UNIVERSIDAD DE CANTABRIA

DPTO. DE ELECTRÓNICA Y COMPUTADORES.

IMPACTO DEL SUBSISTEMA DE COMUNICACIÓN EN EL RENDIMIENTO DE LOS COMPUTADORES PARALELOS: DESDE EL HARDWARE HASTA LAS APLICACIONES.

Presentada por:

Valentin Puente Varona

Dirigida por:

Ramón Beivide Palacio.

SANTANDER, OCTUBRE DE 1999

Capítulo 1

Introducción

En este primer capítulo introduciremos las características fundamentales de los computadores paralelos, exponiendo el papel que juegan frente a diversos problemas de gran calado. A continuación, plantearemos la importancia de la red de interconexión como elemento básico cohesionador de los elementos de proceso que da lugar al computador paralelo. Por último, motivaremos el trabajo planteando los objetivos marcados en la realización de este estudio, así como la estructura de esta memoria.

1.1 Introducción.

El continuo avance en la capacidad de integración de la tecnología microelectrónica ha permitido un incremento notable en la capacidad de cálculo de los procesadores. Esto, junto a las mejoras arquitectónicas, ha propiciado que, en la actualidad, muchos problemas únicamente resolubles hace poco más de una década por los más potentes supercomputadores, sean hoy en día prácticamente abordables con un computador convencional.

De la misma forma, a medida que la capacidad de cálculo de los procesadores se incrementa es factible afrontar nuevos retos en forma de problemas cada vez más complejos. Aún así los requerimientos de cálculo de ciertas aplicaciones siempre se encuentran un paso más allá de las posibilidades que nos ofrece la tecnología monoprocesador. En la mayoría de los casos, para emprender esos desafíos es preciso recurrir a el empleo de computadores paralelos. De acuerdo con la definición genérica de que un computador paralelo se entiende como un conjunto de elementos de proceso independientes, que operan de forma conjunta para resolver grandes problemas de una forma rápida [1], es claro que si esos elementos de proceso independientes están en consonancia con los procesadores ofrecidos por la tecnología actual, también los computadores paralelos se beneficiaran de esos avances permitiendo afrontar esas nuevas exigencias. De cualquier modo, aún existen multitud de casos en los que las potencias de cálculo de los computadores paralelos actuales no llegan a permitir su resolución. Muchos de ellos en campos tan diversos como la medicina (biología molecular, genoma humano), la física (QCD, superconducción), etc..(Ver Figura 1-1). Es por ello que, pese al más que previsible futuro incremento de capacidad de cálculo de los procesadores, los computadores paralelos seguirán siendo de vital importancia dentro del campo de la arquitectura de computadores.

En este sentido, el rendimiento ofrecido por los computadores paralelos actuales se ha incrementado de forma considerable. Actualmente sus elementos básicos de cálculo están constituidos por componentes diseñados para ser empleados, casi siempre, en sistemas monoprocesador. Este cambio de tendencia, ya que hace no demasiados años los computadores paralelos empleaban elementos diseñados específicamente, ha producido un crecimiento similar en las potencias de cálculo de los computadores paralelos con respecto a los monoprocesador. Como se puede apreciar en la Figura 1-2, en poco más de 6 años el rendimiento máximo alcanzado por un computador paralelos ha pasado, de poco más de 50 Gigaflops, a ser en la actualidad de más de 2 Teraflops, lo que representa un incremento en la potencia de cálculo de más de 35 veces en 6 años.

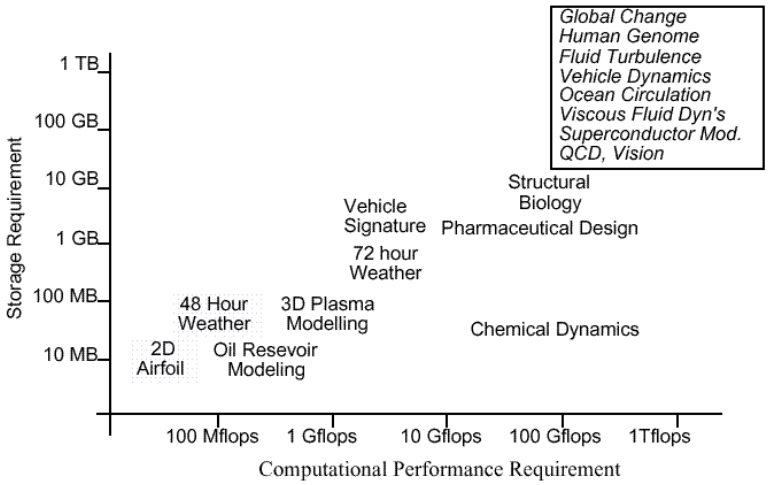

Figura 1-1. Requerimientos de potencia de cálculo para algunas aplicaciones típicas [37].

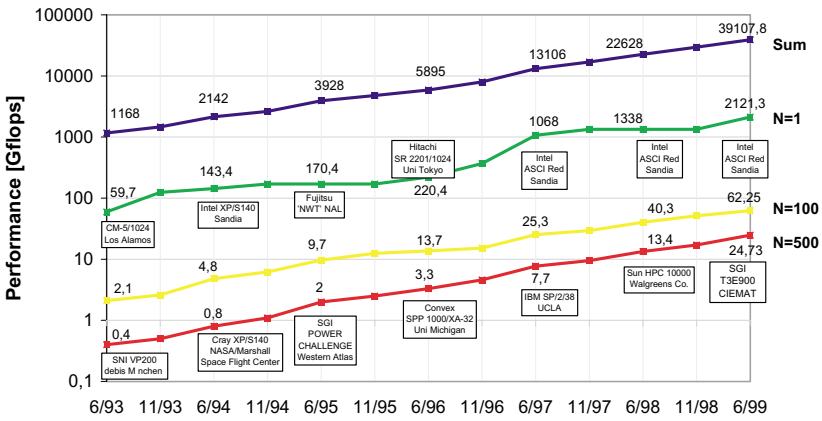

Figura 1-2. Evolución del rendimiento máximo alcanzado por los computadores paralelos [128].

Por otra parte, existen proyectos de desarrollo de sistemas paralelos de elevado rendimiento, denominados ultracomputadores. Entre los más relevantes podemos citar el proyecto ASCI (*Advanced Strategic Computing Initiative*)[84] del DOE (Ver Figura 1-3). En el seno de este proyecto se han logrado desarrollar sistemas que alcanzan rendimientos superiores al Teraflop. Su objetivo final es alcanzar potencias de cálculo por encima de los 100 Teraflops. Para dar lugar a esas elevadas potencias de cálculo, la idea es emplear elementos utilizados en computadores convencionales. Aunque la principal razón que ha motivado este proyecto ha sido el reemplazar los ensayos nucleares reales por simulaciones, en segundo plano, el objetivo que se

persigue es permitir a los constructores incrementar su capacidad tecnológica, lo que se traducirá en un incremento de potencia considerable en los computadores paralelos comerciales.

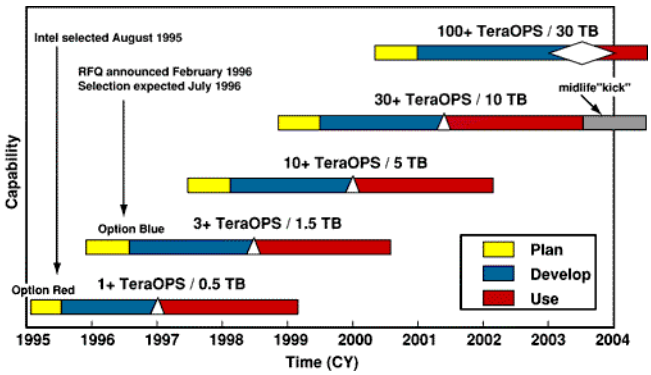

Figura 1-3. *Roadmap* del proyecto ASCI [83].

De modo subyacente a esta iniciativa, esta el proyecto *PathForward* [33]. Su objetivo es extender la tecnología desarrollada en la construcción de estos sistemas y avanzar sobre la meta final del proyecto. En principio, los constructores no pueden invertir grandes sumas de dinero en el desarrollo de esta clase de sistemas por su cuenta ya que poseen un mercado muy reducido. Sin embargo, es preciso el avance en este campo, dado la gran de cantidad de problemas por resolver y la limitación de los sistemas actuales para abordarlos.

