REDES DE INTERCONEXIÓN PARA COMPUTADORES MASIVAMENTE PARALELOS

José M. García Carrasco
Departamento de Informática
Universidad de Castilla-La Mancha
02071 - ALBACETE

Resumen

En este artículo se pretende describir la importancia que tienen las redes de interconexión en el caso de los computadores masivamente paralelos. La tendencia actual para el paralelismo masivo es usar una arquitectura paralela con memoria distribuida, en donde la red de interconexión es la encargada de proveer todo lo necesario para que la máquina tenga un adecuado sistema de paso de mensajes entre los nodos. En este artículo se revisan los principales tópicos relacionados con las redes de interconexión, tales como la topología, la asignación de procesos y el modo de encaminamiento de mensajes. Tras detallar las características principales, se ofrecen también unas consideraciones tecnológicas que hay que tener en cuenta en el diseño de la red.

1. Introducción

La aparición de los ordenadores a mediados de este siglo ha sido sin duda una de las revoluciones más importantes en el campo tecnológico. Tan es así, que se especula con que la aparición de la Informática puede tener un impacto social incluso mayor del que en su día se atribuyó a la revolución Industrial. La historia de los ordenadores apenas tiene 50 años, pero su evolución en este corto período de tiempo ha sido vertiginosa. Conforme el uso de los ordenadores se ha hecho más común en nuestra sociedad, las personas les hemos pedido más y más prestaciones a estos aparatos, al comprobar la ayuda que la Informática presta en la resolución de los mil y un problemas corrientes de cada día. La aparición de los ordenadores personales, en la década de los años '80 ha sido un factor decisivo para favorecer la proliferación del uso de la Informática y ejercer una mayor presión social en el aumento de las prestaciones que desarrollan los ordenadores.

Dicha mejora en prestaciones se abordó en un primer momento apoyándose en diversas innovaciones tecnológicas tales como la integración de circuitos VLSI, el aumento de la frecuencia de reloj en los procesadores, el aumento del tamaño en el bus[1] de control o de datos, etc. Pero estas mejoras tienen un techo impuesto por la naturaleza

[1] Al ser la Informática una ciencia muy reciente, hay un conjunto de palabras inglesas que no tienen traducción clara en castellano y se utilizan con el término inglés sin traducir. Tal es el caso de "bus". Una significado aproximado sería conjunto de hilos o cable.

física de los componentes, lo que hizo que se pensara en otro modo de resolver este problema. Entonces es cuando se dirigió la vista a la arquitectura que presentaban los ordenadores. En la actualidad, los computadores masivamente paralelos con miles de procesadores son considerados como la tecnología mas prometedora para alcanzar una capacidad de procesamiento cercana a los teraflops2. Tales computadores de gran escala se organizan como una replicación o aglomeración de unidades funcionales de tratamiento muy parecidas a las que encontramos en una máquina clásica. Cada una de estas unidades reciben el nombre de nodos. Cada nodo tiene su propio procesador, su memoria local y otros periféricos.

El modo en que los nodos son conectados entre sí varía de unas máquinas a otras. En una máquina con una arquitectura de red *directa*, cada nodo tiene una conexión punto-a-punto con algún número de otros nodos, llamados nodos vecinos. Las redes directas son en la actualidad una arquitectura muy usada para construir computadores masivamente paralelos debido a que escalan bien, es decir, cuando el número de nodos en la máquina aumenta, también aumenta el ancho de banda de comunicación, el ancho de banda de la memoria y la capacidad de procesamiento de toda la máquina en su conjunto.

Los nodos vecinos pueden enviarse mensajes entre ellos directamente, mientras que los nodos que no están conectados deben enviar los mensajes a través de los nodos intermedios que forman la red. En muchos sistemas es frecuente que cada nodo tenga un *encaminador* de mensajes para manejar todos los problemas relacionados con la comunicación.

En este artículo pretendemos esbozar el importante papel que juega la red de interconexión en una máquina masivamente paralela. Para ello, empezaremos presentando las características más importantes que tiene una red de interconexión en una máquina paralela, ya sea con memoria compartida o con memoria distribuida. De todas estas características, nos vamos a centrar en la topología de las redes y el modo de encaminamiento de la información a través de la red. Este último aspecto lo desarrollaremos tan sólo en el caso de las máquinas con memoria distribuida, pues es la arquitectura que en la actualidad tiene más prestaciones.

2. Las redes de interconexión

En una conocida clasificación presentada por Flynn allá por el inicio de la década de los 70 [1], los ordenadores se dividen en función del tipo del flujo de control y flujo de datos que poseen. Es precisamente en esta división cuando aparecen por primera vez las máquinas paralelas divididas en tres clases. De estas tres clases, la más general y que ofrece mayor paralelismo es la que sigue el esquema MIMD, es decir, la que está basada en tener múltiples procesadores cada uno de ellos trabajando sobre

[2]Un teraflop equivale a una potencia de cálculo de 10^{12} instrucciones de coma flotante por segundo.

diversos datos. Aunque este tipo de modelo de paralelismo es el más complejo, es el que se ha impuesto últimamente y sobre el que vamos a desarrollar nuestros conceptos de las redes de interconexión.

Al tener una máquina paralela diversos procesadores surge la necesidad de comunicar información entre ellos de cara a resolver un problema común. Dicha comunicación puede realizarse de dos maneras:

a) Compartiendo un recurso, como por ejemplo un dato común en una memoria común.

b) Mediante el envío de la información por medio de un mensaje.

En ambos casos, será necesario la existencia de una red de interconexión que permita realizar las operaciones descritas. En el primer caso, ya que la memoria está físicamente distribuida en varios bancos, la red de interconexión posibilitará la conexión de cada procesador con todos los bancos de memoria. En el segundo caso, es necesario unir físicamente los procesadores para poder realizar el envío de mensajes.

Estos dos tipos de comunicación da lugar a dos importantes clases de máquinas paralelas según el diseño MIMD: los multiprocesadores y los multicomputadores. Los multiprocesadores son máquinas con memoria compartida o común y los multicomputadores son máquinas con memoria distribuida.

