Balanceo del Encaminamiento Distribuido para la Comunicación en Redes de Interconexión

D.Franco, I.Garcés, E.Luque
Unitat d'Arquitectura d'Ordinadors i Sistemes Operatius - Departament d'Informàtica
Universitat Autònoma de Barcelona - 08193-Bellaterra, Barcelona, Spain
Ph. +34-93-581.19.90 - Fx.+34-93-581.24.78
E-mail: {iarq23,d.franco,iinfd}@cc.uab.es; Web: http://aows1.uab.es

Resumen

En la creación de redes de interconexión para computadores paralelos, es crucial un diseño eficiente debido a su impacto sobre las prestaciones. Por lo tanto, se debería incluir un sistema de encaminamiento de alta velocidad que minimizase la contención y evitase la formación de "hotspots". Los sistemas de encaminamiento estático no son capaces de adaptarse a las condiciones de tráfico. Hemos desarrollado un nuevo método para distribuir uniformemente el tráfico de comunicaciones sobre la red de interconexión llamado Balanceo del Encaminamiento Distribuido (DRB por sus siglas en inglés) que se basa en la expansión de caminos controlada por la carga de comunicaciones con objeto de mantener una baja latencia de mensajes.

El método distribuye uniformemente la carga de comunicación entre todos los enlaces de la red de interconexión y mantiene la latencia controlada siempre que los requerimientos de ancho de banda totales no excedan del ancho de banda total disponible en la red de interconexión. DRB define cómo crear caminos alternativos para expandir los caminos simples (Definición de camino expandido) y cuándo usar los caminos alternativos dependiendo de la carga de tráfico (Selección de caminos expandidos realizada por el Routing DRB). La definición de caminos alternativos ofrece un amplio rango de alternativas a elegir y el Routing DRB está diseñado con el objetivo de minimizar el overhead de monitorizar y decidir. Se presentan resultados de la evaluación realizada en términos de latencia y ancho de banda, las conclusiones de la experimentación y la comparación con otros métodos existentes. Se demuestra que DRB es un método efectivo para balancear el tráfico de la red.

1.Introducción

La computación basada en "clusters" de estaciones de trabajo es una alternativa realista para la Computación de Altas Prestaciones. Actualmente estas configuraciones, que pueden considerarse un tipo de computador paralelo, incorporan una red de interconexión específica compuesta por "switches" a medida. La red de interconexión es una red de alta velocidad que constituye una alternativa a la conexión típica entre las estaciones de trabajo basadas en bus de las LAN. Las redes de alta velocidad están compuestas de conexiones punto a punto entre las estaciones de trabajo formando una cierta topología. Los "switches" tienen varios "ports" a los cuales se conectan otros "switches" o estaciones de trabajo. Los "switches" ATM [DeP95] o Myrinet [Myr98] son ejemplos de tales "switches". Las prestaciones de una red de interconexión son un factor determinante para permitir el cómputo de altas prestaciones en estas configuraciones en "cluster". El cómputo se realiza en las estaciones de trabajo mientras se intercambian mensajes viajando de "switch" a "switch".

Uno de los peores problemas que afecta a las prestaciones de las comunicaciones tanto en computadores paralelos como en "clusters" de estaciones de trabajo es la contención de los mensajes, es decir, cuando dos o mas mensajes quieren usar el mismo enlace al mismo tiempo. La contención de mensajes genera latencia en los mensajes, es decir, el tiempo que el mensaje debe esperar por recursos que necesita. Esta latencia se añade al tiempo de transmisión, que es el tiempo que el mensaje emplea viajando por la red en ausencia de otros mensajes, para dar el tiempo total de comunicación.

Una contención de mensajes sostenida puede producir hotspots [Pfi85]. Un hotspot es una región de la red que está saturada (es decir, existe mayor demanda de ancho de banda del que la red puede ofrecer), y entonces, los mensajes que entran en esta región sufren grandes latencias mientras otras regiones de la red están menos cargadas y disponen de ancho de banda libre. El

problema está en que existe una incorrecta distribución de la carga de comunicaciones sobre la topología de la red y que, aunque el requerimiento de ancho de banda total no supere el ancho de banda total ofrecido por la red, esta distribución no uniforme genera puntos saturados como si la red entera estuviera colapsada. Esta saturación se produce cuando los buffers de los encaminadores de la región hotspot están llenos, mientras otras regiones de la red tienen recursos libres. Además, las situaciones de hotspot se propagan rápidamente a áreas contiguas en un efecto dominó el cual puede colapsar la red entera rápidamente. Este efecto es aún peor en el caso de control de flujo Wormhole porque un paquete bloqueado ocupa un gran número de enlaces a través de la red. Actualmente, para evitar la generación de hotspots, y mantener una latencia estable y uniforme, las redes de interconexión se utilizan a baja carga. Este hecho conduce a una infra-utilización de la red de interconexión.

El programa paralelo que se ejecuta sobre un computador paralelo se describe como una colección de procesos y canales y existe un mapping que asigna cada proceso a un procesador. Los procesos ejecutan concurrentemente y se comunican por canales lógicos. Con respecto al tiempo total de ejecución del programa, la latencia de las comunicaciones debe ser evitada para hacerlas mas rápidas. Es más importante evitar las grandes variaciones de la latencia que una cierta cantidad de latencia. La razón es que una cantidad uniforme de latencia se puede tolerar ocultándola mediante el método de asignar un exceso de paralelismo, es decir, tener suficientes procesos por procesador, y despachar los procesos listos mientras otros procesos esperan por sus mensajes. Pero, si la latencia sufre grandes variaciones imprevistas respecto los valores esperados (debido a hotspots, por ejemplo), aparecerán procesadores ociosos porque todos sus procesos estén bloqueados, esperando por sus mensajes correspondientes, y, consecuentemente, se incrementará el tiempo total de ejecución.

Varios mecanismos se han desarrollado para evitar la generación de hotspots debidos a la contención de mensajes en redes de interconexión, tales como los algoritmos de encaminamiento dinámico o adaptativo, los cuales tratan de adaptarse a las condiciones de tráfico. Algunos ejemplos son Planar Adaptive Routing [CK92], Turn Model [NG92], Algorithm de Duato [Dua93], Compressionless Routing [KLC94], Chaos Routing [Ksn91], Random Routing [Val81] [May93] y otros métodos presentes en [DYN97]. La principal desventaja del encaminamiento adaptativo es el gran overhead debido a la monitorización de la información, al cambio de canales y la necesidad de garantizar la ausencia de deadlock, livelock y starvation. Estos factores han limitado la implementación de estas técnicas en máquinas comerciales.

El trabajo presentado en este artículo se centra en el desarrollo de un nuevo método para distribuir los mensajes en la red de interconexión mediante la expansión de caminos controlada por la carga de la red. El método se llama Balanceo del Encaminamiento Distribuido (DRB, por sus siglas en inglés) y su objetivo es el de balancear uniformemente la carga de tráfico sobre todos los caminos de la red de interconexión. El método se basa en la creación, ante ciertos valores de carga, de varios caminos alternativos simultáneos entre cada nodo fuente y destino para permitir un incremento de uso del ancho de banda y mantener una latencia de mensajes baja. DRB define cómo crear los caminos alternativos para expandir los caminos simples (definición de los caminos multi-carril) y cuándo usarlos, dependiendo de la carga de tráfico (selección de los caminos multi-carril).

El método ofrece un amplio rango de alternativas las cuales van desde el método de encaminamiento estático mínimo hasta el de encaminamiento aleatorio. De hecho, ambos métodos están incluidos en la especificación DRB como casos particulares.

