

# Concorrenca e Paralelismo. Bloque II Paralelismo

## Práctica 4: Asignación dinámica de tareas: ordenación de vectores

Departamento de Electrónica y Sistemas

Primavera 2013



Código secuencial: matvec-order.c

- Partir de una copia de la solución de la práctica 3.
- Modificar el tipo de datos y modo de generar el vector y la matriz:

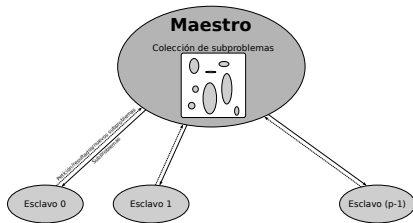
```
double matrix[N][N];
double vector[N];
double result[N];

/* Initialize Matrix and Vector */
gettimeofday( &tv, NULL );
srand( tv.tv_usec );
for(i=0;i<N;i++) {
    vector[i] = (double)rand()/RAND_MAX;
    for(j=0;j<N;j++) {
        matrix[i*N+j] = i*j*((double)rand()/RAND_MAX);
    }
}
```

- Se desea ordenar el vector resultado, que ya se encuentra sustancialmente ordenado.
- El algoritmo de ordenación por inserción directa en estas condiciones es  $O(d \times n)$ , siendo  $d$  el número de elementos desordenados.

## Solución: Paralelización

- El proceso 0 crea una tarea por cada trozo del vector resultado con  $K$  elementos fuera de orden.
- Coste de tarea variable: depende de tamaño de trozo y número de inversiones.
- Esquema de asignación dinámico, máster/slave, para balanceo de carga.



# Solución conceptual

- Implementación maestro/esclavo con  $numprocs \geq 2$ .
- El maestro analiza el vector y crea las tareas atendiendo a dos condiciones:
  - Se ha llegado al máximo número de elementos desordenados, o
  - se ha llegado al tamaño máximo de tarea.
- Se proporciona implementación de la creación de tareas en el fichero `task-creation.c`
- Distribuye inicialmente una tarea por esclavo (envía un trozo de vector a ordenar).
- El esclavo ordena el trozo de vector y lo devuelve al maestro.
- El maestro recibe los resultados y si hay tareas pendientes envía una nueva tarea al esclavo ocioso.
- Si no hay tareas pendientes se envía un código de finalización (`DIE_TAG`) al esclavo.
- El maestro realiza la ordenación global del vector y lo imprime, un elemento por línea.

## Tareas OBLIGATORIAS de la práctica 4

- Implementar el esquema maestro-esclavo propuesto para la ordenación del vector resultado.
- La salida debe ser un vector ordenado. Modo de verificación (el diff no puede devolver nada):

```
./mpi-p4 | grep -v "(PERF)" | tee mpi.out  
sort -n mpi.out | diff mpi.out -
```

## Tareas OPTATIVAS de la práctica 4

- Implementación lo más óptima posible y medición del rendimiento de cada etapa. Comparación con versión secuencial.

## Condiciones de realización

- Deadline: TGR 9-10,15 Mayo (el 08/05 se recuperará la sesión de prácticas que se pierde el 01/05).
- Realización individual.
- Defensa en laboratorio de prácticas.