Como se puede apreciar, el papel de la red de interconexión es tanto más importante cuanto mayor sea el número de elementos físicos que se deben unir y el flujo de información que se desee intercambiar; en el caso de los ordenadores masivamente paralelos, con un gran número de procesadores y una densidad de comunicaciones alta, el papel que desempeña la red es de primer orden [1].

A continuación, vamos a definir algunas características importantes de las redes de interconexión, comunes tanto a los multiprocesadores como a los multicomputadores.

2.1 Redes estáticas y dinámicas

Uno de los criterios más importantes para la clasificación de las redes es el que tiene en cuenta la situación de la red en la máquina paralela, dando lugar a dos familias de redes: redes estáticas y redes dinámicas. Una red estática es una red cuya topología queda definida de manera definitiva y estable durante la construcción de la máquina paralela. La red simplemente une los diversos elementos de acuerdo a una configuración dada. Se utiliza sobre todo en el caso de los multicomputadores para conectar los diversos procesadores que posee la máquina. Por la red sólo circulan los mensajes entre procesadores, por lo que se dice que la red presenta un *acoplamiento débil*. En general, en las redes estáticas se exige poca carga a la red.

Una red dinámica es una red cuya topología puede variar durante el curso de la ejecución de un programa paralelo o entre dos ejecuciones de programas. La red está constituida por elementos materiales específicos, llamados commutadores o *switches*. Las redes dinámicas se utilizan sobre todo en los multiprocesadores. En este caso, la red une los procesadores a los bancos de memoria central. Cualquier acceso de un

procesador a la memoria (bien sea para acceder a los datos o a las instrucciones) debe pasar a través de la red, por lo se dice que la red tiene un *acoplamiento fuerte*. La red debe poseer un rendimiento extremadamente bueno para no demorar demasiado a los procesadores que acceden a memoria.

2.2 Conectividad

Otro criterio para clasificar las redes está basado en su conectividad: la conexión entre los diferentes elementos que necesitan conectarse entre si puede ser total o parcial. El primer caso es el caso ideal, donde todos los elementos están conectados directamente unos con otros; en la práctica, cuando el número de elementos crece, no es posible la conexión total y hay que conformarse con conexiones parciales. En este caso, la red define una topología cuyas propiedades pueden explotarse para encaminar correcta y rápidamente los mensajes desde un procesador origen a un procesador destino. La existencia de una topología permite definir la distancia entre dos procesadores como el número de etapas a franquear o recorrer en la red para ir del procesador origen al destino. Lógicamente, en el caso ideal de topología totalmente conectada la distancia en todos los casos es siempre 1. El problema de la conectividad de la red se refleja en la dualidad *conexiones/contención*. Cuantas más conexiones tenemos entre los procesadores menos contención presenta la red y viceversa. En el caso de topología totalmente conectada el número de cables (hilos) que posee la red es N^2, pero no tiene contención (su valor es 1). En el caso de la topología de bus, es la más sencilla de realización pues sólo utiliza un hilo, pero es la topología que presenta la mayor contención (de valor N^2). Debido a ello se buscan otras topologías que presenten características intermedias en ambos casos.

Para mejorar la conectividad de la red, una de las formas más habitualmente empleadas ha sido la jerarquización de la red. Las topologías jerarquizadas de redes se construyen sobre el principio del bus. En ellas se reagrupan los enlaces para disminuir el número de conexiones físicas. Ejemplos de estos tipos de redes son las jerarquías de buses, las estrellas de varios niveles, las pirámides o los árboles. La jerarquización tiene la ventaja de disminuir el número de elementos conectados para cada subred de la jerarquía. Gracias a ello, los problemas de contención en la red quedan reducidos. Por contra, las dificultades aparecen cuando se cruzan un gran número de mensajes de un nivel a otro de la jerarquía, creándose problemas de cuello de botella (o estrangula-miento) que pueden tener su impor-tancia. Además, a nivel de protocolos de comunicación estas topolo-gías son más dificil de gestionar.

La interrelación de los criterios dinámico/estático, comple-tamente conectado/parcialmente conectado y jerarquizado/no jerarquizado define seis grandes clases de redes. De hecho, las máquinas encontradas en el mercado utilizan redes derivadas de algunas de estas categorías. En la figura 1 se refleja esta división de las redes.

Otro de los aspectos que hay que tener en cuenta al hablar de la conectividad de las redes es si la red es *conexa* o no. Para redes parcialmente conectadas es muy importante asegurar que sean *conexas*, es decir, que entre dos procesadores cualesquiera siempre exista un camino para enviar un mensaje de uno al otro. Esta propiedad es muy importante tenerla en cuenta en el caso de averias en la red. Si una red tiene fallos y la red sigue siendo conexa, esa topología es buena y se denomina que es *tolerante a fallos*. Un ejemplo es la malla, que en el caso de que se estropeen varios enlaces dicha topología ofrece caminos alternativos para poder encaminar los mensajes. Por contra, el anillo en este sentido es una mala topología, pues sólo con que un enlace tenga un fallo la red ya no es conexa y se parte en dos redes disjuntas.

2.3 Los modos de encaminamiento

El encaminamiento es un mecanismo bien realizado por sofware o por hardware que dirige los mensajes entre el procesador origen y el procesador destino, durante la comunicación en una red parcialmente conectada de tipo estático o dinámico. El modo de encaminamiento comprende dos aspectos esenciales: lo que se denomina el *algoritmo de encaminamiento* y lo que recibe el nombre del *control de flujo* en el encaminamiento.

El algoritmo de encaminamiento efectua la elección de caminos cuando existen varios posibles, y gestiona los conflictos que puedan surgir entre los mensajes que quieren tomar el mismo camino. Normalmente se busca un algoritmo de encaminamiento que sea óptimo, es decir, que conduzca a los mensajes por el camino más corto. El algoritmo de encaminamiento depende de la topología de la red.

El control de flujo describe el modo físico de propagación de la información. Hay diversas técnicas para realizar esto, tomadas al principio de las técnicas que se usan en redes locales; últimamente se han producido desarrollos específicos para redes en máquinas paralelas.