Entre los objetivos fundamentales que persigue el proyecto *PathForward* cabe señalar el desarrollo de nuevos sistemas de interconexión que permitan conectar de modo eficiente el elevado número de elementos de cálculo que se requiere para superar los límites actuales. Se pretende desarrollar la tecnología necesaria para conectar hasta 10.000 nodos y que los tiempos de acceso se mantengan en el peor de los casos por debajo de 1000 ciclos de reloj del procesador. De otra forma, la capacidad del sistema global se vería seriamente comprometida. Es claro que la red de interconexión puede representar un claro cuello de botella en este tipo de sistemas, llegando a limitar las posibilidades que dan todos los elementos individuales de cálculo de forma independiente. Las situaciones en las que el número de nodos es muy elevado, aún siendo un caso poco frecuente dentro de los computadores paralelos en la actualidad, nos permite constatar la notable importancia del subsistema de comunicación en los computadores paralelos.

La red de interconexión actúa como medio de comunicación entre todos los elementos que están cooperando para resolver cada aplicación o problema; para que el sistema alcance un rendimiento aceptable es preciso que la comunicación entre sus elementos sea óptima. En este sentido se entiende por óptima, que la capacidad de cálculo del sistema este equilibrada con respecto a las prestaciones exhibidas por la red de interconexión. Es por ello que, a medida que la

capacidad de cálculo se incrementa, las características de la red han de mejorar de forma correlacionada. De la misma forma que en los procesadores, la evolución de la red de interconexión se ve incrementada a medida que la tecnología VLSI avanza. Sin embargo, existen limitaciones inherentes a las características del sistema que puede limitar su desarrollo más que en el caso de otros elementos de los computadores. Entre estos factores podemos citar que el coste de este subsistema es mayor, lo que en la mayoría de los casos impide emplear soluciones prohibitivas tales como usar enlaces individualizados entre todos los elementos de cálculo. Por otro lado, las restricciones físicas en forma de retrasos u organización espacial pueden limitar también el impacto producido por los avances tecnológicos. La evolución constante en el rendimiento de los elementos de cálculo hará que progresivamente la red se convierta en un cuello de botella para el sistema. Por tanto, será preciso aportar nuevas soluciones arquitectónicas que permitan paliar, en la medida de lo posible, sus limitaciones inherentes.

En el estudio presentado en esta memoria, el foco de atención es precisamente este subsistema. Analizaremos y propondremos diferentes alternativas para la red de interconexión, evaluando cual es su influencia en el rendimiento de los computadores paralelos.

1.2 Taxonomía de los Computadores Paralelos.

Cuando definimos un computador paralelo como un conjunto de elementos cooperando para resolver grandes problemas, es preciso clasificar de qué forma se produce esa colaboración. Sabemos que el substrato básico que facilita la comunicación necesaria en la cooperación es la red de interconexión. Sin embargo, es necesario contar con mecanismos en niveles superiores que establezcan de qué manera se produce la comunicación entre cada uno de los elementos de cálculo del sistema. Atendiendo a la forma en que se realiza este intercambio de información podemos clasificar los computadores paralelos en las familias que se señalan a continuación[1].

1.2.1 Computadores de Paso de Mensajes.

También denominados multicomputadores, su característica fundamental es que ha de ser el propio programador quien indique de forma explícita en el código, cómo y cuándo han de ocurrir las comunicaciones. Esta clase de sistemas es denominada de paso de mensajes [37], porque la comunicación se realiza en base a primitivas de envío y recepción de mensajes (*send/recv.*). Las librerías típicas empleadas en el desarrollo de software para estos sistemas son MPI [65] y PVM [58] entre otras.

1. Aunque desde el punto de vista general, esta taxonomía no es estrictamente correcta ya que algunos sistemas de un tipo pueden actuar como el otro incorporando, en la mayoría de los casos, mecanismos basados en *software*, si es una clasificación clarificadora atendiendo a las características *hardware*.

En este tipo de arquitecturas, es habitual emplear computadores completos como bloques básicos. Cada uno de los nodos de cálculo incorpora todos los elementos típicos de un computador (procesador/es, jerarquía de memoria, bus del sistema y bus de entrada/salida). En la mayoría de los casos, la red de interconexión, desde el punto de vista de cada nodo, forma parte de su sistema de entrada/salida. Con estas características es posible construir un computador paralelo de paso de mensajes a partir de computadores personales o estaciones de trabajo sin más que contar con una red de interconexión eficaz y un soporte *software* que facilite la comunicación entre los nodos. Esto es lo que se ha dado en llamar NOW (*Network Of Workstations*). Este es uno de los campos mas atractivos para este tipo de sistemas. De hecho, sistemas construidos empleando esta aproximación como los clusters *CPlant* [60] o *Avalon* [136] se sitúan entre los 150 ordenadores más potentes del mundo.

La principal ventaja de los multicomputadores es su buena escalabilidad y adecuado rendimiento, especialmente bajo determinadas circunstancias. Si embargo, su rango de utilización ha sufrido un pequeño retroceso frente a arquitecturas más amigables al usuario como las de memoria compartida escalables. Este es el inconveniente más serio para estos sistemas: dificultad de uso y versatilidad. Es francamente dificil saber explotar la potencia real del sistema. De cualquier forma, trabajos como [14] indican que determinados problemas con elevadas exigencias siguen haciendo plenamente vigente este tipo de sistemas.

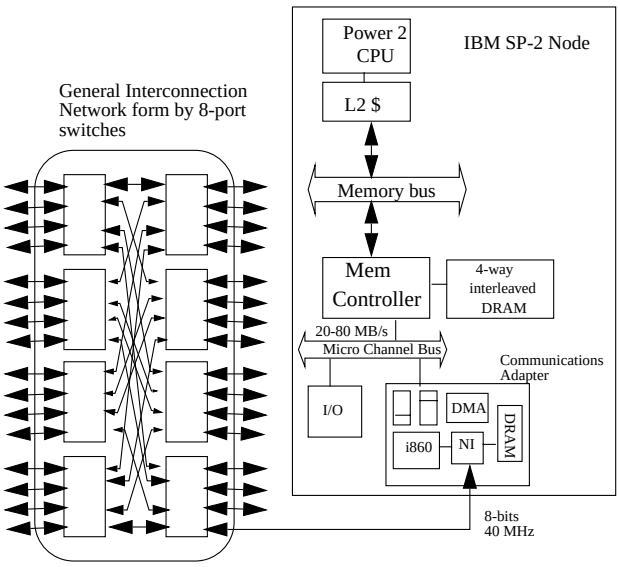

Figura 1-4. Estructura de una arquitectura de paso de mensajes. Caso del IBM SP-2.

Dado que en la mayoría de estos sistemas la red de interconexión se encuentra ubicada como parte del sistema de entrada/salida del computador, es necesario contar con un elemento que facilite la comunicación entre el procesador local y el subsistema de comunicación. Habitual-

mente esta tarea es llevada a cabo por la interface de red. Un ejemplo de este tipo de sistemas es el *IBM SP2* [120]. En la Figura 1-4 se muestra la estructura del sistema de comunicación de esta máquina así como las características de los nodos de proceso.

El papel que juega la red de interconexión es fundamental. Sin embargo, es necesario ser conscientes de que el coste de la comunicaciones depende de factores externos al *hardware* de la red o incluso de la propia interface de red. Uno de los más críticos es el *overhead* que introduce el *software* empleado como librería de paso de mensajes y su interacción con el sistema operativo del nodo. Existen multitud de trabajos que muestran como el coste introducido por las llamadas a las primitivas de envio y recepción pueden llegar a incrementar el coste de las comunicaciones en casi un orden de magnitud para el mismo *hardware* en el subsistema de comunicación [18][106]. Por otro lado, el modo en que se comunica la interface de red con el procesador *host* juega un papel muy importante en este nivel. Es importante indicar que, en muchos casos, un programador experto puede minimizar el impacto de la red en el rendimiento solapando correctamente tiempos de cálculo y comunicación.

1.2.2 Computadores de Memoria compartida

La característica clave de este tipo de arquitecturas es que las comunicaciones entre elementos de proceso se producen de forma implícita y como resultado de operaciones de acceso a memoria convencionales (*load* y *stores*).

