Estado del Arte en Redes de Altas Prestaciones

Lorenzo Fernández y Jose M. García

Resumen— **Hoy en día los clusters de estaciones de trabajo están sustituyendo a los computadores MPP gracias a su relación rendimiento/coste y al aumento de prestaciones de las modernas redes de interconexión. Cada vez se les exige a estas redes más ancho de banda y menores latencias, forzando a la tecnología a alcanzar los límites físicos de la transmisión de datos. Pretendemos mostrar una visión del estado actual de las redes de interconexión de altas prestaciones con el objetivo futuro de llegar a determinar el campo de aplicación de cada una de ellas. De esta forma podríamos decantarnos por una determinada tecnología más barata, simplemente porque el resto no ofrece un aumento de rendimiento, en el tipo de problemas que queremos resolver, como para que sea necesario invertir más dinero en una red más avanzada.**

Keywords—**Redes de altas prestaciones, clusters de estaciones de trabajo.**

I. Introducción

LOS clusters de estaciones de trabajo unidos por redes de interconexión de alta velocidad ofrecen una alternativa escalable y de precio aceptable a los supercomputadores monolíticos. Al contrario que los supercomputadores y máquinas paralelas, los clusters de estaciones de trabajo se basan en hardware de comunicación estandarizado y protocolos de comunicación desarrollados para redes de área local, y no para cálculo paralelo. Conforme el hardware de comunicaciones se hace más y más rápido, el rendimiento de las comunicaciones está limitado por la sobrecarga de procesamiento del sistema operativo y la pila de protocolos más que por la red en sí. Por tanto, cuando pasamos de una Ethernet, por ejemplo, a ATM, no alcanzamos un speedup comparable a la mejora en rendimiento que nos ofrece el hardware. Por tanto, debemos identificar los parámetros críticos de una red adecuada para cálculo paralelo, que son:

- *Latencia:* Debe ser baja para transmitir mensajes pequeños eficientemente.
- *Ancho de banda:* Alto para transmitir mensajes grandes.
- *Escalabilidad* para poder aumentar fácilmente el número de nodos.

Aparte de estos parámetros relacionados con el hardware, el comportamiento funcional de la red debería ajustarse a pilas de protocolos mínimas. Deberíamos exigirle:

- Alta fiabilidad en la transmisión para evitar comprobar errores.
- Entrega de paquetes en orden.

Al implementar tantas funcionalidades como podamos en el interfaz de red, obtenemos un protocolo mínimo y eficiente.

Durante los últimos años, varios fabricantes han introducido conmutadores para interconectar un conjunto de procesadores. Estas redes conmutadas permiten al usuario definir su propia topología. El caso en que cada conmutador tiene un único procesador directamente conectado a él y el resto de los puertos se usan para conectarse a otros conmutadores no se diferencia sustancialmente de una red directa.

La flexibilidad que nos ofrecen los fabricantes para conectar nuestros conmutadores nos permite construir topologías irregulares. Por tanto se desarrollan algoritmos genéricos de enrutamiento. En estos algoritmos, la dirección de destino especifica tanto el conmutador como el puerto de ese conmutador donde se encuentra el procesador que debe recibir el mensaje, y se aplican todos los avances que se encuentran en las topologías regulares (hipercubos, mallas, etc.) a este nuevo enfoque.

El objetivo de este artículo es ofrecer una visión unificada y clara de las distintas opciones que nos ofrece el mercado a la hora de inclinarnos por un tipo de red u otra. Las redes que examinaremos serán las siguientes:

- Fast Ethernet/Gigabit Ethernet/10 Gigabit Ethernet
- Autonet
- HIPPI-800/Gigabyte System Network (GNS)
- Asynchronous Transfer Mode (ATM)
- Servernet I y II
- Myrinet
- Memory Channel
- Scalable Coherent Interface (SCI)
- Synfinity

e-mail:{lfmaimo,jmgarcia}@ditec.um.es. Lorenzo Fernández es Profesor Ayudante en el Departamento de Ingeniería y Tecnología de Computadores. Jose M. García es Profesor Titular en el mismo departamento. Este trabajo se ha realizado dentro del proyecto CICYT: "Desarrollo de una red de estaciones de trabajo de altas prestaciones y bajo coste". TIC97-0897-C04-03

II. Puntos a tener en cuenta a la hora de diseñar una red

Antes de profundizar en las redes de alta velocidad, vamos a echar un vistazo a los principales compromisos que hay que tener en cuenta al diseñar hardware de interconexión. Aparte de la relación rendimiento/precio debemos de tener en cuenta los siguientes aspectos.

A. Escalabilidad

La escalabilidad es la capacidad que tiene la red de crecer casi linealmente cuando aumenta el número de nodos. Los multicomputadores tradicionales tienen una topología de red fija (malla, hipercubo, etc.) que impide que sólo se puedan añadir nodos a la red con unas determinadas restriciones. Sin embargo los clusters son más flexibles. Pueden incorporarse nodos individuales conforme se va necesitando más potencia de cálculo y formar topologías irregulares. Por tanto la red debe tolerar el aumento de la carga y entregar aproximadamente el mismo ancho de banda y latencia para clusters pequeños (8-32 nodos) como para grandes (cientos de nodos).

B. Fiabilidad

Existen aplicaciones que no admiten la pérdida o recepción incorrecta de datos, para evitar lo cual se introducen checksums, reconocimentos, etc. Pero estos protocolos aumentan considerablemente la latencia de la red. Cuando trabajamos con clusters necesitamos la menor latencia posible, para lo cual se intenta minimizar esta sobrecarga. En primer lugar, se considera que las conexiones en los clusters están casi libres de errores. El cálculo de los CRCs puede hacerlo el interfaz de red. Los posibles errores pueden notificarse al software a través de interrupciones o registros de estado. Para liberar al software de tener que almacenar los mensajes, el interfaz de red podría también almacenarlos temporalmente para retransmitirlos en caso de error. El objetivo de todo esto: presentar al usuario una red lo más fiable posible sin sobrecarga software adicional.