El principal problema que se encuentra en el modo de encaminamiento en una red es el bloqueo, lo que puede llegar a inutilizar la red. Es necesario elegir modos de encaminamiento y topologías que resuelvan estos conflictos antes de ejecutar una aplicación.

Tan importante es este aspecto en las redes de interconexión, que más adelante le dedicaremos un punto para explicarlo con más profundidad.

2.4 El tipo de las comunicaciones

Desde el punto de vista de las necesidades de comunicación, las aplicaciones paralelas no son iguales. Unas aplicaciones tienen procesos que intercambian mensajes durante toda su ejecución y establecen un canal de comunicación permanente; por contra, otras aplicaciones tienen procesos que intercambian información con poca frecuencia. Las aplicaciones paralelas no tienen la misma densidad de comunicación y por tanto, no necesitan los mismos tipos de redes.

En una aplicación paralela, definiremos el tiempo que la aplicación interviene en realizar cálculos (T_{cal}) y el tiempo que interviene en la comunicación entre procesos (T_{com}). Para que una aplicación paralela funcione adecuadamente la relación T_{cal} / T_{com} debe ser alta, es decir, debe ser mucho mayor el tiempo que la aplicación gasta realizando cálculos en operaciones que el tiempo que pasa en envío de mensajes.

Habitualmente se utiliza el término de latencia. La latencia de la red puede definirse como el tiempo transcurrido desde que la cabecera de un mensaje entra en la red en el nodo origen hasta que la cola llega al nodo destino. Por tanto, la misión de la red de interconexión es disminuir el valor de la latencia tanto como sea posible para conseguir que la anterior relación sea alta.

Por otra parte, el modelo de comunicaciones que sigue una aplicación paralela se define a nivel temporal y a nivel espacial. A nivel temporal, se distinguen las aplicaciones que se comunican de manera regular o síncrona de aquellas que se comunican de manera irregular o asíncrona. A nivel espacial, distinguiremos las aplicaciones cuyo grafo de comunicaciones es irregular (cualquiera o desconocido en tiempo de compilación), de aquellas con perfil regular.

2.5 Modelización de la red

Ya para acabar, vamos a dar una serie de definiciones en las redes que es importante tener en cuenta. Muchas de ellas se definen a partir de la modelización que se realiza de las redes mediante un grafo no orientado. Ello permite utilizar todo el cuerpo matemático de la teoría de grafos para el estudio de las propiedades de las redes de interconexión. Son las siguientes:

· Distancia entre dos nodos: longitud de uno de los caminos más cortos que unen los dos nodos.

· Diámetro de la red: distancia máxima en la red; indica el peor caso fuera de conflicto.

· Distancia media: media de todos los caminos posibles.

· Grado de la red: número de aristas incidentes (número de vecinos). Si todos los nodos tienen el mismo grado, la red tiene una estructura regular.

A partir de las propiedades matemáticas del grafo subyacente a la red física, podemos comparar las propiedades que se esperan de la red según los criterios de comparación siguientes:

· número pequeño de conexiones en cada procesador

· número elevado de vértices para permitir un paralelismo masivo

· diámetro y distancia media débiles

· algoritmo de encaminamiento simple

3. Topologías clásicas de las redes

A continuación vamos a describir cuales son las topologías que más se usan en las redes de interconexión, comentando algunas de sus propiedades más significativas. Distinguiremos entre topologías para redes estáticas y topologías para redes dinámicas.

3.1 Redes estáticas

Las redes estáticas se usan habitualmente en los multicomputadores. Aunque hay muchas topologías posibles, los multicomputadores comerciales habitualmente usan sólo dos o tres configuraciones. En los multicomputadores de la primera generación la topología preferida era el hipercubo, mientras que en la actualidad, y gracias a usar un encaminamiento de mensajes segmentado, se utiliza con más frecuencia la malla o el toro. Por tanto, únicamente vamos a comentar estas configuraciones. En [1], Reed y Fujimoto ofrecen un análisis de muchas topologías posibles y discuten los principales aspectos concernientes al diseño en multicomputadores.

Para el caso de máquinas desarrolladas para problemas muy sencillos o con pocos nodos, también se ha utilizado a veces la topología de la estrella y el anillo. En la estrella el nodo central se conecta a todos los otros nodos. Para una estrella de grado N el nodo central tiene un grado de N-1, mientras que el resto de nodos tienen un grado de 1. A partir de cuatro o cinco nodos, esta topología ya no es práctica. El algoritmo de encaminamiento es sencillo, pues todos los mensajes se envían al nodo central y este ya los encamina al nodo destino. En el caso del anillo, es también una de las formas de conexión más simple, en el cual los nodos se colocan en una fila conectando los nodos consecutivos incluidos también los dos extremos. El grado del anillo es 2. Para encaminar un mensaje de un nodo a otro, una vez determinado por qué enlace debe enviarse, únicamente hay que transmitirlo hasta que llegue al nodo destino. El diámetro es igual a *N div 2*, donde N es el número de nodos que hay en la red. Esta topología es adecuada para problemas intensivos en cálculo.

Las mallas y los toros

Históricamente, la malla apareció por primera vez en el Illiac IV en 1970. Una malla de dimensión n y de lado k posee k*n vértices, que pueden considerarse como puntos de coordenadas enteras comprendidas entre 0 y k-1 en un espacio euclidiano de dimensión n. Cada vértice esta conectado a aquellos cuyas coordenadas difieren exactamente en 1 en cada una de las dimensiones. La malla no es un grafo regular debido a los bordes: el grado de los vértices internos es 2n, mientras que, cualquier esquina no tiene más que n vértices adyacentes.

El diámetro tiene un valor de n(k-1), mientras que la distancia media vale $n(k^2-1)/3k$.

Una de las características esenciales de la malla es que es infinitamente extensible según k y extensible según n hasta el límite de las interconexiones disponibles; en este sentido, es la más modular de las topologías, y esta es la razón de su gran éxito comercial en los últimos años. El algoritmo de encaminamiento se llama X-Y: se consideran las dimensiones sucesivamente y se recorre la distancia que separa

el origen del destino en esta dimensión. Este algoritmo es óptimo y está exento de bloqueo.

