UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DISCA

Departament d'Informàtica de Sistemes i Computadors
Universitat Politècnica de València

Estudio de la viabilidad de una red híbrida foto-eléctrica

TRABAJO FIN DE MASTER

Máster en Ingeniería de Computadores

Autor

Jose Puche Lara

Directores

Prof. Salvador Petit
Prof. María Engracia Gómez Requena
Prof. Julio Sahuquillo Borrás

September 4, 2015

Resumen

Las altas escalas de integración disponibles actualmente en la fabricación de microprocesadores hacen posible incluir cientos de núcleos de procesamiento dentro del mismo chip. Para reducir la contención en el acceso a memoria principal, estos procesadores *manycore* incluyen varios controladores de memoria que son accesibles desde cualquier núcleo. Para permitir la comunicación entre los diferentes núcleos, así como entre éstos y los controladores de memoria, los manycores suelen utilizar una red de interconexión eléctrica conocida como NoC (del inglés *Network on Chip*).

En este contexto, la red dentro del chip es un elemento fundamental ya que puede incrementar significativamente la latencia de acceso a memoria por dos razones principales. Por un lado, la latencia de un acceso a memoria principal depende, entre otros, de la distancia (es decir, del número de saltos) que el acceso recorre por la red desde el núcleo de ejecución hasta el controlador correspondiente. Por otro, dependiendo de la combinación de aplicaciones que se encuentre en ejecución y de la distribución de éstas en los núcleos, los accesos pueden encontrar una alta contención tanto en la red como en los controladores de memoria. Estos factores pueden potencialmente reducir la escalabilidad de la red.

Una de las soluciones a estos problemas de escalabilidad es el uso de tecnologías alternativas, como la fotónica, en el diseño de la red. En este trabajo se propone el diseño de una red híbrida que combine las tecnologías eléctrica y fotónica. La red eléctrica se mantiene para su uso en distancias cortas, mientras que la red óptica establece comunicaciones directas nodo-controlador, reduciendo significativamente las altas latencias en el acceso a memoria.

Como paso previo al diseño, se realiza un estudio de exploración sobre el impacto de la distancia que separa las aplicaciones del controlador de memoria en las prestaciones de las mismas. El estudio también analiza el tráfico adicional generado por la prebúsqueda hardware y la contención producida por aplicaciones ejecutándose en nodos vecinos del manycore. Estos resultados se han considerado para diseñar el mecanismo de selección de red (híbrida o eléctrica) en tiempo de ejecución. Los resultados obtenidos muestran que la introducción de la red híbrida combinada con una técnica de conmutación eficiente permite reducir la degradación de prestaciones de las aplicaciones hasta un 16 % respecto al peor de los casos.

Palabras Clave: Mecanismos de Prebúsqueda, Redes nanofotónicas, Contención de Memoria, Procesadores Multinúcleo, Redes en Chip

Abstract

The scale of integration of current microprocessor manufacturing processes allows including hundred of cores on a single chip. To reduce main memory access contention, these processors (referred to as manycores) use to implement multiple memory controllers that can be accessed from the distinct cores in the chip. A Network on Chip (NoC) is implemented to enable communication among cores and between cores and memory controllers.

In this context, the NoC becomes a key component that can heavily increase memory access latency due to two main reasons. On one hand, memory access latency depends basically on the distance (i.e. the number of hops) that a memory request has to travel from the requesting core to the corresponding memory controller across the NoC. On the other hand, requests can experience network and memory contention depending on the number of running applications, their memory access patterns and the cores they are allocated to. These factors can potentially reduce network scalability.

A possible solution to overcome these problems relies on the use of alternative network technologies. One example of these emerging technologies is the nanophotonic technology. On this work we propose a hybrid network design that combines electrical and optical technologies. This way, electrical network is used to perform transactions over short distances, while the optical one communicates cores to memory controllers, significantly reducing memory latencies.

First, we study the impact of the distance between memory controllers and the core where applications are executed on the applications' performance. This study also analyzes the additional traffic generated by hardware prefetchers as well as the contention produced by applications co-running on the CMP. The obtained results have been considered in the design of the network (electric or photonic) selection mechanism that applies at run time. Results achieved on the evaluation of our proposal show that the hybrid network jointly with an efficient NoC switching mechanism reduces the performance degradation of the studied applications up to a 16% compared to the worst case.

Keywords: Prefetch Mchanisms, Nanophotonic Network, Memory Hierarchy Contention, Manycore Processors, Networks On Chip

Índice general

Índice de figuras

Índice de tablas

Acronyms

CMOS Complementary Metal Oxide Semiconductor. 17, 18

CMP Chip MultiProcessor. 8–10, 12–14, 16, 56, 63

DOR Dimension Ordered. 14

DWDM Dense Wavelength Division Multiplexing. 17, 19, 21

IPC Instrucciones Por Ciclo. 46, 49, 52, 54, 56, 57, 63

LLC Last Level Cache. 8, 9, 46, 49

MPKI Misses Per KiloInstruction. 46, 49, 50, 52, 60, 64

MWMR Multiple Writer Multiple Reader. 21

MWSR Multiple Writer Single Reader. 20, 23, 24

NoC Network On Chip. 56

SAF Store and Forward. 14, 16, 31, 44, 46, 47, 49, 51, 54, 56, 57, 60–62, 64

SWBR Single Writer Broadcast Reader. 19, 22

SWMR Single Writer Multiple Reader. 20–22, 24

TFM Trabajo Fin de Máster. 8, 12, 16, 44

VCT Virtual Cut Through. 14, 15, 44, 46, 47, 51, 52, 54–62, 64

WDM Wavelength Division Multiplexing. 17, 18

Capítulo 1

Introducción

Este capítulo presenta la motivación de la investigación desarrollada en este Trabajo Fin de Máster (TFM). Para ello, se introducen las ventajas y los problemas con los que se encuentan las redes en chip eléctricas tradicionales. A continuación, se propone la tecnología nanofotónica como una alternativa que puede mitigar los problemas de escalabilidad de estas redes, mejorando las prestaciones y el consumo de las arquitecturas de los Chip MultiProcessor (CMP) actuales. Finalmente, se presenta un resumen de las aportaciones de este TFM.

1.1 Limitaciones de las redes en chip tradicionales

Con el objetivo de satisfacer los requisitos de escalabilidad de la Ley de Moore, las últimas generaciones de la mayoría de microprocesadores han adoptado una arquitectura multinúcleo, también denominada multiprocesador en un chip o CMP. La arquitectura multinúcleo más común se denomina *tiled*, en la que el procesador contiene múltiples *tiles* idénticos. Cada uno de estos tiles está compuesto por un núcleo de ejecución, su caché de L1 privada correspondiente y un banco o fragmento de L2 que puede ser o bien privada o bien compartida entre los diferentes núcleos. Recientemente la caché de L2 tiende a ser privada y los tiles incorporan un banco de caché L3 que suele ejercer de último nivel de caché o Last Level Cache (LLC). Además, cada tile cuenta con un interfaz de red por el que accede a la red que lo comunica con el resto de tiles y componentes del chip.

En este tipo de arquitecturas, por tanto, cuando un núcleo necesita acceder a una parte de la caché que no se encuentra en su propio tile debe utilizar la red de interconexión para llegar hasta ella. La misma situación tiene lugar cuando el acceso es a los controladores de memoria, que se encuentran típicamente ubicados en los extremos del chip.

Por otro lado, el paradigma CMP resulta escalable y beneficioso al ejecutar

aplicaciones multihilo. Sin embargo, estas aplicaciones requieren unos mecanismos de comunicación y sincronización eficientes entre los diferentes hilos dentro del chip. Todo ello, unido a lo expuesto anteriormente, hace del diseño de una red en el chip eficiente un aspecto clave tanto en el rendimiento como en el consumo del CMP.

En lo que respecta a la necesidad de comunicaciones globales eficientes dentro del chip, las redes eléctricas tradicionales presentan buenas prestaciones para un número reducido de nodos. Además, en estos casos, estas prestaciones se corresponden con un nivel razonable de consumo de energía. Sin embargo, diversos aspectos intrínsecos de este tipo de redes comprometen la escalabilidad de los futuros procesadores *manycore*. Con el aumento del número de núcleos se incrementa la distancia recorrida en la red para acceder a los datos; esto provoca el aumento de la latencia y de la contención degradando la productividad de la red, así como incrementando su consumo.

Pese a todas estas razones, las redes en chip convencionales son, actualmente, las más rápidas cuando se trata de comunicar nodos a una distancia relativamente corta. El estudio realizado en este trabajo está enfocado a cubrir las posibles carencias que manifiestan este tipo de redes cuando el número de núcleos crece por encima de varias decenas.

1.2 La tecnología nanofotónica

En los últimos años, los avances en la fabricación sobre silicio de tecnología fotónica han permitido la integración de interconexiones ópticas en los microprocesadores. Esta tecnología promete introducir mejoras en las tres cualidades deseables de toda tecnología de interconexión: alto ancho de banda, alta eficiencia energética y baja latencia. Por otro lado, su capacidad para realizar transmisiones de datos a lo largo del chip con una latencia independiente de la distancia supone una solución a los problemas de escalabilidad de las redes en chip tradicionales.

Debido a estas características, recientemente ha habido una serie de propuestas de redes nanofotónicas con diferentes topologías y para diferentes arquitecturas. La complejidad de estas propuestas es variada y oscila desde simples anillos [1,2] fotónicos hasta topologías más complejas que intentan actuar como *fat trees*, mallas o toros tradicionales [3,4]. Sin embargo, el uso de interconexiones fotónicas complejas puede limitar los beneficios en latencia o consumo energético debido a la cantidad de recursos ópticos necesarios para soportar los requisitos de este tipo de redes.

Otro aspecto clave se encuentra en que la capacidad de este tipo de redes para realizar tareas de encaminamiento dentro del chip es limitada. Si bien en otros contextos como *exascale computing* el encaminamiento es factible, la introducción de

algoritmos de encaminamiento complejos y eficientes en switches ópticos es actualmente un reto. Esta limitación puede introducir complicaciones, por ejemplo, en CMPs que utilicen mecanismos de coherencia por hardware.

Además, el uso de redes nanofotónicas también se ve afectado por las limitaciones de la transmisión de la información entre tecnologías. Puesto que el procesador trabaja con tecnología eléctrica, es necesario para el envío de información realizar la conversión de la señal eléctrica en óptica. Asimismo, para la recepción es necesario volver a convertir la señal óptica en eléctrica. Un diseño adecuado de una red nanofotónica dentro del chip debe controlar el consumo y latencia de estas conversiones de manera que estos se mantengan en unos límites adecuados.

1.3 Aportaciones de este Trabajo Fin de Máster

Teniendo en cuenta las ventajas e inconvenientes de las redes identificadas en los apartados anteriores, en este Trabajo Fin de Máster se propone una nueva red híbrida que aune las ventajas de ambos tipos de tecnologías de red. En primer lugar, dada la velocidad y eficacia de las redes convencionales en distancias cortas, se propone la utilización de una malla eléctrica que interconecta todos los tiles del CMP así como los controladores de memoria. Pero además, se introduce un anillo óptico que proporcione baja latencia y evite posibles contenciones cuando las transmisiones se realizan entre componentes lejanos en la red.

Así, las aportaciones de este Trabajo Fin de Máster son las siguientes:

- Estudio de la escalabilidad de prestaciones de una red eléctrica tradicional utilizando la suite de benchmarks SPEC2006.

- Categorización de las diferentes aplicaciones en función del impacto que presenta en sus prestaciones la distancia que separa el núcleo de cómputo donde se ejecutan del controlador de memoria.

- Estudio de la influencia de técnicas de prebúsqueda agresiva en la degradación ocasionada por la distancia hasta el controlador.

- Estudio del impacto de la contención junto con la distancia hasta el controlador en las prestaciones de las aplicaciones.

- Propuesta y evaluación de una red híbrida fotoeléctrica que selecciona mediante un modelo teórico la red más rápida para realizar una petición a memoria desde un núcleo determinado.

1.4 Estructura de este Trabajo Fin de Máster

El capítulo 2 introduce el campo de las redes en chip, el principal tema de investigación de este trabajo. El capítulo 3 expone y explica diferentes trabajos y artículos previos que guardan relación con el sistema propuesto. El capítulo 4 presenta la propuesta de red híbrida desarrollada en este trabajo. El capitulo 5 introduce el entorno de simulación sobre el que se ha implementado nuestra propuesta. En este capítulo se incluyen también las ampliaciones que se han realizado sobre el código del simulador Multi2Sim y que han hecho posible la simulación de la tecnología nanofotónica. El capítulo 6 muestra y discutee los resultados obtenidos en la evaluación de las propuesta. El capítulo 7 recoge las principales conclusiones de este trabajo y presenta posibles trabajos futuros relacionados con el tema tratado.

Capítulo 2

Redes en chip

En este capítulo se introducen las redes en chip, el principal tema de investigación de este TFM. En primer lugar, se introducen las redes en chip eléctricas utilizadas actualmente en la gran mayoría de CMPs comerciales. Se presentan sus principales virtudes y defectos, así como posibles limitaciones que éstas pueden presentar en el futuro. A continuación se exponen las redes en chip nanofotónicas, enumerando sus principales características y componentes. Además, se presenta un análisis acerca de la viabilidad de este tipo de redes y su coste de implementación sobre silicio en la actualidad.

2.1 Redes eléctricas en chip

A lo largo de la última década, los multiprocesadores en chip o CMPs han dominado el mercado de los microprocesadores. Los esfuerzos realizados anteriormente en conseguir incrementos de prestaciones aumentando la complejidad de los procesadores superescalares han encontrado límites en términos de área y consumo que no pueden ser pasados por alto. Como resultado, las arquitecturas multinúcleo pasan a ejercer un papel clave para conseguir un mayor rendimiento en los procesadores sin incurrir en crecimientos de consumo prohibitivos. Así, uno de los princiaples paradigmas actuales se basa en utilizar hasta cientos de núcleos de arquitecturas relativamente simples y consumo limitado en lugar de un sólo núcleo excesivamente complejo.

El diseño de este tipo de procesadores está basado en la réplica de bloques idénticos denominados *tiles* e interconectados por la red. Un tile típicamente está formado por el núcleo de procesamiento, varios niveles de caché privados y/o compartidos y la interfaz con la red de interconexión. La Figura 2.1 puede observarse un diseño de alto nivel de un CMP de 64 nodos interconectados por una malla bidimensional. En el ejemplo, cada tile consta, además del núcleo y del rou-

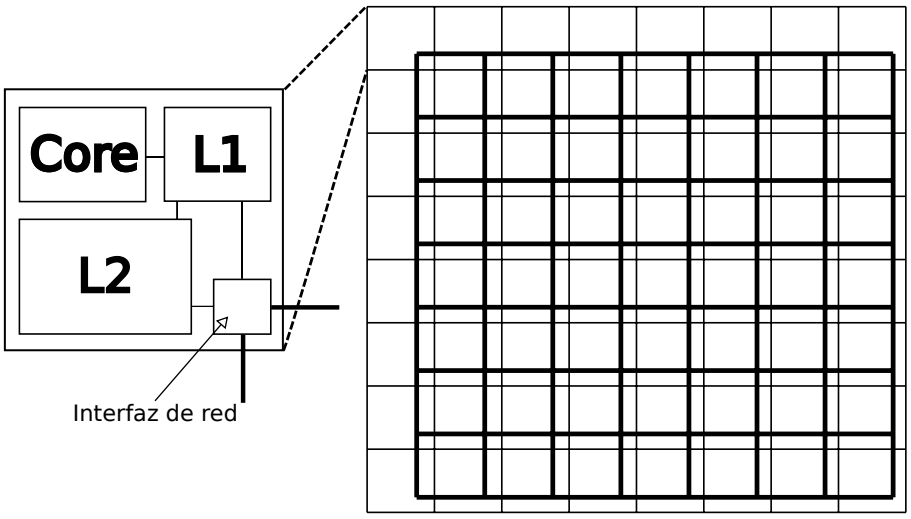


Figura 2.1: Ejemplo de diseño por tiles de un chip multinúcleo.

ter de la red, de una caché L1 y una caché L2. Observando esta figura, se aprecia claramente cómo el diseño puede ser ampliado fácilmente simplemente replicando el tile.

