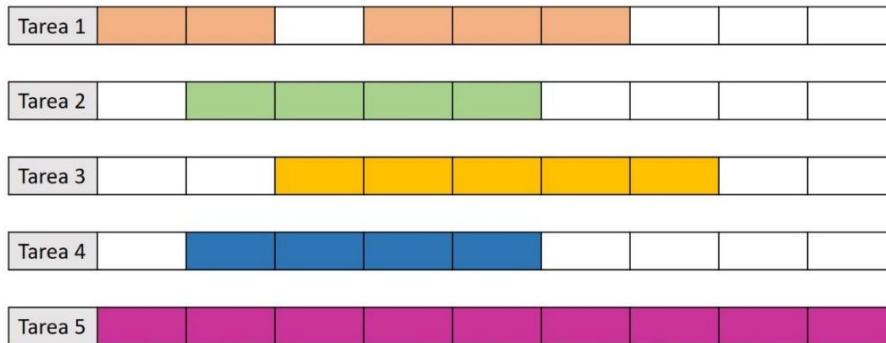
Se proponen las siguientes actividades:

- Calcular el grado de concurrencia.
- Calcular el máximo Speedup posible si se dispone de una cantidad ilimitada de nodos computacionales.
- Calcular Speedup y Eficiencia si el número de nodos computacionales es igual al grado de concurrencia.
- Calcular Speedup y Eficiencia si el número de nodos computacionales es igual a la mitad del grado de concurrencia.
- Calcular Speedup y Eficiencia si el número de nodos computacionales es igual a la tercera parte del grado de concurrencia.
- Graficar el factor de Speedup y la Eficiencia, y explicar los resultados obtenidos.
- ¿Qué observaciones se pueden realizar en cuanto a balanceo de carga? ¿Qué reflexión puede realizarse en cuanto a la escalabilidad?

**Ejercicio N°2** – Suponer que se desea calcular el valor de una variable llamada  $s$ , cuyo valor se obtiene al sumar los primeros 16 números naturales. Resolver los siguientes incisos:

- Suponer que la ejecución de cada suma requiere 1 unidad de tiempo (UT). Determinar cuántas operaciones (sumas) deben realizarse, y determinar cuánto tiempo se requiere si las cuentas se pudieran organizar en 1, 2, 4, 8, y 16 nodos.
- Calcular el Speedup y la Eficiencia en cada caso.
- Construir la correspondiente tabla de valores.
- Graficar y analizar los resultados. Valorarlos e interpretarlos teniendo en cuenta el balanceo de carga y la escalabilidad.

**Ejercicio N°3** – Para el siguiente diagrama:



- a) Proponer un grafo de dependencias para representar las dependencias de **inicio** de cada tarea.
- b) Considerando que la ejecución de cada subtarea (celda) demanda 1 UT (unidad de tiempo), completar la siguiente tabla para distintos valores de  $n$  (nodos involucrados).

$n$	$T(n)$	Speedup( $n$ )	Eficiencia( $n$ )

- c) Realizar el gráfico de speedup correspondiente y analizar los resultados.
- d) Decidir qué consideraciones y decisiones es posible tomar.

**Ejercicio N°4** – Dada la función, donde se supone que las funciones  $T_1$ ,  $T_3$  y  $T_4$  tienen un coste de  $n$  y las funciones  $T_2$  y  $T_5$  de  $2n$ , siendo  $n$  un valor constante:

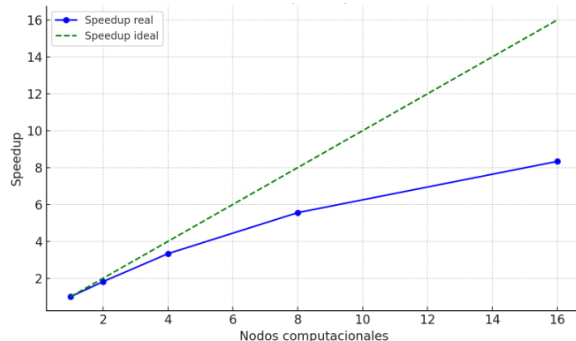
- a) Dibujar el grafo de dependencias y calcular el coste secuencial.
- b) Calcular el tiempo de ejecución paralelo y el speedup con dos procesos.
- c) Analizar y valorar los resultados en base a los índices estudiados.

```
double ejemplo(int i, int j)
{
    double a, b, c, d, e;
    a = T1(i);
    b = T2(j);
    c = T3(a+b, i );
    d = T4(a/c);
    e = T5(b/c);
    return d+e; /*T6*/
}
```

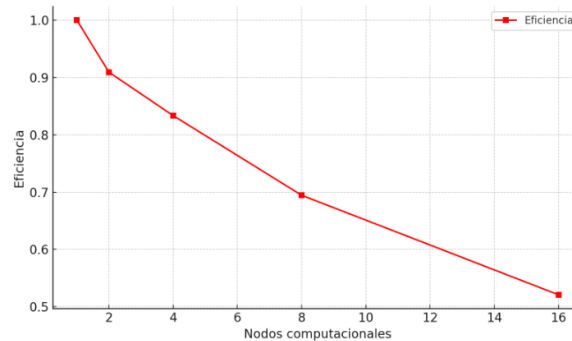
**Ejercicio N°5** - Se presentan los siguientes gráficos obtenidos al resolver una tarea paralela utilizando un sistema con 1, 2, 4, 8 y 16 nodos computacionales. El tiempo de ejecución secuencial de la tarea es de 100 unidades de tiempo (UT).

**Gráfico A Speedup vs Nodos:** Muestra el Speedup para distintos números de nodos, comparado con el Speedup ideal.

**Gráfico B Eficiencia vs Nodos:** Representa cómo varía la eficiencia en función del número de nodos computacionales.



**Gráfico A**



**Gráfico B**

Responder las siguientes preguntas:

- ¿Qué factores podrían explicar la disminución del Speedup observado en el gráfico a medida que se incrementa el número de nodos?
- ¿Por qué la eficiencia puede disminuir cuando el número de nodos computacionales aumenta?
- Analizando los gráficos, ¿en qué momento el aumento de nodos deja de ser beneficioso para el rendimiento?
- ¿Qué estrategias se podrían implementar para minimizar la disminución de eficiencia con un alto número de nodos?