C. Memoria Compartida frente a Memoria Distribuida

La memoria compartida hace la red de interconexión transparente para los procesos a través de un espacio virtual global. El hardware y software de gestión de la memoria virtual (MMU, tabla de páginas, etc) se utiliza para mapear las direcciones virtuales en direcciones físicas locales o remotas. Por otro lado la memoria distribuida al tener que utilizar paso de mensaje hace la red de interconexión visible a las aplicaciones. Los datos se pueden mandar a otro nodo a través de llamadas send/receive del API. Comparado con el modelo de memoria compartida, el usuario tiene que llamar a las rutinas de comunicación explícitamente para transferir los datos por la red. Aparte de Memory Channel y SCI, que soportan el modelo de memoria compartida distribuida, el resto de los sistemas de interconexión están pensados para paso de mensajes.

III. Detalles del Diseño

La implementación de una red de alta velocidad implica mirar más de cerca aquellos detalles que van a influir en el rendimiento global. Pequeñas modificaciones en el hardware pueden mejorar considerablemente las prestaciones de nuestra red. En general la norma a seguir es: *Mantener el caso frecuente lo más simple y rápido posible.*

A. Capa Física

Elegir el medio físico correcto para un canal es un compromiso entre la velocidad de los datos y el coste del cable. Los medios serie ofrecen ancho de banda moderado, pero pueden utilizar niveles de enlace estándar como Gigabit Ethernet ó Fiber Channel. Un cable ancho de 64 bits como el que se usa en HIPPI, permite velocidades muy altas, pero el número de patillas es una limitación en la construcción de conmutadores. Para transmitir señales a una velocidad relativamente alta (100 MHz y más) y con un consumo de potencia razonable, podemos utilizar dos cables transmitiendo niveles de corriente complementarios como señal. De esta forma un conmutador unidirección de 8×8 con 32 líneas de señal diferenciales produciría 1024 patillas sólo para los enlaces. Podemos encontrar medios fabricados con cable o medios ópticos. Ya que las señales eléctricas no se pueden transmitir a lo largo de cables demasiado largos, la fibra óptica sustituirá en un futuro a los cables basados en cobre.

B. Protocolo de Enlace

El término protocolo de enlace define la forma en que construimos los mensajes que se transmitirán sobre la capa física, y la definición de la interacción en la comunicación entre dos puntos conectados. Típicamente un mensaje de datos está delimitado por palabras especiales de control que pueden usarse para detectar el comienzo y final del mensaje y para indicar eventos en el protocolo de enlace (el receptor no puede recibir más datos, solicitud de retransmisión, etc).

C. Conmutación

La conexión entre puertos de entrada y salida y la transferencia de datos entre ellos dentro de una etapa de la red, se denomina conmutación. Hoy día se utilizan dos técnicas principalmente en las redes, conmutación de paquetes y wormhole. El primero almacena un paquete completo en un conmutador antes de mandarlo a la siguiente etapa. Este mecanismo de store-and-forward necesita una cota superior del tamaño del paquete (MTU, Maximum Transfer Unit) y algo de espacio de almacenamiento para contener uno o varios paquetes temporalmente. Con wormhole, los datos son inmediatamente enviados a la siguiente etapa tan pronto como se decodifica la dirección que se encuentra en la cabecera. La baja latencia y la necesidad de muy poca cantidad de almacenamiento son las ventajas de esta técnica. La longitud del mensaje puede ser variable, pero el diseñador debe tener en mente que un mensaje largo puede bloquear varias etapas a la vez en su camino al destino. También la corrección de errores es más difícil. Usando store-and-forward se podría comprobar primero un paquete entero antes de mandarlo a la siguiente etapa. Usando wormhole, los datos corruptos podrían ser mandados antes de que se reconociera que lo son.

D. Enrutamiento

La dirección situada en la cabecera de un mensaje lleva la información necesaria para que el hardware de enrutamiento del conmutador determine el canal correcto de salida que llevará los datos más cerca de su destino. Existen algoritmos de ruteo deterministas y adaptativos, dado que sólo se encuentran deterministas en las redes comerciales, nos centraremos en ellos. Habitualmente se utilizan dos mecanismos de enrutamiento: encaminamiento fuente y enrutamiento basado en tabla.

En el encaminamiento fuente, la cabecera contiene toda la información acerca de qué camino tomar en cada uno de los conmutadores. El conmutador se limita a comprobar la primera dirección, eliminarla de la cabecera y enviar el paquete.

En el basado en tabla, cada conmutador contiene una tabla donde para cada posible dirección de destino se indica el puerto de salida. La tabla de enrutamiento es proporcional al número de nodos.

E. Control de Flujo

El control de flujo se utiliza para evitar exceder la capacidad del buffer de los conmutadores, que puede producir pérdida de datos. Antes de que el emisor pueda comenzar la retransmisión, el receptor debe indicar que va a aceptar los datos. Se puede adoptar un esquema basado en créditos, donde cada emisor obtiene un número de créditos del receptor. En cada transmisión de un paquete, el emisor consume un crédito y se detiene cuando todos los créditos se hayan acabado. También podemos elegir simplemente indicar al emisor si el receptor puede o no aceptar datos. En ambos casos, el flujo de información de control viaja en dirección contraria a los datos.

TABLE I

Características del Enlace Físico

Red	Velocidad	Conmutación	Ruteo
Fast Ethernet	100 Mbit/s	store/forward	Tabla
Gig. Ethernet	1 Gbit/s	store/forward	Tabla
Myrinet	1.28 Gbit/s	wormhole	Fuente
Servernet II	125 Mbyte/s	wormhole	Tabla
Mem. Chann.	100 Mbyte/s	store/forward	Tabla
Synfinity	1.6 Gbyte/s	wormhole	Fuente
SCI	400 Mbyte/s	store/forward	Tabla
ATM(OC-12)	155(622)Mbit/s	store/forward	Tabla
HiPPI	800 Mbit/s	store/forward	Tabla
GSN	6400 Mbit/s	store/forward	Tabla

F. Transferencia de los datos

La transferencia eficiente de los datos del mensaje entre la memoria principal de nodo y el interfaz de red es un factor crítico a la hora de lograr que las aplicaciones se aproximen al ancho de banda real. Para lograr esto, el software de los interfaces de red modernos hace intervenir al SO sólo cuando las aplicaciones abren o cierran el dispositivo de red. Las transferencias de datos reales se realizan completamente en modo usuario por bibliotecas de rutinas para evitar el coste de las llamadas al SO. La meta es un mecanismo sin copia, donde los datos se transfieren directamente entre el espacio de usuario en memoria principal y la red.