Bajo esta aproximación, la programación de aplicaciones para este tipo de sistemas es mucho más amigable que en el caso de los computadores de paso de mensajes, puesto que en la mayoría de las situaciones solo es preciso el incorporar algunas directivas de compilación en el código secuencial para obtener el código paralelizado. Esta facilidad contrasta con los sistemas de paso de mensajes, donde el empleo de comunicaciones explícitas fuerza en muchos casos a emprender grandes cambios en el código secuencial de las aplicaciones e incluso rediseñar el algoritmo de resolución del problema. La mayor facilidad de programación de estos sistemas radica en que el programador siempre tiene un espacio global de direcciones, visto de la misma forma por todos los procesadores del sistema.

Además de esta facilidad de programación, otra ventaja clara de este tipo de sistemas es un uso uniforme de los recursos del sistema, lo que desde el punto de vista del usuario es muy conveniente cuando se pretende usar el sistema en *throughput* de tareas u otros cometidos no estrictamente ligados al cálculo paralelo. En otras palabras, la versatilidad de este tipo de sistemas es muy superior a la de los computadores de paso de mensajes.

En función de cómo está distribuida la memoria del sistema, esta clase de arquitecturas se puede subdividir en sistemas UMA (*uniform memory access*) y NUMA (*non-uniform memory access*).

Los primeros sistemas paralelos en emplear la aproximación UMA datan de principios de los 60, con sistemas como el *Burroughs 5500*, *CDC 3600*, y el *IBM System/360 Model 50* entre otros. Actualmente, en los sistemas de rango medio-bajo son claramente la aproximación más empleada. Dentro de esta categoría podemos encontrar desde computadores personales de alto rendimiento con soporte para dos y cuatro procesadores, hasta servidores departamentales con un número más alto de procesadores. Empleando procesadores usados habitualmente en los computadores personales o estaciones de trabajo se dota al sistema de un medio de comunicación tal que, como se muestra en la Figura 1-5, permite una visión uniforme de la memoria del sistema por parte de los procesadores. De la misma forma el empleo de un medio compartido como red de interconexión, por ejemplo en forma de bus, facilita todos los aspectos relacionados con el mantenimiento de la coherencia entre las caches de cada procesador mediante protocolos como el *snoopy* basados en *broadcast*.

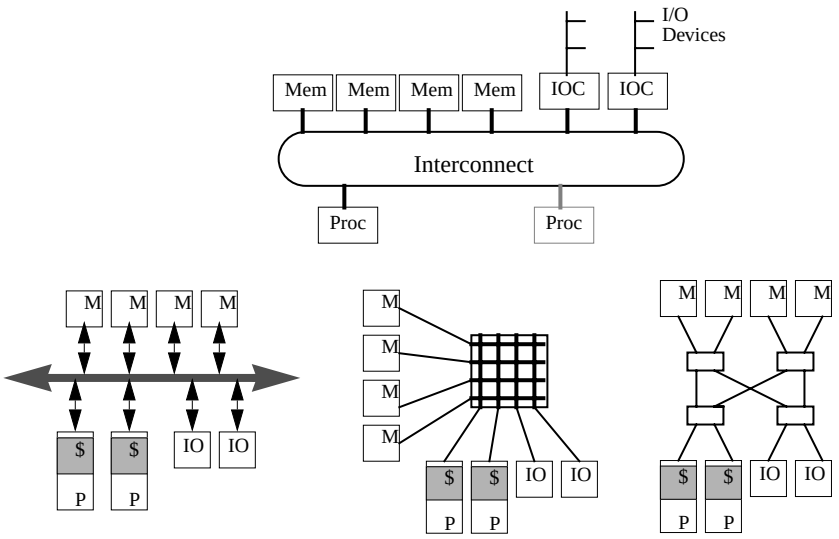

Figura 1-5. Arquitecturas UMA.

El empleo de caches individualizadas por procesador permite reducir considerablemente los requerimientos de ancho de banda por parte de los procesadores hacia el medio de comunicación. Sin embargo, pese a la incorporación de este tipo de mecanismos, habitualmente los sistemas UMA presentan una escalabilidad limitada. Generalmente por encima de un número de elementos de proceso no demasiado elevado, con las tecnologías actuales de implementación no es posible desarrollar medios compartidos con suficiente capacidad. Es habitual que el

número máximo de procesadores este limitado a 16 ó 32. Algunos ejemplos recientes, como el *Sun E10000* [29], logran escalar hasta 64 procesadores bajo la aproximación UMA (Ver Figura 1-6). En este sistema se emplea un bus híbrido (medio compartido en direcciones y crossbar en datos) denominado *Gigaplane-XB* que soporta un ancho de banda de casi 13 Gbytes/sec. Estas prestaciones facilitan que el sistema logre escalar hasta 64 procesadores, pero aún así, superar esta barrera, sigue siendo difícil incluso empleando arquitecturas extremadamente sofisticadas como ésta.

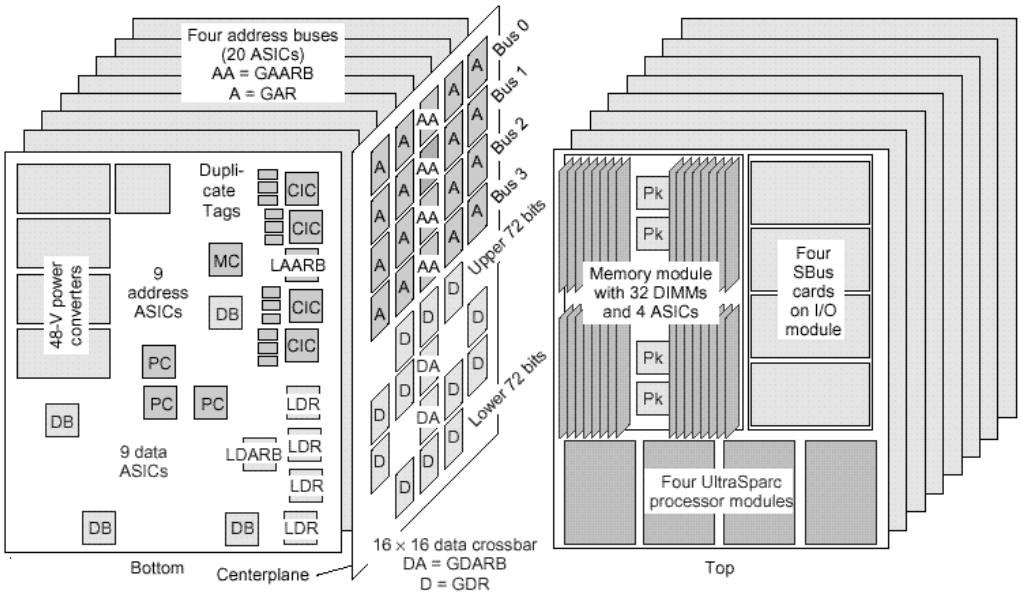

Figura 1-6. Estructura de una arquitectura de memoria compartida UMA. Caso del Sun E10000 Enterprise Server [29].

Los sistemas de memoria compartida de acceso no uniforme, o también denominados DSM (*Distributed Shared Memory*) surgieron como una alternativa a la clase de sistemas anteriores y en donde se pretendía mantener la facilidad de uso, pero permitiendo escalar el número de nodos del sistema hasta cifras similares a los computadores de paso de mensajes. El primer computador que empleó esta aproximación fue el CMU *C.mmp* [138] a principios de la década de los 70. Desde este primer sistema, donde para alcanzar un rendimiento razonable siempre era necesario recurrir al empleo de paso de mensajes [14], se han ido produciendo avances hasta la actualidad, donde este tipo de sistemas están introduciéndose de forma rápida en el mercado de los computadores paralelos e incluso un DSM, como el *Blue Mountain* de SGI, ha llegado a situarse entre los ordenadores más potentes del mundo. Con este computador se espera alcanzar un rendimiento sostenido de más de 3 Teraflops.

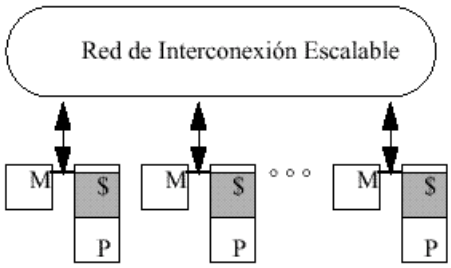

Figura 1-7. Arquitectura de un sistema ccNUMA.