Las dos próximas secciones presentan la técnica DRB. DRB tiene dos componentes: primero, una metodología sistemática para generar los caminos multi-carril según la topología de la red y, segundo, un algoritmo de encaminamiento para monitorizar la carga de tráfico y seleccionar caminos multi-carril para conseguir la distribución de los mensajes según la carga. La sección 2 define algunos conceptos y la metodología de generación de caminos multi-carril. La sección 3 presenta el algoritmo de Routing DRB. La sección 4 muestra la evaluación del método DRB realizada mediante experimentación con un simulador de redes de interconexión.

La sección 5 presenta una discusión del método y comparación con otros métodos existentes. Finalmente, la sección 6 presenta las conclusiones y el trabajo futuro.

2. Balanceo del Encaminamiento Distribuido (DRB)

El Balanceo del Encaminamiento Distribuido (DRB) es un método de creación de caminos alternativos entre fuente y destino en la red de interconexión usando una expansión de caminos controlada por la carga. DRB distribuye cada mensaje de cada par fuente-destino sobre un camino multi-carril formado por varios caminos. Esta distribución esta controlada por el nivel de carga de los caminos. El objetivo de DRB es conseguir una distribución uniforme de la carga de tráfico sobre la totalidad de la red de interconexión con objeto de mantener una latencia de mensajes baja y evitar la generación de hotspots. Esta distribución de mensajes mantendrá una baja latencia uniforme en toda la red de interconexión siempre que la demanda total de ancho de banda no exceda la capacidad de la red de interconexión. Además, debido a la eliminación de los hotspots, se incrementa el throughput de la red y se permite un uso de la red a cotas mayores de carga. Dependiendo de la carga de tráfico y de su patrón de distribución, el método DRB configura los caminos para distribuir la carga de los caminos mas cargados a los menos.

La idea principal del método se basa en el comportamiento de la latencia de los mensajes respecto al nivel de carga en redes de interconexión. Este comportamiento típico ha sido estudiado por muchos autores [Aga91], [Dal92], [DYN97] y puede representarse por una curva no-lineal en la cual se pueden identificar dos regiones: primero, una región plana a bajo nivel de carga con comportamiento cuasi-lineal (donde grandes cambios de la carga de comunicación causan pequeños cambios en la latencia) y, segundo, una pendiente brusca a partir de un valor threshold (en la cual cambios pequeños de la carga de comunicaciones causa grandes cambios en la latencia). Asimismo, se puede definir un valor threshold donde la curva cambia de la región plana a la región pendiente. Esta latencia threshold es el punto de la curva con mínimo radio de curvatura.

La región de la curva con pendiente pronunciada es no deseable porque significa que la latencia no es estable con respecto a pequeños cambios de la carga de tráfico. En concordancia con este comportamiento de la latencia, el método DRB mueve el punto de trabajo de la región saturada a un punto de menor latencia en la parte plana de la curva. Este efecto se consigue modificando la distribución de los caminos para reducir el tráfico sobre los caminos más cargados. El resultado conseguido son grandes reducciones de la latencia en los caminos congestionados y pequeños incrementos de latencia sobre los caminos no congestionados porque estos todavía tienen ancho de banda disponible y , por lo tanto, el efecto global es positivo. La configuración de latencia resultante es uniforme y baja para todos los caminos. El método DRB consigue los siguientes objetivos:

1. La reducción de la latencia de los mensajes bajo un cierto valor threshold mediante el cambio dinámico del número de caminos alternativos usados por el par fuente-destino, mientras se mantiene una latencia uniforme para todos los mensajes. Esto se consigue maximizando el uso de los recursos de la red de interconexión con objeto de minimizar los retardos de comunicación.
2. La minimización del alargamiento de los caminos. Este punto es importante para los controles de flujo Wormhole y Virtual Cut-through porque se incrementan dos factores principales que afectan la latencia como son el consumo de ancho de banda y los puntos de colisión. Para Store&Forward, es también importante porque el retardo de transmisión depende directamente de la longitud del camino de los mensajes.
3. Maximización del uso de los enlaces de los nodos fuente y destino, distribuyendo uniformemente los mensajes sobre todos los enlaces del nodo.

Con objeto de mostrar cómo DRB trabaja para crear los caminos alternativos, se presentan las siguientes definiciones:

Definición 0:

- Una red de interconexión I se define como un grafo dirigido I(N,E), donde N es un

conjunto de nodos $N = \bigcup_{i=0}^{MaxN} N_i$ y *E* un conjunto de arco que conectan pares de nodos. Generalmente, cada nodo esta compuesto de un encaminador y esta conectado a otros nodos mediante enlaces, representados por los arcos. La topología puede ser *regular* o *irregular*, dependiendo de la red. Por ejemplo, para *k*-ary *n*-cubes [Dal90], *n* es la dimensión y *k* el tamaño.

- Si dos nodos N_i y N_j están directamente conectados por un enlace, entonces N_i y N_j son *nodos adyacentes*
- *Distancia(* N_i *,* N_j *)* es el mínimo número de enlaces que deben ser cruzados para ir de *Ni* a *Nj* según el Grafo *I*.

Un *camino P(* N_i *,* N_j *)* entre dos nodos N_i y N_j es el conjunto de nodos seleccionados entre N_i y N_j según el encaminamiento estático mínimo definido para la red de interconexión. (Por ejemplo, Dimensional Order Routing para k-ary n-Cubes). N_i es el nodo *fuente* y N_j el nodo *destino*. La ***Longitud*** de un camino P ***Longitud****(P)* es el número de enlaces entre N_i *y* N_j siguiendo el encaminamiento definido. En el caso de encaminamiento estático mínimo:

Longitud (P(N_i *,* N_j *))= Distancia(* N_i *,* N_j *)* [1]

Definición 1: Un *Supernodo S(tipo, tamaño,* N_0^S *)* $= \bigcup_{i=0}^{l} N_i^S$ se define como una región estructurada de la red de interconexión constituida por *l* nodos adyacentes N_i^S alrededor de un nodo "central" N_0^S tal que:

1) N_i^S cumple con una cierta propiedad especificada en *tipo* y

2) *Distancia(* N_i^S *,* N_0^S *)<=tamaño.*

Como casos particulares, cualquier nodo individual y la red completa pueden ser Supernodos. Un Supernodo que contenga solo un nodo se llama forma canónica mínima, si un Supernodocontiene todos los nodos de la red se llama forma canónica máxima. Un nodo puede pertenecer a mas de un Supernodo.

DRB define dos tipos diferentes de Supernodos aplicables a cualquier topología. El primero se llama Area de Gravedad y el segundo Subtopología. Los parámetros *tipo* y *tamaño* del Supernodos determinan que nodos están incluidos en el Supernodoy además las siguientes propiedades: Forma topológica, número de nodos *l*, Grado del Supernodo(Número de enlaces del los nodos N_i no conectados a otros enlaces de nodos N_i , es decir enlaces conectados al exterior del Supernodo), número de nodos del Supernodoconectados a N_0^S .

Supernodo Area de Gravedad

Un Supernodo *Area de Gravedad* es el conjunto de nodos que se encuentran a una distancia del nodo N_0^S menor o igual que *tamaño*. Este tipo, *S("Gravity Area", tamaño,* N_0^S *),* define, para una red de grado n, un árbol n-ario con raíz en el nodo central N_0^S y profundidad *tamaño*. Dependiendo de la topología, no es siempre posible definir un árbol completo. Este tipo "mapea" un árbol de grado máximo sobre la topología. Este árbol se expande al máximo e incluye todos los nodos que están a una distancia *tamaño* o menor de la raíz. Es adecuado para redes regulares e irregulares. Fig. 1 muestra el concepto de Supernodo.

