

Algoritmo para la simulación paralela de redes de Petri temporizadas

PABLO HERNÁNDEZ MORERA

RESUMEN

La simulación de un modelo de red de Petri temporizada fuerza la ejecución ordenada en el tiempo de las transiciones sensibilizadas. La noción de relación de causalidad es utilizada para averiguar las dependencias entre las transiciones, con objeto de aumentar la cantidad de transiciones disparadas en cada estado de la red. Dicha propuesta está basada en el concepto de distancia temporal. Este concepto permite localizar el conjunto de transiciones mutuamente independientes en cualquier momento (incluso con tiempos disparo diferentes), y dispararlas simultáneamente en el mismo instante de CPU sobre un conjunto de procesadores, disminuyendo el tiempo de ejecución de la simulación de la red de Petri.

ABSTRACT

An algorithm for parallel simulation of temporized PETRI networks

The simulation of a timed Petri net model forces the time ordered execution of the enabled transitions. Causality relation is presented and used to find out the interactions among the transitions, in order to increase the amount of firing transitions. This approach is based on the concept of time-distance. This concept allows to collect the set of transitions mutually independent at any time (even with different firing times), and fire them together in the same CPU time on a set of processors, decreasing the execution time of the simulation of a Petri net.

INTRODUCCIÓN

Actualmente, los sistemas son analizados mediante una representación de los mismos. De este modo, la simulación permite el conocimiento de su comportamiento con un coste bajo y sin riesgo.

Nos centraremos en sistemas de eventos discretos, donde el tiempo es de naturaleza discreta y avanza con la sucesión de eventos, de manera que si en el modelo no sucede nada, el tiempo de simulación permanece inmóvil (Vázquez y Martínez, 1990). Su simulación sobre una arquitectura multiprocesadora será una práctica rápidamente extendida en el futuro debido a la disminución de los costes de procesamiento.

REDES DE PETRI DE ALTO NIVEL TEMPORIZADAS

Las redes de Petri constituyen un método simple y potente para modelar el flujo de información y control de un sistema (Peterson, 1977). Una red de Petri se representa mediante un grafo que modela las propiedades estáticas del sistema. En el grafo se presentan dos tipos de nodos: lugares y transiciones, representando las condiciones y los eventos de un sistema respectivamente. Los lugares se representan mediante círculos y las transiciones mediante líneas rectas. Los lugares y transiciones se interconectan mediante arcos dirigidos, describiendo las

condiciones necesarias para que un evento se ejecute, así como las condiciones resultantes de la realización de un evento.

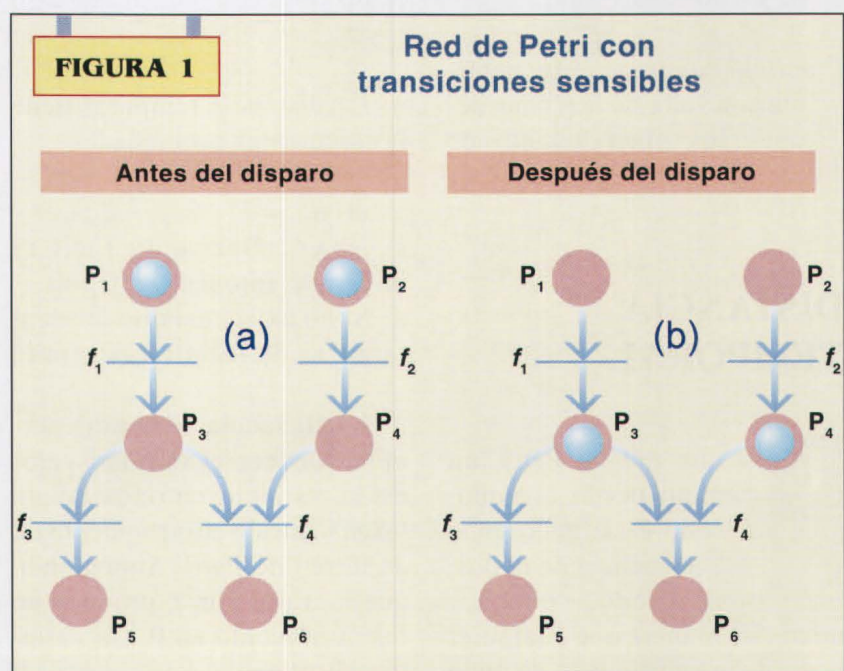
La ejecución de una red de Petri modela las propiedades dinámicas del sistema. El flujo de información resultante de la ejecución del sistema se representa mediante la posición de los llamados token, puntos negros que se alojan en los lugares de la red. La existencia de un token en un lugar indica el cumplimiento de la condición representada por ese lugar.

