SIMULADOR DE SISTEMAS SUPERESCALARES Y PARALELOS

R. Rico López, N. Dueñas Miguel, O. Rodríguez Pascual, J. A. de Frutos Redondo

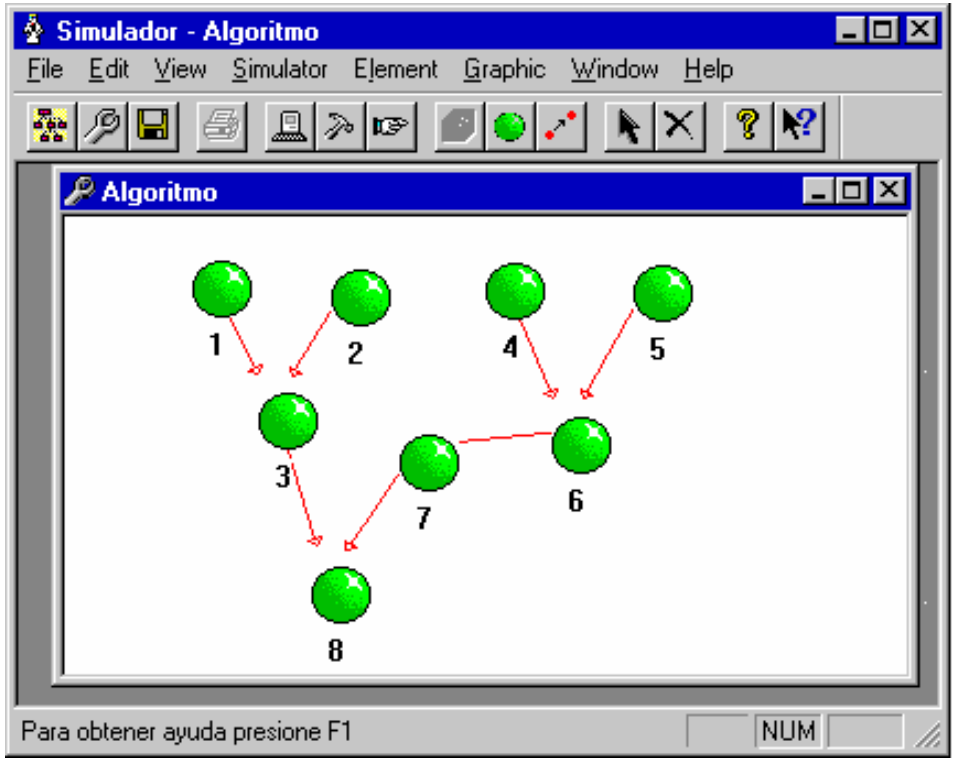
Departamento de Automática
Área de Arquitectura y Tecnología de Computadores
UNIVERSIDAD DE ALCALÁ DE HENARES
{rico@aut.alcala.es}

1. Situación actual y objetivos

La mejora del rendimiento de los procesos computacionales pasa de manera inexorable por el paralelismo que encontramos de muy diversas formas. En los procesadores superescalares, con la multiplicación de las unidades funcionales; en los sistemas multiprocesador, con la interconexión de varios procesadores; en sistemas distribuidos, con las redes de *clusters*.

Todo esto ha hecho que se desarrollen los estudios acerca de las comunicaciones entre procesadores, que se investigue en los protocolos de coherencia y en otros campos relacionados con objeto de conseguir una eficiente conexión de los elementos de cálculo. Sin embargo, no podemos utilizar toda la potencia del *hardware* paralelo si no somos capaces de extraer todo el paralelismo implícito en el particular algoritmo a computar.

Con objeto de estudiar cómo extraer todo el paralelismo subyacente, hemos desarrollado un simulador que posee la capacidad de procesar un algoritmo sobre una red de elementos de procesamiento obteniendo como resultado la temporización de eventos sobre dicha red.



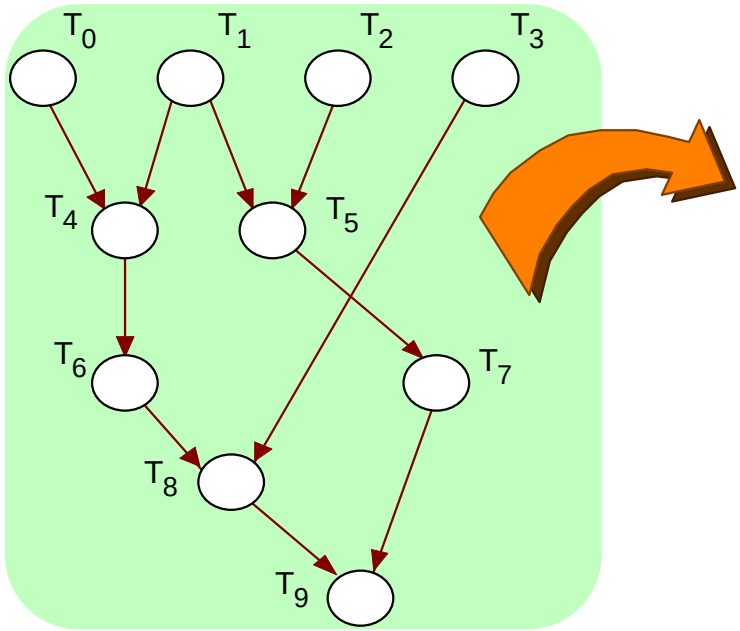
El modelo utilizado permite realizar simulaciones tanto para sistemas multiparalelos, sistemas de tipo *cluster* o procesadores superescalares.