Supernodo Subtopología:

Un Supernodo *S("Subtopology", tamaño,* N_0^S *)* tiene la misma forma topológica completa o parcial que la red de interconexión, pero su *dimensión* y/o *tamaño* se reducen. Por lo tanto, un Supernodo *Subtopología* puede considerarse un tipo de "proyección" topológica de la red. Se

puede aplicar a redes regulares con una topología estructurada , dimensión y tamaño. Por ejemplo, en un k-ary n-cube, un Supernodo *Subtopología* es cualquier j-ary m-cube con j<k y/o m<n.

Una descripción más detallada de estos Supernodos para k-ary n-cubes [Dal90] y redes Midimew [Bei92] y la evaluación de las propiedades antes mencionadas se puede encontrar en [Gar97].

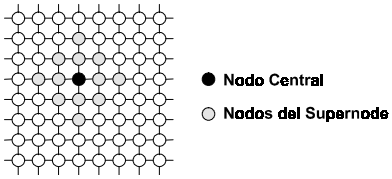


Fig 1. Nodos pertenecientes a un Supernodo Area de Gravedad de tamaño 2.

Definición 2: Un *Multi-step Path MSP*(*SOrigin,* $N_i^{SOrigin}$, N_j^{SDest} , *SDest*) es el camino generado entre dos Supernodos Fuente y Destino (*SOrigin* y *SDest)* como

$$MSP_s = \prod (N_0^{SOrigin}, N_i^{SOrigin}, N_j^{SDest}, N_0^{SDest}) =$$

$$= P1(N_0^{SOrigin}, N_i^{SOrigin}) \bullet P2(N_i^{SOrigin}, N_j^{SDest}) \bullet P3(N_j^{SDest}, N_0^{SDest}),$$

donde • significa concatenación de caminos, y P1, P2 y P3 son caminos mínimos:

Paso 1: Desde el nodo central del Supernodo *Supernodo Fuente,* $N_0^{SOrigin}$, al nodo perteneciente a *Supernodo Fuente,* $N_i^{SOrigin}$.

Paso 2: Desde el $N_i^{SOrigin}$ al nodo perteneciente al *Supernodo Destino,* N_j^{SDest} .

Paso 3: Desde el N_j^{SDest} al nodo central del Supernodo *Supernodo Destino,* N_0^{SDest} .

Los Pasos 1 **y/o** 3 pueden ser nulos si *Supernodo Fuente=*{ $N_0^{SOrigin}$ } (Es canónico mínimo) **y/o** *Supernodo Destino* ={ N_0^{SDest} }(Es canónico mínimo).

Si ambos *Supernodos Fuente* y *Destino* son canónico mínimo, entonces, el *Multi Step Path* se dice que está en forma **canónica**, y es igual al camino siguiendo encaminamiento estático mínimo. Fig.2 muestra un ejemplo de un Multi-step path.

La ***Longitud*** de un *Multi-step path Longitud (MSP)* se define como la suma de la longitud de cada paso individual siguiendo encaminamiento estático:

$$Longitud(MSP) = Longitud(P1(N_0^{SOrigin}, N_i^{SOrigin})) + Longitud(P2(N_i^{SOrigin}, N_j^{SDest})) +$$

$$Longitud(P3(N_j^{SDest}, N_0^{SDest})) \qquad [2]$$

De esta definición se puede ver que algunos Multi-step Paths (MSP a partir de ahora) entre $N_0^{SOrigin}$ y N_0^{SDest} pueden ser de longitud no mínima. Esta longitud es una medida del tiempo de transmisión del MSP .

Latencia del MSP es el tiempo de espera consumido por un mensaje para ir de $N_0^{SOrigin}$ a N_j^{SDest} en colas de los encaminadores debido a contención entre mensajes más el tiempo de transmisión.

$$Latencia(MSP) = \text{Tiempo de Transmisión} + \sum_{\forall nodes \in MSP} Queuingdelay(Nodo) \qquad [3]$$

MSP *Ancho de Banda(MSP)* es el inverso de la Latencia:

$$Ancho\ de\ Banda(MSP) = Latencia(MSP)^{-1} \qquad [4]$$

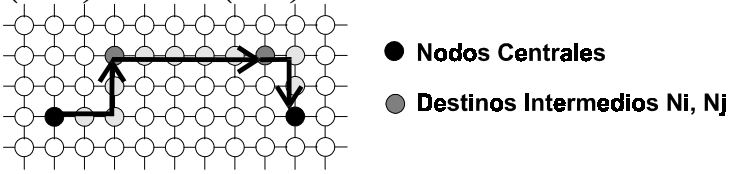


Fig 2. Multi Step Path

Definición 3: Un *Metacamino P*(Supernodo Fuente, Supernodo Destino)* es el conjunto de todos los multi-step paths P_i generados entre los Supernodos *Supernodo Fuente* y *Supernodo Destino:* $P^* = \bigcup_{\forall i,j} MSP\ (N_0^{SOrigin}, N_i^{SOrigin}, N_j^{SDest}, N_0^{SDest})$. Supóngase *l* el número de nodos del *Supernodo Fuente* y *k* el número de nodos del *Supernodo Destino*.

Anchura del Metacamino s es el número de Multi-step Paths que componen el Metacamino:

$$Anchura\ del\ \ Metacamino = s = l*k \qquad [5]$$

Cuando *Supernodo Fuente* y *Destino* están en forma canónica mínima, el Metacamino M* se dice que está en forma **canónica**, es decir, está compuesto únicamente por el camino mínimo estático (s=1). Fig. 3 muestra el Metacamino generado entre dos Supernodos de tamaño 2.

Longitud del **Metacamino** (Longitud(P*)) es la media de todas las longitudes de los multi-step paths individuales que lo componen,

$$Longitud(P^*) = (1/s)\sum_{\forall s} longitud(MSP) \qquad [6]$$

Latencia del Metacamino (Latencia(P*)) es la latencia equivalente definida como la inversa de la suma de las inversas de las latencia de los MSPs individuales. Estas latencia son, de hecho, anchos de banda y su suma es el ancho de banda equivalente. El concepto físico es el mismo que sumas la asociación de elementos en paralelo que se puede encontrar en sistemas electrónicos por ejemplo.

$$Latencia(P^*) = (\sum_{\forall s} Latencia(MSPs)^{-1})^{-1} \qquad [7]$$

Latencia Canónica de un Metacamino P* es el tiempo que un mensaje de un determinado tamaño consume para recorrer un camino en ausencia de otros mensajes en la red y cuando el Metacamino es canónico.

Ancho de Banda del Metacamino se define como el inversa de la Latencia:

$$Ancho\ de\ Banda(P^*) = Latency(P^*)^{-1} = (\sum_{\forall s} Ancho\_de\_Banda(MSPs) \qquad [8]$$

Ancho de Banda canónico del Metacamino se define como el Ancho de Banda del Metacamino canónico en ausencia de otros mensajes en la red. Es la inversa de la Latencia canónica y representa el máximo número de mensajes por unidad de tiempo que un camino puede aceptar

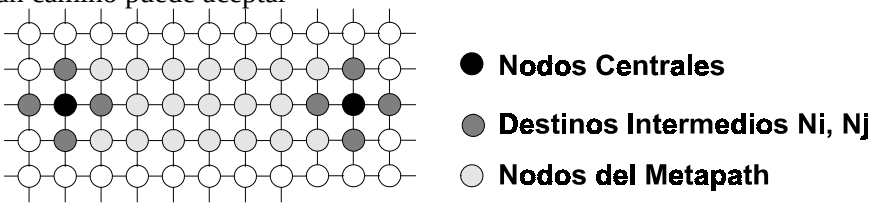


Fig 3. Metacamino

3. Encaminamiento DRB

Tal y como se describió en la introducción, para una aplicación dada, se asigna un *Supernodo Fuente* al nodo fuente y un *Supernodo Destino* al nodo destino de manera que se crea un Metacamino *P** para cada canal lógico. El Supernodo fuente es una **MeSA (Message Scattering Area)** desde el nodo fuente y el Supernodo destino es una **MeGA (Message Gathering Area)** al Destino. En cada proceso fuente se selecciona del Metacamino *P* un Multi-Step Path* P_s (*SOrigin,* $N_i^{SOrigin}$, N_j^{SDest}, *SDest*) para enviar a través de el los mensajes hacia su destino.