El movimiento de los token en una red de Petri se realiza mediante el disparo de las transiciones. Para realizar el disparo de una transición, ésta debe estar sensible, esto es, todos sus lugares de entrada deben contener al menos un token. El conjunto de token que hacen sensible una transición se denomina tupla sensible. El disparo de una transición consiste en desalojar su tupla sensible de sus lugares de entrada, generando un nuevo token en cada lugar de salida de la transición disparada.

En la figura 1(a) se muestra una red de Petri con las transiciones t_1 y t_2 sensibles. El disparo de t_1 desaloja el token de P_1 y genera un nuevo token en P_3 . El disparo de t_2 retira el token de P_2 y crea un nuevo token en P_4 . El resultado de ambos disparos se muestra en la figura 1(b).

Las redes de Petri de alto nivel temporizadas (STER) combinan el modelado del tiempo junto con el uso de expresiones funcionales (Ghezzi, 1991). Tienen las siguientes características:

- Los token son conjuntos de pares <identificador, valor>, donde identificador es una variable con un valor asociado. Uno de estos identificadores es *chronos* y representa el instante de tiempo en el que el token fue creado. El identificador x de un token t es denotado $t.x$.
- Cada transición tiene asociada una acción definiendo las condiciones de los token de entrada para participar en el



disparo (predicado) y los nuevos valores de los identificadores de los token de salida (resultado).

A diferencia de las redes de Petri convencionales, en las redes de alto nivel la existencia de un token en cada lugar de entrada de una transición no implica necesariamente que la transición esté sensible. También es necesario que cada uno de los token satisfaga el predicado de la transición.

La acción de cada transición debe contener una condición temporal para determinar el tiempo de disparo de la transición $C(t)$ en base a los identificadores de la tupla sensible. Se compone de dos elementos: las condiciones a evaluar para obtener el *chronos* $S(t)$ a partir del cual la transición es sensible y la duración de la transición. Para obtener $S(t)$ se deben establecer dos tipos de condiciones:

Condición de lugar: son establecidas para cada lugar de entrada de una transición para determinar el token sensible de dicho lugar.

Condición de transición: determina el valor de $S(t)$ considerando los token sensibles de cada lugar de entrada.

DISTANCIA TEMPORAL

En una red de Petri, un camino desde el nodo n_1 al nodo n_n es definido como una secuencia de nodos, $nodo_1$ $nodo_2$ $nodo_3$... $nodo_n$, de manera que cualquier $nodo_i$ de la secuencia es un nodo

entrada para el nodo n_{i+1} y un nodo salida para el nodo n_{i-1} .

Como consecuencia de la "conexión" entre t_1 y t_n a través del camino indicado, el disparo de la transición t_1 generaría un token de salida en el lugar P_2 , que sensibilizaría a la transición t_2 del camino. Un disparo de t_2 produciría la sensibilización de la siguiente transición t_3 del camino, y así sucesivamente hasta que un token de salida de t_{n-1} sensibilizaría a t_n . En definitiva, el camino entre dos transiciones posibilita que el disparo de la transición origen sensibilice a la transición destino en algún estado posterior de la red. Al token generado en P_i por t_j mediante el proceso descrito se le denomina token inducido.

La distancia temporal desde la transición t_1 a la transición t_n de una red de Petri temporizada, $d(t_1, t_n)$, es definida como la mínima distancia temporal de todos los caminos que enlazan t_1 con t_n . La distancia temporal de un camino es la suma de las duraciones de las transiciones incluidas en el camino exceptuando los extremos, en este caso t_1 y t_n .

La distancia temporal tiene las siguientes propiedades:

- $d(t_i, t_j) \geq 0$
- No es reflexiva $d(t_i, t_j) \neq d(t_j, t_i)$
- Si $t_i = t_j$ entonces $d(t_i, t_j) = 0$
- Si no existe camino desde t_i a t_j se define $d(t_i, t_j) = \text{infinito}$

La distancia temporal permite conocer el mínimo valor de la variable *chronos* de un token inducido en cualquier lugar de la red de Petri. Suponiendo que la transición t_j produce un token inducido en P_x , la variable *chronos* de éste tendrá un

valor igual o mayor que $C(t_j) + d(t_j, t_x)$, donde t_x es cualquier transición de salida de P_x y $C(t_j)$ es el tiempo de disparo de t_j .

ANÁLISIS DE CAUSALIDAD ENTRE TRANSICIONES

Sean t y t' dos transiciones de una red de Petri tal que el disparo de t afecta al de t' , entonces t y t' tienen una relación de causalidad denotada $t \Rightarrow t'$. Dos transiciones t y t' son mutuamente independientes si y sólo si no $t \Rightarrow t'$ ni $t' \Rightarrow t$.

Observando el tiempo de disparo se puede concluir las siguientes propiedades:

- Si dos transiciones t y t' tienen el tiempo de disparo $C(t) = C(t')$, entonces t y t' son mutuamente independientes.
- Dadas dos transiciones t y t' con $C(t)$ y $C(t')$ tal que $C(t) < C(t')$, entonces no $t \Rightarrow t'$.

