

# Test-tema2AC.pdf



**Sanchez01**



**Arquitectura de Computadores**



**2º Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**  
**Universidad de Granada**

## Test tema 2 AC

1. En un multicomputador con 4 procesadores (P0 a P3), mediante la comunicación de recorrido (scan) sufijo paralelo, el procesador P2 envía información los procesadores P0, P1, y al propio P2 (aparte de otras posibles comunicaciones).
2. En un multicomputador con 4 procesadores (P0 a P3), mediante la permutación de rotación el procesador P3 envía información al procesador P0 y recibe del P1 (aparte de otras posibles comunicaciones).
3. La expresión para la ley de Gustafson es  $S=f+p*(1-f)$ , donde  $f$  es la fracción no paralelizable del tiempo de ejecución paralelo y  $p$  es el número de procesadores que intervienen.
4. Un programa paralelo tarda 200 ns. Durante 50 ns solo puede ser ejecutado por un procesador y durante los otros 150 ns intervienen 4 procesadores (todos ellos igual de cargados). La sobrecarga se considera despreciable. El valor de la ganancia de velocidad es menor que 3.
5. Un programa paralelo tarda 40 ns en ejecutarse en un procesador y durante 10 ns de esos 40 ns el programa no es paralelizable, mientras que en el resto del tiempo paralelo intervienen cinco procesadores cargados por igual. El valor de la  $f$  de la ley de Gustafson para ese programa es igual a 0.25.
6. El tiempo de sobrecarga u overhead de un programa paralelo se debe únicamente al tiempo de comunicación entre los procesadores.
7. La ganancia de velocidad que consiguen  $p$  procesadores en un código secuencial que tarda un tiempo  $T_s$  en ejecutarse en un procesador, con una fracción no paralela de  $T_s$  igual a 0, un grado de paralelismo igual a  $n$  y un tiempo de overhead igual a  $p$  (Crece linealmente con  $p$ ) es igual a  $T_s/((T_s/n)+n)$ , para  $p>$ .
8. En un computador MIMD no se puede utilizar el modo de programación SPMD (Single Program Multiple Data).
9. En la asignación dinámica de tareas a procesos, está determinado antes de la ejecución qué tarea va a realizar cada procesador o core.

10. Dado el bucle
- ```
for (i=0;i<Iter;i++) {  
    código para i  
}
```

Mediante

```
for (i=idT;i<Iter;i=i+nT) {  
    código para i  
}
```

Se consigue la distribución estática de las  $Iter$  iteraciones del bucle entre  $nT$  hebras, cuyo identificador es  $idT$  ( $idT=0,1,...,nT-1$ ).