G. Operaciones colectivas

En el cálculo paralelo a menudo necesitamos realizar comunicaciones colectivas, como sincronización por barrera, o multicasts, especialmente en redes para memoria virtual compartida, donde los datos actualizados deben distribuirse al resto de los nodos. Los clusters de hoy día suelen dejar esta tarea al software. Sólo unas pocas redes tienen soporte hardware para operaciones colectivas. Las redes de medio compartido, como Fast Ethernet pueden fácilmente realizar broadcasts de datos mientras que en las basadas en conexiones punto a punto, como Myrinet o ServerNet la operación es más complicada.

IV. Familia Ethernet: Fast Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet

Ethernet es un término genérico correspondiente a una familia de LANs que comparten una serie de características comunes. La principal es utilizar un medio compartido y trabajar con tramas compatibles. Ethernet ha sido con diferencia la LAN más extendida en todos los ámbitos, pero sus bajas prestaciones han obligado a una evolución para adaptarse a las necesidades actuales. La familia Ethernet emplea un protocolo CSMA/CD en su nivel físico y pueden emplearse varios dispositivos de interconexión para configurar la red: repetidores, concentradores, puentes y routers.

A. Fast Ethernet

Fast Ethernet corresponde la primera evolución de Ethernet para convertirse en una red de alta velocidad (100 Mbit/s) sobre UTP o cable de fibra óptica. Utiliza el protocolo CSMA/CD. Se está utilizando hoy día para actualizar la mayoría de redes Ethernet que funcionan a 10 Mbit/s sin que el cambio sea traumático.

Para que el protocolo CSMA/CD siga funcionando, el retardo en el peor caso de la señal transmitida por el medio entre dos puntos no debería ser tan largo como para que un dispositivo terminara de transmitir antes de que pueda detectar una colisión. Esto limita la longitud que puede alcanzar la red. Permite funcionamiento full-dúplex usando un concentrador.

B. Gigabit Ethernet

Gigabit Ethernet se aprovecha del éxito de Ethernet y Fast Ethernet para ofrecer una LAN con sus mismas características, y por tanto compatible con las anteriores a una velocidad mucho mayor de 1 Gbit/s. Sigue utilizando el mismo formato de trama, permitiendo incluso la misma longitud en el campo de datos: de 64 a 1514 bytes. También permite funcionamiento full-dúplex. El método Gigabit Ethernet CSMA/CD se ha mejorado para mantener un diámetro de 200 metros a velocidades de gigabit. Se ha aumentado el tiempo de portadora del CSMA/CD y el tiempo de ranura de su valor actual de 64 bytes a un nuevo valor de 512 bytes. Aunque el valor mínimo de datos, de 64 bytes, no se ha cambiado. Los paquetes menores de 512 bytes tienen una extensión de portadora extra. Se ha añadido la característica de Ráfagas de Paquetes, que permite a los conmutadores y otros dispositivos enviar ráfagas de pequeños paquetes para utilizar completamente el ancho de banda disponible. Los dispositivos que operan en modo full-dúplex (conmutadores y distribuidores con buffer) no están sujetos a la extensión de la portadora, del tiempo de ranura o ráfagas de paquetes. Estos dispositivos continúan usando el Gap Inter-trama Ethernet de 96 bit (IFG) y un tamaño mínimo de paquete de 64 bytes. Su funcionamiento es fundamentalmente conmutado.

C. 10 Gigabit Ethernet

Actualmente el denominado *IEEE P802.3 Higher Speed Study Group* está estudiando un nuevo estándar que amplíe 802.3 y que admita una velocidad de 10 Gigabit/s manteniendo las características de la familia Ethernet.

Los objetivos de este estándar son:

- Emplear 10 Gigabit Ethernet en una primera fase como red backbone, y en posteriores fases como red metropolitana.
- Diseñarla desde el comienzo con ese objetivo en mente.
- Detección de fallos en aproximadamente 10 ms.
- Solamente soporta operación full-dúplex.
- Alta eficiencia en la codificación para aprovechar el medio.
- Deja de utilizar CSMA/CD, ya que hace a la capa MAC dependiente de la velocidad.

V. Autonet

Autonet es una red de área local construida en base a conmutadores interconectados por enlaces punto a punto, full-dúplex de 100 Mb/s. Cualquier puerto de un conmutador puede conectarse a cualquier otro o a una tarjeta de red en un ordenador.

Autonet no está diseñada para proporcionar servicios de voz o vídeo integrados y no ofrece ancho de banda garantizado para soportar tráfico en tiempo real. Está pensada para una carga consistente en tráfico de datos a ráfagas, que es tolerante a reordenaciones, pero necesita una tasa de pérdida de paquetes muy baja. En cada conmutador, un procesador monitorea la configuración física de la red. Un algoritmo distribuido, ejecutándose en el conmutador recalcula automáticamente las tablas para incorporar enlaces reparados o nuevos enlaces y conmutadores. Los controladores de red en los hosts tienen puertos alternativos, uno activo, y otro que sirve para seguir trabajando si el puerto activo deja de funcionar. La unidad de conmutación es el paquete; los paquetes son de longitud variable y pueden tener varios kilobytes de longitud. Los conmutadores envían los paquetes usando wormhole, técnica que minimiza la latencia de conmutación. Autonet tolera retardos de transmisión suficientes como para permitir enlaces de fibra óptica de 2 km de longitud.