El toro de base k y de dimensión n puede considerarse como una malla en la que se han cerrado los bordes sobre sí mismos. Es muy frecuente el uso del toro de dimensión 1, denominado anillo, o k-ciclo. Un toro de dimension n y de base k contiene kn vértices, cada uno de ellos conectado a un k-ciclo en cada una de sus n dimensiones. Es un grafo regular de grado 2n, de diámetro n(k/2) y de distancia media nk/4 (si k es par). El algoritmo de encaminamiento de la malla sería óptimo, pero no es aplicable pues puede engendrar interbloqueos.

El hipercubo

Un hipercubo de dimensión n puede definirse considerando sus vértices etiquetados con la representación binaria de los números 0 al 2^{n-1} ; dos vertices estan conectados siempre y cuando sus etiquetas difieran en un solo bit. Su grado y su diámetro son iguales a n.

Pueden construirse recursivamente a partir de hipercubos de dimensión inferior si consideramos dos hipercubos de dimensión n-1 cuyos vértices se etiquetan de 0 a 2^{n-1}-1, obtenemos un hipercubo de dimensión n, uniendo los vértices de etiqueta igual. Esto representa ventajas evidentes desde el punto de vista de la modularidad, puesto que es posible aumentar progresivamente el tamaño de la red; sin embargo, esta modularidad está limitada por el número máximo de conexiones inicialmente previstas.

El hipercubo posee un algoritmo de encaminamiento óptimo simple y adaptado a una realización cableada. Es suficiente enviar el mensaje por los enlaces correspondientes a las dimensiones en las que las direcciones del procesador considerado y del procesador destino no coincidan. Si se fija el orden de recorrido, por ejemplo, números de bit decrecientes, el algoritmo está exento de bloqueo.

Desde el punto de vista de criterios geométricos, el hipercubo ofrece una topología mucho más eficaz que la malla. En efecto, para una red de tamaño N, el diámetro y la distancia media del hipercubo evolucionan según Log_2 N, mientras que en el caso de una malla 2D, son del orden de $\sqrt{N}$.

3.2 Redes dinamicas

Las redes dinámicas son redes que pueden cambiar la topología de comunicación durante la ejecución de los programas o entre dos ejecuciones de programas. Las redes dinámicas se han utilizado esencialmente en los multiprocesadores de memoria compartida: la red dinámica soporta, por consiguiente, la carga de unir los N procesadores a los M bancos de la memoria central.

Como en las redes estáticas, podemos realizar redes dinámicas a base de enlaces punto- a-punto o de bus; nuevamente se presenta un problema en la relación conexión/contención. Desde el momento en que el número de procesadores sobrepase algunas decenas, debemos elegir soluciones que se situen entre estos dos extremos. Por ejemplo, podemos construir un crossbar, o matriz de puntos de cruce utilizando

conmutadores. Con un *crossbar,* el número de enlaces crece solamente en 2N, y el control de la red es extremadamente simple. Sin embargo, el número de conmutadores crece a su alrededor de manera cuadrática. Los *crossbar* son también muy interesantes, pues actualmente el coste de un conmutador realizado en VLSI es más bajo que el de un enlace. Un amplio estudio de las redes dinámicas de interconexión, tanto para el caso SIMD como para el MIMD, se puede encontrar en [1]. Las más importantes son las siguientes:

Las redes multietapa

El número de conmutadores utilizados en los *crossbar* los convierte en impracticables para tamaños de máquinas importantes. Por ello se introducen las redes multietapa. Estas tienen por objetivo acercar, tanto como sea posible, los rendimientos de la red *crossbar,* haciendo uso de un número menor de conmutadores, pero, eso sí, con más tiempo para su recorrido. Las redes multietapa se componen de conmutadores ensamblados bajo la forma de un cuadro rectangular cuyas dimensiones son generalmente de N líneas por Log_2N columnas, es decir, $NLog_2N$ conmutadores; las líneas se corresponden con el número de procesadores. Los mensajes se propagan siguiendo las Log_2N columnas y necesitan Log_2N etapas intermedias antes de llegar a su destino.

Desde el punto de vista de la arquitectura, las redes multietapa se clasifican según dos criterios:

· Funcionalidad: la operación típica de la red es una permutación de las entradas sobre las salidas. Las permutaciones que pueda realizar la red la caracterizan funcionalmente.

· Control: el encaminamiento de mensajes implica el posicionamiento de los conmutadores en cada etapa.

La red Omega

Para las arquitecturas MIMD, donde las comunicaciones son asíncronas y dinámicas, se utilizan redes que tienen la propiedad del camino único: entre un origen y un destino, existe un solo camino. De esta propiedad se deduce, en general, una cierta simplicidad en el control. Por contra, estas redes tienen facultad de bloqueo. A este tipo de red pertenecen las redes Banyan Omega, hipercubo-indirecto y otros muchos. La mayor parte de estas redes son topológicamente equivalentes: se deducen las unas de las otras intercalando una permutación fija en cada lado de la red.

La red Omega se basa en la espacialización del principio de Shuffle/ Exchange. Cada conmutador esta cableado para realizar la función de *Exchange* (Cambio): consiste en invertir el bit menos significativo. Los conmutadores están conectados entre ellos en modo *Shuffle,* lo que conlleva el desplazamiento de un bit a la izquierda. El algoritmo de encaminamiento es muy simple: en la etapa i, se comparan los bits i-ésimos (yendo desde los más significativos hacia los menos significativos) del origen y del destino: si son iguales, no se cruzan; en caso contrario, se cruzan *(Exchange).*

En una red multietapa de tipo Omega, es posible llegar a varios destinos a partir de un solo origen. Esto permite realizar difusiones en la red. Estas redes son, por el

contrario, sensibles a los fallos ya que, por ejemplo, si un hilo se corta no serán posibles muchas de las rutas.

4. Tecnología de las redes en los multicomputadores

En los dos últimos capítulos de este artículo nos vamos a centrar en las redes de interconexión para los multicomputadores, ya que en la actualidad son las máquinas donde se puede conseguir más fácilmente la computación masivamente paralela.