En este nuevo paradigma, las redes de interconexión *on-chip* juegan un papel fundamental en el rendimiento de estos procesadores. Las redes en chip permiten además que las prestaciones se incrementen de un modo escalable, ya que éstas se encuentran asociadas a aumentar el número de núcleos (*i.e.* el nivel de paralelismo) con los que cuenta el CMP.

2.1.1 Aspectos de diseño de una red dentro del chip

Las redes en chip heredan la mayoría de las técnicas y mecanismos ya diseñados para el campo de las redes de altas prestaciones. Sin embargo, a pesar de las similitudes, existen ciertas limitaciones que obligan a establecer diferencias entre ambos contextos. Estas limitaciones están relacionadas con las altas escalas de integración *on-chip* y provocan que las redes en chip se encuentren mucho más expuestas a efectos físicos que otro tipo de redes no llegan a experimentar. Aspectos como el tamaño de los búferes, la ubicación y longitud de los enlaces (que está directamente relacionada con la elección topología) o el área de los switches utilizados son puntos clave en el diseño de una red en el chip eficiente.

A lo largo de este apartado se presentan los principales aspectos a tener en cuenta en el diseño de una red en el chip. Además, se discuten las opciones contempladas y seleccionadas en el diseño del sistema base utilizado en este trabajo.

Topologías: La topología constituye un aspecto de diseño clave en el rendimiento y coste de cualquier red dentro del chip. La industria de multiprocesadores tiende a utilizar topologías de malla en las redes dentro del chip. Este tipo de topologías es actualmente el más utilizado gracias a que se ajusta a la superficie bidimensional del silicio. Esto permite una fabricación en serie sencilla de *tiled*-CMPs a la vez que ofrece unas prestaciones aceptables en términos de ancho de banda y latencia media. Por este motivo, esta topología de red será utilizada como base en los estudios y análisis realizados.

Algoritmos de encaminamiento: La elección de una determinada topología ofrece múltiples rutas de comunicación entre los nodos fuente y destino. Sin embargo, se debe realizar una correcta selección entre las diferentes rutas disponibles que permita un rendimiento sostenido de la red a la vez que evite situaciones de bloqueo y colapso en la misma. Es en este punto donde el algoritmo de encaminamiento establecido en cada uno de los *switches* de la red adquiere importancia. Durante la realización de este trabajo, el algoritmo de encaminamiento utilizado ha sido el algoritmo bien conocido XY o Dimension Ordered (DOR). En una malla 2D que utiliza este algoritmo, los mensajes alcanzan primero la coordenada X del destino y posteriormente circulan sobre la coordenada Y del mismo. Se trata de un algoritmo determinista que garantiza la ausencia de interbloqueos en la red y cuya implementación sobre una malla requiere un coste mínimo.

Técnicas de conmutación: Las técnicas de conmutación determinan cómo los paquetes circulan por la red y qué recursos (como búferes o enlaces) utilizan a lo largo del tiempo. El principal objetivo de estas técnicas es proporcionar un uso equilibrado de los recursos de la red y evitar contenciones innecesarias en paquetes listos para continuar su ruta. Las tres técnicas de conmutación bien conocidas son: Store and Forward (SAF), Virtual Cut Through (VCT) y *wormhole*. En SAF, cada nodo debe esperar la recepción del paquete completo antes de comenzar el reenvío del mismo al siguiente nodo. Obsérvese que este esquema supone un incremento lineal de la latencia respecto a la distancia del origen al destino incluso en ausencia de contención en la red. Con VCT, un paquete puede comenzar a ser reenviado tan pronto como es recibida la cabecera del mismo. Este esquema permite solapar en el tiempo la recepción del paquete con el reenvío al nodo siguiente. Como alternativa a VCT, *wormhole* controla el flujo de paquetes a nivel de flit en lugar de a nivel de paquete. Este cambio en el esquema permite utilizar búferes de tamaño más reducido pero como contrapartida puede presentar situaciones complejas en las que los mensajes quedan bloqueados en varios switches de la red simultáneamente. *Wormhole* aparece debido a la necesidad de reducir el área consumida por los switches dentro del chip; sin embargo, dado que es-

te aspecto no es relevante en el presente trabajo y *wormhole* ofrece prestaciones similares a VCT, no se han realizado pruebas con esta técnica.

Técnicas de control de flujo: Para garantizar la ausencia de pérdidas de paquetes en la red (i.e. descarte de paquetes en presencia de bloqueo) el mecanismo de conmutación debe cooperar con técnicas de control de flujo a nivel de enlace. Estas técnicas proporcionan un modo de controlar el flujo de datos entre el emisor y el receptor de manera que el switch receptor no se ve obligado a descartar paquetes por falta de espacio en sus búferes. Las dos técnicas principales de control de flujo utilizadas son el control por créditos y la señalización *stop&go*. La primera de ellas mantiene una cuenta del número de *slots* disponibles en el búfer del receptor que se actualiza según la entrada o salida de nuevos flits en el búfer. El número de créditos se transmite a los switches anteriores y de esta forma el emisor cuenta con información acerca del estado del switch receptor. Por otro lado, la técnica *stop&go* envía una de estas dos señales cuando corresponde para permitir o impedir el envío de nuevos paquetes al receptor.

La toma de decisiones adecuadas respecto a estos componentes permite obtener diseños de redes en chip eficientes que ofrecen buenas prestaciones a costes aceptables.

Sin embargo, la red en el chip no está exenta de problemas en términos de escalabilidad. Conforme el número de nodos de la red crece, esta se puede ver expuesta a problemas de contención, cuellos de botella o variaciones de latencia. Además, estos problemas se acentúan cuando se trabaja con las topologías de malla 2D tan habituales dentro del campo de las redes en chip.

En las redes con topología de malla bidimiensional existen varios aspectos que se ven comprometidos conforme crece el número de nodos. Uno de los aspectos principales se encuentra en el aumento de colisiones que se produce entre los mensajes en la red. Debido a que estos pasan más tiempo dentro de la misma, son más propensos a generar contención, lo que afecta a la productividad de la red. Además, al incrementar el número de nodos, la distancia media de la red no escala linealmente, lo que conlleva que la latencia de las transmisiones crece incluso en ausencia de contención.

Veamos un ejemplo para ilustrar este problema. Supongamos una malla bidimensional de 64 nodos distribuidos en 8 filas y 8 columnas. En el primer y último nodo se encuentran conectados los controladores de memoria a los que deberán acceder los 64 nodos de la red según corresponda. Además, la red utiliza switches segmentados de 3 etapas y 16 bytes/ciclo de ancho de banda en sus enlaces. Así, si el nodo 0 pretende almacenar un paquete de 72 bytes (formado por 64 bytes de tamaño de un bloque de caché más 8 bytes de cabecera) en el controlador ubica-

do junto al nodo 63, este paquete deberá realizar un total de 14 saltos por la red. La traducción en ciclos correspondiente con ese número de saltos depende de la técnica de conmutación utilizada por la red. En el peor de los casos la red utilizará conmutación SAF, lo que supondría que en cada nodo el paquete debe ser almacenado y serializado por completo hasta comenzar su reenvío. Así, el número de ciclos de latencia en este caso sería $14 \times (3 + \frac{72}{16}) = 112$ ciclos.

El incremento en la latencia de las comunicaciones puede suponer una degradación de prestaciones del sistema completo cuando éste ralentiza el servicio de las peticiones de memoria. Por tanto, para evitar esta degradación en las prestaciones se debe investigar en mecanismos que permitan paliar las dificultades de escalabilidad de la red en el chip. En este TFM se realiza un estudio de exploración que cuantifica los problemas ya identificados y se propone una solución basada en la tecnología nanofotónica.

2.2 Redes ópticas en chip

Las redes ópticas han sido tradicionalmente utilizadas en áreas metropolitanas debido a que la tecnología fotónica es muy poco dependiente a la distancia entre los elementos que establecen comunicación. Esta latencia independiente de la distancia, unida a la mayor eficiencia energética que presentan estas redes frente a las redes tradicionales, hacen de ellas un componente que potencialmente puede mejorar las prestaciones y consumo del CMP.

Gracias a los avances en tecnología nanofotónica sobre el silicio, en la actualidad esta tecnología se plantea como una alternativa real a las tradicionales redes eléctricas dentro del chip [5]. Las redes ópticas cuentan con la capacidad de transmitir varios flujos de información simultáneamente a altas velocidades, lo que se traduce en una reducción significativa del número de cables necesarios para llevar a cabo la comunicación entre múltiples emisores y receptores.

2.2.1 Componentes de una red óptica

Para conseguir establecer una comunicación óptica completa entre dos componentes dentro del chip se necesita de la integración de varios elementos propios de este tipo de redes. A continuación se exponen los principales dispositivos que se requieren para integrar una red fotónica completamente operativa dentro del chip.

- **Láser:** Los láseres son los encargados de introducir el haz de luz en el chip. Este componente puede ubicarse dentro o fuera del chip, aunque típicamente son emplazados en el exterior del encapsulado. Esto se debe a que el consumo y el área que requieren los láseres on-chip son mucho mayores y supone por tanto un desperdicio de recursos para el resto de componentes.

- ***Waveguides*:** Los waveguides se acoplan a los láseres para transportar la señal luminosa. El proceso de fabricación de estos componentes es de vital importancia para evitar pérdidas de señal significativas (y por tanto desperdicio de potencia) cuando se introducen cambios de dirección y giros en el waveguide. Estos elementos combinan dos materiales de alto y bajo índice de refracción en su parte interna y externa respectivamente, confinando el haz luminoso y guiándolo hasta su destino. La elección de estos materiales así como de un proceso de fabricación compatible con CMOS son cruciales para evitar que los waveguides presenten pérdidas de energía excesivas.

  Por otro lado, una de las mayores ventajas de las redes fotónicas radica en que la señal óptica se puede multiplexar en un rango finito de longitudes de onda o *wavelengths*. Este proceso se conoce como Multiplexado por División de Longitudes de Onda o Wavelength Division Multiplexing (WDM). Cuando el número de wavelengths en el que se divide la señal es elevado (típicamente hasta 64 longitudes de onda) este proceso recibe el nombre de Dense Wavelength Division Multiplexing (DWDM). Esta característica es la que dota a las redes ópticas de una densidad de ancho de banda elevada.

- **Anillo resonador:** Un anillo resonador es un componente formado por una waveguide que toma la forma de una circunferencia de diámetro reducido (entre 3 y 5 μm) [6]. Los resonadores son componentes ópticos que por defecto sólo reaccionan a una determinada longitud de onda, determinada por su diametro. Sin embargo, los resonadores pueden calibrarse para reaccionar a diferentes longitudes de onda alterando la temperatura mediante la aplicación de corriente. Por tanto, cuando se usa WDM o DWDM se deben calibrar los resonadores involucrados en una comunicación óptica. Los anillos resonadores son el elemento base de dos componentes conocidos como moduladores y detectores:

  - **Modulador:** Un modulador o transmisor es un anillo resonador que se encarga de imprimir la señal digital en la luz extraída por el resonador y que circula por el waveguide. El material utilizado en los moduladores es el Germanio, ya que puede ser utilizado en proceso CMOS y presenta un alto porcentaje de absorción luminosa. Generalmente, un modulador es simplemente un resonador que absorbe señal eléctrica y la modula en un determinado wavelength λ_i.
  - **Detector:** Un detector o receptor es un anillo resonador que se encuentra calibrado a una determinada longitud de onda. Actúa como filtro de una wavelength λ_i y dirige los haces de luz extraídos a un fotodetector. Obsérvese que en la recepción se requieren tantos detec-

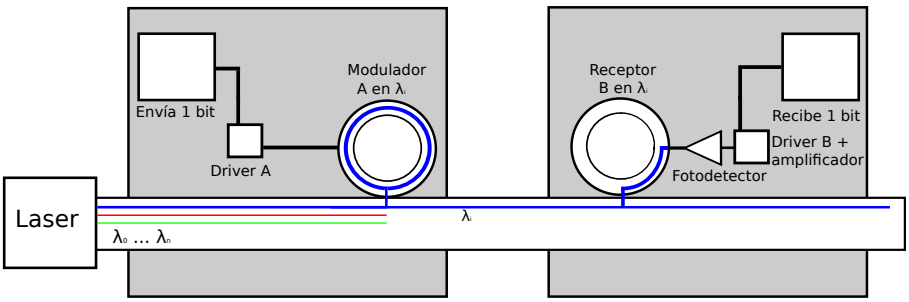


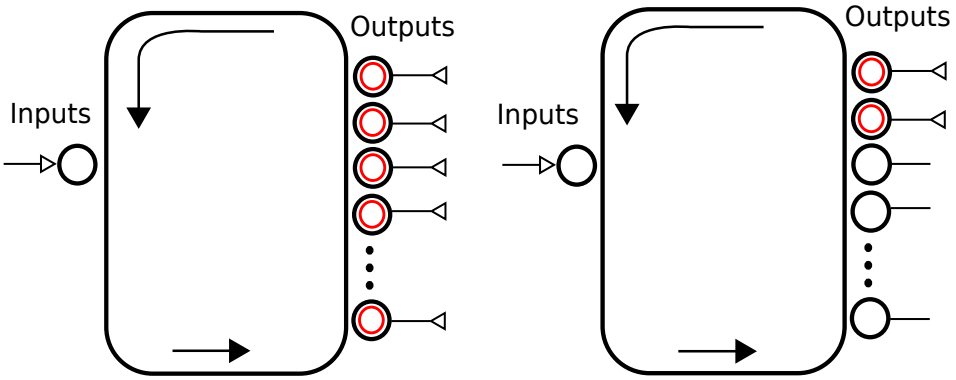
Figura 2.2: Ejemplo de comunicación mediante un enlace óptico.

tores como longitudes de onda esten asociadas a una comunicación óptica determinada.

- **Fotodetector:** Tras filtrar las longitudes de onda correspondientes a un destino, se les debe aplicar una operación de conversión óptico-eléctrica. Esta labor es realizada por el fotodetector, componente que extrae los fotones de la señal óptica y los transforma en corriente eléctrica. El fotodetector debe tener acoplado un amplificador de señal eléctrica para que la salida producida pueda ser tratada correctamente.

En la Figura 2.2 se puede observar un ejemplo de comunicación que emplea todos los componentes descritos anteriormente. La figura muestra cómo se realiza la transmisión de un flujo de bits entre un emisor y un receptor, utilizando un láser y un waveguide que permite WDM. En primer lugar, el modulador del emisor recibe la orden de transmitir un determinado flujo de bits. Para ello, el modulador A que opera a una determinada λ_i comienza a codificar y modular dicho flujo en la longitud de onda i. Obsérvese que las posibles comunicaciones que usan el resto de longitudes de onda no se ven afectadas por este proceso.

Posteriormente, la señal luminosa modulada que circula por el waveguide pasa por el modulador B que debe ser previamente calibrado para resonar en la longitud de onda λ_i. De esta manera el resonador reaccionará al paso de la luz y la filtrará, realizando ésta un movimiento circular en el interior del anillo. A continuación, el fotodetector acoplado al anillo convierte la luz que circula en el interior del receptor. Finalmente, la señal eléctrica obtenida pasa a ser manejada por el driver B que almacena los valores correspondientes en los biestables del receptor.



(a) Esquema SWBR (b) Esquema SWMR

Figura 2.3: Esquemas de comunicación single writer.