Un conmutador Autonet tiene 12 puertos full-dúplex que están interconectados internamente por unas barras cruzadas de 13×13. Se eligió unas barras cruzadas porque su estructura es sencilla, su latencia es baja, y su rendimiento es fácil de comprender. El decimotercer puerto se utiliza para que el procesador del conmutador pueda enviar y recibir mensajes. El pequeño número de puertos es un resultado directo de querer tener en funcionamiento el sistema de rápidamente. Todo el hardware de Autonet está construido con componentes comerciales, y 12 puertos era todo lo que podíamos encajar en un conmutador razonablemente grande sin utilizar circuitos integrados de propósito específico. El diseño del conmutador de Autonet podría fácilmente escalarse a 32 ó 64 puertos por conmutador.

Autonet usa un buffer FIFO en cada puerto del conmutador con un esquema de control de flujo start/stop. Funcionando normalmente no se descartan paquetes en el conmutador receptor. Una cola de 1024 bytes es suficiente para absorber la latencia de "roundtrip" de 2 km de enlace de fibra óptica, aunque realmente usamos una cola de 4096 bytes. El control de flujo es independiente de los límites del paquete así que un único paquete puede estar en varios conmutadores a la vez.

A continuación se muestran el resto de las características más relevantes de Autonet.

- *Puertos alternativos.* Permite que un host esté conectado a dos conmutadores diferentes. El host usa uno de los puertos y cambia al otro si deduce que el inicial no está funcionando bien.
- *Algoritmo distribuido para cálculo del árbol de expansión.*
- *Ruteo multicamino libre de bloqueo.* Permite varias rutas entre un fuente y destino concretos.
- *Enrutamiento Up/Down.* Consiste en etiquetar los enlaces del árbol de expansión como Up o Down según si se acercan a la raíz o se alejan. Definimos una ruta legal como una que nunca usa un enlace en la dirección Up después de haber usado uno en la dirección Down.
- *Reconfiguración automática.* Cuando el software que se ejecuta en los conmutadores detecta un cambio en la red se dispara la reconfiguración automática, recalculándose las tablas de enrutamiento en los conmutadores.

VI. High Performance Parallel Interface (HiPPI-800) y Gigabyte System Network (GSN)

HiPPI es un estándar de comunicación de datos basado en cable capaz de transferir datos a 800 Mbit/s sobre 32 líneas paralelas o 1.6 Gbit/s sobre 64 líneas paralelas. Se diseñó para proporcionar comunicación de alta velocidad entre computadores de alto rendimiento, y por tanto para intentar adaptarse a sus requerimientos de E/S.

HiPPI es un enlace punto a punto formado por un par trenzado de cobre para distancias de hasta 25 metros. Los datos se transfieren por líneas de 32 bits en ráfagas de 1 a 256 palabras. Una o más ráfagas constituyen un paquete, mientras que uno o más paquetes constituyen una conexión.

Hay varios estándares HiPPI. El estándar HiPPI-PH define la capa física, HiPPI-FP describe el formato y contenido (incluyendo cabecera) de cada paquete, y HiPPI-SC permite construir un mecanismo de conmutación que permite multiples conexiones punto a punto a la vez. Tiene dos desventajas principales, la distancia máxima entre nodos (25 metros) y el número de conexiones que están muy limitadas.

A. Gigabyte System Network (GSN)

En febrero de 1996, un grupo de trabajo ANSI comenzó a especificar una versión más rápida y capaz de HIPPI, añadiendo características tales como corrección de errores, control hardware de flujo y previsión de extremadamente bajas latencias (del orden de un microsegundo) que el estándar HIPPI-800 no tenía.

En noviembre de 1997, SGI testeó el estándar HIPPI-6400 en un circuito, el primer ejemplo del circuito integrado SuperHIPPI Media Access Controler (SUMAC) de aplicación específica. Se le denominó Gigabit System Network (GSN).

En el documento ISO/IED 11518 se especifica el nivel físico de un interfaz con enlace punto a punto y full-dúplex para transmisión de datos de usuario fiable y con control de flujo a 6400 Mbit/s, en cada sentido, a distancias de hasta 1 km. También se especifica el interfaz con cable de cobre para distancias de hasta 40 m. Se proporcionan también conexiones a un interfaz óptico para mayores distancias. Sus micropaquetes de pequeño tamaño fijo proporcionan una estructura eficiente y de baja latencia para pequeñas transferencias, y un componente para grandes transferencias.

- Ancho de banda de transferencia de 6400 Mbit/s (800 MBytes/s).
- Enlace full-dúplex capaz de realizar transferencias independientes en ambos sentidos simultáneamente.
- Cuatro circuitos virtuales que proporcionan la posibilidad de una multiplexación limitada.
- Una unidad de transferencia fija, un micropaquete de 32 bytes, para mejorar la eficiencia del hardware.
- Una unidad de transferencia pequeña, resultando en una baja latencia para mensajes cortos, y utilizándose como componente para formar mensajes largos.
- Control de flujo basado en créditos que evita el overflow del buffer.
- Comprobación de errores extremo a extremo

así como enlace a enlace.

- Retransmisión automática para corregir datos defectuosos, proporcionando una entrega de datos garantizada, en orden y fiable.
- Un interfaz eléctrico paralelo acoplado en alterna para manejar cable de cobre sobre distancias limitadas.
- Un interfaz eléctrico paralelo para manejar un interfaz óptico local con vistas a alcanzar mayores distancias.

VII. Asynchronous Transfer Mode (ATM)

ATM ha sido adoptado por la *International Telecommunication Union* (ITU) como la tecnología de conmutación de paquetes indicada para integrar todos los servicios de telecomunicaciones y redes especializadas en una infraestructura mundial común, conocida como *Broadband Integrated Service Digital Network* (B-ISDN). Los estándares ATM y B-ISDN del ITU definen una arquitectura que puede crecer con la tecnología y está diseñada para ser adaptable y ajustarse eficientemente a los requerimientos de diferentes aplicaciones.