4.1 Características principales de los multicomputadores

Los multicomputadores [1] son computadores con varios procesadores, cada uno con su propia memoria local, que se comunican entre sí a través de una red de interconexión. Estas máquinas apenas tienen una década de existencia, pues el primer prototipo (el Cosmic Cube) se desarrolló en 1981. A pesar de su corta historia, han sufrido un desarrollo muy rápido, distiguiéndose ya dos generaciones.

Su tardía aparición cabe atribuirla a tres factores fundamentales: los requisitos de memoria, la comunicación entre procesadores y la dificultad de programación. En cierta manera, esta arquitectura puede considerarse como una extension de las redes de estaciones de trabajo. Sin embargo, se diferencia por el grano de paralelismo: el orden de magnitud del tamaño de la red es de, al menos, cien o mil nudos; por el contrario, las memorias locales no sobrepasan en mucho el tamaño de algunos megaoctetos. La red de intercomunicación es mucho más densa que en las redes de estaciones de trabajo y las demoras producidas en la comunicacion se miden en centenas de microsegundos.

La mayoría de las máquinas de paso de mensaje comercializadas actualmente tienen un grano de paralelismo medio. Pero la disminución del grano es objeto de trabajos universitarios e industriales muy numerosos. Este estilo arquitectónico representa una de las vías más prometedoras hacia un paralelismo extremadamente masivo. En efecto, en las arquitecturas de memoria compartida, los accesos a memoria pasan por la red de interconexión. El tiempo de acceso a memoria no puede ser peor que el de una máquina secuencial y crecerá con el tamaño del sistema, puesto que aumenta el número de etapas de la red a atravesar. Ahora bien, cada procesador efectúa al menos un acceso a memoria por instrucción, para la adquisición de ésta. Los mecanismos de *caché* y de combinación que pueden paliar este problema implican un aumento de la complejidad del *hardware* y del *software* que se opone al objetivo del paralelismo extremadamente masivo.

A la inversa, en las máquinas de memoria local, los intercambios de información más frecuentes (adquisición de la instrucción y manipulación de datos privados) se efectúan a corta distancia, por un acoplamiento clásico procesador-memoria. La demora introducida por la transmisión a través de la red no penalizará más que los intercambios, relativamente menos frecuentes, de procesos a procesos.

Aunque las máquinas de paso de mensaje estén deliberadamente organizadas para minimizar el uso de la red, las comunicaciones interprocesos continúan siendo el

cuello de botella del sistema. Esta situación se debe esencialmente a dos factores: las limitaciones tecnológicas sobre el número de ligazones físicas que deben pasar por la máquina, y la interface procesador/red.

Una solución es la adecuada asignación de los procesos a los procesadores para minimizar las comunicaciones. En la literatura se denomina a esta solución con el término mapeo. Como se define en [1], el mapeo es el problema de asignar los diferentes procesos a los procesadores físicos de tal forma que se maximice que el número de procesos que comunican caigan en pro-cesadores directamente conectados en la red. Un libro interesante de redes de interconexión donde se trata de forma extensa el problema del mapeo es el escrito recientemente por Hilbers [1].

Una novedosa solución para mejorar la comunicación entre procesos es la reconfiguración dinámica de la red de interconexión. Aunque en el apartado anterior se comentó que los multicomputadores poseían una red de tipo estático, últimamente se están desarrollando trabajos con redes dinámicas para aliviar el cuello de botella que representa la red de interconexión [1], [1]. Esta solución es adecuada sobre todo para aquellos problemas cuyo patrón de comunicaciones (a nivel temporal) no es regular.

4.2 Aspectos tecnológicos

El estudio de las redes de interconexión, en términos puramente topológicos (diámetro y distancia media), presupone una anchura de canal constante, es decir, que la anchura del camino de los datos entre dos procesadores unidos físicamente en la red sea la misma, cualquiera que sea la topología estudiada. Esta considera igualmente una demora constante de propagación por las conexiones. Bajo estas hipótesis, los resultados del modelo gráfico tienden a presentar como preferibles las redes donde los nudos tienen un grado fuerte (hipercubos) en relación a aquellas otras de grado débil (mallas).

Pero la limitación esencial de los sistemas basados en VLSI reside en las conexiones y no en los dispositivos que conmutan (lógica y memoria). A menudo, se modeliza esta limitación por la sección de la red, que es el número de líneas (de 1 bit de ancho) que atraviesan una mediatriz de la red.

Consideremos el caso de una aplicación en la que las comunicaciones están distribuidas uniformemente, es decir, las probabilidades de comunicación entre dos procesadores cualesquiera son iguales. La red será estable si el número medio de bits que pueden atravesar la mediatriz no sobrepasa la sección. La toma de conciencia de esta limitación en el campo de los multicomputadores es muy reciente. En efecto, para cientos de procesadores el recurso en conexiones es, en todo momento, suficiente. Para órdenes de magnitud superiores, deben hacerse dos preguntas:

· ¿Cuál es la posibilidad de realizar físicamente la red en dos o tres dimensiones, dado un número limitado de conexiones posibles entre componentes, de contactos en las tarjetas, etc.?

· ¿Cómo repartir la banda global de la red entre las aristas abstractas del grafo?.

El primer problema se conoce con el nombre de *packaging*, es decir, de la encapsulación de la pastilla y de sus conectores. Los progresos a este nivel son mucho más lentos que los correspondientes a la escala o densidad de integración, pero las perspectivas de la encapsulación de componentes en tres dimensiones (3D) es más clara cada día.

La segunda cuestión significa que, para realizar físicamente una red de dimensión elevada, es necesario trabajar sobre un espacio de dos o tres dimensiones, y distribuir luego las dimensiones de la red entre las dimensiones del espacio físico. Las dimensiones lógicas suplementarias crean líneas largas que, a su vez, aumentan la superficie de interconexión y el tiempo de propagación para el conjunto de la red. La comunicación puede basarse, entonces, en dos esquemas:

· Líneas largas, y por consiguiente estrechas y lentas, por efecto del tiempo de propagación por la línea y también por la secuenciación (serie) que deberá experimentar cada mensaje. Las líneas largas permiten topologías cuyo diámetro lógico es débil, por ejemplo los hipercubos.