Bajo este esquema, la comunicación entre un nodo fuente y un nodo destino se puede ver como si se estuviera utilizando un Metacamino multi-carril más extenso, de un ancho de banda

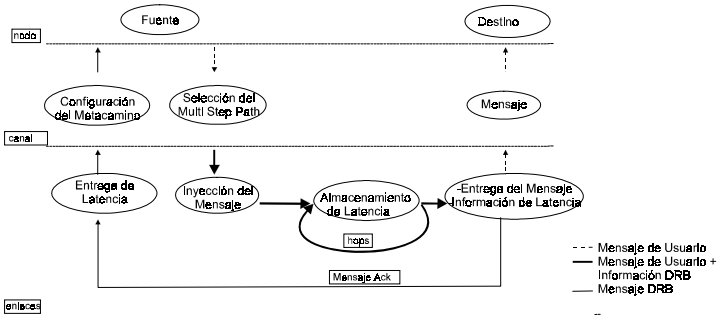
potencialmente más grande que el camino original desde un supernodo fuente a un supernodo destino. Este camino multi-carril se asemeja a una autopista, cuyos accesos de entrada y salida estarían representados por las áreas de MeSA y MeGA, respectivamente.

Aún cuando ciertos caminos puedan compartir algunos enlaces, el hecho de utilizarlos alternativamente produce el efecto de maximizar el uso del ancho de banda disponible para el metacamino. Al utilizar DRB se logran dos efectos importantes para la comunicación: primero, se reduce la latencia en los caminos únicos debido a que éstos están menos cargados. Esta reducción es alta debido a la conducta no lineal de la latencia (tal y como se explicó en la Sección 2). Segundo, para cada par fuente-destino, se utilizan caminos en paralelo lográndose aumentar el throughput.

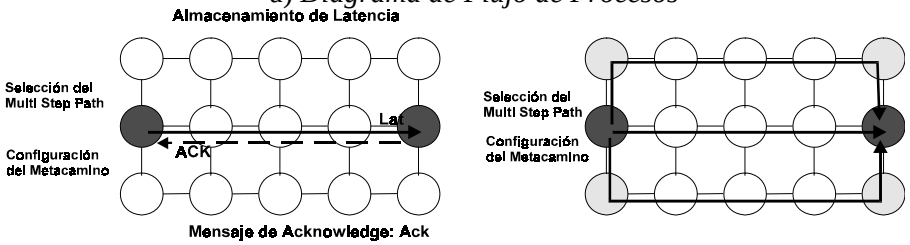
Encaminamiento DRB:

Las funciones principales del encaminamiento DRB son: recoger la latencia detectada por los mensajes en la red, configurar dinámicamente los metacaminos dependiendo de ésta y distribuir los mensajes entre los Multi-Step Paths del Metacamino. De esta manera, DRB se subdivide en tres partes diferenciadas: Monitorización de la Carga, Configuración Dinámica de Metacaminos y Selección del Multi-Step Path. Estas fases son ejecutadas independientemente por cada canal de la aplicación. La actividad de monitorización es llevada a cabo por los mensajes y su objetivo es registrar la latencia que encuentran en su camino. La Configuración Dinámica del Metacamino se ejecuta a nivel del canal cada vez que un mensaje llega a su destino y la selección del Multi-Step Path se realiza a nivel de canal cada vez que un mensaje se inyecta.

La Fig. 4a muestra las acciones ejecutadas por DRB a nivel de diagrama de flujo de procesos. La Fig. 4b muestra el procedimiento para un mensaje que viaja siguiendo el encaminamiento estático y la latencia se recibe por medio de un mensaje de acknowledge en el nodo fuente. La Fig. 4c. muestra el procedimiento una vez que el Metacamino se ha expandido y los mensajes se envían a través de diferentes Multi-Step Paths.



a) *Diagrama de Flujo de Procesos*



b) *Fases de Monitorización y Configuración* c) *Fase de expansión del camino*

Fig. 4. Algoritmo DRB

Para llevar a cabo esta funcionalidad, DRB desarrolla las siguientes acciones:

1- Monitorización de la Carga:

Los propios mensajes ejecutan las operaciones relacionadas con la monitorización de la carga. Cada mensaje registra y lleva consigo la latencia. (Almacenamiento de latencias en la Fig. 4). El mensaje guarda información sobre la latencia que experimenta en cada nodo que atraviesa cuando se bloquea debido a la contención con otros mensajes (por ejemplo, L1, L2, L3 en la Fig. 4). La monitorización determina la Latencia(Ps) de acuerdo a la expresión [3].

Cuando un mensaje llega a su destino la información de latencia del MSP que había registrado se envía de vuelta al nodo origen por medio de un mensaje de acknowledge. Este mensaje de acknowledge tiene máxima prioridad en la red.

El objetivo de la actividad de monitorización es detectar el patrón de tráfico actual con el fin de identificar las regiones más y menos cargadas de la red.

El siguiente pseudo-código muestra la fase de Monitorización de la carga:

Monitorización de la Carga() /*Se ejecuta en cada encaminador intermedio*/
Begin
1.***For*** cada hop,
1.1.Acumular latencia(Tiempo en colas) para calcular Latencia(MSP)
End For
2.En el destino la Latencia(MSP) se envia de vuelta al origen y se entrega a la función de ***Configuración del Metacamino***.
End Monitorización

2- Configuración Dinámica del Metacamino:

El objetivo de esta fase es determinar el *tipo* y *tamaño* del Metacamino de acuerdo a la Latencia(P*).

Cuando el origen recibe una latencia MSP calcula la nueva Latencia(P*) (utilizando [7]) y toma la decisión de incrementar o reducir el tamaño de los Supernodos dependiendo si la Latencia(P*) esta fuera del intervalo definido por [Thl-Tol, Thl+Tol]. El valor umbral identifica el punto de saturación de la latencia (el cambio desde la región plana a la vertical), como se explicó en la Sección 2. *Tol* define la desviación estándar tolerada entre ancho de banda actual y el ancho de banda canónico. El intervalo determinado por Tol define el rango donde el Metacamino no se modifica.

Los tamaños del Supernodo se modifican para encontrar el nuevo ancho del Metacamino dependiendo de la relación entre el Ancho de Banda Canónico (P*) y el Ancho de Banda (P*) definido por la siguiente fórmula:

-Encontrar el $\Delta size$ que

$$ABc < AB + ABc * Tol * 1/k * \sum_{i=1}^{\Delta size} i < ABc(1+1/k) \qquad [9]$$

incrementar el Ancho del Metacamino = Ancho del Metacamino + $\Delta size$; si AB<Abc*Tol

$$ABc < AB - ABc * Tol * 1/k * \sum_{i=1}^{\Delta size} i < ABc(1+1/k); \qquad [10]$$

y decrementar Ancho del Metacamino = Ancho del Metacamino - $\Delta size$; if AB>Abc*Tol

- k es un parámetro que define el uso del canal y depende del número de canales lógicos de la aplicación y del tamaño de la red.

La configuración de los Supernodos toma en cuenta los valores de latencia, la topología de la red de interconexión y la distancia física de los nodos fuente y destino para equilibrar el ancho de banda y el alargamiento del Metacamino.