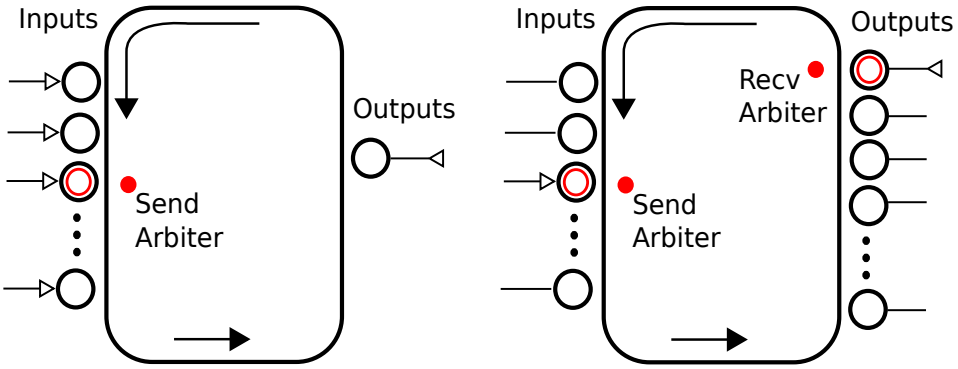
2.2.2 Esquemas de comunicación y asignación de *wavelengths*

Pese a que la red óptica puede operar a frecuencias de hasta 10 GHz, ésta solo puede transmitir un bit en cada wavelength por ciclo. Esto significa que el número de wavelengths que se asignan a cada nodo es crítico, ya que repercute en el ancho de banda agregado con el que dicho nodo contará para sus transmisiones y recepciones.

Para la comunicación entre los diferentes nodos utilizando un único canal óptico existen numerosas propuestas en la literatura. Los esquemas que se explican a continuación permiten interconectar diferentes entradas con diferentes salidas mediante el acceso a un canal de comunicación compartido. Más adelante, mediante la utilización de DWDM, el número de wavelengths asociadas a cada nodo se incrementará y, por tanto, estos esquemas podrán realizar envíos simultáneos en diferentes longitudes de onda.

- **Single Writer Broadcast Reader (SWBR)**: En este esquema se realiza una difusión entre todos los receptores de una señal que ha introducido el emisor en una determinada longitud de onda. Se trata de un esquema poco habitual ya que requiere del calibrado de los moduladores de todos los receptores cada vez que se realiza una transmisión, lo que supone un desperdicio de energía en la red. En la Figura 2.3a puede apreciarse un diagrama sobre la comunicación en este esquema. Obsérvese cómo todos los anillos correspondientes a las salidas se encuentran calibrados para recibir el mensaje.

- **Single Writer Multiple Reader (SWMR)**: En este esquema los moduladores de los receptores se encuentran sin calibrar, por lo que por defecto ninguno de ellos extrae señal alguna del waveguide. Así, cuando el emisor



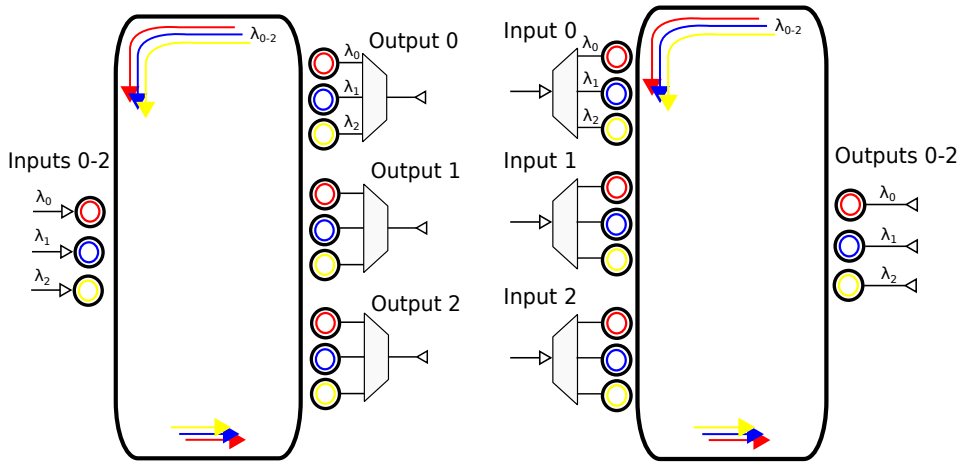
(a) Esquema MWSR (b) Esquema MWMR

Figura 2.4: Esquemas de comunicación multiple writer.

quiere enviar un paquete a un receptor, éste primero ha de comprobar que su anillo resonador se encuentra activo en la wavelength correspondiente. Esto requiere por tanto una lógica adicional eléctrica u óptica que permita al emisor activar y calibrar el filtro del receptor cuando corresponda. En la Figura 2.3b se puede observar un diagrama de comunicación de este esquema. Se trata de un esquema similar al correspondiente a SWBR pero en este caso sólo los receptores interesados en la comunicación son los que activan sus anillos resonadores.

- **Multiple Writer Single Reader (MWSR):** El esquema MWSR resulta de utilidad cuando varios emisores quieren comunicarse con un mismo destino. Sin embargo, este esquema precisa del uso de técnicas de arbitraje entre los diferentes emisores para decidir quién accede al medio óptico. Estas técnicas de arbitraje pueden implementarse bien eléctricamente o bien mediante óptica. En la Figura 2.4a puede observarse este esquema de comunicación. La utilización de este esquema cuando se cuenta con DWDM permite asignar diferentes wavelengths a cada emisor. De esta manera, el arbitraje solo es necesario cuando varios emisors pretenden comunicarse con el mismo destino.

- **Multiple Writer Multiple Reader (MWMR):** En el caso de MWMR se permite la comunicación entre cualquier emisor con cualquier receptor, por lo que se debe arbitrar en ambos lados antes de realizar la transmisión para evitar colisiones. La cantidad de moduladores en este esquema es del orden de $O(N \times b_\lambda)$ donde N es el número de nodos tanto emisores como receptores y b_λ es el número de wavelengths utilizados. Se trata por tanto de un esquema de comunicación que permite gran flexibilidad pero que supone



(a) Esquema SWMR (b) Esquema MWSR

Figura 2.5: Esquemas de comunicación SWMR y MWSR con WDM de 3 longitudes de onda.

un coste elevado de recursos ópticos. En la Figura 2.4b se puede observar el diagrama correspondiente a este esquema.

Estos esquemas de comunicación se corresponden con una red fotónica en la que el canal óptico no se encuentra multiplexado por longitudes de onda. Sin embargo, cuando se cuenta con DWDM, estos esquemas pueden ampliarse teniendo en cuenta que cada waveguide puede ser dividida en un total de hasta 64 λ_i. Este tipo de esquemas se utilizan para la implementación de *crossbars* ópticos, ya que permiten conectar un determinado número de entradas y salidas entre sí. En este punto se encuentra una de las principales decisiones de diseño de una red fotónica ya que el esquema elegido tiene un impacto directo en el número de posibles comunicaciones paralelas y su ancho de banda.

En la Figura 2.5a se puede observar el esquema de comunicación SWMR correspondiente a una multiplexación de la señal luminosa en tres longitudes de onda. En este caso, las tres entradas cuentan con una λ_i propia para comunicarse con cada una de las tres salidas. Sin embargo, se puede apreciar que en caso de que dos o más entradas pretendan comunicarse con el mismo terminal de salida se presenta un conflicto en la red. Por tanto, se debe utilizar una técnica de arbitraje que determine qué entrada gana el acceso hacia la salida. De esta forma, el proceso de comunicación a seguir en este esquema sería el siguiente:

1. Las entradas I_1 e I_2 intentan enviar un paquete a la salida O_2.

2. Una función de arbitraje determina qué entrada es la escogida. Supongamos I_2 como la ganadora del medio.

3. La entrada I_2, al saberse ganadora del arbitraje, envía una señal eléctrica al terminal O_2 para que calibre correctamente el receptor que le corresponde en λ_2 (identificada con color amarillo).

4. El transmisor en I_2 modula la longitud de onda λ_2 para enviar el paquete.

5. Finalmente, el terminal de salida recibe el paquete y lo convierte a señal eléctrica, terminando así la comunicación.

Este tipo de *crossbar* óptico también permite implementar el esquema SWBR mediante el reenvío del paquete a todas las terminales de salida. En este caso, cada terminal de salida convierte el paquete recibido y comprueba si es responsable del mismo. La utilidad de este esquema se reduce a la distribución de información de arbitraje redundante, ya que para comunicaciones entre pares consume significativamente más energía que el esquema SWMR.

En el esquema SWMR existe además una alternativa que permite evitar utilizar arbitraje global antes de realizar el envío. La solución recibe el nombre de *buffered*-SWMR y consiste en incluir búferes junto a cada receptor óptico en cada terminal de salida. Así, el emisor solo debe conocer si cuenta con espacio suficiente en el búfer destino antes de enviar el paquete. En el lado del receptor, si este cuenta con paquetes disponibles en varias colas, realizará un arbitraje local entre las mismas e irá sirviendo los paquetes sucesivamente. Este esquema resulta de utilidad cuando no se cuenta con mecanismos de arbitraje global eficientes (i.e. mecanismos de arbitraje por tecnología óptica que no ralentizan los envíos). En la Figura 2.6 se puede observar una ilustración de este esquema de comunicación.

Independientemente del tipo de esquema SWMR que se utilice, las comunicaciones bajo este patrón necesitarán al menos un transmisor por entrada y $O(N^2 \times b_\lambda)$ receptores donde N es el número de nodos tanto emisores como receptores y b_λ es el número de wavelengths utilizados en el bus. Por tanto, para evitar disparar el consumo y coste de los recursos ópticos, es recomendable utilizar este esquema en situaciones en las que el número de receptores es reducido.

Por otro lado, en la Figura 2.5b se presenta el esquema de comunicación MWSR que, al igual que en el caso anterior, utiliza tres longitudes de onda para interconectar tres entradas con tres salidas. En este caso, las longitudes de onda (o canales ópticos formados por varias λ_i) se asocian al número de salidas del *crossbar*. Como resultado, en este caso también es necesario un arbitraje global que evite colisiones cuando dos o maś entradas pretenden comunicarse con una misma salida. De esta forma, el proceso de comunicación resulta prácticamente análogo al anterior:

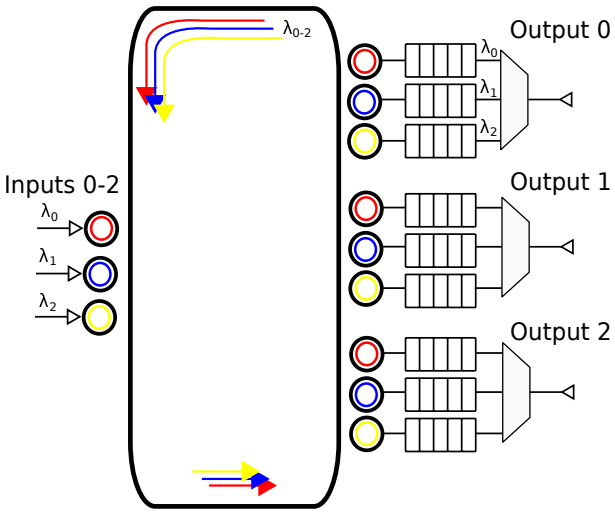


Figura 2.6: Esquema de comunicación buffered-SWMR.

1. Las entradas I_1 e I_2 intentan enviar un paquete a la salida O_2.
2. Una función de arbitraje determina qué entrada es la escogida. En este caso se supone I_2 como la ganadora del medio.
3. En este caso, la salida O_2 ya se encuentra calibrada por defecto en λ_2, por lo que no es necesario realizar el calibrado activo en este esquema.
4. El transmisor en I_2 modula la longitud de onda λ_2 para enviar el paquete.
5. Finalmente, el terminal de salida recibe el paquete y lo convierte a señal eléctrica, terminando así la comunicación.

La principal ventaja del esquema MWSR se encuentra en que no es necesario realizar el calibrado activo de las diferentes salidas. Gracias a esto, es posible evitar el consumo necesario para activar los receptores, a diferencia del caso del esquema SWMR que sí lo necesita.

En lo que respecta a componentes ópticos, MWSR requiere de al menos un receptor por entrada y $O(N^2 \times b_\lambda)$ transmisores. Esto quiere decir que, al igual que el esquema SWMR, el número de componentes (transmisores en el caso de MWSR y receptores en el caso de SWMR) crece cuadráticamente. Por tanto estos esquemas deben ser utilizados en situaciones en las que el tamaño y requisitos de la red no impliquen un crecimiento desmedido de la cantidad de componentes ópticos necesarios. En caso contrario, el consumo derivado de estos componentes y el área utilizada por los mismos se convertirán en limitaciones para la red.

Capítulo 3

Trabajo relacionado

Este capítulo describe trabajos relacionados con la tecnología fotónica así como su aplicación a las redes en chip.

El desarrollo de la tecnología fotónica así como su integración en el silicio han sido campos ampliamente investigados en la última década. En [7], S. Abadal *et. al* estudian diferentes posibilidades de aplicación de la tecnología fotónica en arquitecturas híbridas y multiprocesadores. En lo que respecta a la utilización de los componentes ópticos en redes en chip, numerosos trabajos como [8–10] estudian diversas alternativas acerca de cómo abordar el diseño de una red fotónica dentro del chip, así como sus oportunidades y retos.

Como resultado de estas investigaciones previas, a lo largo de los últimos años se han presentado numerosas propuestas de redes ópticas como solución a los problemas cada vez mayores de escalabilidad en las prestaciones de las redes dentro del chip. En previsión de los requisitos de ancho de banda que necesitarán las arquitecturas futuras, D. Vantrease *et. al* proponen Corona [1], una arquitectura manycore 3D que utiliza comunicación mediante tecnología fotónica tanto dentro como fuera del encapsulado.

Por otro lado, en [11] G. Kurian *et. al* presentan ATAC, una arquitectura que cuenta con una red óptica dentro del chip que permite la interconexión de 1000 núcleos dentro del mismo. La principal diferencia que presenta con Corona radica en la asignación de los recursos ópticos a los emisores y receptores de la red. Como alternativa a Corona y ATAC, en [2] se propone Firefly, una arquitectura híbrida que agrupa los núcleos del chip en clusters e interconecta dichos clusters mediante conexiones ópticas.

La integración de la tecnología fotónica dentro del chip, sin embargo, no está exenta de dificultades. Debido a las características especiales y a la reducida escala de los componentes ópticos, el desarrollo de NoCs fotónicas es muy sensible a los errores de fabricación; este propiedad se conoce como variabilidad en el proceso de fabricación. Respecto a este problema, en [12] se proponen diferentes solucio-

nes para mitigar el efecto de la variabilidad en la fabricación en las prestaciones de la red fotónica.

En [13] encontramos la red FlexiShare, un anillo fotónico con un esquema MWMR propuesto para un CMP de 64 núcleos. Esta red utilia un mecanismo de arbitraje basado en flujos de tokens para incrementar la utilidad de la red. Los autores utilizan diferentes cantidades de canales y DWDM de 64 longitudes de onda por canal.

Si bien Corona, FlexiShare y Firefly presentan resultados y prestaciones aceptables, lo hacen a cambio de una utilización significativa de los recursos ópticos. En [14], A. García-Guirado y S. Bartolini realizan un estudio de los componentes ópticos utilizados por estas tres redes y proponen una serie de políticas para administrar estos recursos de un modo más adecuado. Los autores proponen políticas basadas en el tamaño de los mensajes que circulan por la red, la disponibilidad de los recursos ópticos necesarios para la comunicación y la distancia que separa al emisor del receptor.

Las redes anteriores y los esquemas de comunicación comentados en el Apartado 2.2.2 requieren de técnicas de arbitraje a la hora de compartir los canales ópticos en comunicaciones simultáneas. En [15], D. Vantrease *et. al* proponen la utilización de la tecnología óptica para realizar las tareas de arbitraje y control de flujo. Los autores presentan dos clases de técnicas de arbitraje basadas en tokens y evalúan las mismas con objetivos relativos a latencia, utilización y *fairness*.

Aunque este trabajo se centra en la integración de la tecnología fotónica dentro del chip, otros autores como Batten *et. al* proponen en [16] una estrategia de diseño para redes ópticas a nivel interchip. Además, realizan una propuesta sobre cómo utilizar una red óptica para interconectar el procesador con la memoria principal.