Para que el disparo de una transición t afecte al disparo de otra transición t' , el token inducido por t debe satisfacer las siguientes condiciones:

- Debe llegar al lugar de entrada de t' antes de su disparo.
- Debe modificar la tupla sensible de la transición t' .

El concepto de distancia temporal permite evaluar el cumplimiento del primer requisito comparando el valor de la variable *chronos* del token inducido por

t con el tiempo de disparo de t' . La comprobación del segundo requisito es más laboriosa pues es necesario conocer el valor del resto de los identificadores del token inducido con objeto de evaluar el predicado de t' .

El trabajo desarrollado sólo contempla la comprobación de la primera condición, y restringe la segunda condición a la evaluación de la condición temporal, por lo que su cumplimiento sólo manifiesta una "posible" relación de causalidad. De esta forma se pone de manifiesto una posible relación de causalidad independiente de la semántica asociada a cada transición.

El análisis de los dos tipos de condición de lugar posibles: "mínimo del lugar" y "máximo del lugar", proporciona los valores de chronos para que el disparo de una transición t afecte a t' .

Condición "mínimo del lugar"

En la figura 2 se representa la sensibilización y disparo de la transición t' en el eje temporal delimitando cuatro zonas distintas, donde $S_p(t')$ representa el chronos del token sensible del lugar de entrada P.

La transición t genera un token inducido en el lugar de entrada P de la transición t' , y se consideran los cuatro casos que se pueden producir dependiendo del valor del chronos del token inducido, que denotaremos como $TI(t)$.

Caso A (zona 1)

$TI(t) < S_p(t') < C(t')$

La condición "mínimo del lugar" garantiza que será el token

de menor chronos el que forme parte de la tupla sensible de t' .

La aparición del token inducido en las condiciones descritas $TI(t) < S_p(t')$, puede conducir a que la evaluación de la condición de lugar proporcione el token inducido como token sensible de t' , en vez de cualquiera de los token presentes en ese lugar. En estas circunstancias en las que el disparo de la transición t' pueda verse afectado por el disparo de la transición t , es obligado mantener una secuencialidad en los disparos de ambas transiciones, debiendo disparar la transición t antes que t' , quedando de manifiesto una posible relación de causalidad entre ambas transiciones.

Caso B (zona 2)

$S_p(t') \leq TI(t) \leq S(t) < C(t')$

El token inducido sólo puede alterar el resultado de la condición de lugar si $TI(t) = S_p(t')$, pero el cambio del token sensible no produce ningún cambio sustancial en el disparo de la transición t' por lo que se considera que la influencia de la transición t sobre t' es nula.

Caso C (zona 3)

$S_p(t') \leq S(t') < TI(t) < C(t')$

Siguiendo el mismo razonamiento empleado en los casos anteriores se concluye que el token inducido no perjudicará al disparo de la transición t' .

Caso D (zona 4)

$S_p(t') \leq S(t') < C(t) \leq TI(t')$

En este caso, ya no existe posibilidad de modificar el disparo de una transición que ha sido previamente realizado.

De esta discusión se puede concluir que la transición t puede afectar a la transición t' sólo en el caso A, donde

$$TI(t) < S_p(t') \quad (1)$$

Condición "máximo del lugar"

Caso A' (zona 1)
 $TI(t) \leq S_p(t') \leq S(t') < C(t')$
 El token inducido modifica la tupla sensible si $TI(t) = S_p(t')$, pero el cambio del token sensibilizador no altera el disparo de la transición t' .