El siguiente pseudo-código muestra la fase de Configuración del Metacamino:

Configuración_del_Metacamino (Threshold Latencia Th, Tolerance Tol)
/*Se ejecuta en los nodos fuente cuando llega una nueva latencia */
/*Latencia Threshold es la latencia de cambio entre la región plana a la vertical definida en la Sec. 2
Tol define el intervalo dentro del cual el Metacamino no cambia */
Var MSP_Latencias: Array[1..SSNSize*DSNSize] of int;
Begin
1.Recibir Latencia(MSP)

2.Calcular Latencia(P*) utilizando [7]
*3.**If*** (Latencia(P*) > Thh+Tol) Incrementar Tamaños del Supernodo según [9]
ElseIf (Latencia(P*) < Thl-Tol) Reducir Tamaños del Supernodo según[10]
EndIf
End **Configuración del Metacamino**

3- Selección del MSP:

Esta función selecciona un MSP para que cada mensaje distribuya la carga entre los Multi-Step Paths del Metacamino. Para cada mensaje enviado, se selecciona un MSP dependiendo del Ancho de Banda del MSP que es el inverso de la Latencia del MSP; a mayor disponibilidad de ancho de banda, el uso se hace más frecuente. Supongamos que MSP(k) es el $k^{ésimo}$ MSP del Metacamino y BW(MSP(k)) es su ancho de banda asociado. Los Anchos de Banda del MSP se utilizan como los valores de una distribución discreta de los MSPs. El procedimiento es el siguiente:

1. Convertir la distribución de latencias en una función de distribución de latencias P, obteniendo las proporciones P[MSP(k)]=P[MSP(k)<=k] añadiendo y normalizando los Anchos de Banda discretos de cada MSP.

$$P(MSP(k))= \frac{\sum_{k=1}^{i} AB(MSP(k))}{\sum_{k=1}^{s} AB(MSP(k))}; \quad P(MSP(i)) = 1 ; (s = Number\ of\ MSPs) \quad [11]$$

2. Generar un valor aleatorio con distribución uniforme con valor R entre [0,1).

3. Encontrar el valor k que

$$P[MSP(k-1)<k-1] < R <= P[MSP(k)<k] \quad [12]$$

El siguiente pseudo-código muestra la fase de selección del Multi Step Path:

Selección_del_Multi_Step_Path() /* Ejecutada en el nodo fuente cuando inyecta un mensaje */
Begin
1.Construir la función de distribución acumulativa sumando y normalizando los Anchos de Banda del MSP siguiendo[11]
2.Generar un número aleatorio entre [0,1)
3.Seleccionar un MSP de acuerdo a la función de distribución acumulativa [12]
End *Selección_del_Multi_Step_Path*

DRB ha sido diseñado con el objetivo de minimizar los overheads y de manera que sea escalable. En este sentido, no hay un intercambio periódico de información y es totalmente distribuido. Tiene la característica que al actuar con cargas bajas de comunicaciones, la actividad de monitorización es mínima y los mensajes siguen encaminamiento mínimo.

El espacio de memoria y el overhead del tiempo de ejecución del algoritmo es muy bajo debido a que las acciones implicadas son muy sencillas. Además, el número de veces que se ejecutan estas actividades es linealmente dependiente del número de canales lógicos de la aplicación y el número de mensajes enviado.

En cuanto al overhead de tiempo, la tarea de monitorización es sólo la latencia registrada por el propio mensaje, es decir, almacenamiento de un valor entero, y el algoritmo de Configuración del Metacamino es un cómputo local y sencillo que se aplica sólo en el caso que se detecte un aumento de latencia.

Con respecto al overhead de espacio, el registro de la latencia es uno o pocos enteros que el mensaje lleva en su cabecera, y la única información que se mantiene en el nodo fuente es el array de latencias MSP para cada canal lógico.

Es importante resaltar que para lograr una distribución uniforme de la carga, se necesita implementar una acción global y que por esta razón, todos los nodos fuente-destino son capaces de expandir sus caminos dependiendo de la carga de mensajes durante la ejecución del programa.

DRB Routing toma ventaja de la localidad espacial y temporal de las comunicaciones de los programas, tal y como los sistemas de memoria cache hacen con las referencias de memoria. El algoritmo adapta las configuraciones de los Metacaminos al patrón de tráfico actual. Mientras este patrón es constante, las latencias serán bajas y las configuración de los Metacaminos no se activa. Si la aplicación cambia a un nuevo patrón de tráfico y las latencias de los mensajes cambia, la configuración de Metacamino de DRB adaptará los Metacaminos a la nueva situación. DRB es útil para patrones de comunicaciones repetitivos que son los que pueden causar las peores situaciones de hot-spots. También, DRB reduce la latencia de inyección configurando varios caminos disjuntos entre cada par fuente-destino. Además, la adaptabilidad del Metacamino es específica y puede ser diferente para cada par fuente-destino dependiendo de la distancia estática o de las condiciones de latencia, de manera que pueda adaptarse a cada patrón.

El efecto de este método es permitir un alto nivel de tráfico aceptado, es decir, que sólo se sature la red a niveles muy altos de tráfico. Esto significa que la granularidad de los procesos de la aplicación (relación cómputo/comunicaciones) puede ser menor y presentar mayores variaciones porque estas fluctuaciones son mejor toleradas. Dado este marco de referencia, el único punto que la aplicación debe tomar en cuenta es que los requerimientos del ancho de banda total no excedan el ancho de banda de la red: los requerimientos de distribución del ancho de banda no son motivo de preocupación por parte del usuario.

4. Evaluación de DRB

Hemos desarrollado un simulador de redes conducido por tiempos para estimar las prestaciones del encaminamiento DRB. En esta sección mostraremos los resultados para varios patrones de tráfico y con variación de la carga dada una longitud de mensaje fija. Estos resultados los compararemos con los del Encaminamiento estático. El simulador implementa el Encaminamiento aleatorio DRB presentado en la sección 3. También se puede elegir encaminamiento estático tipo DOR (Dimensional Order Routing). Se pueden simular diferentes topologías de red de cualquier tamaño (k-ary n-cubes, midimews) y los tipos de control de flujos utilizados comúnmente (Store&Forward, Wormhole y Cut-through).

Las simulaciones consistieron en envío de paquetes a través de los enlaces de la red dependiendo de un patrón de tráfico específico. Las simulaciones fueron realizadas para varias topologías y tamaños. Las topologías seleccionadas fueron: Toros, Hipercubos y Midimews variando en rangos desde 16 hasta 256 nodos. Hemos asumido control de flujo Wormhole y diez flits por paquete. Cada enlace es bi-direccional y tiene asociado sólo un buffer de un flit. La generación de paquetes sigue una distribución exponencial cuya media es el tiempo de llegada de los mensajes. Los resultados fueron ejecutados varias veces con diferentes semillas y se observó que presentaban valores consistentes. Las simulaciones se llevaron a cabo para 100,000 paquetes, los efectos de los primeros 20,000 paquetes entregados no se incluyen en los resultados para minimizar el efecto del estado inicial en las simulaciones.