ATM es una tecnología de conmutación de paquetes orientada a conexión que emplea celdas de longitud fija formadas por una cabecera de 5 bytes y un campo de datos de 48 bytes. El protocolo ATM define tres capas:

- *Capa física.* Especifica varias opciones para el medio físico, velocidades de transmisión y esquemas de codificación. Los estándares para la capa física preferidos son los denominados *Synchronous Digital Hierarchy* (SDH) y *Synchronous Optical Network* (SONET) que usan un rango de velocidades de 155.52, 622.08 ó 2488.32 Mbit/s. El medio físico incluye UTP-5 (pares trenzados sin apantallar de categoría 5 para velocidades de hasta 155 Mbit/s) así como fibra óptica mono y multi-modo.
- *Capa ATM.* Define el formato de celda y cómo se envían dichas celdas entre enlaces usando un camino virtual de 8 bits (VPI) y un identificador de canal virtual de 16 bits (VCI) dentro de la cabecera de 5 bytes de la celda. Los 48 bytes del campo de datos no están protegidos por códigos de detección de errores en esta capa, pero la cabecera está protegida por un código de detección y corrección de 8 bits. Un identificador de tipo de datos de 3 bits (PTI) distingue entre celdas que llevan datos de usuario y las que llevan distintos tipos de información de control y manejo usada por la propia red. Un bit de pérdida de prioridad de celda (CLP) define la prioridad que debe usarse cuando se descartan celdas en una red congestionada.
- *Capa de Adaptación ATM (AAL).* Adapta la capa ATM a las necesidades de las clases específicas de servicio. Los estándares ITU definen cuatro clases de servicio (servicios de tasa de bits constante y variable, con o sin requerimientos de tiempo real) para diferentes tipos de aplicaciones. Se define un protocolo de capa de adaptación para cada clase de servicio (de AAL1 hasta AAL5). El protocolo AAL5, propuesto por el ATM Forum para aplicaciones de movimiento de datos, ha sido adoptado ahora por el ITU y por el resto de compañías, y representa la especificación más apropiada para aplicaciones LAN. Un paquete puede ser de hasta 64 Kbytes y es dividido (y reensamblado más tarde) en un flujo de celdas de la capa ATM. La cola del paquete lleva un código CRC de detección de errores, pero la recuperación ante un error deben realizarla los protocolos de nivel superior. La capa física, funciones de capa ATM y la mayor parte de la capa AAL5 están implementadas normalmente en hardware.

Se utilizan conmutadores para construir sistemas escalables. Los conmutadores ATM de gran tamaño se construyen como redes de conmutación multietapa, típicamente formadas a partir de chips de 16×16 ó 32×32 puertos funcionando a 155 Mbit/s y 622 Mbit/s. Las redes de conmutación ATM permiten varias conexiones virtuales en cada enlace interno.

Hay dos formas de actuar que pueden adoptarse cuando aparece la congestión. La primera es descartar celdas, la segunda es utilizar un protocolo de control de flujo por hardware en el enlace interno para evitar la llegada de más celdas hasta que haya disponible suficiente espacio de almacenamiento. Los buffers internos de los elementos de conmutación se dimensionan de forma que la probabilidad de congestión bajo ese tráfico aleatorizado es aceptablemente baja (la probabilidad típica de pérdida de una celda es de 10^{-10} o menos).

Los patrones de tráfico naturales generados por aplicaciones reales pueden concentrar las celdas ATM, produciendo una acusada congestión. Por tanto hay que tomar precauciones adicionales cuando se utilizan conmutadores pensados para telecomunicaciones. ATM sufre serias ineficiencias y altas latencias cuando un conmutador intermedio descarta un paquete que es parte de un bloque de datos. Un conmutador con control de flujo evita la pérdida de celdas y el coste del crecimiento no lineal de la latencia.

Debido a la sobrecarga de los controladores software, no hay disponibles interfaces de baja latencia.

VIII. ServerNet

En 1995, Tandem introdujo la primera implementación disponible comercialmente de una SAN denominada ServerNet. En 1998, Tandem anunció la aparición de ServerNet II, la continuación de la primera versión, que aumenta el ancho de banda y añade nuevas características mientras conserva una completa compatibilidad con ServerNet. Tandem trata uno de los principales problemas de los servidores: ancho de banda de E/S limitado.

ServerNet es una red conmutada full-dúplex y con conmutación por wormhole. La primera implementación usa conexiones paralelas de 9 bits con señalización LVDS/ECL a 50 MHz. ServerNet II aumenta el ancho de banda a 125 Mbyte/s manejando serializadores/deserializadores 8b/10b para conectarse a cables estándar 1000BaseX (de Gigabit Ethernet). Con el soporte de cables serie de cobre, ServerNet puede extenderse a través de grandes distancias. Por razones de compatibilidad, los componentes de ServerNet II también implementan el interface de la primera versión. Juntos siempre y cuando se utilice lógica adicional de conversión, los componentes de ServerNet y ServerNet II pueden mezclarse en un sistema, permitiendo al cliente actualizar fácilmente un cluster ya existente con componentes de la nueva generación sin la necesidad de reemplazar componentes antiguos.

Los enlaces operan asíncronamente y utiliza control de flujo para asegurar que los datos no tiene que ser descartados por falta de espacio en el buffer.

A. Transferencia de datos

El mecanismo de transferencia de datos básico soportado es la lectura/escritura en memoria remota basada en DMA. Un nodo final puede ser programado para que lea o escriba un paquete de datos de hasta 64 bytes (512 bytes en ServerNet II) desde/a una posición remota de memoria. La dirección de un paquete se compone de un ID de 20 bits y un campo de dirección de 32/64 bits.

Una característica principal de ServerNet es su entrega ordenada y libre de fallos de los datos a varios niveles. Cada extremo asegura la corrección de la transmisión mandando de vuelta reconocimientos al emisor. En caso de errores, el hardware invoca a las rutinas del controlador para manejo de errores.