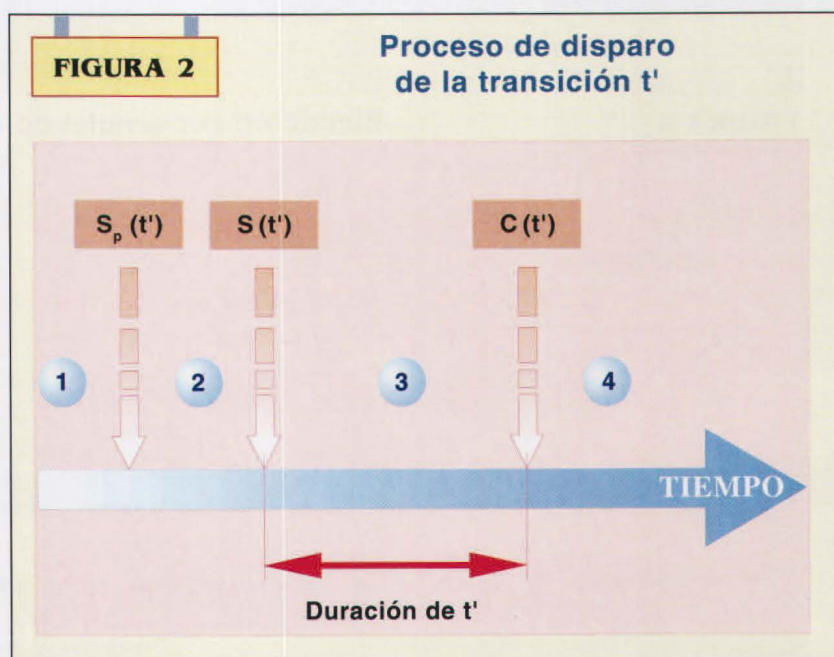
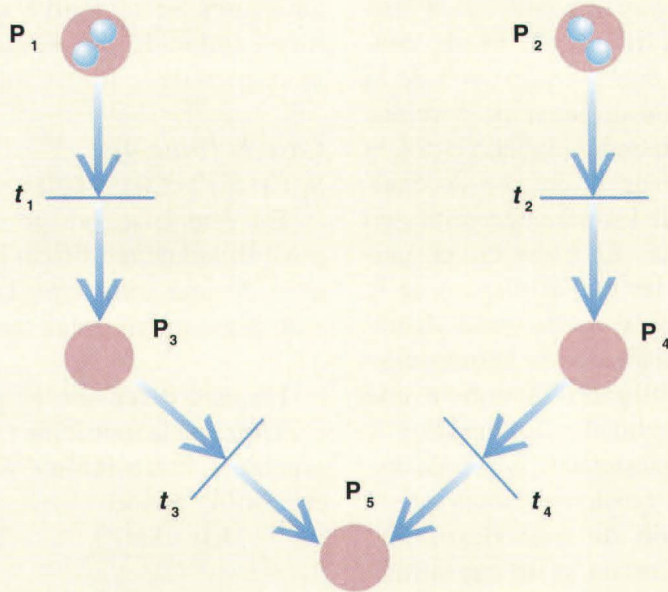


FIGURA 3

Red de Petri temporizada



$$f_1 = \{ \langle P_1, P_3 \rangle | P_3.chronos = \min(P_1.chronos) + 1 \}$$

$$f_2 = \{ \langle P_2, P_4 \rangle | P_4.chronos = \min(P_2.chronos) + 2 \}$$

$$f_3 = \{ \langle P_3, P_5 \rangle | P_5.chronos = \min(P_3.chronos) + 2 \}$$

$$f_4 = \{ \langle P_4, P_5 \rangle | P_5.chronos = \min(P_4.chronos) + 2 \}$$

Caso B' (zona 2)

$$S_p(t') < TI(t) \leq S(t') < C(t')$$

En este caso el token inducido es el de mayor chronos de los presentes por lo que es seleccionado como token sensible, en vez de cualquiera de

los token presentes en ese lugar. En estas circunstancias se pone de manifiesto una posible relación de causalidad, por lo que el disparo de la transición t debe preceder al de la transición t' .

Caso C (zona 3)

$$S_p(t') \leq S(t') < TI(t) < C(t')$$

En este caso los resultados son similares al B'.

Caso D (zona 4)

$$S_p(t') \leq S(t') < C(t') \leq TI(t)$$

El token inducido por la transición t no puede afectar al disparo de una transición que ha sido previamente realizado.

De los casos analizados con la condición "máximo del lugar" se concluye que la transición t puede afectar a la transición t' sólo en los casos B' y C, donde

$$S_p(t') < TI(t) < C(t') \quad (2)$$

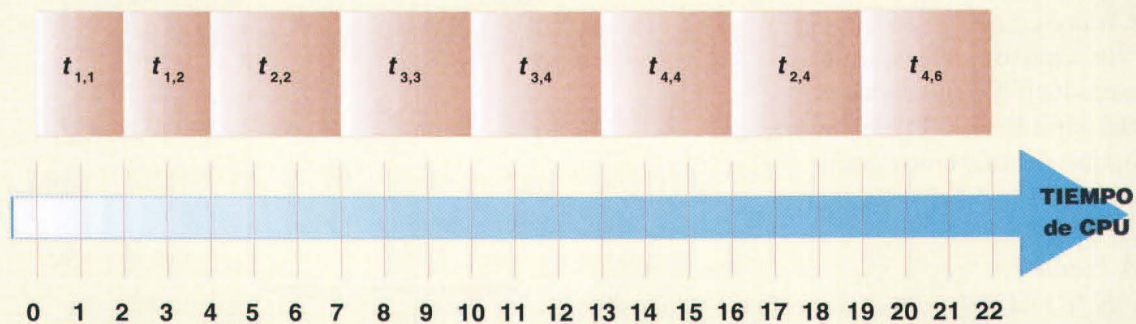
Yuxtaponiendo las condiciones 1 y 2 se obtiene la condición que revela una posible relación de causalidad entre dos transiciones, aplicable con ambos tipos de condiciones de lugar.

$$C(t') + d(t, t') < C(t') \quad (3)$$

En el supuesto de que la expresión 3 no se satisfaga, el disparo de t puede hacer sensible otras transiciones que deban ser

FIGURA 4

Ejecución secuencial de la simulación



disparadas antes de la transición t' , pero ninguna de ellas puede afectar a t' .