En realidad, este tipo de sistemas ha experimentado un crecimiento explosivo a partir de principios de los 90. El primer sistema de estas características comercial fue el *KSR-1* [52]. Soportaba hasta 32 procesadores en anillo y fue el primer computador NUMA que era capaz de mantener coherencia por hardware (ccNUMA). Posteriormente surgieron prototipos como el *Stanford DASH* [85] de los que derivaron computadores que comenzaron a ser comercializados a mediados de la década de los 90, como el *SGI Origin 2000* [82] (Ver Figura 1-8). Existen otros ejemplos de sistemas que emplean esta aproximación como el *Sequent NUMA-Q* [87], *General NUMALiine* [34] o *Convex Exemplar* [127], todos ellos basados en el estándar *SCI* [110].

Frente al número de procesadores que nos podemos encontrar en un sistema UMA en este caso es posible escalar el tamaño del sistema hasta más de 512 nodos[1], cifra completamente impensable para un sistema SMP(*Symmetric Multi-Processor).*

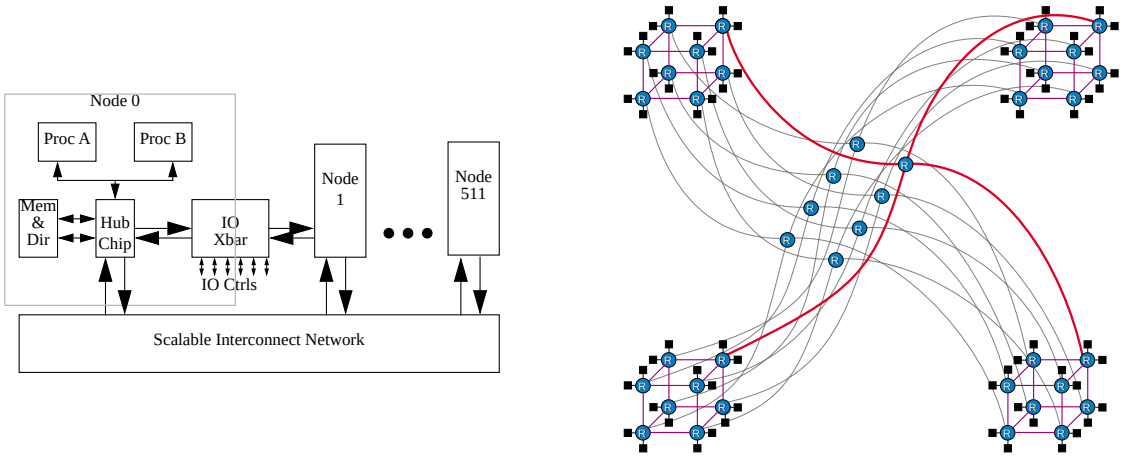

Figura 1-8. Estructura de una arquitectura de memoria compartida NUMA. Caso del SGI Origin 2000.

1. En la actualidad existen algunos sistemas DSM con coherencia *hardware* con hasta 2048 procesadores [128].

En general, la aproximación NUMA no requiere del empleo de mecanismos que permitan que todos los nodos de proceso tengan una visión coherente del espacio de direcciones global. Es posible manejar un espacio común de direcciones y que sea el programador el encargado de manipular el contenido de esas direcciones de memoria de forma segura, evitando cualquier posibilidad de error derivado de problemas de falta de coherencia. Esta es la aproximación empleada por el *Cray T3E* [112].

En el caso de las arquitecturas NUMA la red de interconexión juega un papel crítico. A diferencia del caso de los multicomputadores, la interface de red y la propia red de interconexión forma parte del sistema de memoria del computador. De hecho, el papel de la interface de red en muchos casos se confunde con el propio controlador de memoria, es por ello que en ocasiones se denomine agente de comunicación [64]. La situación de la red en ese contexto hace que en muchos casos juegue un papel más importante en el rendimiento del sistema que el caso de los computadores de paso de mensajes. Es necesario señalar que dado que cada procesador ve un espacio común de direcciones a nivel lógico pero distribuido a nivel físico, cada una de los accesos a posiciones de memoria que se encuentran fuera del rango mantenido por el nodo local se convertirán en accesos remotos. De esta forma, de manera transparente al programador, la comunicación entre nodos se realiza a través de peticiones de bloques. Para que el sistema no vea afectado su rendimiento es preciso que la latencia exhibida por la red de interconexión no incremente los accesos remotos excesivamente (pese a que porcentualmente el número de accesos remotos es muy inferior al de accesos locales, una elevada penalización en cada fallo puede afectar de forma notable al rendimiento del sistema completo). Además, dado que se trata de sistemas escalables, es preciso que el susbsistema de comunicación escale correctamente al incrementar el número de elementos de proceso.

1.3 Arquitecturas S-SMP.

Bajo la clasificación establecida en el punto previo, en este trabajo nos centraremos en evaluar de qué manera influye el subsistema de comunicación en una arquitectura de tipo NUMA con coherencia mantenida por hardware. Este tipo de sistemas se denominan también S-SMP (*Scalable Symmetric Multi-Processors*)[86].

Cuando en un sistema multiprocesador de memoria compartida se emplean caches individualizadas por procesador, aparece el problema del mantenimiento de la coherencia de datos. Es claro que en este caso cuando varios procesadores se encuentran accediendo a una misma posición de memoria y uno de ellos modifica su contenido, es necesario proveer al sistema de un mecanismo que evita una visión incoherente de esa posición por parte del resto de procesadores.

Bajo estas condiciones, un sistema sera coherente si bajo cualquier circunstancia todos los procesadores ven el mismo contenido en la memoria [119]. Un modo de mantener una visión coherente de la memoria es a través de llamadas de invalidación o actualización por parte del procesador que va a modificar el valor de la posición de memoria hacia todos aquellos que lo tengan almacenado en su cache. En el caso de los sistemas SMP es sencillo realizar cualquiera de estas operaciones si el sistema de interconexión esta basado en un medio compartido, ya que la operación de *broadcast* es inherente al sistema. Sin embargo en un sistema NUMA, con redes de interconexión más complejas, no es posible recurrir a esta aproximación.

Dependiendo de qué manera es compartida la información de forma segura por cada uno de los elementos de cálculo podemos clasificar, a grandes rasgos, estos sistemas en: ccNUMA y COMA.

ccNUMA

En este tipo de sistemas, en cada nodo reside una porción de la memoria total del sistema. Los datos que ha de manejar la aplicación, en forma de variables compartidas, son distribuidos a lo largo del sistema por el programador, cargador o sistema operativo. La distribución de esta información se hará de tal forma que solo existirá una única copia de cada variable en la memoria principal del sistema. El modo de mantener la coherencia, cuando una misma posición de memoria esta siendo accedida por dos o más procesadores, se realiza habitualmente en base a directorios [28]. Históricamente el control de coherencia basado en directorio es anterior a los protocolos de coherencia basados en *snoopy* [119].

Habitualmente, cada nodo puede contar con uno o más procesadores y sus respectivas caches, una memoria principal y un directorio. Las variables compartidas están almacenadas en un único nodo denominado *home*. El directorio es el encargado de mantener el estado de todas las variables compartidas por dos o más procesadores. El *home* es el encargado de que el resto del sistema mantenga una visión coherente de dicha variable y en el caso de surgir alguna operación de escritura en esa posición de memoria, será el encargado de actualizar o invalidar, dependiendo del caso, el resto de copias que existan a lo largo del sistema. La idea es la misma que la empleada en sistemas *snoopy* pero sin recurrir a operaciones de *broadcast*.

La complejidad que requiere esta clase de sistemas es perfectamente abordable desde el punto de vista de su coste. De hecho, como se ha comentado previamente, existen varios sistemas que empleando estas técnicas han logrado un éxito comercial importante.

COMA

La idea clave en una arquitectura COMA (*Cache Only Memory Access*) es emplear la memoria local de cada nodo del multiprocesador como si fuese una cache del resto del sistema, migrando y replicando los datos a manejar en cada nodo en función de las demandas de la aplicación, de la misma forma que ocurre en la cache de un monoprocesador. Bajo este planteamientos para cualquier nodo, la memoria local actúa como cache y el conjunto de la memoria local de los nodos remotos actúa como un equivalente a su memoria principal. La ventaja fundamental de esta aproximación es que el sistema es capaz de reaccionar a los fallos de acceso remoto distribuyendo de forma automática, a lo largo del sistema, los datos que está manejando la aplicación. En este caso, si un procesador tiene que acceder repetidamente a una posición de memoria alojada en un nodo remoto es posible replicar la posición de memoria remota en la memoria local y en accesos consecutivos no será preciso efectuar un acceso remoto para acceder al dato.