B. Conmutadores

ServerNet ofrece conmutadores de 6 puertos, que pueden conectarse en cualquier topología. Router II, la siguiente generación de conmutadores ServerNet, aumenta el número de puertos a 12. Los puertos tanto de entrada como de salida contienen FIFOs para almacenar una cierta cantidad de datos y se conectan a través de unas barras cruzadas de 13×13. El puerto adicional se usa para inyectar o extraer paquetes de control. Una característica especial de los conmutadores ServerNet es la posibilidad de formar las denominadas Fat Pipes (Tuberías Anchas). Varios enlaces físicos pueden usarse para formar un enlace lógicos, conectándolos a los mismos extremos.

IX. Myrinet

Myrinet es un tipo de red de área local basada en la tecnología usada para comunicación de paquetes y conmutación en procesadores masivamente paralelos. Surge a partir de dos proyectos de investigación de la US Advanced Research Projects Agency: Mosaic y Atomic LAN.

Un enlace Myrinet está compuesto por un par full-dúplex de canales a 1.28 Gbps, obteniendo 2.56 Gbps. Utiliza el hecho de que los errores en los bits son extremadamente raros para no sobrecargar con software las transmisiones. Puesto que la comunicación es fiable (una tasa de error bastante inferior a 10^{-15} en cables de hasta 25 metros de longitud), en el enrutamiento se utiliza wormhole con control de flujo del tipo Go/Stop para evitar saturar al receptor.

El cable estándar de Myrinet tiene 18 pares trenzados, nueve en cada dirección y funciona en modo diferencial. La transmisión es síncrona a una velocidad de 80 millones de caracteres de 9 bits/s. El carácter de 9 bits puede ser o bien un byte de datos de 8 bits o un símbolo de control de entre los cinco que tiene. Se decide usar 25 metros como longitud máxima para cable eléctrico. Para mayores distancias se usará fibra óptica.

A. Conmutadores

Myricom está actualmente vendiendo conmutadores de 4, 8 y 16 puertos, y desarrollando los de 32 puertos. Estos conmutadores emplean el ya mencionado wormhole como método de envío. La latencia del peor caso al formar el camino a través de un conmutador de 8 puertos es 550 ns. El corazón del conmutador es una barra cruzada segmentada que no introduce ningún conflicto interno entre los flujos de los paquetes.

El principal criterio en el diseño de los conmutadores es su pequeño tamaño, baja potencia y ser dispositivos auto-inicializables que pueden residir en cualquier espacio conveniente para cablear la red. El conmutador de 4 puertos requiere $\sim$ 15W de $+12V\pm3V$ de alimentación de una o dos fuentes de alimentación. Estos conmutadores funcionan correctamente a temperaturas de hasta 55žC. Debido a que el conmutador no contiene software, no hay posibilidad de que los usuarios vuelquen programas o tablas de enrutamiento en el conmutador. El conmutador simplemente dirige paquetes.

B. Software

Dos tipos de software proporcionan acceso y control de una Myrinet: El MCP ejecutándose en los procesadores de los interfaces de los hosts y los controladores de dispositivo y sistemas operativos ejecutándose en los hosts.

El controlador del dispositivo inicialmente carga el MCP cuando el host arranca. El MCP comienza a ejecutarse tan pronto como el dispositivo libera una señal de reset y después interactúa concurrentemente con su host y la red.

Por la parte del host, un conjunto de colas del tipo productor-consumidor asignadas a comandos y reconocimientos controlan el interfaz. Ejecutar los comandos del host incluye controlar el mecanismo de DMA, una traducción de dirección de red a una ruta, incluir la ruta y el tipo de paquete en el paquete de salida y controlar el interfaz de paquetes. Desde el punto de vista del receptor, el MCP comprueba la validez del paquete entrante, interpreta las cabeceras y transfiere los datos del paquete a los buffers especificados en la memoria del host.

Entrelazado con estas interacciones con el host, el MCP realiza continuamente mapeo, monitorización y enrutamiento seleccionando lo que provoca que la red se autoreconfigure y autorestituya.

X. Memory Channel

Memory Channel de Digital es un enfoque completamente diferente a las SANs. Proporciona una porción de memoria virtual compartida global a base de mapear porciones de memoria física remota como memoria virtual local (también llamada memoria reflejada). Las bases de Memory Channel se obtuvieron a partir de Encore y muchos otros proyectos como pueden ser VMMC o SHRIMP. La primera versión se puso en venta en abril de 1996, y casi un año después se introdujo Memory Channel 2 que mejora el rendimiento global y la escalabilidad de la arquitectura Memory Channel.

Memory Channel se compone de dos componentes: un adaptador PCI y un concentrador. La primera versión de Memory Channel implementa la red como un medio compartido, en el que sólo una transmsión puede estar activa en un momento dado. Esto se vio que era un cuello de botella y la segunda implementación se convirtió en unas barras cruzadas de 8×8 conmutadas, full-dúplex y punto a punto. Los adaptadores también pueden conectarse directamente entre sí sin usar un concentrador.

Al pasar de la versión inicial a la segunda cambiaron muchas cosas:

A. Transferencia de los datos

Para permitir las comunicaciones sobre la red de Memory Channel, las aplicaciones mapean páginas para sólo lectura o sólo escritura en su espacio de direcciones virtuales. Cada interfaz del host contiene dos tablas de control de páginas (PCT), una para los mapeos de escritura y otra para los de lectura. Para las páginas de sólo lectura, se fija una página en memoria física local. Se pueden especificar varios atributos de página: si permite recepción, si se debe provocar una interrupción tras la recepción, si permite lectura remota, etc. Si una página está mapeada como de sólo lectura, se crea una entrada en la tabla de páginas para una página apropiada en el espacio de direcciones de 128Mbytes del interfaz PCI. Los atributos de página se pueden usar para:

- Almacenar una copia local de cada paquete.
- Solicitar mensajes de reconocimiento del receptor para cada paquete.
- Definir los paquetes como de tipo broadcast o punto a punto.

Los broadcasts se envían a cada nodo de la red. Si un paquete de broadcast entra en un concentrador de barras cruzadas, la lógica de arbitraje espera hasta que todos los puertos de salida estén disponibles. Los nodos, que han mapeado la página direccionada como un área con permiso de lectura, almacenan los datos en su región de memoria local fijada. El resto de los nodos simplemente ignoran el dato. De esta forma, una vez que las regiones de datos están mapeadas y configuradas, sencillas instrucciones de almacenamiento transfieren los datos a los nodos remotos sin intervención del sistema operativo.