Capítulo 4

Propuesta: Red híbrida fotoeléctrica

En este capítulo se describe la red híbrida propuesta en este Trabajo Fin de Máster. Teniendo en cuenta las limitaciones de las redes en chip eléctricas ya explicadas en el Capítulo 2, en este trabajo presentamos una red formada por una malla bidimensional eléctrica acompañada por un anillo fotónico que permitirá reducir la latencia de las largas distancias dentro del chip.

4.1 Modelo de red híbrida

El modelo de red híbrida propuesto en este trabajo puede observarse en la Figura 4.1. Los 64 nodos del CMP están conectados por una malla bidimensional eléctrica, a la que se conectan también a través del primer y último nodos (nodos 0 y 63) los controladores de memoria. Además, se incluye un anillo implementado con tecnología fotónica.

Para conseguir que todos los nodos sean alcanzados por el anillo óptico, en esta propuesta el diseño de los tiles es simétrico en lugar de totalmente idéntico. De esta forma se consigue conectar los switches de 4 tiles para proporcionar un acceso cercano al anillo óptico. Esto es necesario ya que la tecnología óptica no permite la implementación de waveguides que realicen un gran número de cambios de dirección. De lo contrario, la potencia requerida para transmitir el haz de luz por el waveguide se incrementaría en exceso [17].

Por otra parte, la decisión de ubicar los controladores a ambos extremos de la red se corresponde con la situación habitual de estos componentes en las redes en chip tradicionales. Obsérvese que ambos controladores se encuentran separados por un total de 14 saltos en la red, lo que coincide con el diámetro de la misma.

El objetivo es, por tanto, reducir la latencia de las comunicaciones de los nodos que se encuentran más alejados de los controladores de memoria. La red híbrida propuesta permite a los nodos que se encuentran a elevadas distancias de los con-

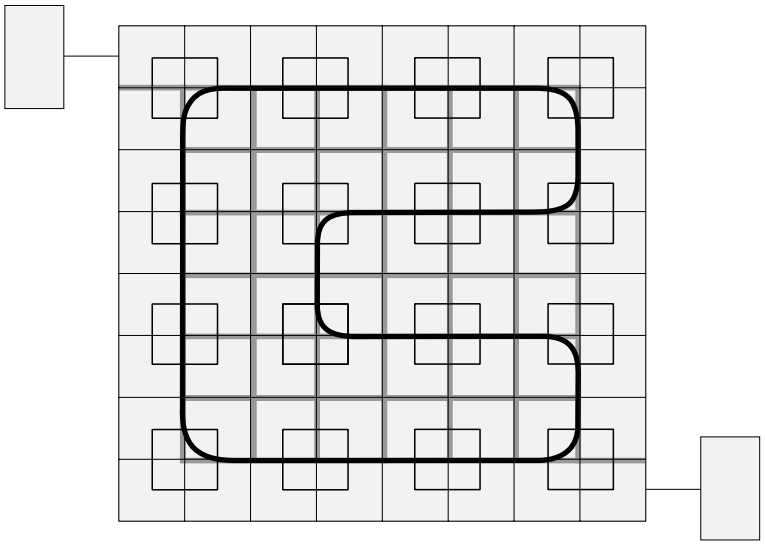


Figura 4.1: Modelo de red híbrida propuesta.

troladores utilizar un camino alternativo y más veloz a través del cual realizar las peticiones y obtener los bloques de memoria. Sin embargo, este no será el único criterio que determine cuándo utilizar el anillo óptico. En el Apartado 4.1.1 se presentan los criterios contemplados a la hora de determinar qué red es utilizada para llevar a cabo un acceso a los controladores.

El esquema de comunicación utilizado en el anillo óptico requiere diferenciar entre la comunicación de nodo a controlador (nodo-controlador) y en sentido inverso (controlador-nodo). En lo que respecta al sentido nodo-controlador, se utiliza un esquema *buffered*-SWMR en el que cada nodo cuenta con cuatro[1] longitudes de onda asignadas para el envio de paquetes a los controladores de memoria. Debido a que el sistema cuenta con 64 nodos, se requieren al menos 256 wavelengths para satisfacer este requisito. Teniendo en cuenta que cada waveguide se puede descomponer mediante DWDM en 64 wavelengths, se requieren al menos 4 waveguides para implementar la comunicación nodo-controlador.

En lo que respecta al sentido controlador-nodo, se utiliza igualmente un esquema *buffered*-SWMR en el que cada controlador tiene asignados un total de 32 wavelengths para el envío de paquetes a los nodos. Puesto que el procesador cuenta con dos controladores, el sentido controlador-nodo requiere un waveguide completo adicional. En total, para la comunicación en ambos sentidos se requieren 5 waveguides (es decir, 320 wavelengths) 4 para el envío de paquetes en el sentido

[1]Este valor ha sido determinado tras un estudio teórico del ancho de banda requerido para el anillo óptico.

Núcleos de procesamiento	
Núcleos	64
ISA	x86
Frequencia	3GHz
Política de issue	Fuera de orden
Predictor de saltos	Combinado
Ancho de issue/commit	4 instrucciones/ciclo
Tamaño del ROB	256 entradas
Cola de Load/Store	64/48 entradas
Jerarquía de caché	
L1 Icache (privada)	32KB, 8 vías, 64B-línea, 2cc
L1 Dcache (privada)	32KB, 8 vías, 64B-línea, 2cc
L2 (privada)	256KB, 16 vías, 64B-línea, 11cc, 16 MSHR
Red de interconexión eléctrica	
Topología	2D Mesh 8x8
Frecuencia	2 GHz
Encaminamiento	X-Y
Tamaño de los búferes	256B
Ancho de banda de enlace	16Bytes/ciclo
Conmutación	Store & Forward y Virtual Cut-Through
Router	Segmentado 3 etapas, 3 ciclos/hop
Red de interconexión óptica	
Topología	Anillo
Frecuencia	10 GHz
DWDM	Sí
Nº de waveguides	5, 64 wavelengths/waveguide
Ancho de banda/wavelengh	1bit/ciclo
Esquema de comunicación	Buffered-SWMR asimétrico
Conversión Eléctrico-Óptica	1 ciclo
Conversión Óptico-Eléctrica	1 ciclo
Memoria principal	
Latencia fija	200 ciclos

Tabla 4.1: Configuración del sistema completo

nodo-controlador y 1 para el envío de paquetes en el sentido controlador-nodo.

El resto de los parámetros del sistema base se pueden consultar en la Tabla 4.1.

4.1.1 Criterios de elección de red

El funcionamiento de la red híbrida propuesta depende de ciertos criterios relativos al estado de las redes en el momento de la comunicación. El objetivo primario de la red híbrida es proporcionar a los nodos un acceso rápido a la memoria. Para conseguir esto, se han escogido una serie de criterios que permiten a un nodo determinar qué red utilizar en el momento de iniciar una petición al controlador de memoria. Estos criterios son los siguientes:

- **Umbral de latencia:** En este criterio se tienen en cuenta dos factores. El primer factor se debe a la comparación de los resultados obtenidos por el modelo teórico de la latencia de la red eléctrica en ausencia de contención y el modelo teórico de la latencia del anillo óptico. En caso de que la latencia teórica del anillo óptico resulte inferior, la transmisión es realizada por el mismo.

  Para ambas redes, el modelo teórico de la latencia en ausencia de contención depende de la técnica de conmutación de paquetes utilizada. La Tabla 4.2 expone las expresiones correspondientes a los modelos teóricos de latencia de Store & Forward y Virtual Cut-Through. En ambos modelos, la variable d se corresponde con la distancia en número de saltos que separa al emisor del receptor; t_{link} es el tiempo que lleva a un paquete atravesar un enlace y por tanto depende del tamaño del paquete y del ancho de banda del enlace; y t_{hop} es la cantidad de ciclos necesaria para atravesar el *switch* segmentado. Finalmente, la última parte de la expresión hace referencia al tiempo de serialización necesario para reintroducir el paquete en un nuevo enlace y reenviarlo al siguiente destino.

Store & Forward
$lat_{SAF} = t_{link} \times (d+1) + t_{hop} \times d + \frac{packetSize}{bandwidth_{link}} \times d$
Virtual Cut-Through
$lat_{VCT} = t_{link} \times (d+1) + t_{hop} \times d + \frac{packetSize}{bandwidth_{link}}$

Tabla 4.2: Modelos de latencia utilizados

Resulta llamativa en estas expresiones la diferencia de latencia derivada de la utilización de una técnica de conmutación u otra. En el caso de SAF, la latencia de serialización se incrementa proporcionalmente con el número de saltos, mientras que en VCT esta latencia permanece constante. Es importante destacar que para la red eléctrica el número de saltos depende de las posiciones relativas entre nodo y controlador, mientras que para la red óptica la distancia es constante ya que todos los nodos conectados al anillo óptico se comunican directamente a través de este con ambos controladores.

El segundo factor tiene en cuenta la latencia de la red eléctrica en presencia de contención. Esta latencia no se calcula teóricamente sino que se obtiene a partir del valor de la latencia observado en la red por el último fallo de caché generado por la ejecución de una instrucción de *load*. Este valor se compara con la latencia teórica del anillo óptico y determina qué red debe utilizarse. Nótese que no se tiene en cuenta la contención en la comunicación a través del anillo óptico debido a que el esquema *buffered*-SWMR garantiza ausencia de contención en la comunicación entre nodos y controladores.

- **Privacidad de los datos:** Este criterio sólo se aplica a cargas paralelas y permite utilizar la red óptica para garantizar una rápida obtención de los datos privados desde la memoria principal, mientras que los datos compartidos son proporcionados por las distintas cachés que se encuentren en posesión del bloque a través de la red eléctrica.

Capítulo 5

Entorno experimental

En este capítulo se presenta el entorno de simulación utilizado para evaluar la propuesta de este Trabajo Fin de Máster. Además, se describen las principales incorporaciones y ampliaciones que se han desarrollado sobre este entorno de trabajo con el objetivo de soportar la simulación de la tecnología fotónica. Finalmente, se introducen las cargas multiprogramadas que se han utilizado durante las simulaciones para obtener los resultados.

5.1 El framework de simulación Multi2Sim

Multi2sim [18] es un entorno de simulación dirigido por eventos y con precisión a nivel de ciclo diseñado para computación heterogénea CPU-GPU. Está escrito en lenguaje C e incluye modelos para CPUs superescalares, multinúcleo y multihilo, así como para arquitecturas GPU. El simulador se encuentra en su versión 4.2, soporta la ejecución cualquier *suite* de *benchmarks* de la que se disponga del código fuente e incluso de código pre-compilado de usuario en diversas arquitecturas sin necesidad de realizar tareas de portabilidad.

El entorno de simulación CPU se divide en dos componentes software principales: el simulador funcional y el simulador arquitectural. La simulación funcional emula la ejecución de un programa en un procesador x86 nativo, interpretando el código binario del programa y reproduciendo dinámicamente su comportamiento a nivel de ISA. Por otro lado, la simulación arquitectural obtiene una traza de las instrucciones x86 a partir del simulador funcional y sigue la ejecución de las estructuras hardware del procesador ciclo a ciclo. Este tipo de simulación modela procesadors superescalares multinúcleo segmentados fuera de orden, una jerarquía de memoria completa con protocolo de coherencia caché y diferentes redes de interconexión. Sin embargo, Multi2Sim no ofrece un modelo detallado de memoria principal, así como del controlador de memoria, sino que las latencias mo-

deladas en el acceso a memoria principal son fijas independientemente del patrón de accesos a memoria.

5.2 Ampliaciones realizadas sobre Multi2Sim

Durante la implementación de este Trabajo Fin de Máster se han realizado una serie de ampliaciones en el simulador Multi2Sim. Las extensiones implementadas tienen el objetivo de permitir en primer lugar la simulación realista de las características únicas de la tecnología fotónica y, además, dotar al simulador de la capacidad de utilizar la mejor opción entre las redes disponibles.

Así, cabe destacar el estado del simulador Multi2Sim en su versión 4.2 antes de comenzar este trabajo. En lo que respecta a las características relativas a los objetivos de este trabajo, éstas eran:

- Multi2Sim permite declarar varias redes en sus archivos de configuración. Sin embargo, no admite más de una interconexión entre los módulos de memoria por lo que estos no pueden seleccionar diferentes redes para acceder a otros niveles de la jerarquía de memoria.

- Multi2Sim define un sistema de red con una frecuencia única, por lo que todas las redes declaradas en sus archivos de configuración presentan la misma frecuencia.

- Multi2Sim no utiliza VCT. Calcula el t_{link} y el $t_{serializacion}$ dividiendo el tamaño del paquete entre el ancho de banda del enlace, por lo que implementa SAF por defecto.

- Multi2Sim mantiene en la Unidad de Gestión de Memoria (MMU) una lista sobre las páginas de memoria utilizadas. No realiza sin embargo clasificación alguna sobre estas páginas que permita diferenciar si contienen datos privados o compartidos.

Para alcanzar los objetivos de este trabajo, las características anteriores del simulador deben ser modificadas. En los siguientes apartados se describen las implementaciones realizadas sobre el framework Multi2Sim que permiten utilizar la red híbrida propuesta.

5.2.1 Múltiples redes de interconexión

La red híbrida propuesta en el Capítulo 4 requiere por parte del sistema la capacidad de soportar dos redes completamente operativas. Además, el anillo óptico

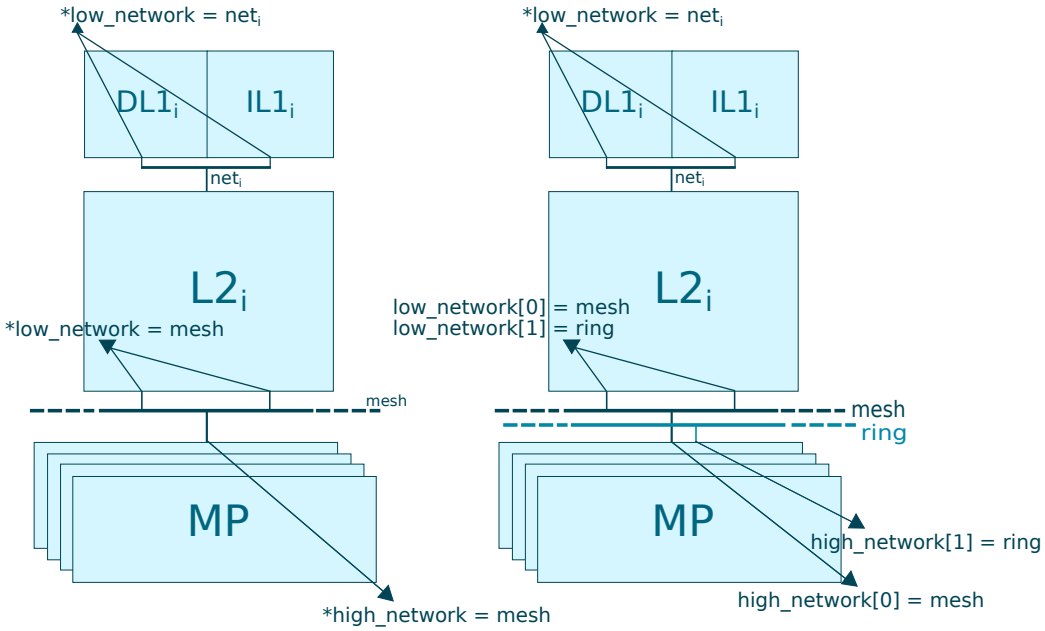


Figura 5.1: Interconexión entre módulos de memoria antes y después de la ampliación de Multi2Sim.

debe contar con características específicas de la tecnología fotónica como son la multiplexación en longitudes de onda o la frecuencia a 10 GHz. Por ello, la primera extensión realizada sobre el simulador consiste en permitir que el sistema opere con dos redes simultáneamente.

Multi2Sim soporta la declaración de varias redes en el archivo de configuración correspondiente a la red. Partiendo de este punto, el siguiente paso consiste en habilitar a los diferentes módulos de memoria la utilización de cualquiera de las redes. En la Figura 5.1 se observa la diferencia entre cómo se realizaban las conexiones de los módulos de memoria en la versión 4.2 del simulador y cómo se realizan tras la ampliación.