Como es fácil de intuir, mantener la coherencia de multitud de copias de las variables a lo largo del sistema es realmente complejo ya que no existe una única copia segura, como en el caso ccNUMA. De hecho las ventajas que incorpora parecen no justificar su coste. Actualmente tan solo existen algunos prototipos en desarrollo como el *StanFord Flash* [79] o *I-ACOMA*[130] entre otros y en el campo de los sistemas comerciales únicamente el *KSR-1* y el *KSR-2* podrían ser incluidos dentro de esta categoría.

Por otro lado, es preciso señalar que esta clasificación no es estricta ya que existen multitud de propuestas en las que se opta por una u otra aproximación en función de las características de la aplicación, como en el caso de la propuesta *Reactive-NUMA* [50], u otras variantes que facilitan la replicación y migración a nivel de páginas como la empleada en el *SGI Origin 2000*.

1.4 Importancia de la Red de Interconexión.

Partiendo del hecho de que un computador paralelo se puede reducir a un conjunto de elementos que cooperan para lograr un fin común, es evidente que ha de existir un medio que facilite el intercambio de información entre ellos de forma eficiente. En cualquiera de las clasificaciones establecidas en puntos previos, resulta claro que el rendimiento ofrecido por la red de interconexión puede afectar de forma considerable al rendimiento global del sistema.

La red puede limitar el rendimiento del sistema desde dos puntos de vista: bien porque el tiempo de acceso sea demasiado elevado o la cantidad de información que puede manejar antes de desbordarse sea demasiado baja. En ambos casos, el intercambio de información entre los elemen-

tos de cálculo se verá afectado y por tanto, independientemente de su capacidad individual, el sistema verá reducido considerablemente su rendimiento global.

Los aspectos de la red de interconexión que más influyen en el rendimiento del sistema son básicamente cuatro: la topología, el mecanismo de conmutación, el control de flujo y el algoritmo de encaminamiento.

Respecto a la topología, sus características han ido variando desde el empleo de estructuras sencillas, de bajo costo y poco escalables, como las redes de medio compartido (ejemplo típico es el bus), hasta grafos más complejos. Y dentro de éstos últimos, han sido dos los grupos que han dominado las máquinas más significativas: las redes directas y las redes indirectas. En las primeras, donde pueden incluirse entre otras, las redes *k-ary n-cube*, los árboles, las redes de Bruijin, etc., cada nodo del grafo con el que puede modelarse la red tiene asociado, al menos, un elemento de proceso y los arcos del grafo representan los enlaces de comunicación entre dichos elementos de proceso. Su característica esencial es que la latencia de los mensajes no es uniforme ya que la distancia a recorrer depende del par origen-destino. Ello permite explotar la localidad de las comunicaciones de gran parte de las aplicaciones paralelas (lo que beneficia enormemente al rendimiento) y posibilitar una buena escalabilidad. Estas características tan atractivas son las principales razones que han motivado el gran número de sistemas que han utilizado las redes directas. Algunos de estos sistemas son *Intel Paragon* [61](Malla 2-D), *MIT J-Machine*[39] (Malla 3-D), *Cray T3D*[113] *y T3E* [114](Toro 3-D), etc...

Por otro lado, son numerosos los sistemas que han empleado redes indirectas, también denominadas redes basadas en conmutadores. Características diferenciadoras son que los elementos de cómputo solo están asociados a algunos de los vértices del grafo y que la latencia de los mensajes es, normalmente, uniforme dado que, independientemente del par origen-destino, la distancia a recorrer por los paquetes es fija. Ejemplos son las redes en Crossbar (*Cray X/Y-MP, Myrinet*[13]), redes MIN (*IBM SP2* [4]) o las redes de Benes y Clos. No obstante, existen redes de interconexión híbridas que incorporan propiedades de dos o más tipos mencionados. Algunos ejemplos son el *Stanford DASH* [85], *Convex Exemplar* [35]o *SGI Origin 2000* [82].

Respecto al mecanismo de conmutación empleado en las redes de interconexión de computadores paralelos cabe señalar que, aunque en sus comienzos se utilizaron técnicas de conmutación de circuitos, casi todas las máquinas emplean conmutación de paquetes. El principal motivo es que la eficiencia de la primera técnica, es decir abrir y reservar un camino físico entre el par origen-destino, solo es grande si la cantidad de información a enviar es elevada y/o la fluctua-

ción en el ritmo de transferencia es muy baja. Es decir casi todo lo contrario de lo que ocurre usualmente entre los nodos de un sistema multiprocesador.

En la conmutación de paquetes por el contrario, no se realiza reserva del medio, sino que la información avanza, en forma de paquetes, siguiendo bien un camino previamente establecido o buscando su propia ruta al nodo destino. Aunque esta técnica implica la incorporación de mecanismos de arbitrio para gestionar los fenómenos de contención, permite elevar considerablemente el nivel de ocupación de los recursos de la red.

Otro factor que influye considerablemente en el rendimiento del sistema es el control de flujo empleado. De nuevo, en los primeros sistemas multiprocesadores y heredada de las redes de área local, se utilizó la técnica de control de flujo *store-and-forward*. Es decir, en el camino hacia su destino cada paquete de información debía ser almacenado íntegramente en los nodos intermedios antes de ser reenviado al siguiente nodo de la ruta. Esto provoca que la distancia topológica de la red sea un factor multiplicativo de la latencia de los mensajes y convierta este control de flujo en prácticamente inviable para las redes de interconexión de sistemas paralelos.

En 1979, *Kermani* y *Kleinrock* introdujeron la técnica de control de flujo *virtual cut-through*, que constituyó una notable mejora respecto al método anterior [77]. Ahora, cada *phit* o unidad mínima de información intercambiable entre routers, puede ser reenviada al siguiente router en el camino al destino sin haber recibido todo el paquete. Esto provoca la necesidad de espacio de almacenamiento dentro de cada encaminador, pero convierte la distancia topológica en un factor aditivo de la latencia (menor sensibilidad a la topología) y mejora por tanto enormemente el rendimiento de las aplicaciones del sistema multiprocesador.

Si la unidad mínima sobre la que se realiza el control de flujo, o *flit*, es más pequeña que el paquete (puede estar constituida por uno o varios *phits*) el control de flujo anterior se convierte en el denominado *wormhole* [40]. Ello permite reducir el espacio de almacenamiento necesario en cada encaminador a un solo *flit* y por lo tanto manejar longitudes de paquete arbitrarias. Estas dos características han provocado que éste sea el mecanismo de control de flujo más empleado. No obstante, en los últimos años parece claro que una longitud fija en el tamaño de paquete disminuye la complejidad del encaminador e incrementa el throughput de la red, máxime teniendo en cuenta que los avances en VLSI ya han superado ampliamente las restricciones del espacio de almacenamiento de principios de los años 80.

El cuarto factor señalado y que tiene una gran influencia sobre el rendimiento de la red es el método utilizado para elegir el camino entre cada par origen-destino. Influencia aún no del todo

aclarada a pesar del elevado número de años bajo discusión. Así, en las máquinas que utilizan encaminamiento determinista, es decir una ruta predeterminada, los encaminadores son simples y por tanto rápidos y de bajo coste. Sin embargo, bajo determinados patrones de tráfico esta técnica puede provocar grandes desbalanceos en los niveles de ocupación de los recursos. Por el contrario, bajo encaminamiento adaptativo, es decir cuando la elección del camino depende, además del par origen-destino, del estado actual de la red, la complejidad asociada a cada encaminador es más elevada y por tanto su coste y su tiempo de paso. El efecto positivo es que amplían el rango de operación de la red, en algunos casos, mucho más allá que el tolerado por los métodos deterministas.

Por último, y como una buena muestra de la complejidad asociada al diseño óptimo de una red de interconexión, conviene mencionar que numerosas combinaciones de las soluciones a cada uno de los factores que han sido mencionados, conducen a la aparición de anomalías que pueden provocar la no viabilidad práctica de dichas soluciones. La más importante, por la dificultad de su resolución, es sin duda el *deadlock* o interbloqueo en el avance de paquetes que se presenta como consecuencia del tipo de topología, control de flujo y encaminamiento elegido y que provoca la inutilización del subsistema de interconexión. Las otras anomalías que pueden presentarse son la inanición o *starvation* y el *livelock*. La primera se elimina cuando se realiza una asignación de recursos equitativa. El *livelock* esta ligado a la utilización de encaminamiento no mínimo, es decir que existe porque la progresión de los paquetes por la red no siempre es a través del camino más corto.