Además de este mecanismo básico de transferencia de datos, Memory Channel soporta una sencilla primitiva de lectura remota, una sincronización de barrera basada en hardware y una primitiva de bloqueo rápido. Para asegurar un comportamiento correcto, Memory Channel implementa una entrega de los datos escritos en un estricto orden. También

TABLE II
Comparación de Memory Channel I y II

Características	MC I	MC II
Anchura en bits	37 s-dúplex	16 f-dúplex
Reloj	33 MHz	66 MHz
Long. máxima	4 m.	10 m.
Vel. máxima	133 Mbyte/s	133 Mbyte/s
A.B. punto a punto	66 Mbyte/s	100 Mbyte/s
Máximo paquete	32 bytes	256 bytes
Lectura remota	no	sí
Tamaño de página	8 Kbytes	4/8 Kbytes
Arquitec. del hub	bus	cross-bar

una escritura invalida las entradas de la caché del lado del lector, proporcionando por tanto coherencia de caché en el cluster.

XI. Scalable Coherent Interface (SCI)

SCI, también conocido como Local Area Multiprocessor (LAMP), comenzó en 1988 en el proyecto de estándar Futurebus del IEEE, donde observaron que un diseño de bus no podría soportar las necesidades de las siguientes generaciones de multiprocesadores. Para poder aumentar el rendimiento hasta niveles mayores, SCI abandonó las partes específicas de los buses que no podían ser escaladas (tales como transacciones de lectura no divididas, que mantienen el bus desde el comienzo hasta el final y coherencia de caché por snooping del bus), pero mantuvo la arquitectura general tanto como fue posible para que fuera fácil de usar como red de interconexión. Se diseñó para trabajar con sistemas heterogéneos que pudieran crecer incrementalmente. SCI se convirtió en un estándar aprobado por ANSI/IEEE en 1992 y actualmente está siendo aplicado a computadores comerciales (p.e. Hewlett-Packard/Convex Exemplar).

SCI opcionalmente proporciona coherencia de caché controlada completamente por hardware basada en directorio distribuido para una memoria compartida global que sigue el modelo NUMA (con lista doblemente enlazada) con extensiones para reducir las latencias de actualización de las cachés para sistemas grandes.

SCI reduce la latencia de la comunicación interprocesador en un factor considerable. Una comunicación remota en SCI se produce como parte de la ejecución de una sencilla instrucción de carga o almacenamiento en el procesador que se traduce en un fallo de caché provocando que el controlador de caché direccione la memoria remota vía SCI.

La forma habitual de enviar datos obliga a construir un paquete e ir moviéndolo de un lugar a otro por la memoria a base de llamadas al sistema operativo, normalmente provocando latencias entre diez y mil veces mayores que SCI. Podemos utilizar protocolos como *Active Messages* para reducir estas sobrecargas, pero aún son mucho más lentos que SCI. Para portar los programas fácilmente, los antiguos protocolos pueden ser implementados sobre SCI. Comparado con FC o HiPPI, SCI es menos eficiente para grandes bloques, especialmente para grandes distancias.

GSN garantiza la entrega fiable de los datos. La estructura fundamental es una conexión en anillo que comparte el ancho de banda entre varios dispositivos. Con el uso de conmutadores, los anillos pueden conectarse juntos para formar otras topologías como pueden ser mallas. Tanto un dispositivo como un anillo entero pueden conectarse a un puerto de un conmutador SCI, dando al usuario la facilidad de equilibrar coste frente a rendimiento.

El estándar definía originalmente enlaces serie a 1.25 Gbit/s que pueden usarse con fibra óptica para largas distancias y otros enlaces de 18 bits de ancho a 1 GByte/s con señales en ECL diferencial para distancias cortas. Además se han especificado otras tecnologías, como LVDS (Low Voltage Differential Signals), así como otras anchuras funcionando a velocidades que van desde 250 MByte/s hasta 8 GByte/s.

Un nodo SCI puede recibir, transmitir y reenviar paquetes concurrentemente con una tasa de bits erróneos, que aunque depende de la capa física, es extremadamente baja (típicamente menos de 10^{-14}).

XII. Synfinity

La SAN Synfinity está desarrollada y comercializada por Fujitsu System Technologies, una unidad de HAL Computer Systems y Fujitsu. Synfinity ofrece soporte para los dos modelos de programación paralela: paso de mensajes y memoria compartida. Hay tres componentes disponibles:

- Synfinity NUMA es un adaptador para conectar nodos SMP juntos en un cluster ccNUMA.
- Synfinity CLUSTER es un adaptador para host pensado para paso de mensajes.
- Synfinity NET es un conmutador de seis puertos para conectar varios interfaces juntos en un cluster.

Synfinity es una aproximación agresiva hacia el mayor ancho de banda físico posible. Se usan dos tipos de cables de 34 bits de anchura: cables de 2 metros a 200 MHz y cables de 10 metros a 100 MHz. Con ambos flancos de reloj usados para transmitir datos, esto consigue unas transferencias de datos de 1.6/0.8 Gbyte/s.

Hay disponibles interfaces para hosts con bus PCI de 32/64 bits, a 33/66 MHz. El interfaz se compone de dos chips: el PCI to Mercury Bridge (PMB) y el Mercury Interface Chip (MIC).

El MIC proporciona el interfaz eléctrico en la red Mercury e implementa los servicios de fiabilidad extremo a extremo. Acepta paquetes de datos del PMB, mira en la tabla de enrutamiento, genera el código detector de errores e inyecta el paquete en la red. PMB y MIC están sincronizados con el reloj PCI a 66 MHz, MIC convierte el flujo de datos a los 200/100 MHz de la interconexión Mercury.