La aplicación cuenta con un editor de redes de procesadores, un editor de algoritmos, un sistema de asignación de algoritmos a redes, una base de datos de características descriptivas de los procesadores y otra para las tareas que componen los algoritmos. La simulación calcula una previsión del tiempo de programa y genera un informe (tabulado y gráfico) en el que obtenemos la carga de trabajo de cada unidad de procesamiento en función del tiempo, así como el porcentaje de tiempo que ha estado trabajando y otros datos de interés.

La herramienta tiene utilidad en el campo didáctico ya que genera información clara y fácilmente comparable entre configuraciones. Ha sido desarrollada inicialmente bajo D.O.S. y posteriormente se ha escrito en Visual C++ para ser ejecutada bajo *Windows 95* y *Windows NT* en plataformas tanto *Intel* como *Alpha*.

2. Conceptos básicos

Un algoritmo se define como un conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. Cada una de esas operaciones se relaciona con las demás por dependencia en los datos y se puede representar por un grafo de flujo de datos. Nosotros hablamos de tareas en lugar de operaciones para hacer más general al concepto y poder adaptarlo tanto a sistemas multiprocesador como a sistemas superescalares. Así, las tareas pueden ser simples instrucciones de un programa secuencial, o bien un procedimiento más extenso, independiente en sí mismo en cuanto a los datos manejados, que se entrega a un procesador en un sistema multiparalelo o de *clusters*.



tarea	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0	0	0
5	0	1	1	0	0	0	0	0	0	0
6	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	0	1	0	0	0	0
8	0	0	0	1	0	0	1	0	0	0
9	0	0	0	0	0	0	0	1	1	0

dependencia de

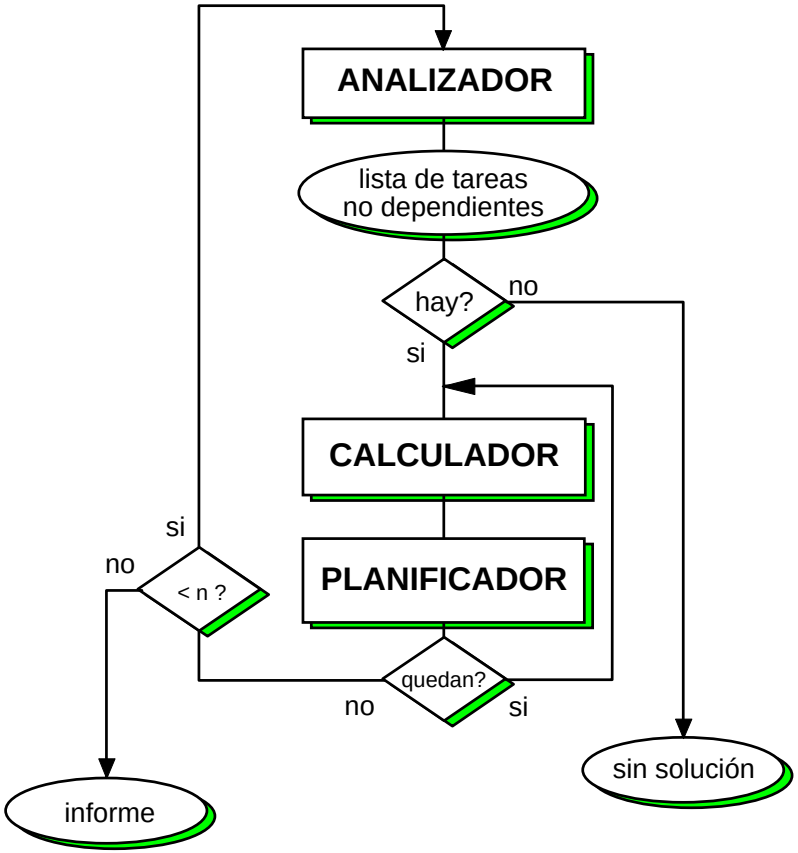
El algoritmo viene descrito por una matriz de dependencias (*D*) en la que cada fila es un vector de dependencias. La componente i-ésima del vector determina la dependencia respecto a la tarea i.

Existe una regla empírica que puede formularse así: "la matriz de dependencias describe un algoritmo solucionable siempre que D^n sea nula" siendo *n* la dimensión de la matriz.

3. Características de la aplicación

Las características de los procesadores y de las tareas se han almacenado en sendas bases de datos implementadas con *Microsoft Access* y accesibles desde la aplicación. Los ficheros que se generan en el volcado de datos tienen un formato compatible con *Excel*. La herramienta se compone estos bloques:

- Editor gráfico de red, con asignación de procesadores.
- Editor gráfico de algoritmo, con asignación de tareas.
- Asignación de algoritmos a redes.
- Simulación y volcado de resultados.



Comenzamos diseñando la topología de la red colocando nodos y estableciendo los enlaces entre los mismos. Se asignan los procesadores a los nodos. Por otra parte, se diseña el algoritmo. Las esferas representan las tareas, y se unen mediante enlaces que indican las dependencias de datos. Se determinan tareas de la base de datos. Cuando se tiene la red y el algoritmo diseñados se pasa a realizar la asignación de tareas a elementos de procesamiento. Posteriormente se puede simular el sistema completo y sacar informes. También, se puede hacer una simulación de una tarea sobre un procesador.