Hemos escogido algunos de los patrones de comunicaciones utilizados comúnmente para evaluar las redes de interconexión [DYN97]. Los siguientes patrones de comunicaciones fueron considerados en nuestro estudio: Uniforme, hot-spot, bit-reversal, butterfly, perfect shuffle y matriz transpuesta. Los patrones bit-reversal, butterfly, perfect shuffle y matriz transpuesta toman en cuenta las permutaciones que se ejecutan usualmente en muchos algoritmos paralelos

Bajo el patrón de tráfico uniforme, cada nodo envía mensajes a los otros con la misma probabilidad. Bajo el tráfico de hot-spot algunos destinos están fijos de manera que se incremente el tráfico en una zona particular de la red y cause un área de saturación de la ocupación de los enlaces, es decir un hot-spot. Bajo el patrón bit-reversal, el nodo con coordenadas binarias a_{n-1}, $a_{n-2,}$ ..., a_1, a_0 se comunica con el nodo a_0, $a_{1,}$ ..., a_{n-2}, $a_{n-1.}$ El tráfico Butterfly se forma intercambiando los bits mas y menos significativos: el nodo con coordenadas binarias a_{n-1}, $a_{n-2,}$ ..., a_1, a_0 se comunica con el nodo a_0, $a_{n-2,}$ ..., a_1, $a_{n-1.}$ En el patrón de Matriz Transpuesta el nodo con coordenadas binarias a_{n-1}, $a_{n-2,}$ ..., a_1, a_0 se comunica con el nodo $a_{n/2-1,}$

..., a_0, a_{n-1},...,$a_{n/2}$. El patrón Perfect Shuffle se forma rotando un bit a la izquierda: el nodo con coordenadas binarias a_{n-1}, $a_{n-2,}$ ..., a_1, a_0 se comunica con el nodo a_{n-2}, $a_{n-3,}$ ..., a_0, $a_{n-1.}$

Hemos estudiado la latencia promedio de comunicaciones, el throughput promedio de la red y la distribución de carga en la red. La latencia de comunicaciones se midió como el tiempo total de espera de los paquetes para acceder a los enlaces en su camino desde el nodo fuente al destino. El throughput se calculó como la relación porcentual entre la carga aceptada (cantidad de información enviada y recibida) y la carga aplicada (carga inyectada). Estas cargas de comunicaciones se midieron como el número de mensajes por unidad de tiempo. Con el fin de mostrar la distribución de la carga, calculamos la latencia promedio en cada enlace de la red. Los experimentos se llevaron a cabo para un rango de tráfico en la red desde baja carga hasta la saturación. En la evaluación de DRB, tomando en cuenta la ocupación de los caminos, hemos seleccionado algunos supernodos para formar un metacamino compuesto de tres MSP, que incluye el original y dos caminos adicionales. Con esta configuración se obtendrá una gran reducción de las latencias debido a la baja ocupación de los canales.

A continuación, se presentarán y compararán los resultados cuantitativos de la experimentación que se ha realizado para medir las prestaciones del encaminamiento estático utilizando DOR y DRB para Toros 8x8 e Hipercubos de 6 dimensiones. Los valores obtenidos para otros tamaños no se muestran debido a que se observa la misma tendencia en los gráficos.

Análisis de Resultados

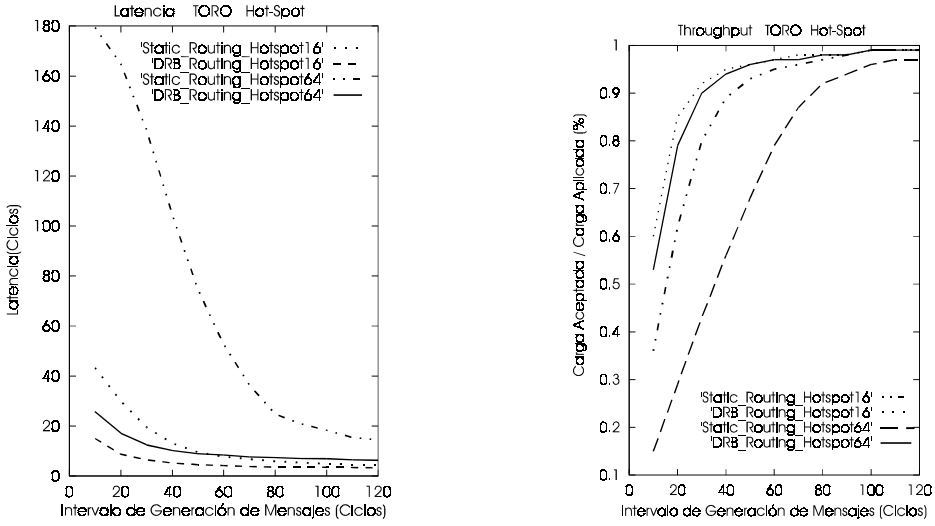
Bajo un tráfico uniforme las curvas muestran valores similares de latencia y throughput promedio debido a que no existe un desequilibrio en la carga y por lo tanto, DRB no modifica la distribución de tráfico. DRB no mejora esta situación por ser ésta la tendencia esperada (la uniformidad) de acuerdo a la definición dada de DRB. En cambio, para los otros cuatro patrones evaluados el desequilibrio de las comunicaciones es mayor y el uso del encaminamiento estático o de DRB afecta de manera significativa la tendencia de las curvas.

Las siguientes gráficas muestran dos curvas por cada patrón, una para encaminamiento estático y otra para DRB mostrando la latencia medida tal y como se explicó anteriormente. La carga se representa en valores decrecientes de izquierda a derecha.

La Fig. 5 (a) muestra los resultados de latencia para dos toros de 16 y 64 nodos, con el patrón de hot-spot. Las figuras 6 (a) y 7 (a) muestran los resultados de latencia utilizando los patrones bit-reversal, butterfly, perfect shuffle y matriz transpuesta para un toro de 64 nodos y un hipercubo de 6 dimensiones respectivamente. Los resultados son similares para todos los patrones de manera que los comentaremos conjuntamente. En general, se obtienen mejores resultados con el encaminamiento DRB. La diferencia entre las curvas de encaminamiento estático y DRB se incrementa a medida que aumenta la carga. Las curvas del encaminamiento estático muestran un aumento mayor en la latencia a medida que la carga se incrementa comparada con las de DRB. DRB efectúa una configuración automática del Metacamino dependiendo de la carga.

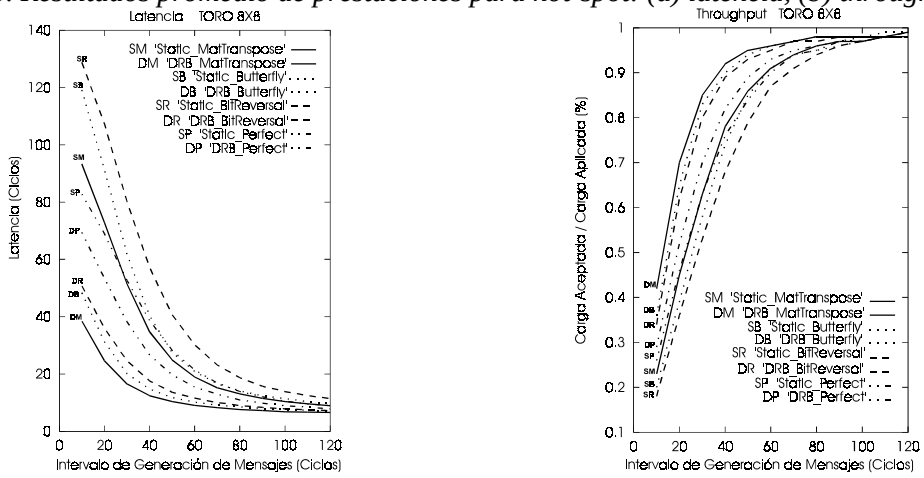
Se puede observar que a cargas bajas (carga >100), DRB se comporta casi igual que el encaminamiento estático. Esto significa que el método que sigue DRB no sobrecarga la red cuando no es necesario. A cargas intermedias (cargas entre 60 y 100) DRB comienza a utilizar dos MSPs, reduciendo la latencia. Con cargas altas (carga<60) utiliza el número máximo de MSPs permitidos, decrementando al máximo la latencia. A medida que la carga se incrementa, las mejoras de latencia se incrementan también, lo que da como resultado reducciones de latencia de hasta un 50 % o más en el punto más alto de carga.