· Líneas anchas en cuanto al número de líneas físicas en paralelo, pero que, en todo caso, deben ser cortas y rápidas. Las topologías correspondientes son regulares y ampliables, típicamente mallas o toros 2D ó 3D. Es el objetivo actual, tendente a alcanzar un paralelismo masivo. Por ejemplo, la red del futuro sucesor de los Hipercubos de Intel (proyecto Touchstone) está constituido por una malla 2D. Un análisis [1] muestra que, bajo la hipótesis de utilizar un número constante de hilos a través de la bisección de la red, una topología bidimensional minimiza la latencia para redes de hasta 1024 nudos. Para redes mayores, parece preferible una topología tridimensional.

4.3 Algunas máquinas comerciales

Por último, vamos a citar brevemente algunas de las máquinas comerciales que tienen la arquitectura de un multicomputador.

Las máquinas de paso de mensajes más populares se realizaron por Intel, con dos arquitecturas sucesivas: los iPSC/l y los iPSC/2. Son máquinas de grano bastante grueso, que tienden a alcanzar los rendimientos de las supercomputadoras, utilizando tecnologías mucho menos agresivas, y por tanto con un coste menor.

El otro ejemplo típico de multicomputador es el construido a partir del Transputer, un microprocesador diseñado especialmente para computación paralela. Un Transputer es un microprocesador que reúne en un solo chip un procesador, una memoria local y canales de comunicación que proporcionan una conexión punto-a-punto con otros Transputers. La arquitectura de la CPU es de tipo RISC. El control complejo se realiza por microcódigo. En particular, cada Transputer puede soportar varios procesos que se ejecutan concurrentemente. El ordenamiento de los procesos se lleva a cabo enteramente por microcódigo, lo que permite realizar una conmutación de contexto muy rápido (alrededor de 1 μs). El Transputer que más difusión ha tenido hasta el momento es el T800. Para posibilitar las redes de comunicación de alta velocidad, hace

pocos años se ha desarrollado la segunda generación de transputers, denominado T9000. Este nuevo transputers tiene un procesador superescalar, un planificador de tareas *hardware*, 16 Kbytes de memoria *caché* en el propio chip y un procesador de comunicaciones autónomo [1]. Sobre todo a nivel de la Comunidad Económica Europea, el diseño de máquinas basadas en el Transputer es muy importante.

5. El modo de encaminamiento

Para cualquier red de interconexión pero especialmente para las redes en los multicomputadores, uno de los factores de diseño más importante es el modo de encaminamiento que tiene la red. Como ya se comentó, la topología totalmente conectada es impracticable, por lo que se debe implementar algún mecanismo que se encargue de propagar los datos de un procesador a otro en función de la dirección contenida en dichos datos.

Como viene recogido en [1], el modo de encaminamiento de datos debe satisfacer un número de requerimientos, siendo los más importantes los siguientes:

* El protocolo de encaminamiento debe estar libre de bloqueo (*deadlock*).
* Ningún paquete de datos puede ser retrasado infinitamente en la red.
* Un mensaje siempre debe tomar el camino más corto para llegar a su destino.
* Dicho mecanismo se debe adaptar a las condiciones de tráfico de la red y debe explotar al máximo su ancho de banda.
* El protocolo de encaminamiento debe conseguir la más baja latencia para la red y el más alto rendimiento.
* Se debe asegurar que no haya adelantamientos en los mensajes. Es decir, dos mensajes enviados por un nodo a otro no pueden llegar en orden diferente a como fueron enviados.

Al hablar del modo de encaminamiento, hay que detallar por una parte el algoritmo de encaminamiento y por otra el control del flujo de mensajes. Con respecto a los algoritmos de encaminamiento, hay dos grandes grupos: los estáticos o deterministas y los dinámicos o adaptativos. Por *estático* se entiende un algoritmo que tiene un comportamiento definido y constante a lo largo del tiempo, habiéndose desarrollado varios modelos que garantizan la ausencia de bloqueos. Son algoritmos más sencillos de implementar pero tienen el problema de provocar en ocasiones puntos de congestión en la red. Por contra, los algoritmos *adaptativos* son aquellos que cambian su comportamiento a lo largo del tiempo en función de diversos parámetros de la red, mejorando por tanto la velocidad y el rendimiento en el envío de mensajes. Lógicamente, la complejidad de estos algoritmos es mucho mayor y requieren una circuitería adicional. Los principales objetivos del encaminamiento dinámico son reducir las colisiones entre mensajes y el tiempo total de transferencia e incrementar la tolerancia a fallos.

Una solución para mejorar las prestaciones de estos algoritmos es basa en el uso de canales virtuales. Un *canal virtual* es un canal de comunicación que se define entre dos procesadores de la red y que está soportado por un canal físico, pero sin coincidir con él, ya que por cada canal físico de comunicación se suelen definir varios canales virtuales. Los canales virtuales son multiplexados en el canal físico, por lo que comparten el ancho de banda de dicho canal. Las ventajas que se obtienen con los canales virtuales derivan del hecho de que cuando un mensaje está retenido en la red, no colapsa el canal físico por el que está viajando, dejándolo para aquellos mensajes que vayan por los canales virtuales que están compartiendo el mismo canal físico. Esta solución es sobre todo interesante para modos de encaminamiento que poseen un control de flujo del tipo *wormhole routing*. Lógicamente, se tiene que garantizar en el diseño del algoritmo de encaminamiento que es libre de bloqueo, realizando para ello una ordenación entre los canales y definiendo un grafo de dependencias que garantice la anterior propiedad. Cada canal virtual tiene su propio *buffer* a nivel de flit, su propio control y su propio camino para los datos.