Tal y como se indica en la Figura 5.1, esta modificación ha consistido en ampliar el puntero utilizado anteriormente por los módulos de memoria para acceder a su red asociada. En la jerarquía de memoria de Multi2Sim, cada módulo de memoria cuenta con punteros para acceder a su red superior e inferior si la hubiera. Por ello, en esta ampliación se han convertido los punteros *low_network* y *high_network* del LLC y de la memoria principal respectivamente en vectores de punteros. De esta forma, tanto los módulos de LLC como los controladores de memoria cuentan con tantas redes accesibles como elementos tiene su vector asociado. Así, tras esta extensión, Multi2Sim soporta actualmente declarar cualquier

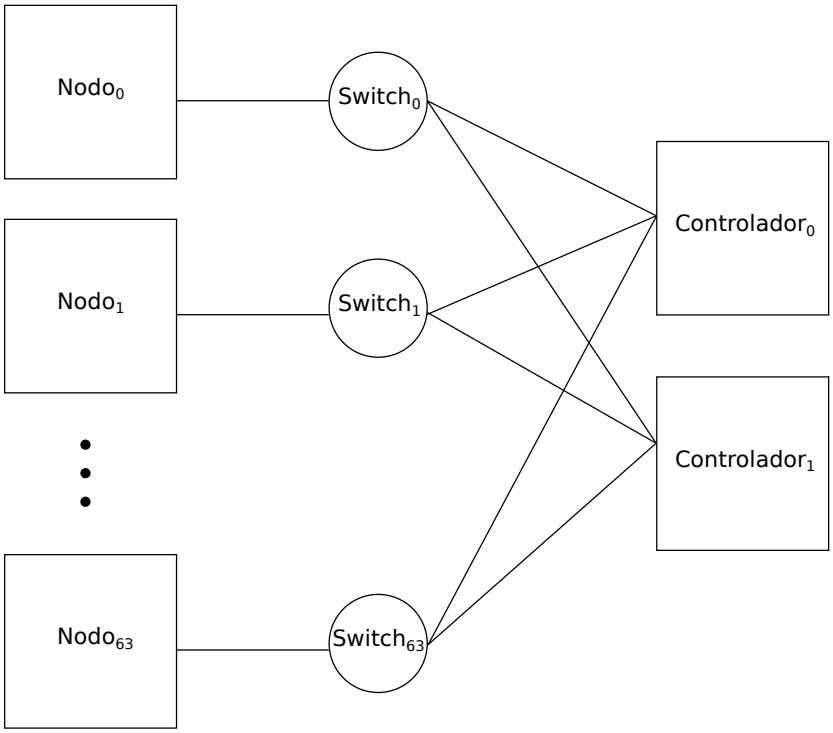


Figura 5.2: Modelo de anillo óptico en Multi2Sim.

número de redes de interconexión entre los módulos de LLC y memoria principal.

Gracias a esta implementación, el anillo óptico se puede simular en Multi2Sim mediante conexiones directas entre cada nodo y cada controlador de memoria, limitando el ancho de banda de cada una de estas conexiones al determinado por el esquema de comunicación óptico seleccionado. El modelo de anillo óptico en Multi2Sim se presenta en la Figura 5.2.

5.2.2 Virtual Cut-Through

La técnica de conmutación Virtual Cut-Through es muy habitual en el campo de las redes en chip. Por tanto, para obtener resultados realistas relativos al impacto de diversos factores como distancia o contención en la red en el chip es necesario contar con un simulador que implemente esta técnica.

Para introducir la implementación de VCT es necesario explicar varios conceptos previos acerca de cómo opera la red en el framework Multi2Sim. En primer lugar, Multi2Sim diferencia dos tipos de nodos: *endpoints* y *switches*. Los endpoints son nodos que se asocian a los módulos de memoria, mientras que los switches son los que llevan a cabo las tareas de interconexión entre nodos, control

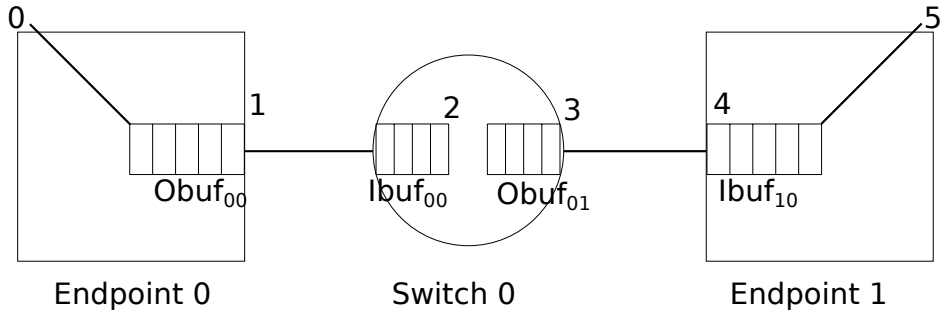


Figura 5.3: Esquema de conexión entre dos nodos de red adyacentes en Multi2Sim.

de flujo, conmutación, etc.

El funcionamiento de la red se dirige por medio de una máquina de estados que se activa por medio de eventos. En la Figura 5.3 se muestra el esquema básico de conexión entre dos nodos. Los eventos que tienen lugar para llevar a cabo una comunicación entre ambos son los siguientes:

- Evento 0 - **SEND**: El evento SEND es el que inicia una transmisión. Introduce un paquete en la red, comprueba que el búfer de salida del endpoint tiene espacio disponible para el paquete y programa un evento OUTPUT para el ciclo siguiente.

- Evento 1 - **OUTPUT**: El evento OUTPUT realiza la transmisión por el enlace contiguo al búfer de salida en el que se encuentra el paquete. Comprueba que el búfer de entrada del siguiente switch o endpoint tiene espacio suficiente para almacenar el paquete y programa el evento INPUT para n ciclos después, donde $n = \frac{packetSize}{bandwidth_{link}}$.

- Evento 2 - **INPUT**: El evento INPUT simula la segmentación del switch en etapas. Este evento cuenta con un parámetro que recibe el nombre de $bandwidth_{node}$ que, junto con el tamaño de paquete, determina los ciclos de latencia que se tarda en atravesar el switch. El siguiente evento OUTPUT o RECEIVE que corresponda se programa tras n ciclos, donde $n = \frac{packetSize}{bandwidth_{node}}$.

- Evento 3 - **OUTPUT**: Este evento realiza las mismas acciones que el evento 1, dirigiendo esta vez el paquete al endpoint final.

- Evento 4 - **INPUT**: Este evento realiza las mismas acciones que el evento 2. Además, se programa el evento RECEIVE para el mismo ciclo en el que se ha realizado este evento.

Ciclos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Eventos	S	O	O	O	O	O	I	I	I	O	O	O	O	O	I & R

Tabla 5.1: Transmisión ciclo a ciclo de un paquete de 72B en Multi2Sim con SAF.

Ciclos	0	1	2	3	4	5	6	7	8	9	10
$Frag_0$	S	O	I	I	I	O	-	-	-	-	I & R
$Frag_1$		S	O	I	I	I	O	-	-	-	I & R
$Frag_2$			S	O	I	I	I	O	-	-	I & R
$Frag_3$				S	O	I	I	I	O	-	I & R
$Frag_4$					S	O	I	I	I	O	I & R

Tabla 5.2: Transmisión ciclo a ciclo de un paquete de 72B en Multi2Sim con VCT.

- Evento 5 - **RECEIVE**: Este evento lee el mensaje de la red y actualiza las estadísticas correspondientes al mismo. La latencia final de la comunicación se calcula obteniendo la diferencia entre el ciclo de envío y el ciclo de recepción del paquete.

Siguiendo este esquema se aprecia la presencia implícita de conmutación Store & Forward en Multi2Sim. Para ilustrar esto con un ejemplo, supongamos la transmisión de un paquete de 72 bytes (62B de datos más 8B de cabecera) entre dos nodos adyacentes siguiendo este esquema y contando con un ancho de banda de enlace de 16 bytes/ciclo; el evento INPUT consume un total de 3 ciclos en atravesar el switch.

La sucesión de ciclos correspondiente a este ejemplo se muestra en la Tabla 5.1. Tal y como se observa, la transmisión requiere un total de 14 ciclos debido a que, con 16 bytes/ciclo de ancho de banda, la serialización del paquete consume 5 ciclos tanto en el endpoint origen como en el switch que lo retransmite.

Para superar esta limitación de Multi2Sim, la implementación de VCT se ha basado en la división de los paquetes en fragmentos de tamaño igual o menor al ancho de banda del enlace expresado en bytes/ciclo. De esta forma, un paquete de 72 bytes queda dividido en 4 fragmentos de 16 bytes y un fragmento de 8 bytes. De esta forma se produce una segmentación de los eventos en el tiempo y se oculta la latencia de serialización del reenvío por parte de los switches intermedios. Siguiendo este esquema, la sucesión de ciclos del ejemplo anterior queda tal y como se observa en la Tabla 5.2. En este caso, la latencia de la transmisión se reduce a 10 ciclos gracias a la operación segmentada de la red.

Obśervese que los ejemplos mostrados en las Tablas 5.1 y 5.2 se corresponden con comunicaciones a un salto de distancia. Conforme se incrementa la distancia de la comunicación, SAF aumenta la latencia en 8 ciclos por salto mientras que

VCT lo hace en 4 ciclos por salto. De esta forma, la introducción de VCT en Multi2Sim consigue reducir la latencia de las comunicaciones.

5.2.3 Múltiples dominios de frecuencia a nivel de red

La red híbrida propuesta asume que el anillo fotónico opera a una frecuencia de 10 GHz. Multi2Sim administra las frecuencias del sistema mediante dominios de manera que cada subsistema de la arquitectura cuenta con un dominio propio. En nuestro caso, Multi2Sim define dominios para el sistema de memoria, el sistema de red y las redes locales que interconectan los módulos de memoria local con los módulos de caché L2. Por tanto, para soportar múltiples frecuencias en el sistema de red, el sistema de dominios ha de ser modificado.

En la implementación actual, cada red declarada en el archivo de configuración define su propia frecuencia. Posteriormente, Multi2Sim crea un dominio por cada frecuencia declarada en el sistema y actualiza el dominio de simulación a la mayor frecuencia declarada en la arquitectura.

5.2.4 Conversiones electrico-óptica y óptico-eléctrica

Los componentes descritos en la Sección 2.2 realizan operaciones de conversión entre las señales óptica y eléctrica. Estas operaciones consumen cada una un ciclo en la red óptica (i.e. un ciclo a 10 GHz) y deben ser incluidas en el modelo de red fotónica para obtener resultados realistas.

En Multi2Sim la introducción de estas latencias de conversión se ha implementado añadiendo dos nuevos eventos a la máquina de estados del simulador: los eventos E2O y O2E. Cada uno de ellos tiene lugar a la entrada y salida de los switches que conectan cada para nodo-controlador y se encargan de añadir la latencia correspondiente a cada conversión, de manera que el siguiente evento se programa después de los ciclos correspondientes a esta latencia.

Para calcular la latencia de conversión, se ha incorporado a Multi2Sim un nuevo componente denominado fotoconversor que cuenta con una frecuencia y ancho de banda de conversión determinados. La frecuencia del fotoconversor es habitualmente la frecuencia a la que opera la red óptica, mientras que el ancho de banda de conversión hace referencia a la cantidad de bits o longitudes de onda que el componente es capaz de convertir en un ciclo. En nuestro caso de trabajo este parámetro toma el valor del número de longitudes de onda asignadas a cada nodo, típicamente 4. De esta forma, cada evento introduce un retardo de un ciclo debido a estas conversiones.

5.2.5 Clasificación de páginas de memoria

Adicionalmente, para dotar al modelo de mayor flexibilidad cuando el sistema ejecuta cargas paralelas, es necesario contar con una clasificación de los datos de memoria que se solicitan. Obteniendo información acerca de la privacidad de los datos se abre la puerta a implementar políticas que favorezcan el tráfico de datos privados por el anillo óptico y que a su vez permitan compartir datos entre varios núcleos por la red eléctrica.

La clasificación de páginas en memoria se hace habitualmente a nivel de TLB. Sin embargo, dado que Multi2Sim carece de este componente de la arquitectura, en la versión actual el mecanismo de clasificación se ha implementado en la Unidad de Gestión de Memoria (MMU, del inglés *Memory Management Unit*). El mecanismo consiste en establecer todas las páginas como privadas inicialmente y detectar cuándo más de un núcleo ha accedido a la misma página, momento en el que pasará a clasificarse como compartida.

5.2.6 Modelo de selección de red

Teniendo en cuenta los criterios de selección de red expuestos en la Sección 4.1.1 se ha incorporado a Multi2Sim un modelo de elección que determina sobre qué red se va a enviar el paquete. En el Algoritmo 1 se expone la cadena de acciones que realiza el simulador cuando recibe una petición de envío de un paquete en cualquier sentido (nodo-controlador o controlador-nodo).

En primer lugar, el algoritmo debe conocer si existe una red alternativa que utilizar. En caso afirmativo, se calcula la latencia umbral de selección como la latencia teórica del anillo fotónico. A continuación, se obtiene la latencia teórica de la malla eléctrica teniendo en cuenta la técnica de conmutación empleada. Utilizando estos datos el algoritmo calcula el primer criterio de selección, que consiste en determinar qué red presenta una latencia teórica menor.

El segundo criterio se utiliza cuando el sistema ejecuta cargas paralelas. De ser así, el bloque de memoria cuenta con información que indica si pertenece a una página privada o compartida. En caso de pertenecer a una página privada, el bloque será enviado por el anillo.

Si finalmente ninguno de los dos criterios es satisfecho, el algoritmo selecciona la malla eléctrica como medio de envío del paquete.

5.3 Benchmarks utilizados para simulación

Las propuestas de este Trabajo Fin de Master han sido evaluadas utilizando las cargas de la *suite* SPEC 2006 [19], que se explican a continuación.

Algorithm 1: Algoritmo de selección de red.

```
if existe una red fotónica adicional then
    establecer la latencia del anillo fotónico como latencia umbral;
    if conmutacion de la red eléctrica es SAF then
        calcular latencia teórica con la fórmula de latencia de SAF;
    else
        calcular latencia teórica con la fórmula de latencia de VCT;
    end
    if umbral < latencia teórica eléctrica or
    (estamos ejecutando cargas paralelas and es bloque privado) then
        enviar paquete por anillo óptico;
        return;
    end
end
enviar paquete por malla eléctrica;
```

5.3.1 Aritmética de Enteros

perlbench: Lenguaje C, derivada a partir de Perl V5.8.7. La carga incluye SpamAssasin, MHonArc y specdiff (herramienta de SPEC que comprueba la salida del benchmark).

bzip2: Lenguaje C, herramienta de compresion. Se trata de la herramienta bzip2 V1.0.3 de Julian Seward modificada para realizar su mayor parte de trabajo en memoria en lugar de E/S.

gcc: Lenguaje C, compilador C. Está basado en gcc V3.2.

mcf: Lenguaje C, optimiza operaciones combinacionales para programación de vehículos. Utiliza el algoritmo de redes simplex para programar el transporte público.

gobmk: Lenguaje C, aplicación de inteligencia artificial. Juega al juego Go, de complejidad profunda.

hmmer: Lenguaje C, aplicación de búsqueda de secuencia genética. La carga realiza análisis de secuencias utilizando perfiles de modelos ocultos de Markov (perfiles HMM).

sjeng: Lenguaje C, aplicación de inteligencia artificial, ajedrez. Un programa de ajedrez de alta clasificación que además permite ciertas variantes del juego del ajedrez.

libquantum: Lenguaje C, aplicación de física, Computación Cuántica. Simula un computador cuántico ejecutando el algoritmo de Shor de factorización tiempo-polinomial.

h264ref: Lenguaje C, aplicación de compresión de vídeo. Una implementación de referencia del estándar H.264/AVC, sustituto de MPEG2. Codifica *streams* de vídeo utilizando dos conjuntos de parámetros.

omnetpp: Lenguaje C++, aplicación de simulación discreta de eventos. Utiliza el simulador OMNet++ para modelar una red Ethernet en un campus de gran tamaño.

astar: Lenguaje C++, aplicación de algoritmos de búsqueda de rutas. Se trata de una librería de búsqueda de rutas para mapas 2D, incluyendo el algoritmo bien conocido A*.

xalancbmk: Lenguaje C++, aplicación de procesamiento XML. Se trata de una versión modificada de Xalan-C++, que transforma documentos XML en otro tipo de documentos.