A medida que la latencia baja, el throughput se incrementa como se puede ver en la Fig. 5 (b) para el patrón de hot-spot, en la Fig. 6 (b) y Fig. 7 (b) para los patrones bit-reversal, butterfly, perfect shuffle y matriz transpuesta. El throughput mejora hasta un 50% para DRB mientras que el encaminamiento estático se satura más rápidamente.



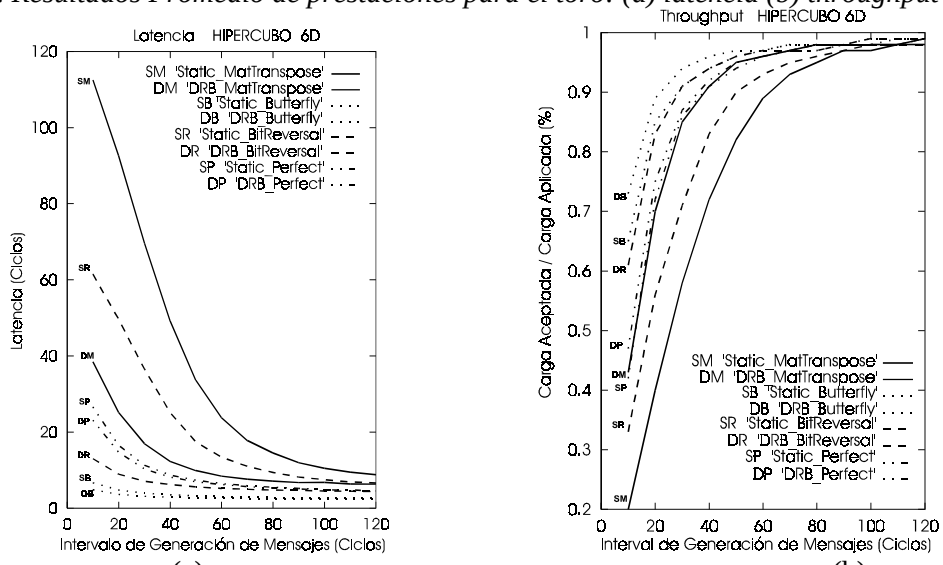
(a) (b)

Fig. 5. Resultados promedio de prestaciones para hot-spot: (a) latencia, (b) throughput..



(a) (b)

Fig. 6. Resultados Promedio de prestaciones para el toro: (a) latencia (b) throughput .

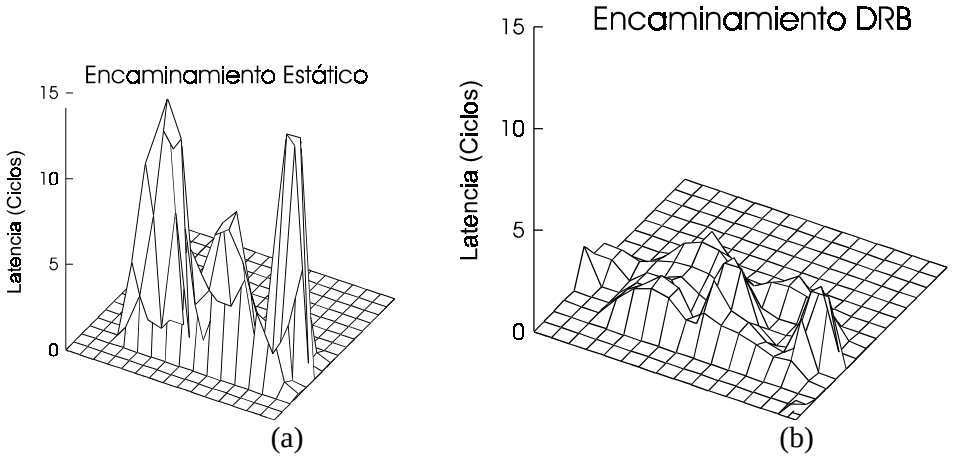


(a) (b)

Fig. 7. Resultados promedio de prestaciones para el toro: (a) latencia, (b) throughput .

En la Fig. 8 se muestran las superficies de latencia para los enlaces de la red utilizando el encaminamiento estático y DRB con el patrón de hot-spot. Se observa claramente como DRB distribuye la carga y elimina los hot-spots. El intervalo de generación de mensajes es de 30 ciclos. Cada punto de la rejilla representa la latencia promedio de los enlaces de cada nodo del

toro. Se puede ver que utilizando el encaminamiento estático (Fig. 8a), grandes hot-spots aparecen en la red, mientras otras regiones están siendo poco usadas. La latencia promedio máxima en los hot-spots es cercana a los 15 ciclos. Cuando utilizamos encaminamiento DRB (Fig. 8b), los hot-spots desaparecen debido a que el exceso de carga en los enlaces ha sido distribuida hacia otros enlaces menos usados. La latencia promedio máxima en este caso es sobre los 3.5 ciclos.



(a) (b)

Fig. 8 Distribución de Latencias para hot-spot (a)Encaminamiento estático (b)DRB

Las conclusiones son que, utilizando DRB, mas mensajes son envíados y con menor latencia. DRB mantiene una distribución uniforme obteniéndose un mejor uso de los recursos de la red y el punto de saturación de la red se alcanza a mayor carga, minimizándose la aparición de hot-spots. Estos resultados se producen porque DRB envía mensajes por caminos nuevos y diferentes que están menos cargados y los utiliza paralelamente.

5. Comparación con otros Métodos

Muchos métodos adaptativos tratan de modificar los caminos existentes cuando un mensaje llega a un área saturada. Este es el caso, por ejemplo, del Chaos Routing [KSn91] que utiliza un algoritmo aleatorio para enviar los mensajes por caminos alternativos. La diferencia con DRB es que éste no actua a nivel individual de cada mensaje, sino que trata de adaptar el fllujo de información entre los nodos fuente y destino de los caminos no saturados.

Los algoritmos de encaminamiento Aleatorio [Val81] [May93] distribuyen uniformemente los requerimientos del ancho de banda sobre toda la máquina, independientemente del patrón de tráfico generado por la aplicación, pero con el costo del alargamiento del camino. Examinando la latencia promedio, se observa que los caminos de longitud máxima no se alargan, en cambio, los caminos de longitud uno se incrementan, en promedio, hasta la distancia media de la red en topologías regulares, por lo que los caminos más cortos son los más afectados. Esto se debe a que el método es "ciego" ya que no toma en cuenta el tráfico actual y distribuye todos los mensajes "a la fuerza" sobre toda la máquina.

Aunque DRB comparte algunos objetivos con el encaminamiento aleatorio, la diferencia es que DRB no sólo trata de mantener el throughput, sino que también mantiene una latencia individual por mensaje controlada porque el alargamiento del camino puede ser limitado. En cambio, en promedio, el encaminamiento aleatorio dobla el alargamiento con el efecto negativo sobre la latencia mencionado anteriormente. Se puede ver que el encaminamiento estático es un caso extremo cuando ambos Supernodos, fuente y destino, contienen solo el nodo fuente o destino, respectivamente, y que el encaminamiento aleatorio es el otro extremo en el que el supernodo fuente contiene todos los nodos de la red de interconexión.

IBM ofrece una solución similar pero más restringida, menos flexible y no adaptable con el algoritmo de encaminamiento RTG (Route Table Generator) utilizado en la SP2, que selecciona estadísticamente cuatro caminos para cada nodo fuente-destino, éstos caminos se usan con una política de "round-robin" con el fin de uniformizar el uso de la red [Sni95]. La CS-2 de Meiko

establece de antemano todos los caminos fuente-destino y selecciona cuatro caminos alternativos para balancear el tráfico de la red [Bok96].