La principal dificultad en el diseño de los algoritmos de encaminamiento es que se debe asegurar que éste sea libre de bloqueo. Un bloqueo en una red de interconexión ocurre cuando ningún mensaje puede avanzar hacia su destino debido a que todas las colas de mensajes de los nodos del sistema están llenas. El tamaño de las colas tiene una gran influencia en la probabilidad de alcanzar una configuración que esté bloqueada. De todas formas, el modo más práctico para evitar el bloqueo es desarrollar algoritmos de encaminamiento libres de bloqueo. En estos últimos años, ésta es una de las áreas de más estudio e investigación. Varios desarrollos libres de bloqueo existen en redes con encaminamiento del tipo almacenamiento y reenvio [1]. En el caso de encaminamiento segmentado la situación es más complicada, aunque se han propuesto metodologías en el caso de encaminamiento estático [1]. Para evitar el incremento de la congestión de la red, se pueden emplear algoritmos de encaminamiento adaptativo, los cuales tienen que ser libres de bloqueo. Se han desarrollado varios algoritmos de encaminamiento adaptativos libres de bloqueo para redes con encaminamiento segmentado: en el caso del hipercubo se ha diseñado el algoritmo "Hyperswitch". También se han desarrollado teorías que permiten garantizar la ausencia de bloqueo en presencia de dependencias cíclicas entre canales, lo cual permite una mayor flexibilidad y mayores prestaciones [1].

Junto a los algoritmos de encaminamiento hay otro importante aspecto a tener en cuenta: los mecanismos de control de flujo. En la actualidad se han desarrollado una diversidad de mecanismos para, una vez determinado el camino que deben seguir los mensajes, manejar el flujo de mensajes de un nodo origen a uno destino. Los principales mecanismos de control de flujo son los siguientes:

<u>La conmutación de circuitos</u>

En esta técnica -la primera que se desarrolló- los nodos que se comunican en un instante dado se unen por un camino físico que no se modifica mientras dura la comunicación. Esta técnica es similar a lo que ocurre en la red telefónica.

Fundamentalmente se emplea en las redes dinámicas de los multiprocesadores, no teniendo mucho interés en el caso de los multicompu-tadores.

La conmutación de mensajes

También se denomina de almacenamiento y reenvio. Esta técnica almacena cada mensaje completamente en un nodo y entonces lo transmite al siguiente nodo. El mensaje construye su camino paso a paso en la red. Esta técnica es similar a la empleada en las redes telemáticas. Cada nodo debe disponer de un *buffer* donde retener los mensajes que le llegan, perdiendo una cantidad de tiempo apreciable en almacenar y recuperar de la memoria estos mensajes. Esta técnica provoca una alta latencia en la red de interconexión, estando influida por la distancia que hay entre los nodos que comunican. La latencia de la red es (L/B)D, donde L es la longitud del mensaje, B es el ancho de banda del canal y D es la longitud del camino entre los nodos fuente y destino. Otra desventaja de este mecanismo de control es que aumenta el tamaño de la memoria local requerida en cada procesador. Esta técnica fue adoptada por la mayoría de los multicompu-tadores comerciales de la primera generación, tales como iPSC-1, Ncube-1, Ametek 14 o FPS-T.

Wormhole o encaminamiento segmentado con espera

En esta técnica cada mensaje se descompone en pequeños fragmentos denominados unidades de control de flujo o *flits*. El primer flit es la cabecera y contiene la dirección de destino del mensaje. El último flit es la cola y los flits intermedios contienen solamente datos. Cuando el flit de cabecera llega a un nodo, se encamina inmediatamente. A medida que el flit avanza, va reservando la ruta y los demás lo siguen. Los canales reservados por la cabecera serán liberados cuando pase el flit de cola. Si la cabecera no encuentra un canal libre para poder continuar, se detiene temporalmente hasta que se libere. El mecanismo de control de flujo detiene los restantes flits del mensaje. Conforme los *flits* van avanzando, el mensaje está partido a lo largo de los canales entre el nodo fuente y el nodo destino. Es posible que el *flit* de cabeza haya llegado al nodo destino mientras parte del mensaje aún no ha salido del nodo fuente. Debido a que los *flits* no contienen información acerca del destino (excepto el de cabeza), los diversos *flits* de un mensaje no se pueden mezclar con los *flits* de otro mensaje. Por tanto, cuando la cabeza de un mensaje es bloqueada, todos los *flits* de ese mensaje paran de avanzar y bloquean a su vez cualquier otro mensaje que necesite transitar por los canales que ellos ocupan. El nombre de este mecanismo de control le viene debido a que el avance del mensaje recuerda el avance de un gusano. Sus características más importantes son:

* No se almacenan los mensajes en los nodos intermedios.
* El encaminamiento es distribuido.
* Los algoritmos de encaminamiento son implementados por hardware.
* La posibilidad de bloqueo es más acentuada.
* El tiempo total de transferencia es mucho más reducido.

* El mensaje es descompuesto en bloques (cabecera, bloques de información y cola).

* Se encamina la cabecera, sin esperar al resto del mensaje.

* Los enlaces se reservan hasta que pasa la cola.

En este caso, la latencia de la red es $(L_f/B)D + L/B$, donde L_f es la longitud de cada flit, B es el ancho de banda del canal, D es la longitud del camino y L es la longitud del mensaje. Como $L_f << L$, la latencia de la red no se ve prácticamente afectada por la distancia que tiene que recorrer el mensaje. El primer multicomputador comercial que adoptó esta técnica de control de flujo fue el Ametek 2010, el cual usaba una topología de malla 2D. El Ncube-2 también usa este encaminamiento con una topología hipercubo. Asimismo, el Intel Touchstone Delta y el Intel Paragon usan esta técnica en una malla 2D. Por último, el proyecto que se está desarrollando en el MIT denominado J-Machine también usa el encaminamiento segmentado en una malla 3D.

Virtual cut-through

Es una técnica similar a la anterior [1]. Se diferencia en que ésta almacena el mensaje en un *buffer* cuando se bloquea, quitándolo de la red y permitiendo el paso de otro mensaje. Esto alivia el cuello de botella del algoritmo anterior y mejora el rendimiento de la red, a costa de una circuitería más compleja.