SIMULACIÓN SECUENCIAL DE REDES STER

La simulación de una red STER se realiza mediante el algoritmo Event Scheduling/Time Advance (Banks, Carsons y Nelson, 1996), que garantiza que todos los disparos se ejecuten en orden cronológico mediante la implementación de una lista de futuros disparos (LFD). En un instante de tiempo t , la LFD contiene todas las transiciones cuyo tiempo de disparo ha sido previamente programado para ocurrir en el futuro.

La simulación comienza localizando las transiciones sensibilizadas, calculando su tiempo de disparo e inicializando la LFD. De la LFD se selecciona la transición con el tiempo de disparo menor. El disparo de la transi-

ción seleccionada puede dar lugar a nuevas sensibilizaciones, lo que obliga a reconstruir la LFD.

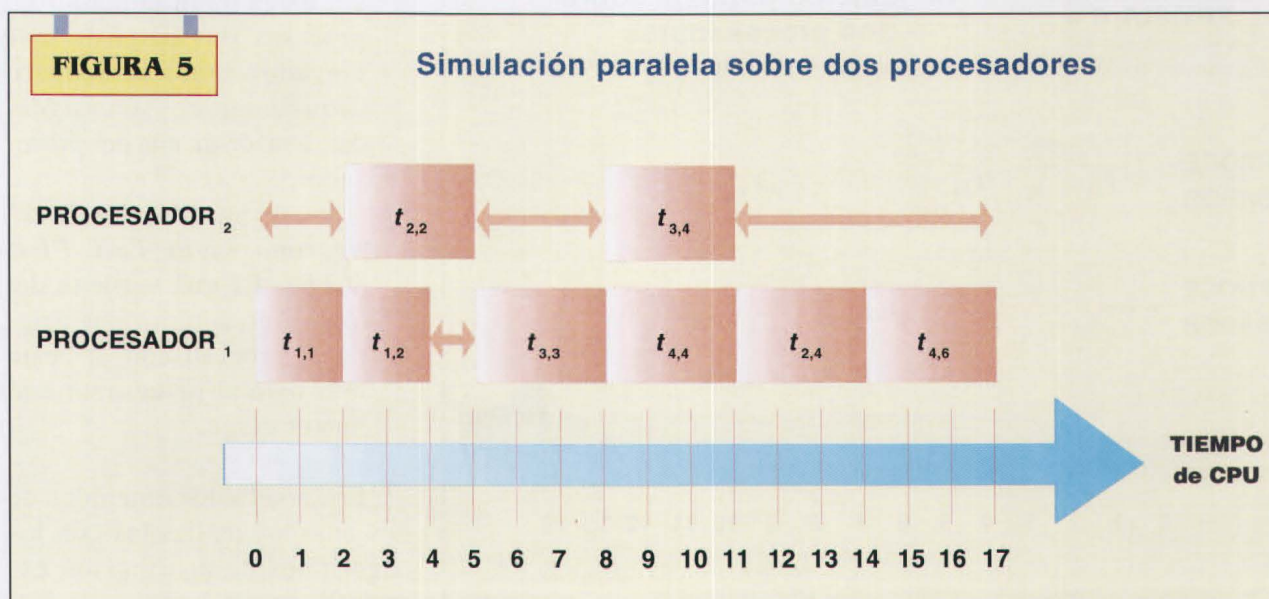
La ejecución de la red de Petri de la figura 3 en una máquina monoprocesadora consistirá en la realización secuencial de los disparos como se muestra en la figura 4, donde se supone que el tiempo de ejecución del disparo de t_1 es 2 y el del resto de transiciones es 3. $t_{i,j}$ simboliza el disparo de la transición i en el instante de simulación i (tiempo de disparo).

SIMULACIÓN PARALELA DE REDES STER

El algoritmo Event Scheduling/Time Advance expuesto en el punto anterior no impone ningún orden de realización para la ejecución de las transiciones con igual tiempo de disparo. Ese grado de libertad permite su realización simultánea en un conjunto de procesadores.

En la figura 5 se representa gráficamente la ejecución de la red de la figura 3 sobre dos procesadores mediante la implementación comentada en el párrafo anterior. Aunque el tiempo de ejecución de la simulación paralela ha disminuido considerablemente con respecto a la ejecución secuencial, se observa una infrautilización de los recursos disponibles (zonas delimitadas mediante flechas).

Los estados de inactividad de los procesadores pueden ser reducidos disparando un mayor número de transiciones en cada momento. Mediante la expresión 3 se comprobará la existencia de causalidad entre transiciones, de manera que aquellas que no mantengan una relación de causalidad sean ejecutadas en un instante de tiempo anterior que el correspondiente a su tiempo de disparo. El disparo de una transición que deba esperar a la realización de un disparo previo, puede ser adelantado hasta el instante de ejecución inmediatamente posterior al instante de ejecución del disparo previo.