La combinación de todos estos aspectos, en los que para cada uno de ellos existen numerosas soluciones, conduce a un problema de elevada complejidad y hacia cuya disminución y entendimiento va dirigida esta tesis. Dado que no es posible llegar a desarrollar una red ideal, es necesario plantear estructuras para la red de interconexión que minimicen el coste del subsistema de comunicación y obtengan un rendimiento balanceado respecto a las demandas de los nodos de cálculo. Además, el comportamiento de la red ha de ser tal que la escalabilidad del sistema no se vea comprometida. Bajo estas circunstancias, el diseño de la red de interconexión se presenta como un problema de ingeniería, dado que se centra en minimizar la relación coste-rendimiento. Sin embargo, la cantidad de aspectos que confluyen en este campo lo convierten en una materia de gran atractivo. Un estudio adecuado ha de contemplar desde las demandas que imponen los computadores paralelos, hasta las restricciones tecnológicas, pasando por interesantes aspectos matemáticos.

1.4.1 Características Básicas de Algunos de los Encaminadores/redes Actuales más Significativos.

Para entender de qué manera los aspectos citados previamente son considerados a la hora de proponer la red de interconexión de un sistema paralelo, a continuación pasaremos a comentar brevemente las características de este sistema en el caso de un conjunto significativo de computadores paralelos. En primer lugar, en la Tabla 1-1 se resumen algunas propiedades de las redes de interconexión de diversos computadores comerciales y prototipos académicos. A partir de estos datos, se puede intuir el impacto que han tenido en la propia red de interconexión los continuos avances de las tecnologías VLSI. Como se puede observar, la evolución ha conducido de emplear enlaces de tan solo un bit, como en el caso del *nCUBE/2*, hasta enlaces de 20 bits como en el caso del *Origin*. La evolución en los tiempos de ciclo son también claras, pero sin llegar a alcanzar los logrados en el caso de los procesadores. Las limitaciones físicas impuestas por los propios canales de comunicación o la distribución espacial del subsistema hacen que la evolución no llegue a tales niveles. En este sentido, otro factor que influye de manera notable es el menor volumen de producción y utilización de estos sistemas frente a los procesadores.

Sistema (Año[a])	Topología	Tiempo de Ciclo (ns)	Anchura física de los canales (bits)	Tiempo de paso por encaminador (ciclos)	Tamaño de un Flit (bits)
nCube/2 (1992)	Hipercubo	25	1	40	32
TMC CM-5 (1993)	Fat-Tree	25	4	10	4
IBM SP-2 (1994)	Banyan	25	8	5	16
Intel Paragon (1993)	Malla 2-D	11.5	16	2	16
Meiko CS 2 (1993)	Fat-Tree	20	8	7	8
Cray T3D (1993)	Toro 3-D	6.67	16	2	16
Dash (1992)	Malla 2-D	30	16	2	16
J-Machine (1991)	Malla 3-D	31	8	2	8
SGI Origin (1997)	Hipercubo jerar.	2.5	20	16	160
Myrinet (1995)	Irregular	6.25	16	50	16
Cray T3E (1996)	Toro 3-D	13.5	14	4	70

Tabla 1-1. Características de la red de interconexión de varios sistema comerciales y prototipos académicos [37]

a. Las fechas corresponden a las primeras referencias bibliograficas disponibles sobre cada sistema.

A continuación se resumen, para los tres computadores más recientes de los mostrados en la tabla, las características más importantes de su red de interconexión.

1.4.2 Cray T3E.

El *Cray T3E* [114] representó la segunda generación de una familia de sistemas multiprocesador que comenzó con el *Cray T3D* [74]. La red de interconexión en ambos sistemas es un toro 3-D. En este sistema, muchas características del encaminador y la red se mantienen con respecto a la primera generación. Sin embargo, otras son radicalmente distintas. La idea fundamental perseguida en su desarrollo fue intentar tolerar la latencia de la comunicación para un rango de cargas de trabajo más amplio que en la primera generación. Bajo este punto de vista, el objetivo fundamental perseguido fue incrementar, en la medida de lo posible, la productividad de la red.

Para alcanzar este fin se introdujo la utilización de canales de comunicación segmentados con una mayor capacidad real. La implementación del encaminador opera a 75 MHz y durante un solo ciclo de reloj 5 *phits* son enviados a lo largo del canal. Los canales de comunicación poseen un ancho de 14 bits, por lo que la unidad mínima de información que puede manejar el router es de 70 bits. De esta forma cada *flit* puede transportar una palabra de 8 bytes más 6 bits de información adicional. Por otro lado, el control de flujo empleado en la comunicación es *wormhole*. Los canales emplean un protocolo de comunicación basado en créditos.

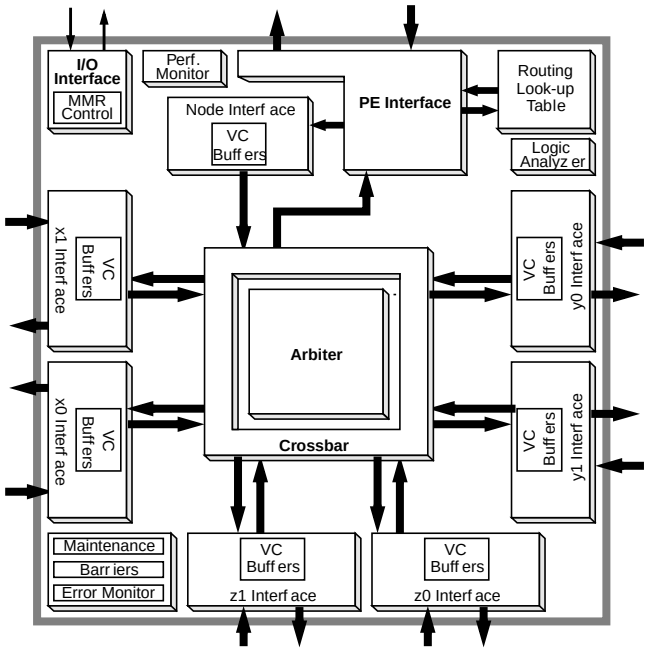

Figura 1-9. Arquitectura del encaminador del T3E [113].

Otra modificación incorporada sobre el encaminador empleado en el *Cray T3D* es el mecanismo de encaminamiento. En este caso se incorpora un canal virtual adicional completamente adaptativo. Este canal junto con cuatro canales virtuales deterministas evitan cualquier aparición de *deadlock* en la red de interconexión aplicando la teoría descrita en [43]. Los canales deterministas emplean un mecanismo de encaminamiento en orden de dirección. Este tipo de encaminamiento y la incorporación de adaptatividad mejoran notablemente la tolerancia a fallos del sistema. Dos de los canales virtuales deterministas son empleados exclusivamente por el tráfico de peticiones y los otros dos por el de respuestas. De esta forma se logra evitar cualquier posibilidad de interbloqueo como consecuencia de la capacidad limitada de las colas de consumo [86].

El tamaño de los paquetes que puede manejar este router es de hasta 10 *flits*. Los espacios de almacenamiento temporal por canal van desde los 22 *flits* de los canales adaptativos hasta los 12 *flits* de los canales deterministas.

En la Figura 1-9 se muestra una representación de los bloques básicos de este encaminador.

1.4.3 *SGI Origin 2000*: SGI Spider.

Se trata de un computador de memoria compartida escalable [82]. La topología empleada en la red de interconexión de este sistema es un hipercubo jerárquico (Ver Figura 1-8). El papel clave de la red lo desempeña su encaminador, denominado SPIDER.

El SGI SPIDER (*Scalable Pipelined Interconnect for Distributed Endpoint Routing*) [55] no ha sido diseñado exclusivamente para su empleo en sistemas multiprocesador, sino como un bloque básico para construir conmutadores no bloqueantes de gran escala, como conmutadores para aplicaciones gráficas distribuidas. Los seis enlaces físicos del encaminador son completamente bidireccionales incorporando 20 bits de datos y dos señales adicionales de sincronización. Los datos son enviados en el flanco ascendente y descendente de su reloj. El reloj de los enlaces opera a 200 MHz. El núcleo del encaminador opera a 100 MHz, manejando en cada ciclo bloques de información de 80 bits que es serializada en el canal de comunicación en cuatro *phits* de 20 bits. La estructura de los bloques básicos de este encaminador es la mostrada en la Figura 1-10.