Synfinity ofrece varios servicios de transferencia de datos y comunicación. Se pueden iniciar meras transferencias Send/Receive para mandar directamente datos hacia otros nodos. El nodo receptor almacena los datos en colas de mensajes que se encuentran en memoria principal. Las operaciones de memoria remota están soportadas a través de llamadas Put/Get. Especialmente para cálculos paralelos de grano fino, Synfinity ofrece operaciones denominadas *Remote Atomic Read-Modify-Write* (RMW). El usuario puede elegir entre varias operaciones tales como *Fech and Add* o *Compare and Swap*. Este modo de comunicación puede ser extremadamente útil para implementar bloqueos globales. Para las operaciones colectivas pueden programarse registros especiales de barrera para colaborar en una sincronización en el cluster con un número arbitrario de nodos.

Mercury usa control de flujo basado en créditos. El PMB también puede generar reconocimientos *Busy* o *Retry* que le indica al emisor que reintente la transacción más tarde.

Synfinity ofrece un conmutador de seis puertos con virtual cut-through a 200 Mhz. Puede usarse para conectar hasta 64 nodos en cualquier topología. Utiliza enrutamiento fuente con una latencia de 42 ns.

XIII. Trabajos Futuros

A lo largo de este artículo hemos mostrado de una forma muy condensada las características de las principales redes de interconexión de altas prestaciones. Pretendemos fijar la base para una serie de estudios acerca de la adecuación de cada una de las redes comentadas a cada tipo de problema. Habrá que considerar las facilidades que ofrece cada una para broadcasting, envío simultáneo de todos a todos, etc, así como el tamaño de los paquetes que maneja el algoritmo, para determinar si pedimos una latencia menor o un mayor ancho de banda.

También podemos plantearnos el utilizar el paradigma de paso de mensajes o el de memoria compartida. Gracias a SCI y a Memory Channel se nos ofrece la posibilidad de tener una memoria virtual global o una memoria compartida distribuida que nos permita programar con un modelo de memoria compartida en mente, pasándole la mayor parte del trabajo sucio, es decir, las comunicaciones, al hardware. Incluso aprovechar la baja latencia de estos protocolos para estudiar una capa de paso de mensaje construida sobre ellos, que explotaría su eficiencia en el envío de paquetes de pequeño tamaño.

Nuestro objetivo final, establecer de forma cuantitativa unas bases para elegir la red que interconectará nuestro cluster de estaciones de trabajo ofreciendo la mejor relación rendimiento/precio para el tipo de problemas que pretendemos resolver.

Bibliografía

[1] Thomas E. Anderson, David E. Culler, and David A. Patterson, "Case for NOW (Networks of Workstations)," *IEEE Micro*, pp. 54–64, February 1995.

[2] Marco Fillo and Richard B. Gillett, "Architecture and Implementation of Memory Channel 2," *Digital Technical Journal*, vol. 9, no. 1, 1997.

[3] Nanette J. Boden et al., "Myrinet: A Gigabit-per-Second Local Area Network," *IEEE Micro*, vol. 15, no. 1, pp. 29–36, February 1995.

[4] Robert W. Horst, "TNet: A Reliable System Area Network," *IEEE Micro*, vol. 15, no. 1, pp. 37–45, February 1995.

[5] Michael D. Schroeder et al., "Autonet: A High-Speed, Self-Configuring Local Area Network Using Point-to-Point Links," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1318–1335, October 1991.

[6] Scott Pakin, Mario Lauria, and Andrew Chien, "High Performance Messaging on Workstations: Illinois Fast Messages (FM) for Myrinet," *Association for Computing Machinery*, 1995.

[7] José Duato, Sudhakar Yalamanchili, and Lionel Ni, *Interconnection Networks, An Engineering Aproach*, IEEE Computer Society Press, 1997.

[8] Rajkumar Buyya, Ed., *High Performance Cluster Computing: Architectures and Systems*, vol. I, Prentice Hall PTR, NJ, USA, 1999.

[9] David García and William Watson, "ServerNet II," in *Proceedings CANPC Workshop*, 1998, Lecture Notes in Computer Science 1362.

[10] Jeff Larson, "The HAL Interconnect PCI Card," in *Proceedings CANPC Workshop*, 1998, Lecture Notes in Computer Science 1362.

[11] V. Puente, B. Torón, J.A. Gregorio, and R. Beivide, "Plataformas Paralelas Basadas en Redes Myrinet," in *VII Jornadas de Paralelismo*, 1997.

[12] Anthony Alles, "Atm internetworking," Cisco Systems, available at http://cell-relay.indiana.edu/cell-relay/ docs/cisco.html.

[13] "The HIPPI Protocol," Available at http://www.cis.ohio-state.edu/~jain/ cis788-95/hippi/.

[14] "SCI Standardization," http://www1.cern.ch/RD24/ specification.html.

[15] Manfred Liebhart, *Optimization of SCI Systems*, Ph.D. thesis, Technical University of Austria, February 1998.

[16] IEEE802.3 Higher Speed Study Group, "Cabling Survey Ad Hoc," Tech. Rep., Institute of Electrical and Electronics Engineers, Inc. (IEEE)., http://grouper.ieee.org/groups/802/3/ 10G_study/public/index.html.

[17] Arie Van Praag, "What is GSN and can it be used for high-energy physics data acquisition?," Tech. Rep., CERN European Organization for Nuclear Research, Available at: http://www.cern.ch/HSI/gsn/reports/Gsn64rp7.PDF, January 1998.

[18] Gigabit Ethernet Alliance, "Gigabit Ethernet. Accelerating the Standard for Speed," Tech. Rep., Gigabit Ethernet Alliance, http://www.gigabit-ethernet.org/technology/ whitepapers/gige_0698/gea1999_rev.pdf, May 1999.

[19] Maximilian Ibel, Klaus E. Schauser, cris T. Scheiman, and Manfred Weis, "High Performance Cluster Computing Using SCI," Tech. Rep., University of California, Santa Barbara, Available at http://www.cs.ucsb.edu/research/sci.

[20] Mario Lauria and Andrew Chien, "MPI-FM: High Performance MPI on Workstation Clusters," Tech. Rep., University of Illinois at Urbana-Champaign, Available at http://www-csag.cs.uiuc.edu/papers/index.html.