Double buffering

Este método intenta evitar el problema del *deadlock* de la red. Para ello lo primero que hace el nodo fuente es almacenar en un *buffer* el mensaje que va a ser enviado. Despues intenta establecer una conexión con el nodo destino por formar un camino *rígido* a través de los nodos intermedios. Si se encuentra un bloqueo o un fallo en alguno de estos nodos intermedios, se vuelve de nuevo al nodo origen, intentando volver a formar dicho camino después de un retraso aleatorio. Una vez que se ha conseguido la conexión rígida entre el nodo fuente y el nodo destino, el mensaje entero es transferido, siendo almacenado en un *buffer* en el nodo destino antes de ser despachado. De ahí el nombre de esta técnica.

Mad Postman

Esta técnica está basada en el concepto de *redes virtuales*, un método para evitar el *deadlock* en la red de interconexión. Su funcionamiento en cada nodo consiste en sacar cada paquete de datos (cada *flit*) que está de paso por la misma dimensión por la que ha llegado, tan pronto como llega el primer bit al nodo, decidiendo más tarde cuál era el encaminamiento más adecuado. Debido a ello se pueden producir un número de *flits* enviados a nodos *equivocados*, pero que rápidamente desaparecerán de la red. Como máximo, este número de *flits* es igual a la dimensión de la red. El nombre de esta técnica le viene precisamente de esta *extraña* forma de encaminar los mensajes. A pesar de todo, ofrece una latencia muy reducida en la red, siendo adecuado para mallas con canales con poco ancho de banda.

Pipelined Circuit Switching o conmutación segmentada de circuitos

Es una nueva modalidad de la técnica de conmutación de circuitos desarrollada por Gaughan y Yalamanchili en la Universidad de Atlanta. El objetivo es desarrollar un control de flujo que sea más resistente a los fallos que pueda haber en la red, y que en los computadores masivamente paralelos, al tener un número elevado de nodos, es más probable que haya un cierto número de fallos en la red. Se trata de enviar la cabecera del mensaje por delante, construyendo el camino mínimo entre el nodo origen y el destino (utiliza un algoritmo de encaminamiento adaptativo), volviendo atras sobre un enlace cuando no puede continuar por alguna causa (por ejemplo, un fallo en la red). Una vez el camino ha sido establecido, los datos son enviados de forma segmentada. Esta técnica es muy resistente a fallos y tiene unas prestaciones similares a las de *wormhole*.

6. Conclusiones

En este artículo se ha presentado el estado del arte de las redes de interconexión para computadores masivamente paralelos. Estos computadores, que tienen un diseño MIMD según la clasificación de Flynn, son los que se están usando cada dia más para alcanzar la potencia de cálculo y las prestaciones que hoy en dia se les exige a los ordenadores. Fundamentalmente, nos hemos centrado en los multicomputadores como máquinas que, debido a mejor escalabilidad y relación prestaciones/coste, se están utilizando cada dia más. El gran avance que se ha producido en la tecnología VLSI también ha facilitado mucho el desarrollo de los multicomputadores. Los principales tópicos de las redes de interconexión han sido tratados, exponiendo en que consiste cada problema y en qué estado se encuentran las investigaciones acerca de ese punto. Especial realce se ha hecho en las consideraciones topológicas y en el modo de encaminamiento de los mensajes en la red.

Referencias

[1] Flynn, M.J. **Some computer organizations and their efectiveness**. *IEEE Trans. on Computers*, Vol. C-21, pp. 948-960, 1972.

[1] Germain-Renaud, C. y Sansonnet, J.P. *Ordenadores Masivamente Paralelos*. Paraninfo, Madrid, 1993.

[1] Reed, D.A. and Fujimoto, R.M. *Multicomputer Networks: Message-Based Parallel Processing*, MIT Press, Cambridge, Mass., 1987.

[1] Siegel, H.J. *Interconnection Networks for Large-scale Parallel Processing*. McGraw-Hill, New York, 1990.

[1] Athas, W.C. and Seitz, C.L. **Multicomputers: Message-passing concurrent computers**. *IEEE Computer*, Vol. 21, No. 8, pp. 9-24, August 1988.

[1] Bokhari, S.H. **On the mapping problem**. *IEEE Trans. on Computers*, Vol. C-30, No. 3, pp. 207-214, March 1981.

[1] Hilbers, P.A. *Processor Networks and Aspects of the Mapping Problem*. Cambridge University Press, Cambridge 1991.

[1] Bauch, A.; Braam, R. and Maehle, E. **DAMP: A dynamic reconfigurate multiprocessor system with a distributed switching network**. *2nd European Distributed Memory Computing Conference,* Munich, April 1991.

[1] García, J.M. and Duato, J. **Dynamic reconfiguration of multicomputer network: Limitations and Tradeoffs**, in P. Milligan and A. Nuñez (Eds.), *Euromicro Workshop on Parallel and Distributed Processing*, IEEE Computer Society Press, pp. 317-323, 1993.

[1] Dally, W.J. **Performance analysis of k-ary n-cube interconnection networks**. *IEEE Trans. on Computers*, Vol. C-39, No. 6, pp. 775-785, June 1990.

[1] May, M; Thompson, P. and Welch, P. *Networks, Routers and Transputers: Function, Performance and Application*. IOS Press, Amsterdam 1993.

[1] Jesshope, C.R., Miller, P.R. and Yantchev, J.T. **High performance communications in processor networks**. *Proc. 16th. Int. Symp. Computer Architecture,* Jerusalem, Israel, May 1989.

[1] Gelernter, D. **A DAG-based algorithm for prevention of store-and-forward deadlock in packet networks**. *IEEE Trans. Computers*, Vol. C-30, No. 10, pp. 709-715, October 1981.

[1] Dally, W.J. and Seitz, C.L. **Deadlock-free message routing in multiprocessor interconnection networks**. *IEEE Trans. Computers*, Vol. C-36, No. 5, pp. 547-553, May 1987.

[1] Duato, J. **On the design of deadlock-free adaptive routing algorithms for multicomputers: theoretical aspects**. *Proc. 2nd European Distributed Memory Computing Conference*, Munich, FRG, April 1991.

[1] Kermani, P. and Kleinrock, L. **Virtual cut-through: a new computer communication switching technique**. *Computer Networks*, Vol. 13, pp. 267-286, 1979.