El mecanismo de encaminamiento empleado en este encaminador es determinista. Incorpora cuatro canales virtuales por canal físico. A diferencia del encaminador del T3E, incorpora un sistema de arbitrio evolucionado que permite mejorar considerablemente los niveles de productividad de la red. Se centra en el empleo de colas de almacenamiento DAMQ (*Dynamically*

Allocated Multi-Queue) para la evitación del bloqueo por parada de la cabeza [125]. Además, emplea un sistema de encaminamiento, denominado *routing look-ahead*, que permite determinar el puerto de salida de los mensajes (basado en tabla) de forma simultánea al arbitraje del conmutador [132]. El control de flujo es *wormhole* y la unidad de información mínima en este encaminador es el *micropaquete* conceptualmente muy similar al *flit*. Cada micropaquete contiene un total de 160 bits de información de los cuales 128 corresponden a *payload*. De la misma forma que el encaminador del *Cray T3E* la gestión de los bufferes de almacenamiento sigue una política basada en créditos. Cada uno de los 24 canales virtuales del encaminador posee un buffer asociado con capacidad para cinco micropaquetes.

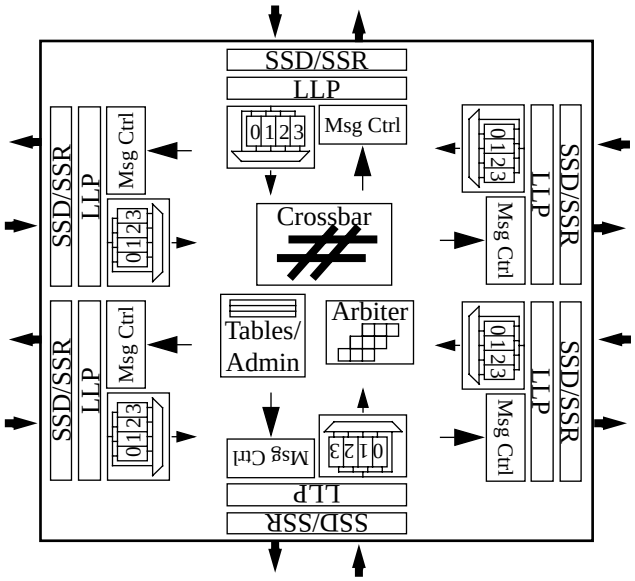

Figura 1-10. Arquitectura del encaminador SGI SPIDER [55].

1.4.4 Myrinet.

Myrinet [13] es una red de interconexión ideada para ser empleada como subsistema de comunicación en NOWs (*Network Of Workstations*). Los computadores paralelos basados en este tipo de aproximación han experimentado un crecimiento considerable. El continúo incremento en la capacidad de cálculo de los computadores personales o estaciones de trabajo ha propiciado su expansión. Sin embargo, a la hora de construir un computador paralelo es preciso contar con una red de interconexión que permita extraer las posibilidades de todos los elementos de cálculo. La potencia de cálculo ofrecida por este tipo de sistemas puede llegar a superar incluso la alcanzada por supercomputadores de coste mucho más elevado [133]. Este es precisamente el papel a jugar por este tipo de red: se trata de una red LAN con unas características peculiares,

especialmente orientadas hacia las construcción de NOWs. Entre ellas podemos citar que el control de flujo es segmentado (*Virtual Cut-through*) y posee otras muchas características empleadas en diversas redes de interconexión de computadores paralelos. La red esta constituida por dos elementos básicos: las interfaces de red y los conmutadores. Ambos elementos pueden ser combinados para dar lugar a topologías irregulares, como se muestra en la Figura 1-11.

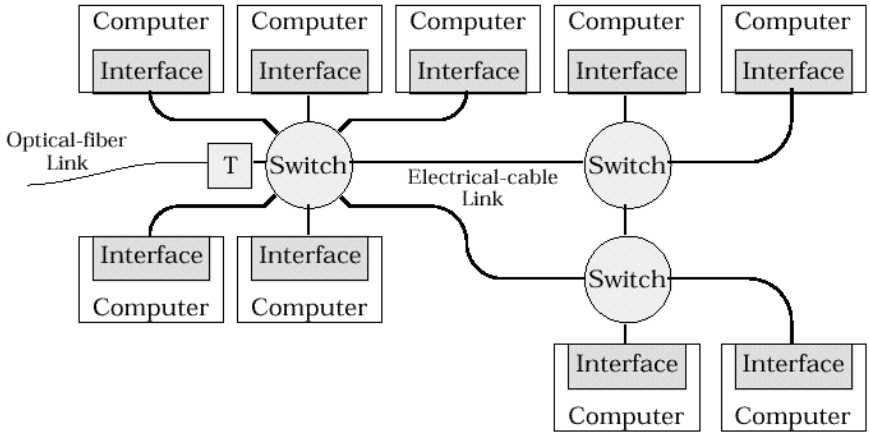

Figura 1-11. Ejemplo de una red de interconexión basada en Myrinet.

La interface de red incorpora un pequeño procesador específico denominado *LANai* encargado de controlar el flujo de información entre el *host* y la red. La estructura de éste, es la mostrada en la Figura 1-12. Está situado en el bus de entrada/salida del *host*. Como se puede apreciar, la interface incorpora una memoria para almacenar los mensaje a procesar y el código a ejecutar por el *LANai*. El software que controla y facilita el acceso a Myrinet esta repartido entre el sistema operativo del *host*, el controlador del dispositivo y el programa de control de la interface. El programa de control de la interface se denomina MCP (*Myrinet Control Program*) y es ejecutado por el procesador *LANai*. El MCP es cargado inicialmente por el controlador de dispositivo. El MCP interactúa concurrentemente con la red y el *host*. Las tareas típicas que lleva a cabo el MCP son cálculo y chequeo del CRC, control del DMA, recepción y envío de paquetes, control de encaminamiento del paquete y control en la transferencia de mensajes entre la memoria del computador y los buffers de envío y recepción. En general, cada librería de paso de mensajes tienen su propio MCP y en función de las características de éste, el rendimiento ofrecido por el sistema puede diferir considerablemente [106].

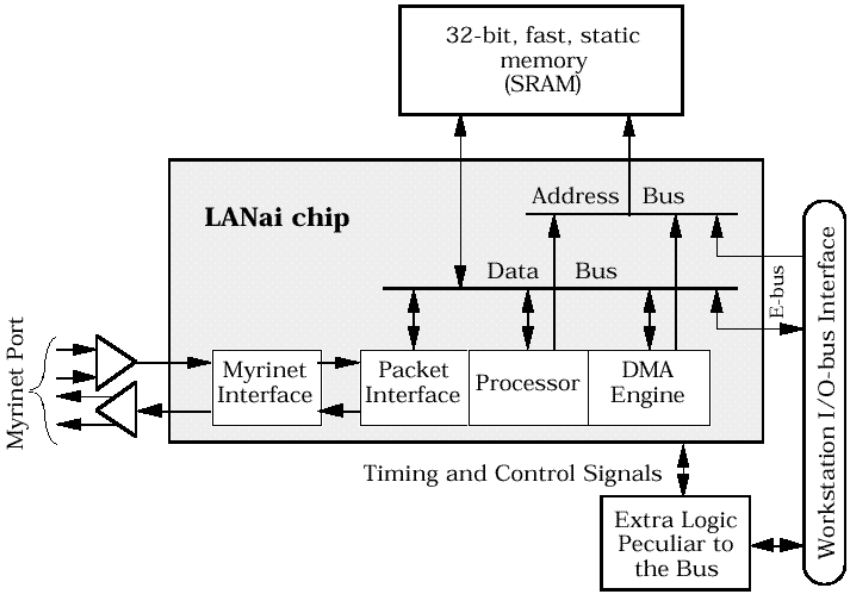

Figura 1-12. Estructura de una interface de red Myrinet. Procesador LANai.

Los conmutadores están implementados mediante crossbars. Pueden llegar a tener hasta 16 puertos y tiempos de paso en torno a los 550 ns. La anchura de los canales de comunicación es de 9 bits, 8 de ellos de datos y una línea dedicada para el envío de comandos. De la misma forma que en el caso de los encaminadores descritos previamente, los canales de comunicación son segmentados. El encaminamiento es “basado en fuente” lo que facilita el encaminamiento en topologías no regulares.