Aplicando este método en la simulación paralela de la figura 3 se obtiene una disminución del tiempo de ejecución de la simulación como se muestra en la figura 6.

A continuación se indican los pasos del algoritmo de simulación paralela propuesto, donde DT es el conjunto de transiciones mutuamente independientes que pueden ser disparadas simultáneamente.

1. Inicializar la LFD mediante la obtención del conjunto de transiciones sensibilizadas.
2. Ordenar la LFD en orden ascendente según los tiempos de disparo de las transiciones que lo componen.
3. Inicializar DT con la transición con menor tiempo de disparo.
4. Seleccionar la siguiente transición de la LFD y comprobar la existencia de relación de causalidad entre las transiciones de la LFD con un tiempo de disparo menor y

la seleccionada. En caso afirmativo desechar la transición, y en caso contrario añadirla a DT.

5. Volver al paso 4 hasta haber seleccionado todos los elementos de la LFD.
6. Disparar simultáneamente las transiciones incluidas en DT.
7. Ir al paso 1.

ALGORITMOS DE DISTRIBUCIÓN DE PROCESOS

Una vez obtenido el conjunto de transiciones, que pueden ser disparadas simultáneamente, es necesario realizar una distribución óptima de los procesos de manera que se igualen las cargas de cada uno de los procesadores.

El problema es una generalización del problema de parti-

ción del conjunto, cuyo enunciado es: sea Z el conjunto de los números enteros. Dado un conjunto finito A de Z, encontrar un subconjunto de A (A') tal que cumpla:

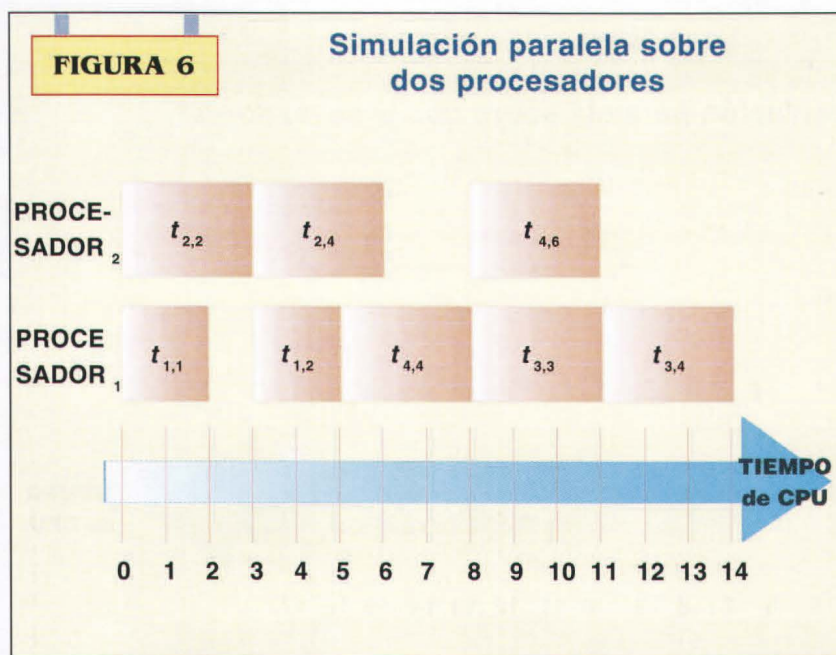
$$\sum_{a \in A'} a = \sum_{b \in (A-A')} b$$

La solución del problema conlleva un coste computacional muy elevado (Sarkar, 1989), siendo posible generar una solución sólo en casos muy concretos (2 procesadores y procesos de igual duración). Como alternativa existen algoritmos para localizar una solución subóptima con un coste computacional muy razonable:

- **Algoritmo Heavy Task First (HTF):** Este algoritmo es una versión modificada del descrito en (Shirazzi, Wang y Pathak, 1990). A cada proceso se le asigna una prioridad que es función de su tiempo de ejecución, que se denominará coste del proceso. El algoritmo asigna el proceso con el coste más alto al procesador con menor carga. Al comienzo, todos los procesadores se encuentran sin carga y un conjunto de procesos (las de coste más elevado) igual al número de procesadores son asignadas, a razón de una por procesador.

- **Algoritmo Light Task First (LTF):** Es una variante del algoritmo anterior que asigna el proceso con el coste más bajo al procesador con menor carga.