5.3.2 Aritmética de Coma Flotante

bwaves: Lenguaje Fortran, aplicación de Dinámica de Fluídos. Calcula el flujo viscoso laminar transitorio transónico en tres dimensiones.

gamess: Lenguaje Fortran, aplicación de Química Cuántica. Gamess implimenta un amplio rango de cálculos de química cuántica.

milc: Lenguaje C, aplicación de Física y Cromodinámica Cuántica. Un aplicación que genera campos gague para programas basados en la teoría de campos gauge en retículos con quarks dinámicos.

zeusmp: Lenguaje Fortran, aplicación de Física. ZEUS-MP es un código de computación de Dinámica de Fluídos desarrollado por el Laboratorio de Computación Astrofísica (NCSA, Universidad de Illinois) para la simulación de fenómenos astrofísicos.

gromacs: Lenguajes C y Fortran, aplicación de Bioquímica y Dinámica de Moléculas. Simula ecuaciones newtonianas del movimiento de cientos de millones de partículas. Los casos de prueba simulan la proteína Lisozima en una disolución.

cactusADM: Lenguajes C y Fortran, aplicación de Física y Relatividad General. Resuelve las ecuaciones de evolución de Einstein utilizando el método numérico Staggered-Leapfrog.

leslie3d: Lenguaje Fortran, aplicación de Dinámica de Fluídos. Dinámica de Fluídos Computacional (CFD) utilizando Simulaciones de Large-Eddi con el Modelo Linear-Eddy en 3D. Utiliza el esquema de integración temporal MarCormack-Predictor-Corrector.

namd: Lenguaje C++, aplicación de Biología y Dinámica de Moléculas. Simula sistemas biomoleculares de gran tamaño. El caso de prueba cuenta con 92224 átomos de apoliproteína A-I.

dealII: Lenguaje C++, aplicación de Análisis de Elementos Finitos. Librería de C++ dirigida a la adaptatividad de elementos finitos y a la estimación de errores. El caso de prueba resuelve una ecuación del tipo Helmholtz con coeficientes no constantes.

soplex: Lenguaje C++, aplicación de Programación Lineal y Optimización. Resuelve un programa lineal utilizando el algoritmo simplex y álgebra lineal dispersa. Los casos de prueba incluyen planificación de raíles y modelos de puentes aéreos militares.

povray: Lenguaje C++, renderizado y trata de imágenes. El caso de prueba es una imagen suavizada de 1280×1024 de un paisaje con objetos abstractos y diferentes texturas basadas en la función de ruido de Perlin.

calculix: Lenguaje C y Fortran, aplicación de Mecánica de Estructuras. Código de elementos finitos para aplicaciones de estructuras 3D lineales y no lineales. Utiliza la librería SPOOLES.

GemsFDTD: Lenguaje Fortran, aplicación de Electromagnética Computacional. Resuelve las ecuaciones de Maxwell en 3D utilizando el método de tiempo-dominio de diferencias finitas (FDTD).

tonto: Lenguaje Fortran, Química Cuántica. Un paquete de química cuántica de código abierto que utiliza un diseño orientado a objetos en Fortran 95.

lbm: Lenguaje C, aplicación de Dinámica de Fluídos. Implementa el Método Lattice-Boltzmann para simular fluidos incomprimibles en 3D.

wrf: Lenguaje C y Fortran, aplicación de clima. Modela el clima en escalas desde metros hasta miles de kilómetros. El caso de prueba consiste en un área de 30km durante dos días.

sphinx3: Lenguaje C, aplicación de reconocimiento de voz. Sistema de reconocimiento de voz ampliamente conocido diseñado por la Universidad Carnegie Mellon.

Capítulo 6

Resultados experimentales

En este capítulo se presentan y analizan los resultados obtenidos. En primer lugar, se presenta un estudio de exploración acerca de las limitaciones de escalabilidad de una malla eléctrica comúnmente utilizada. Para ello, se evalúan las prestaciones de cargas multiprogramadas a diferentes distancias del controlador de memoria y bajo diferentes configuraciones del sistema. Tras identificar las limitaciones, se analizan las mejoras obtenidas a partir de la red híbrida propuesta en el Capítulo 4.

6.1 Estudio de limitaciones de la red eléctrica

Tal y como se ha expuesto en el apartado 2.1, la topología de red más empleada en el ámbito de las redes en chip es la malla bidimensional. Sin embargo, este tipo de redes presenta límites de escalabilidad conforme crece el número de nodos, ya que el diámetro de la red aumenta significativamente. Esto supone un incremento de la latencia de las comunicaciones entre extremos de la red así como del número de colisiones que se producen en la misma, lo que finalmente implica una degradación de las prestaciones de la red. En este apartado se llevan a cabo varios estudios relativos a la ejecución de cargas alejadas progresivamente del controlador de memoria, con el objetivo de cuantificar el impacto de las distancias de la red en las prestaciones del sistema.

Figura 6.1: Topología utilizada para medir el impacto de la distancia hasta el controlador.

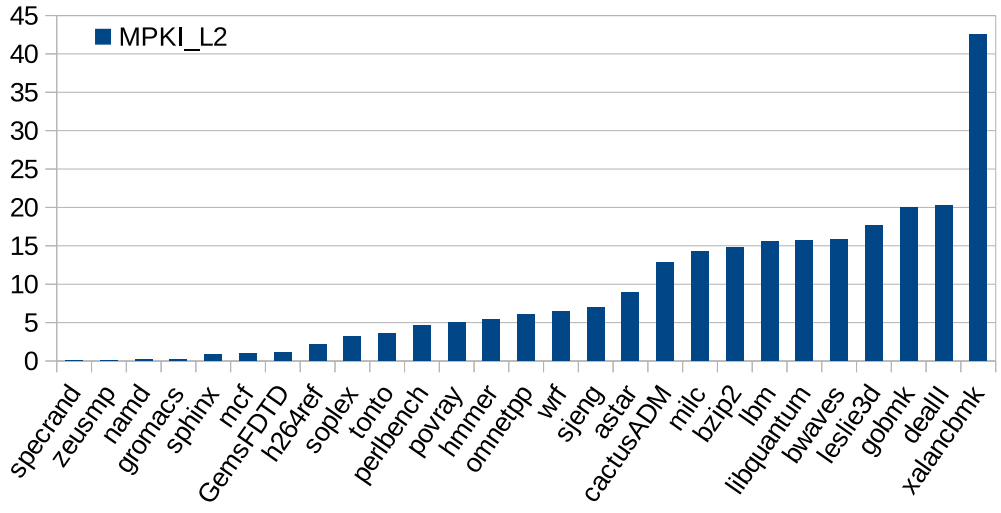
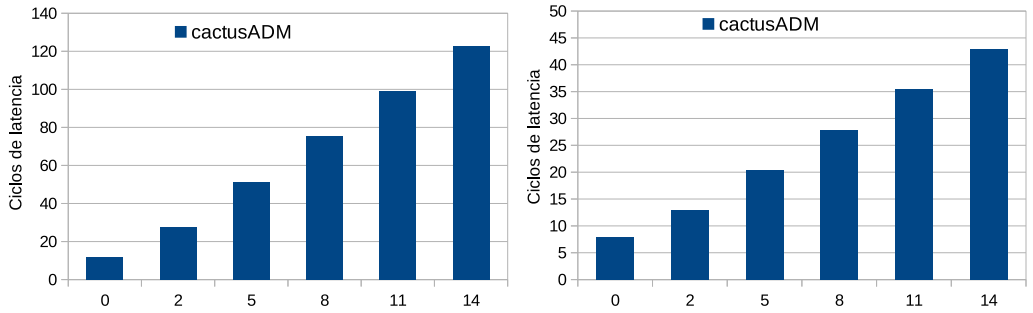


Figura 6.2: MPKI de L2 de las aplicaciones estudiadas.

Como topología para este estudio preliminar se ha seleccionado un array como el que puede observarse en la Figura 6.1. Esta topología permite simular de un modo eficaz elevadas distancias hasta el controlador para un bencmark en ejecución individual, por lo que resulta ideal para estas evaluaciones. El diámetro del array se ha establecido en 14 ya que coincide con el diámetro del sistema base de 64 nodos (véase Capítulo 4) que se evaluará posteriormente. Las características de los componentes del sistema empleado en estas simulaciones son las indicadas en la Tabla 4.1. Dado que la latencia de las transmisiones está condicionada por la técnica de conmutación empleada por la red, se evaluarán tanto el caso de SAF como de VCT.

Por otro lado, los microprocesadores actuales implementan mecanismos de prebúsqueda que tratan reducir el número de ciclos de espera mediante la anticipación de bloques de memoria que la aplicación puede necesitar en un futuro. Estos mecanismos suponen un incremento en la utilización del ancho de banda ya que aumentan el número de peticiones y bloques que circulan por la red. Por esta razón se ha considerado relevante incluir en este estudio el impacto adicional del prefetch junto con la distancia hasta el controlador de memoria.

Finalmente, otro aspecto que no debe pasarse por alto en la evaluación de las prestaciones de una red es el efecto de la contención. La ejecución de cargas paralelas o de múltiples cargas simultáneamente sobre el sistema hace que los recursos de la red se vean saturados, con el consiguiente impacto negativo en las prestaciones. Además, este efecto se ve realzado también conforme aumenta el diámetro de la red debido al aumento del número de colisiones. Este caso, por



(a) Evaluación con *Store-and-Forward*. (b) Evaluación con *Virtual-Cut-Through*.

Figura 6.3: Impacto de la distancia en los ciclos de latencia de la aplicación `cactusADM`.

tanto, también será contemplado en este estudio de exploración.

6.1.1 Impacto de la distancia en las prestaciones

Las Figuras 6.3a y 6.3b muestran, respectivamente, el incremento de latencia de red conforme crece la distancia al controlador utilizando las ténicas de conmutación SAF y VCT. Los resultados se corresponden con la ejecución de la aplicación `cactusADM`, aunque el resto de cargas presentan resultados similares debido a que se trata de valores de latencia media. Tal y como se observa, en el caso de utilizar SAF, el número de ciclos de media por transmisión asciende de 12 ciclos a 0 saltos del controlador hasta 122 ciclos a 14 saltos, lo que supone un incremento del 90 % entre ambas ubicaciones. El número de saltos indica el número de nodos de cómputo por los que debe pasar la petición. A esto se le añade el propio nodo del controlador de memoria.

Por otro lado, cuando se utiliza la técnica de conmutación VCT, los ciclos totales se ven significativamente reducidos respecto a los obtenidos con SAF. Sin embargo, los porcentajes de incremento de latencia entre distancias se mantienen similares. Así, cuando `cactusADM` se encuentra a 0 saltos de controlador presenta una media de 8 ciclos en sus transmisiones, mientras que cuando se encuentra a 14 saltos esta cifra crece hasta un total de 43 ciclos. Esto supone, por tanto, un incremento de latencia media de red del 81.5 % entre ambas distancias.

Estos resultados pueden afectar, por tanto, a las prestaciones de las aplicaciones que se ejecutan en el CMP. En la gráfica de la Figura 6.4 se presenta la degradación de prestaciones que experimentan las aplicaciones cuando se ejecutan en diferentes posiciones del array con SAF respecto a su ejecución separadas 0 saltos del controlador. Aquí se puede observar que no todas las aplicaciones se ven afectadas de igual forma por la distancia al controlador. El pico de degra-

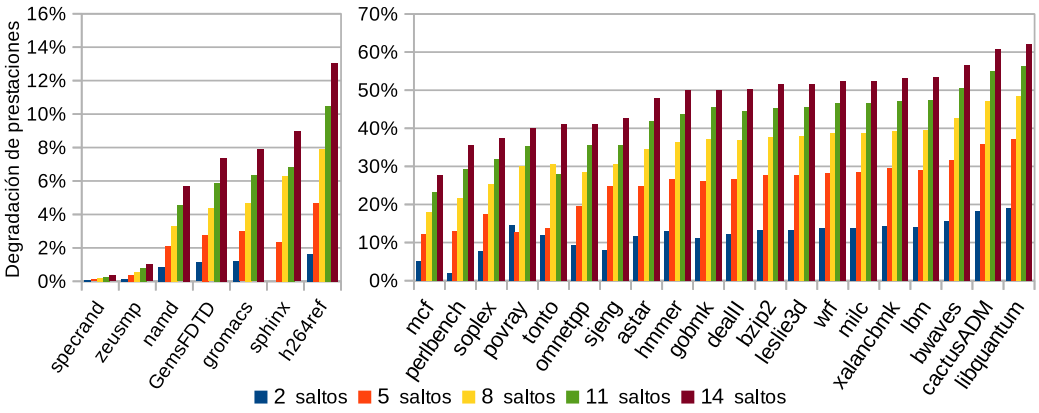


Figura 6.4: Impacto de la distancia y SAF en las prestaciones (IPC) de las aplicaciones en orden creciente de izquierda a derecha.

dación de prestaciones se encuentra en la aplicación `cactusADM`, que alcanza el 61 % cuando se ha de recorrer el diámetro completo hasta alcanzar el controlador. Sin embargo, aplicaciones como `zeusmp` o `specrand` apenas degradan sus prestaciones en un 1 %.

Las diferentes aplicaciones se han categorizado teniendo en cuenta su sensibilidad al efecto del diámetro en las prestaciones. Así, las cargas menos sensibles (7 de 26) presentan, a distancia 14 del controlador, una degradación de IPC media cercana al 4 %. Por otro lado, en lo que respecta a las cargas más sensibles, esta degradación media se eleva hasta un porcentaje que ronda el 43 %.

Estos resultados se justifican teniendo en cuenta la frecuencia de accesos a memoria que presenta que presenta cada aplicación. En el caso del primer grupo de aplicaciones, éstas son cargas que fallan con poca frecuencia en la LLC, de manera que presentan escasos accesos a memoria respecto al total de instrucciones ejecutadas. Se trata de aplicaciones con un MPKI de L2 reducido, tal y como puede observarse en la Figura 6.2.

Sin embargo, el resto de aplicaciones requieren de un uso mucho más frecuente de la red de interconexión debido a su mayor cantidad de accesos a memoria. Esto significa que el impacto negativo de los incrementos de latencia indicados anteriormente se experimenta con mucha más frecuencia en este tipo de cargas. Por consiguiente, el IPC se ve degradado significativamente cuando el diámetro de la red comienza a ascender por encima de 5 nodos.

Los resultados correspondientes a las ejecuciones realizadas con la técnica de conmutación VCT se presentan en la Figura 6.5. En este caso, gracias a que VCT supone un incremento mucho menor en los ciclos por salto en la red, la degradación de prestaciones se ve reducida hasta un 27 % en el caso extremo de

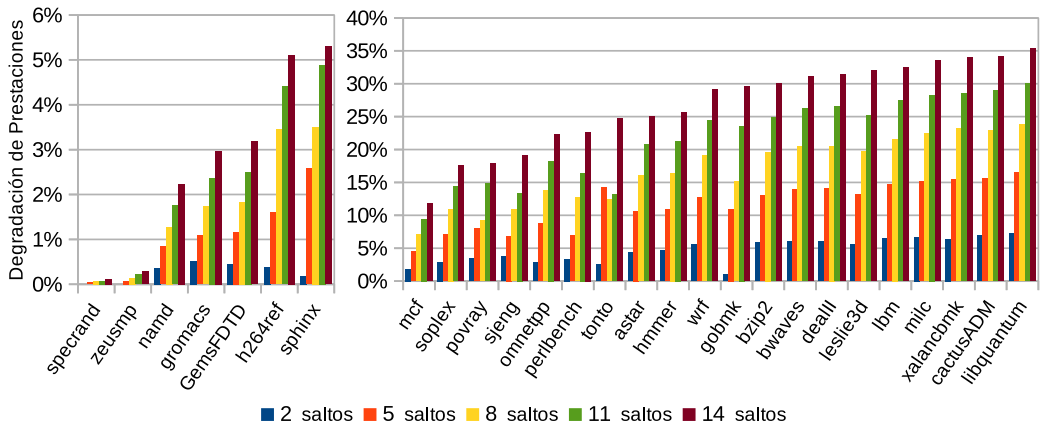


Figura 6.5: Impacto de la distancia y VCT en las prestaciones de las aplicaciones en orden creciente de izquierda a derecha.