1.5 Motivación y Objetivos.

Como se puede apreciar en el caso de las sistemas comentados previamente, los planteamientos seguidos por los diseñadores en cada caso difieren. Son notables las discrepancias relativas a topología, encaminamiento, control de flujo, etc... Estos casos son solo una muestra de la falta de consenso en este área: no es posible señalar con rotundidad ninguna alternativa en cuanto a las combinaciones adoptadas a la hora de diseñar o proponer la red de interconexión. El carácter abierto y la complejidad del problema no hace sino incrementar su atractivo. Por ello, en este trabajo se pretende analizar en conjunto planteamientos relativos a los aspectos claves de la red de interconexión y lograr entender en qué medida cada uno de ellos es relevante en su rendimiento final.

El cometido que desempeña la red de interconexión es sencillo desde el punto de vista conceptual: es la encargada de desplazar información desde un punto o nodo del sistema a otro. Sin embargo, el espacio de diseño que se abre para cumplir este objetivo de la forma más rápida y ofreciendo la mayor capacidad, es enorme. Desde la topología hasta la propia implementación de los encaminadores pueden afectar de manera notable al rendimiento de la red y por extensión al sistema. Cada uno de los innumerables factores que afectan al rendimiento no pueden ser analizados de modo individualizado, ya que bajo esta aproximación se corre el riesgo de extraer conclusiones sesgadas y en la mayoría de los casos erróneas. Es decir, no es posible mejorar determinadas facetas de la red de interconexión sin considerar de qué manera estas variaciones pueden afectar a otros aspectos. Esta fuerte interrelación, hace difícil determinar hasta qué grado las propuestas realizadas en este campo de la arquitectura de computadores pueden tener una validez genérica.

Frente a este problema, el principal esfuerzo seguido en este trabajo ha sido intentar aproximarse a algunos de los aspectos más críticos de la red de interconexión de un forma unificada. En este análisis hemos partido desde la toma en consideración de las restricciones que imponen las tecnologías de implementación hasta el impacto causado por ellas en los sistemas paralelos cuando ejecutan determinadas aplicaciones. Entre esos dos extremos hemos considerado oportuno analizar diversos factores críticos como los mecanismos de encaminamiento, las características estructurales de los encaminadores, las dependencias de la implementación o las características topológicas de la propia red.

El análisis unificado del problema requiere contar con un marco de trabajo preciso y claramente delimitado. Este marco de trabajo facilitará la tarea de estudiar de modo comparativo el rendimiento alcanzado por determinadas propuestas realizadas en este nivel. Para establecer este marco, en nuestro caso, hemos fijado los siguientes objetivos:

- Establecer un banco de pruebas que permita reproducir, de la forma más fiable posible, las condiciones de trabajo de la red en un sistema real. En este contexto, es fundamental abordar el análisis de rendimiento de las redes de interconexión no solo desde el punto de vista de las cargas de trabajo sintéticas si no también desde las aplicaciones reales. A la hora de establecer como serán esas cargas reales, tendremos que optar por alguno de los paradigmas previamente expuestos. En esta caso hemos optado por emplear una arquitectura DSM con coherencia hardware como la ccNUMA. Las cargas de trabajo sintéticas tan solo modelan las condiciones de trabajo de la red de interconexión de un modo aproximado. Es por ello

necesario estudiar en qué modo pueden afectar determinados elementos de la red de interconexión al rendimiento global del sistema. Para cumplir este objetivo es fundamental disponer de:

  - Una herramienta de evaluación que permita modelar de forma precisa el comportamiento del tipo de sistema paralelo considerado.
  - Un conjunto de aplicaciones o *benchmarks* que sean representativos de la gran variedad de comportamientos que pueden llegar a exhibir las aplicaciones paralelas para el tipo de sistema considerado.

- Analizar y determinar en cada caso las posibles restricciones que imponen las tecnologías VLSI en la red de interconexión. Bajo este punto de vista, es claro que a la hora de proponer posibles mejoras en la red en cualquiera de sus múltiples aspectos, es necesario saber, al menos de forma aproximada, de qué manera afecta al coste de la red. Para alcanzar este objetivo, será preciso:
  - Emplear una metodología de análisis que considere estos aspectos. Esta metodología pasa por el diseño e implementación *hardware* de cada una de las propuestas de la red de interconexión a considerar.
  - Disponer de herramientas de simulación precisas que permitan manejar de modo unificado las características establecidas mediante la metodología expuesta previamente y poder analizar conjuntamente el comportamiento de las aplicaciones reales con estos costes.

Teniendo en cuenta el marco de trabajo establecido, en esta tesis nos moveremos en, al menos, cuatro ejes del espacio de diseño. Intentaremos determinar de qué manera afecta cada uno de estos parámetros y sus interrelaciones en el rendimiento final de la red. Este análisis parte de los siguientes objetivos:

- Analizar cómo los mecanismos de encaminamiento afectan al rendimiento del sistema. En este punto se introducirá un algoritmo de encaminamiento original orientado a mejorar la productividad de la red de interconexión. Con esta propuesta se pretende ampliar la zona de trabajo de la red de interconexión sin perjudicar excesivamente la velocidad del sistema.
- Conocer cómo los detalles de bajo nivel influyen en la productividad. Se estudiará de qué modo afectan los detalles de implementación en el funcionamiento del subsistema de comunicación.
- Saber cómo influyen los aspectos relacionados con la topología de la propia red en las aplicaciones paralelas.

- Entender y proponer nuevas organizaciones a nivel interno de los encaminadores que permitan mejorar su rendimiento. Se estudiará cómo influyen determinados efectos que surgen del tráfico que ha de soportar la red. Propondremos nuevas organizaciones que intentarán mejorar el rendimiento con un coste añadido reducido.

El objetivo final del trabajo, será proponer una arquitectura para la red de interconexión a partir de las conclusiones extraídas de los puntos previos. Para sugerir esta propuesta, será preciso abordar los objetivos expuestos previamente de un modo unificado. Si establecemos de forma correcta el marco de trabajo que describe las restricciones de implementación y las cargas de trabajo reales, podremos alcanzar el objetivo satisfactoriamente. Sin embargo, es importante señalar que el entorno de trabajo no es en sí mismo un fin sino un medio para alcanzar el objetivo final. Es por ello que, a lo largo del trabajo, centraremos más nuestra atención en la propia red de interconexión y los aspectos colaterales del estudio, si bien son importantes, solo les daremos una relevancia en su justa medida, sin profundizar más allá de lo estrictamente necesario.

1.6 Estructura.

La estructura del trabajo presentado en esta memoria es la siguiente.

- En el **Capítulo 2** centraremos el contexto en el que se encuentra esta tesis, describiendo el modo en que se establecen, en el resto del trabajo, los costes *hardware* de cada una de las propuestas arquitectónicas para la red. A continuación describiremos el banco de pruebas empleado para evaluar el rendimiento de cada red de interconexión. En especial, realizaremos un análisis detallado de varias aplicaciones reales que permitirá establecer su validez a la hora de emplearlas como *benchmarks.*
- En el **Capítulo 3** analizaremos el impacto en el rendimiento de las redes de interconexión, de ciertos aspectos arquitectónicos. Propondremos un nuevo algoritmo de encaminamiento que permitirá incrementar la productividad del subsistema de comunicación con un coste contenido. Analizaremos una implementación *hardware* de este nuevo mecanismo frente a otros preexistentes.
- En el **Capítulo 4** estudiaremos la influencia en el rendimiento del sistema de determinadas decisiones arquitectónicas de bajo nivel. Partiendo del encaminador descrito en el capítulo precedente efectuaremos un análisis de varias alternativas de arbitrio en el encaminador. Analizaremos de qué forma estas decisiones pueden afectar al rendimiento de los computadores paralelos.

- En el **Capítulo 5** examinaremos aspectos relacionados con la topología de la red de interconexión. Analizaremos una topología óptima desde cierto punto de vista teórico y que nunca ha sido considerada bajo condiciones de estudio realistas. Posteriormente, compararemos los resultados ofrecidos frente a otras redes comúnmente empleadas en una amplia variedad de sistemas.
- En el **Capítulo 6** observaremos de qué modo afecta al rendimiento del sistema aspectos relacionados con la organización o estructura interna de los encaminadores. Partiendo del encaminador propuesto en capítulos previos se estudiará cómo influye la disposición y ubicación de los espacios de almacenamiento temporal en el encaminador.
- En el **Capítulo 7** expondremos las conclusiones más importantes que se desprenden de este trabajo así como las líneas futuras de investigación.
- Finalmente, en el **Apéndice A** describiremos de manera detallada la infraestructura de simulación empleada a lo largo de todo el trabajo. Se expondrá la estructura de las herramientas de simulación empleadas así como una nueva metodología de análisis basada en cargas de trabajo reales.