Los resultados obtenidos de los ensayos realizados con los algoritmos de distribución expuestos son:



- El algoritmo HTF consigue un mayor porcentaje de aciertos (soluciones óptimas) que el algoritmo LTF.
- En ambos casos el porcentaje de aciertos disminuye si aumentan los procesos para un número de procesadores fijo. Sin embargo, aparecen máximos intermedios cuando la cantidad de procesos es múltiplo del número de procesadores.
- La diferencia entre la solución óptima y la subóptima es mayor en el LTF que en el HTF.
- La diferencia entre la solución óptima y la subóptima aumenta según crece el número de procesos para una cantidad de procesadores determinada. Este comportamiento mejora cuando el número de procesos y procesadores son múltiplos.

y Lin, 1995) muestran técnicas para analizar el paralelismo de una simulación. Ambos insertan algunas instrucciones en el programa de simulación secuencial para contabilizar el tiempo de ejecución de la simulación paralela en "tiempo real".

Se supone que el programa de simulación secuencial está formado por tareas independientes, que pueden ser trasladadas directamente a los procesadores en una simulación paralela.

Se trabajará con una arquitectura de memoria compartida compuesta de n procesadores homogéneos (Peterson y Silberschatz, 1985). Se distinguen dos tipos de procesadores: maestro y esclavo. El procesador maestro controla el sistema y

determina las tareas que deben ser realizadas por el resto de procesadores esclavos. Finalizada la identificación de las tareas a realizar, el procesador maestro puede participar junto con los procesadores esclavos, en la ejecución de las tareas pendientes de realización.

En el programa de simulación secuencial se insertan unas instrucciones que contabilicen el tiempo de ejecución de la simulación paralela correspondiente a n procesadores. El método implementa n relojes $\{t_1, \dots, t_n\}$ que contabilizan los tiempos de ejecución correspondientes a cada uno de los n procesadores existentes. A cada procesador se le asigna el tiempo de ejecución de las tareas que realiza. Las relaciones de precedencia obligan a que previa-

PRESTACIONES DEL SIMULADOR PARALELO

El algoritmo de simulación paralela fue utilizado en el desarrollo de un simulador de redes de Petri temporizadas. El simulador fue ensayado con un modelo de una red local con protocolo CSMA/CD, que pusiera de relieve las mejoras obtenidas con respecto a una simulación secuencial. El modelo de la red local comprende 10 nodos representando 10 estaciones ethernet interconectadas.

Algunos autores (Sinclair y Dawkins, 1994; y Wong, Hwang

FIGURA 7

Tiempos de ejecución paralela para diferente número de procesadores

NÚMERO DE PROCESADORES	T_p (seg.)	GANANCIA	EFICACIA
1	9.881	1.00	1.0
2	5.301	1.86	0.93
3	3.935	2.51	0.837
4	3.262	3.03	0.757
5	2.887	3.42	0.684
6	2.637	3.75	0.625
7	2.482	3.98	0.569
8	2.370	4.17	0.521
9	2.291	4.31	0.479
10	2.221	4.45	0.445

mente a realizar una actividad los relojes sean incrementados al valor $\max(t_i) \ i=\{1, \dots, n\}$, reflejando de esta manera la espera que deben realizar algunos procesadores hasta que se haya ejecutado la última tarea de la anterior actividad.

La distribución de tareas entre los procesadores se ha realizado mediante el algoritmo Heavy Task First.

El modelo fue simulado hasta haber disparado 50.000 transiciones. Los valores obtenidos se indican en la tabla 7 junto con la ganancia de procesamiento G y la eficacia E_f que responden a las siguientes expresiones (Sun y Zhu, 1995):

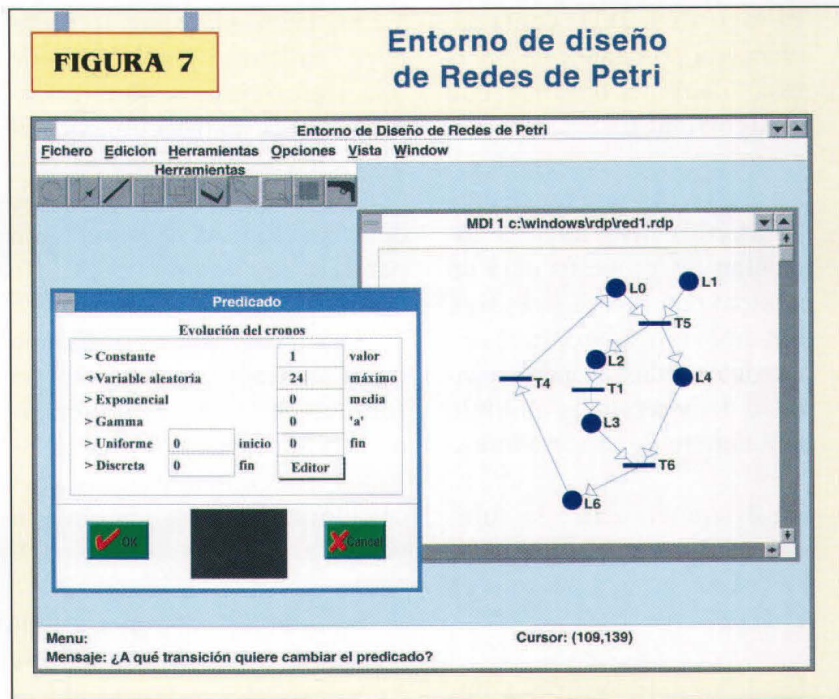
$$G_k = \frac{T_{p=1}}{T_{p=k}}$$

$$E_f = \frac{G_k}{k}$$

donde k es el número de procesadores y $T_{p=k}$ es el tiempo de ejecución de la simulación paralela con k procesadores.