`cactusADM` y un 20 % de media. La reducción más significativa la experimenta la aplicación `bwaves` que disminuye su degradación de IPC de un 58 % en la ejecución con SAF a un 30 % en la ejecución con VCT.

Esta mejora en la degradación de prestaciones es también experimentada por las aplicaciones que no utilizan en exceso la red, si bien la reducción respecto a SAF en estos casos es limitada. Algunos ejemplos de estas reducciones limitadas los encontramos en `h264ref` que pasa de un 13 % de degradación a un 5 % y en `gromacs`, con una disminución del 7 % al 3 %. En general, la degradación de prestaciones media de las aplicaciones menos sensibles a la distancia se ha reducido en un 3 % cuando éstas se encuentran a 14 saltos del controlador. Estas cargas, por tanto, requieren del estudio de otros factores que afecten a su rendimiento y que justifiquen la utilización de la red fotónica.

Por tanto, como conclusión a este apartado, cabe destacar que las prestaciones de ciertas aplicaciones se ven comprometidas conforme aumenta el diámetro de la red y, por tanto, el número de nodos que las separan del controlador de memoria. Además, las prestaciones presentan una escalabilidad drásticamente menor cuando la técnica de conmutación empleada es SAF, mientras que VCT consigue reducir significativamente el impacto negativo de la distancia en las aplicaciones que más la acusan. Sin embargo, pese a la utilización de VCT, cuando la distancia al controlador supera el umbral de 10 saltos, 11 de las 26 aplicaciones experimentan degradaciones de IPC superiores al 25 %.

6.1.2 Impacto de la prebúsqueda en la degradación de prestaciones por distancia

Un aspecto que puede condicionar negativamente el rendimiento de la red dentro del chip a distancias elevadas es el uso de técnicas de prebúsqueda agresiva. Este tipo de técnicas permite incrementar las prestaciones de algunas cargas a costa de un incremento de la utilización del ancho de banda y los recursos de la red. Para observar y cuantificar el impacto de este tipo de técnicas, en este apartado se repiten las ejecuciones anteriores activando el *prefetch*.

El prefetch utilizado cuenta con una agresividad de 4, lo que significa que para cada bloque de caché solicitado el prefetch estima cuáles serán los 4 siguientes bloques que necesita la aplicación, y pide los 4 bloques a memoria en caso de que se detecte un patrón regular de accesos. De esta forma, teniendo en cuenta que el prefetch puede fallar en la predicción de los bloques, el ancho de banda requerido para satisfacer las peticiones de memoria aumenta significativamente respecto a las ejecuciones sin prefetch. Así, este estudio presenta un doble objetivo: observar el beneficio que introduce el prefetch en las prestaciones de las aplicaciones y comprobar cómo afecta éste al rendimiento de la red, debido al incremento de tráfico que supone.

Los distintos efectos de la prebúsqueda y la distancia en las prestaciones de las cargas ejecutadas con SAF se muestran en la Figura 6.6. Se puede observar que el prefetch afecta de forma diferente a las distintas aplicaciones. En primer lugar, cargas como `bzip2`, `milc` o `tonto` apenas presentan diferencias entre las ejecuciones con o sin prefetch, ya que sus porcentajes de degradación apenas se han reducido en torno a un 2 %.

Por otro lado también encontramos aplicaciones que se ven ampliamente beneficiadas por el uso de prefetch. El ejemplo más claro es la aplicación `wrf` a distancia 14, cuya degradación de prestaciones ha pasado de un 52.5 % a un 41.2 %. Así, únicamente mediante la activación del prefetch la carga ha visto reducida su degradación de IPC en un porcentaje superior al 10 %. Existen otras cargas que también han mejorado sus prestaciones gracias a la utilización de prebúsqueda. Estas aplicaciones (*e.g.* `astar`, `soplex`, `perlbench`, `mcf`) reducen la degradación de prestaciones en porcentajes cercanos al 5 %.

El impacto positivo en las prestaciones por parte del prefetch se justifica en la cantidad de fallos de L2 que consigue evitar. En los casos anteriores, el prefetch reduce el número de fallos de L2 por cada mil instrucciones (*i.e.* el MPKI) de las aplicaciones beneficiadas. Por consiguiente, los benchmarks realizan menos peticiones a memoria lo que reduce tanto la cantidad de mensajes que circulan por la red como el número de ciclos de red y memoria asociados a ellos. Todo esto se consigue gracias a los bloques que son anticipados correctamente por medio del prefetch.

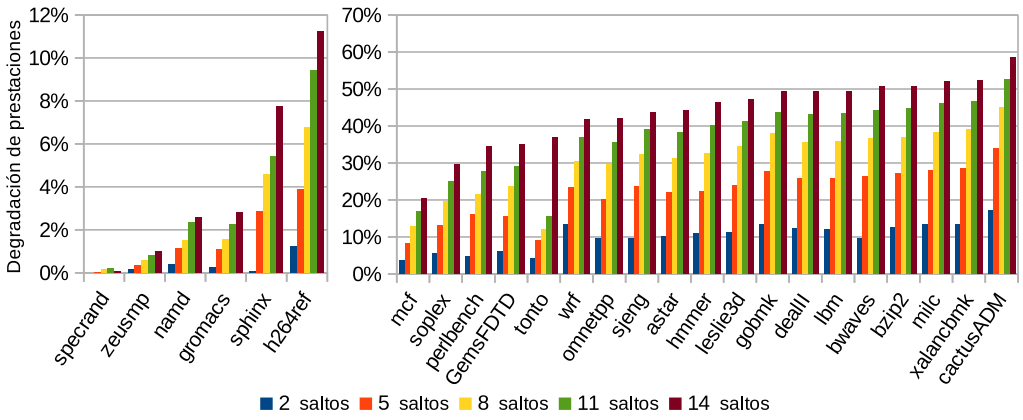


Figura 6.6: Impacto de la prebúsqueda y SAF en las prestaciones de las aplicaciones en orden creciente de izquierda a derecha.

Sin embargo, hay casos en los que la utilización del prefetch no resulta beneficiosa para las prestaciones de la carga. El ejemplo más paradigmático de este grupo es `GemsFDTD`, cuya degradación de prestaciones aumenta desde el 8 % hasta el 34 %, lo que supone un incremento del 26 %. La explicación de esto se encuentra en que `GemsFDTD` es un benchmark esencialmente de cómputo (véase apartado 5.3.2) que presenta un MPKI cercano a cero. Por tanto, en este caso el prefetch únicamente aporta sobrecarga a la red y reemplazos innecesarios en caché, sin otorgar ningún beneficio a la aplicación.

Otro aspecto que puede influir negativamente en los resultados obtenidos es la localidad de cada aplicación. Dependiendo de ésta, el prefetch puede perjudicar las prestaciones en caso de no anticipar los bloques adecuados. El estudio acerca de qué tipo de aplicaciones requiere del uso de técnicas de prebúsqueda dinámicas que resuelva este problema no se encuentra dentro de los objetivos de este trabajo.

En lo que respecta a las ejecuciones realizadas con la técnica VCT, los resultados obtenidos muestran una degradación de prestaciones media a 14 saltos del controlador del 18 %, un 3 % menor respecto a las ejecuciones con VCT sin prefetch. Se trata de la menor degradación de prestaciones media obtenida entre los 4 tipos de ejecuciones realizados hasta el momento.

Sin embargo, tal y como se ha indicado anteriormente, si bien VCT supone un benefecio global para todas las aplicaciones, el prefetch no afecta de igual modo a todas ellas. Por ello, análogamente a como ocurría en las ejecuciones con SAF, casos como `wrf` y `tonto` presentan una reducción en su degradación de prestaciones cercana al 10 %. Por el contrario, las aplicaciones `GemsFDTD, h264ref, povray, sjeng` y `omnetpp` muestran una degradación de prestaciones mayor que en las ejecuciones sin prefetch. El caso de `GemsFDTD` es nuevamente el

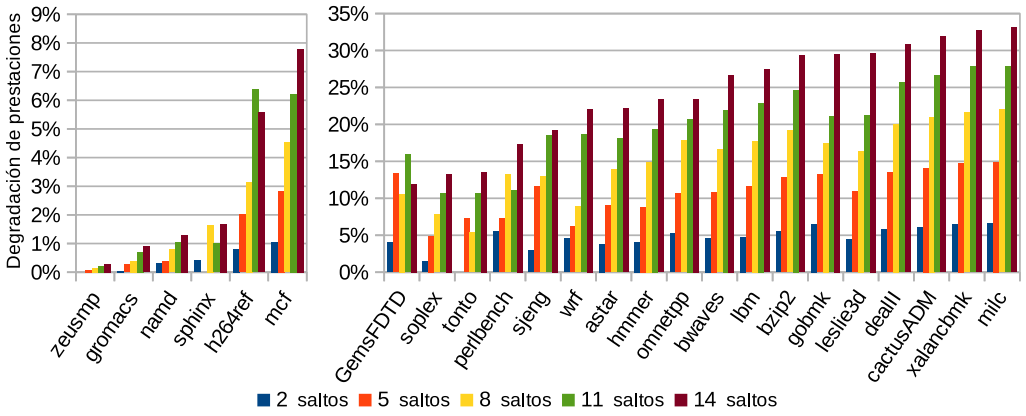


Figura 6.7: Impacto de la prebúsqueda y VCT en las prestaciones de las aplicaciones en orden creciente de izquierda a derecha.

que más acusa esta degradación, incrementándose un 8 % respecto a la ejecución sin prebúsqueda.

Como conclusión a este apartado, la utilización de una técnica de prebúsqueda estática agresiva no permite reducir de forma significativa los efectos negativos de las distancias elevadas dentro de la red. Además, el uso de estas técnicas en ciertas aplicaciones agrava aún más estos efectos, ya que se sobreutiliza la red sin obtener beneficios reales en las prestaciones de la aplicación. Para conseguir una mejora de prestaciones escalable por medio de prefetch en la red en el chip se requiere por tanto de una técnica de prebúsqueda que evite transferir bloques innecesarios por la red. En caso contrario, el coste de transferir estos bloques por la red provocará un incremento de utilización del ancho de banda de la red, lo que en presencia de contención puede llevar a una pérdida de prestaciones aún mayor.

6.1.3 Impacto de la contención de la red en la degradación de prestaciones según la distancia

Los resultados de las ejecuciones en los apartados anteriores se corresponden con la ejecución de las aplicaciones en solitario en la red. En este apartado se presentan los resultados obtenidos tras la ejecución de diversas aplicaciones en compañía de dos cargas que aumentan el tráfico en la red, denominadas *corunners*. Con el objetivo de aumentar el impacto de los corunners y apreciar el impacto en las prestaciones de la aplicación estudiada, éstos serán ubicados entre el núcleo donde se ejecuta la aplicación y el controlador de memoria.

Para limitar el tiempo de estos experimentos se ha seleccionado un subconjunto de aplicaciones que presentan diferentes grados de degradación de prestaciones

en función de la distancia. Las cargas seleccionadas han sido `astar`, `mcf` y `namd` que, utilizando VCT y situadas a 14 saltos del controlador, presentan una degradación de IPC del 25 %, 12 % y 3 %, respectivamente, en ejecución individual.

En lo que respecta a la elección de los corunners, el criterio de selección de los mismos ha sido el MPKI de la caché L2. Esta métrica permite evaluar cómo varía la contención en función de la cantidad de tráfico que introducen los corunners hasta el controlador de memoria. De este modo, los corunners seleccionados han sido `lbm` y `zeusmp`, que presentan un MPKI de 0.03 y 15.54 respectivamente (véase Figura 6.2).

Las simulaciones se han llevado a cabo ubicando dos instancias de cada uno de los corunners junto al núcleo donde se ejecuta la aplicación. Esto significa que, en caso de que la aplicación se ejecutara sobre el núcleo 2 del array, la primera simulación ubicaría una instancia de `zeusmp` en el núcleo 0 y otra en el núcleo 1 del array. A continuación se repetiría el proceso ubicando instancias de `lbm` en lugar de `zeusmp`. Durante estas simulaciones, se han estudiado las prestaciones de la aplicación ubicada en los núcleos 12, 8, 4 y 0 del array[1].

La Figura 6.8 muestra los resultados correspondientes a las simulaciones de la aplicación `astar`. La degradación experimentada por la ejecución con `zeusmp` respecto a la ejecución en solitario ve incrementado el porcentaje medio en 14 puntos en cualquier distancia cuando se utiliza SAF. Esta degradación aumenta aun más (hasta 16 puntos por encima del porcentaje base) cuando el corunner empleado es `lbm`, debido a que se trata de una aplicación que introduce mayor sobrecarga.

Por otro lado, los resultados con VCT presentan un incremento similar, ya que los porcentajes de degradacion base se incrementan entre 15 y 12 puntos porcentuales en las diferentes posiciones. Esto significa que la degradación por contención se mantiene prácticamente constante en todas las ejecuciones, independientemente de la técnica de conmutación empleada. Sin embargo, la degradación total se ve reducida gracias a la reducción de latencia de red que consigue VCT.

La siguiente aplicación estudiada es `mcf`, que presenta una sensibilidad respecto a la distancia al controlador media-baja. Para esta aplicación, tanto en la ejecución realizada con SAF como en la realizada con VCT, el efecto de la contención aumenta la degradación de prestaciones respecto al caso anterior. Así, en el caso de la ejecución junto a `zeusmp`, la degradación media debida únicamente a contención (esto es, degradación total menos el porcentaje de degradación en solitario) supone el 38 %, lo que significa que la degradación en solitario se ha visto multiplicada en un factor de 5. En el caso del corrunner `lbm`, estas cifras son aún mayores ya que la contención incrementa de media en 46 puntos la degradación de prestaciones en las distintas posiciones.

[1] El controlador de memoria se encuentra ubicado en la posición 15 del array (véase Figura 6.1)

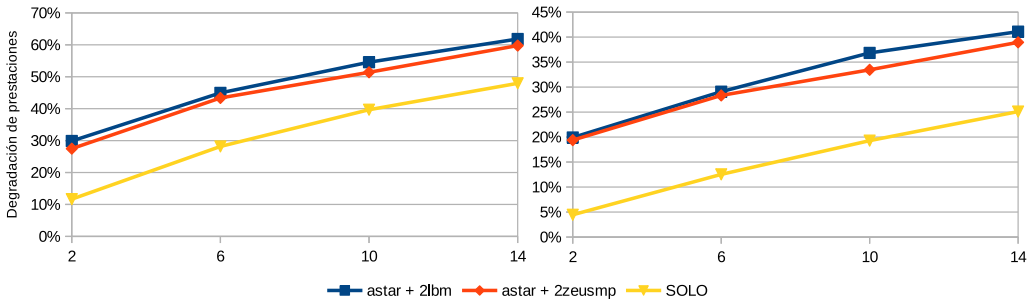


Figura 6.8: Degradación de prestaciones en la aplicación `astar` en su ejecución con dos corunners. A la izquierda, evaluado con SAF; a la derecha, VCT.