DRB puede ser aplicado a cualquier red directa o indirecta con cualquier topología y control de flujo (Store and Forward, Wormhole, Virtual Cut Through, etc.). Dependiendo de la topología y el control de flujo usado, DRB puede introducir "deadlock" en la red. Se debe utilizar alguna técnica para evitarlo, por ejemplo el método de Duato[DYN97]. Esta técnica asigna canales virtuales adicionales [Dal87] para evitar los ciclos en el grafo de dependencias de canales extendido.

Además, se puede ver que por definición DRB nunca produce alargamientos infinitos del camino (libre de live-lock), y que a ningún nodo se le priva de enviar sus mensajes indefinidamente (libre de starvation). También, el orden de los mensajes se debe preservar y bajo DRB sólo los mensajes que pertenecen al mismo canal lógico tienen que ser ordenados. Esto es fácil de hacer numerándolos. La técnica de Message Pre-fetching puede ser utilizada para ocultar el desordenamiento de los mensajes.

6. Conclusiones y Trabajo Futuro

DRB es un nuevo método para la distribución de mensajes en las redes de interconexión. Ha sido desarrollado con la meta de cumplir los objetivos de diseño de las redes de inteconexión de computadores paralelos. Estos objetivos son una conexión todos-con-todos y una latencia baja y uniforme entre cualquier par de nodos bajo cualquier carga.

La distribución del tráfico se logra definiendo caminos alternativos para enviar mensajes entre cada par fuente/destino. Los caminos alternativos son creados definiendo un conjunto de nodos llamados Supernodos que actuarán como destinos intermedios donde se envían los mensajes primeramente antes de enviarlos a su destino final. Se definen dos supernodos, el primero está centrado en el nodo fuente y el segundo en el nodo destino. Se puede utilizar uno sólo o ambos como destinos intermedios para cada par fuente-destino.

DRB tiene dos componentes. El primer componente es la definición de los Supernodos y el segundo es el encaminamiento DRB.

Se ha explicado la definición del Supernodo y los tipos de Supernodos. El nuevo tipo de Supernodo de Area de Gravedad es más interesante que los definidos por analogías topológicas, porque maximiza el uso de los enlaces para los nodos fuente y destino. DRB ofrece un conjunto de caminos alternativos de donde escoger, dependiendo de los requerimientos entre throughput y latencia.

El segundo componente de DRB es el encaminamiento DRB para seleccionar Supernodos específicos a cada par fuente-destino. DRB monitoriza la carga y configura dinámicamente los parámetros del Supernodo dependiendo de los requerimientos actuales de carga de mensaje en la red. El método no pierde recursos significativos de cómputo o comunicaciones porque están totalmente distribuidos, y el overhead de monitorización y decisión es linealmente dependiente del número de mensajes en la red.

En la evaluación hecha para validar DRB se han encontrado grandes mejoras en la latencia y la eliminación de los hot-spots de la red. DRB es útil para patrones de comunicaciones repetitivos que son los que producen las peores situaciones de hot-spot.

La latencia se reduce hasta un 50% y el throughput se incrementa en la misma medida. Los overheads son mínimos porque a baja carga las prestaciones no se reducen.

Con respecto a nuevas evaluaciones del método, hemos diseñado una extensión de DRB para trabajar en dos niveles de manera que se creen los caminos alternativos para balancear el tráfico de la red. El primer nivel es el método aquí presentado, el cual es un nivel inter-nodo que cambia los caminos de los mensajes sin mover las posiciones de los fuentes y destinos, y el segundo es un nivel intra-nodo que migra procesos. El nivel intranodo se necesita en el siguiente caso: supóngase que un nodo tiene un mayor requerimiento de ancho de banda para enviar o recibir que lo que pueden aceptar sus propios enlaces. La distribución de los caminos no puede mejorar este caso, entonces, los procesos fuente o destino deben ser movidos.

Actualmente estamos implementando y haciendo simulaciones para probar esta nueva característica y sus overheads.

Bibliografía

[Aga91] Agarwal A. ”Limits on Interconnection Network Performance”. IEEE Transactions on Parallel and Distributed Systems, Vol. 2, N. 4, Oct 1991, pp.398-412.

[Bei92] R. Beivide. E. Herrada. J.L. Balcázar y A. Arruabarrena. “Optimal Distance Networks of Low Degree for Parallel Computers”. IEEE Trans. on Computers. Vol. 40. N. 10.. Oct 1992, pp. 1109-1124.

[Bok96] Bokhari S. “Multiphase Complete Exchange on Paragon, SP2 and CS2”. IEEE Parallel and Distributed Technology, Vol.4, N.3, Fall 1996, pp. 45-49

[CK92] Chien AA, Kim JH, “Planar Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors”. Proc. of the 19th Symposium on Computer Arch. May 1992, pp. 268-277

[Dal87] Dally WJ. Seitz CL. "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks" IEEE Trans. On Comp. Vol. C-36. N. 5, May 1987. pp 547-553.

[Dal90] Dally WJ. “Performance analysis of k-ary n-cube interconnection networks”. IEEE Trans. On Comput. Vol. 39. Jun. 1990, pp. 775-785.

[Dal92] Dally WJ. “Virtual-Channel Flow Control”. IEEE Transactions on Parallel and Distributed Systems, Vol. 3, N. 2, Mar 1992, pp. 194-205.

[DeP95] De Prycker M, “Asynchronous Transfer Mode, Solutions for Broadband ISDN“, Prentice-Hall, 1995 (3rd Ed.)

[Dua93] Duato J. “A new theory of Dead-lock free adaptive routing in wormhole networks” IEEE Transactions on Parallel and Distributed Systems, 4(12), Dec 1993, pp.1320-1331

[DYN97] Duato J, Yalamanchili S, Ni L. “Interconnection Networks, an Engineering Approach”. IEEE Computer Society Press. 1997.

[Gar97] Garces I, Franco D, Luque E. “Improving Parallel Computer Communication: Dynamic Routing Balancing”. Proceedings of the Sixth Euromicro Workshop on Parallel and Distributed Processing. (IEEE-Euromicro) PDP98. Madrid. Spain. January 21-23, 1998. pps. 111-119

[KLC94] KIM J, Liu Z, Chien A. “Comprensionless Routing: A Framework for Adaptive and Fault-Tolerant Routing”. Proc. of the 21st International Sysmposium onComputer Architecture, Apr 1994, pp.289-300

[Ksn91] Konstanyinidou S, Snyder L. “Chaos Router: Architecure and Performance”. Proc. of the 18th International Symposium on Computer Architecture, May 1991, pp.212-221

[May93] May MD, Thompson PW, PH Welch Eds. “Networks, Routers and Transputers: Function, Performance and application”. IOS Press 1993

[Myr98] Myricom Home Page, http://www.myri.com

[NG92] Ni L, Glass C. “The Turn model for Adaptive Routing”. Proc. of the 19th International Symposium on Computer Architecture, IEEE Computer Society, May 1992, pp. 278-287

[Pfi85] Pfister GF. Norton A. "Hot-Spot Contention and Combining in Multistage Interconnection Networks". IEEE Trans. On Computers. Vol. 34. N.10 Oct 1985, pp.943-948.

[PG+94] Pifarre GD, Gravano L, Felperin SA, Sanz JLC “Fully adaptive Minimal Deadlock Free Packet Routing in Hypercubes, Meshes and Other Networks: Algorithms and Simulations” IEEE Transactions on Parallel and Distributed Systems, V. 5, N.3, Mar 1994, pp. 247-263.

[Sni95] Snir M, Hochschild P, Frye DD, Gildea KJ ”The communication software and parallel environment of the IBM SP2”. IBM Systems Journal. Vol.34, N.2, pp. 205-221.

[Val81] Valiant LG. Brebner GJ. "Universal Schemes for Parallel Communication". ACM STOC. Milwaukee 1981. pp. 263-277