CONCLUSIONES

Se observa una disminución del tiempo de simulación con el aumento del número de procesadores como era de esperar, aunque esta relación no



es proporcional. Las simulaciones realizadas con un número mayor de 4 procesadores no suponen una mejora significativa comparado con el coste del procesador adicional. La relación entre la disminución del tiempo de ejecución de la simulación y el coste de los recursos utilizados se indica mediante el parámetro eficacia que también es mostrado en la tabla 7. Será labor del usuario del simulador decidir un umbral mínimo para este parámetro.

El motivo por el cual la ganancia de procesamiento dista de la ideal es que aún siendo la simulación paralela, el simulador cuenta con una parte de código que debe ser ejecutado secuencialmente por el procesador maestro (ordenar las transiciones, inicializar variables,

etc.). Esto puede ser considerado como el límite inferior para T_p . Aún así, este límite es inalcanzable puesto que habría que añadirle los tiempos de espera debidos a las relaciones de precedencia.

Además de la mejora obtenida con el aumento del número de procesadores, es obligado constatar la disminución del tiempo de ejecución con respecto a la simulación secuencial. El tiempo de ejecución de la simulación secuencial del modelo durante 50.000 disparos de transiciones es de 18'2227 segundos. La simulación paralela realizada con un solo procesador representa una disminución de un 45'78% con respecto a la simulación secuencial.

BIBLIOGRAFÍA

- Banks, J.; Carsons, J.S.; Nelson, B.L. (1996): Discrete-Event system simulation. Ed. Prentice-Hall.
- Fidge, C. (1991): "Logical Time in Distributed Computing Systems" IEEE Computer, Agosto, pp. 28-33.
- Ghezzi, C. et al (1991): "A Unified High-Level Petri Net Formalism for Time-Critical Systems", IEEE Trans. on Soft-

ware Eng. vol 17, Febrero, pp. 160-172.

- **Peterson, J.L.** (1977): "Petri Nets". Computing Surveys, vol.9, nº 3.
- **Peterson J.L.; y Silberschatz, A.** (1985): "Operating System Concepts". Ed. Addison-Wesley.
- **Sarkar, V.** (1989): "Partitioning and Scheduling Parallel Programs for Multiprocessor". MIT Press.
- **Shirazi, B.; Wang, M.; Pathak, G.** (1990): "Analysis and

Evaluation of Heuristic Methods for Static Task Scheduling" J. of Parallel and Distributed Computing, vol.10, pp. 222-233.

- **Sinclair, J.B.; Dawkins W.P.** (1994): "ES: A Tool for Predicting the Performance of Parallel Systems" Proc. of the Second International Workshop on Modelling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 350-353.
- **Sun, X.; Zhu, J.** (1995): "Performance Considerations of Shared Virtual Memory

Machines" IEEE Trans on Parallel and Distributed Systems, vol. 6, nº 11, pp. 1185-1194.

- **Vázquez Gallo, E.; Martínez, A.** (1990): "Simulación de Sistemas". Programa de Postgrado en Sistema y Redes de Comunicaciones. Departamento de Ingeniería de Sistemas Telemáticos de la U.P.M.
- **Wong, Y.; Hwang,S.; Lin, J.B.** (1995): "A Parallelism Analyzer for Conservative Parallel Simulation" IEEE Trans on Parallel and Distributed Systems, vol.6, nº 6, pp. 628-328.

BIOGRAFÍA

Pablo Hernández Morera

Nació en Valencia en 1964. Es Ingeniero Superior de Telecomunicaciones por la Universidad de Madrid. Ha realizado diversas estancias en el extranjero recibiendo formación en productos de comunicaciones. Ha cursado el Programa de Doctorado "Ingeniería Electrónica", y se encuentra realizando su tesis doctoral "Metodología para la simulación paralela de redes de Petri de alto nivel temporizadas" en la División de Comunicaciones del Centro de Microelectrónica Aplicada, bajo la dirección del Dr. D. Juan Do-

mingo Sandoval González. Es autor y colaborador en diversos artículos presentados en congresos internacionales en materias referentes a simulaciones paralelas.

Dirección:

Dpto. Electrónica, Telemática y Automática
Universidad de Las Palmas de Gran Canaria
Campus Universitario de Tafira
35017 - Las Palmas de Gran Canaria
e-mail: pablo@teleco.ulpgc.es

Este trabajo ha sido patrocinado por:

UNIÓN ELÉCTRICA DE CANARIAS (UNELCO)