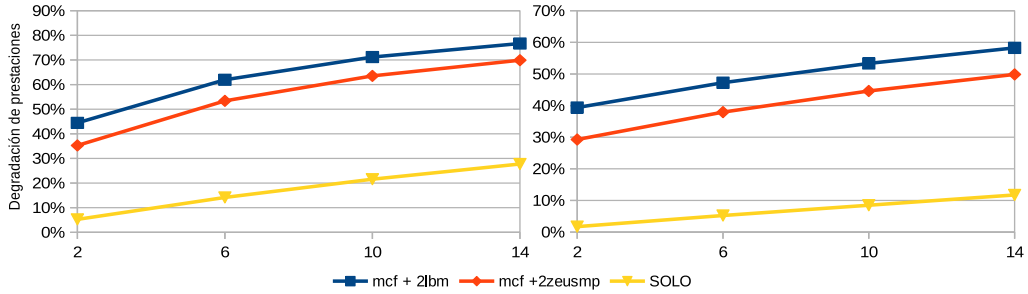


Figura 6.9: Degradación de prestaciones en la aplicación `mcf` en su ejecución con dos corunners. A la izquierda, evaluado con SAF; a la derecha, VCT.

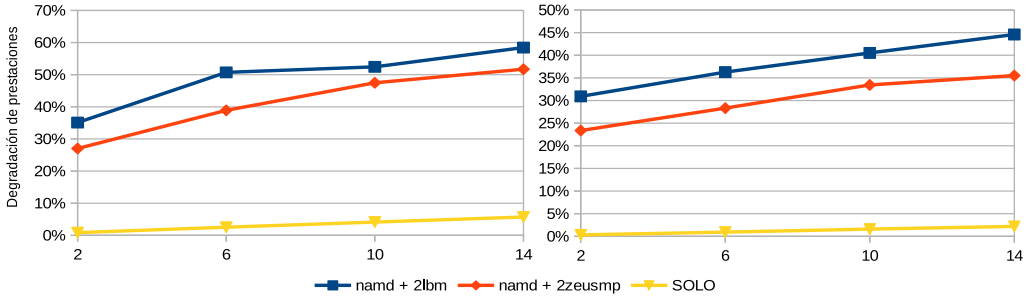


Figura 6.10: Degradación de prestaciones en la aplicación `namd` en su ejecución con dos corunners. A la izquierda, evaluado con SAF; a la derecha, VCT.

A priori este resultado no parece lógico, ya que al contar con menor número de accesos a L2 `mcf` utiliza menos la red y por tanto se ve menos expuesta a sufrir contenciones de tráfico. Sin embargo, la explicación se encuentra en las

IPC utilizando SAF			
Num. saltos	astar	mcf	namd
0	0.683	2.189	2.498
2	0.608	2.074	2.477
4	0.544	1.972	2.456
6	0.494	1.879	2.435
8	0.451	1.794	2.413
10	0.415	1.717	2.395
12	0.384	1.647	2.375
14	0.358	1.584	2.356
IPC utilizando VCT			
Num. saltos	astar	mcf	namd
0	0.711	2.214	2.503
2	0.679	2.176	2.495
4	0.648	2.137	2.487
6	0.622	2.098	2.481
8	0.596	2.061	2.472
10	0.574	2.026	2.463
12	0.552	1.996	2.455
14	0.532	1.954	2.448

Tabla 6.1: IPCs de `astar`, `mcf` y `namd` ejecutadas en solitario

cifras relacionadas con el IPC absoluto de las cargas, que se puede consultar en la Tabla 6.1. Como se observa en esta tabla, el IPC de `mcf` y `namd` es mucho mayor que el de `astar`. Por esta razón, ambas cargas se ven mucho más expuestas a sufrir pérdidas de prestaciones por contención que `astar`, cuyas prestaciones son bajas incluso cuando se ejecuta en solitario.

El mismo análisis se puede realizar teniendo en cuenta los resultados de la ejecución de `mcf` con VCT. En general, VCT permite atenuar la degradación de prestaciones gracias a los efectos positivos que tiene sobre la latencia de la red. Sin embargo, los beneficios de VCT no afectan de igual forma a la degradación debida a la contención. Así, mientras que ejecutándose en solitario la degradación de prestaciones por distancia en `mcf` a distancia 14 alcanza el 11 %, cuando se ejecuta con `zeusmp` y `lbm` este porcentaje se eleva hasta el 49 % y el 59 % respectivamente. Esto significa que la contención a distancia 14 es responsable de más del 80 % de la degradación de prestaciones que experimenta la aplicación.

Los resultados obtenidos para la aplicación `namd` se muestran en la Figu-

ra 6.10. Esta carga presenta resultados similares a los obtenidos con `mcf`, e incluso más acentuados. Esto se debe a que `namd`, tal y como se observa en la Tabla 6.1 es la aplicación que mejor IPC absoluto presenta de todas las seleccionadas. Así, las prestaciones de esta aplicación se degradan drásticamente, suponiendo la contención porcentajes de degradación medios cercanos al 40 % tanto con `zeusmp` como con `lbm`. Obsérvese que en el caso de `namd`, dado que esta carga apenas es sensible a la distancia, la degradación de prestaciones por contención es prácticamente la degradación total que experimenta la aplicación.

El empleo de VCT consigue reducir la degradación media de `namd` en torno a un 8 %. Los beneficios obtenidos por VCT son en este caso menores que los que experimenta `astar`, como resultado de los distintos niveles de tráfico en la red que presentan ambas cargas. La técnica de conmutación no consigue por tanto reducir en este caso de forma significativa la degradación de prestaciones, sino que necesita de algún mecanismo que permita reducir la contención provocada por los corunners.

En resumen, y como conclusión a este apartado, cabe destacar que la importancia de la contención en las prestaciones de las aplicaciones es mayor que la de la distancia que separa a éstas del controlador. Esto se acentúa aún más en casos de aplicaciones que presentan accesos poco frecuentes a memoria, ya que este tipo de cargas presentan IPCs muy altos que se ven muy reducidos cuando son entorpecidas en sus accesos a memoria. Es por tanto necesaria la existencia de mecanismos que permitan un acceso rápido a los controladores de memoria con los que aplicaciones de este tipo no vean tan limitadas sus prestaciones.

6.2 Evaluación de red híbrida

En este apartado se muestran los resultados correspondientes a la evaluación de la propuesta de red híbrida expuesta en el Capítulo 4. La evaluación ha consistido en ubicar las aplicaciones en solitario en varios núcleos del CMP situados a distancias diferentes de los controladores de memoria. Así, los núcleos seleccionados se encuentran en los nodos 0, 2, 4, 6, 15, 31, 47 y 63 de la NoC. Las distancias que separan a estos núcleos de ambos controladores se pueden consultar en la Tabla 6.2. Por otro lado, para garantizar un acceso equilibrado a ambos controladores el sistema cuenta con memoria entrelazada a nivel de página.

El objetivo de estos experimentos es reducir, por medio de la red óptica, el impacto en la latencia que sufre una aplicación cuando accede a un controlador lejano. De esta forma, las prestaciones de dicha aplicación pueden verse beneficiadas al obtener el bloque de memoria mucho antes.

Para observar el impacto en las prestaciones, se han realizado cuatro tipos de simulaciones diferentes. En primer lugar, se han realizado ejecuciones tanto con

Nodo	Controlador 0	Controlador 1
N0	0	14
N2	2	12
N4	4	10
N6	6	8
N15	8	6
N31	10	4
N47	12	2
N63	14	0

Tabla 6.2: Distancia desde cada posición estudiada hasta los controladores de memoria.

SAF como con VCT en una malla bidimensional de características idénticas a la utilizada en la red híbrida. Esta red será utilizada para medir las prestaciones en ausencia de fotónica.

Por otro lado, se han ejecutado las cargas sobre la red híbrida utilizando SAF y VCT en la malla subyacente. Estos resultados junto a los obtenidos en las ejecuciones anteriores serán contrastados con el mejor caso posible, que se corresponde con las prestaciones obtenidas por cada carga ejecutada en solitario junto a un controlador de memoria. De esta forma se podrá apreciar con claridad qué esquemas presentan más degradación de prestaciones respecto a este mejor caso, así como qué soluciones permiten reducir esta degradación.

6.2.1 Degradación de prestaciones en malla y red híbrida

Los resultados correspondientes a la degradación de prestaciones de la malla con SAF se muestran en la Figura 6.11. A diferencia de los resultados mostrados en el capítulo anterior, la degradación de prestaciones en este caso se presenta equilibrada en las distintas posiciones de la malla. Esto se debe a la presencia de dos controladores de memoria, lo que equilibra el número de accesos que se realizan a distancia elevada y distancia reducida. Pese a esto, la degradación de prestaciones general es significativamente elevada, con una media del 27 % entre todas las cargas. Además, cargas como `cactusADM` alcanzan picos del 43 % de degradación de IPC en todas las posiciones.

En el límite inferior encontramos la carga `zeusmp`. Tal y como se ha comentado anteriormente, ésta es una carga que apenas presenta accesos a memoria por lo que apenas modifica su IPC respecto al mejor caso. Esto queda patente en la gráfica, donde `zeusmp` presenta una degradación de prestaciones casi nula, en

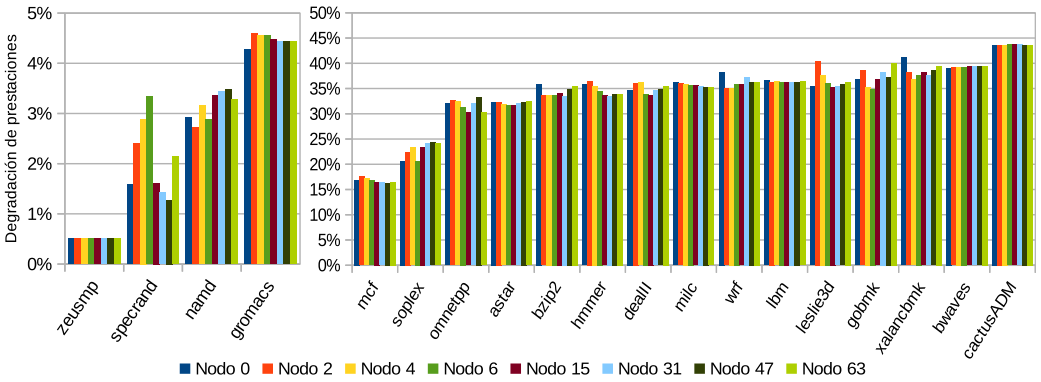


Figura 6.11: Degradación de prestaciones respecto al mejor caso en los nodos 0, 2, 4, 6, 15, 31, 47 y 63 de la malla con SAF.

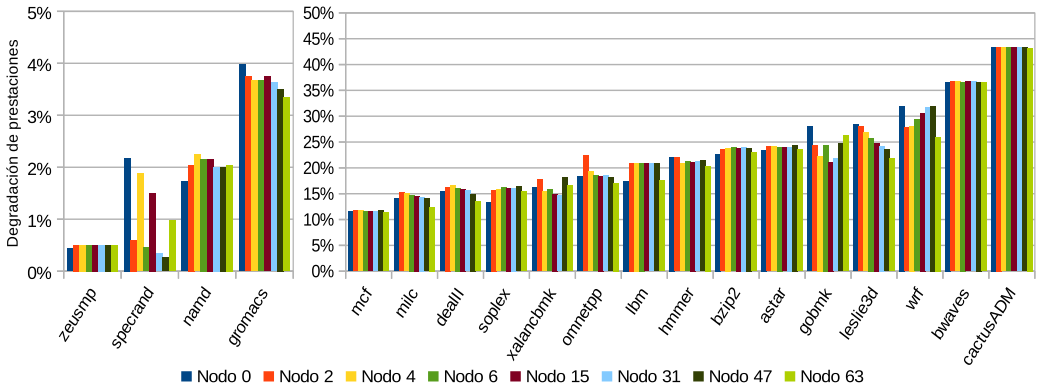


Figura 6.12: Degradación de prestaciones respecto al mejor caso en los nodos 0, 2, 4, 6, 15, 31, 47 y 63 de la red híbrida con SAF.

concreto del 0.5 %.

En la Figura 6.12 se presenta la degradación de prestaciones que experimenta la red híbrida con SAF respecto al mejor caso. Tal y como se puede apreciar, el porcentaje de degradación se ve reducido en prácticamente la totalidad de las cargas. Concretamente, la red híbrida consigue reducir el porcentaje de degradación de la malla anterior de un 27 % a un 18 %, una reducción muy significativa teniendo en cuenta que se está utilizando la técnica de conmutación VCT.

Resulta llamativo que la reducción de la degradación de IPC tiene lugar también incluso en aquellas aplicaciones que ya presentan una baja degradación. Estas aplicaciones reducen la degradación de prestaciones media en porcentajes que oscilan entre el 1 % y el 2 %, reduciendo prácticamente en su totalidad el efecto

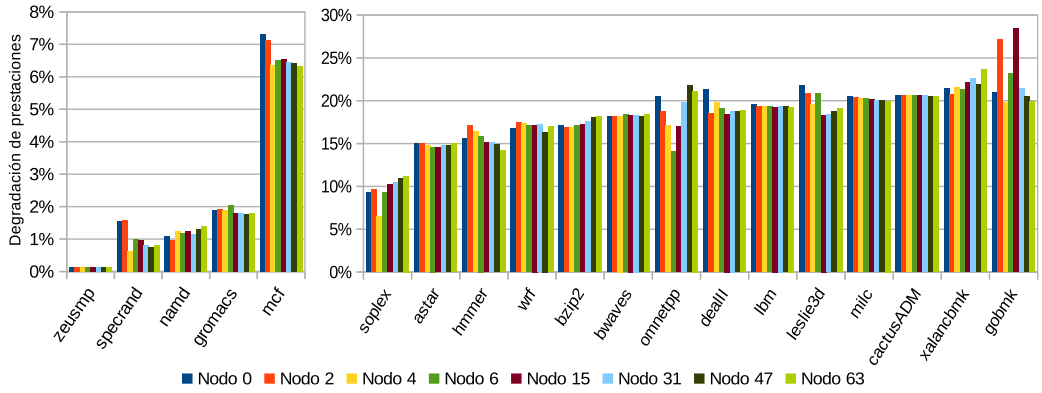


Figura 6.13: Degradación de prestaciones respecto al mejor caso en los nodos 0, 2, 4, 6, 15, 31, 47 y 63 de la malla con VCT.

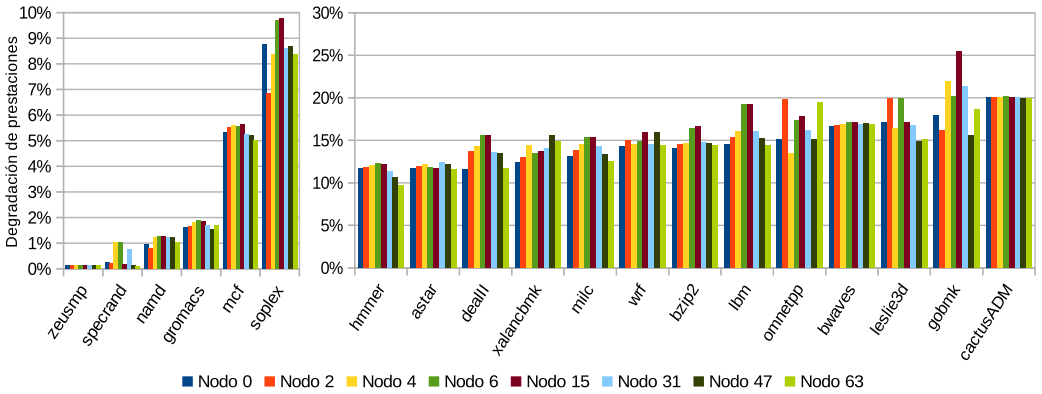


Figura 6.14: Degradación de prestaciones respecto al mejor caso en los nodos 0, 2, 4, 6, 15, 31, 47 y 63 de la red híbrida con VCT.

negativo de la distancia respecto al controlador.

En lo que respecta a las cargas que más acusan la distancia, la mayor reducción de la degradación la encontramos en la carga `milc`. En esta aplicación se consigue reducir el porcentaje medio de degradación de prestaciones desde un 35 % hasta un 14 %. Además, tal y como se observa en la gráfica, el porcentaje medio en este caso es una métrica fiable ya que las aplicaciones presentan porcentajes similares en cualquiera de las posiciones en las que han sido ubicadas. Nuestra propuesta permite, por tanto, seleccionar la mejor red en función de dónde se sitúa la aplicación en ejecución dentro de la malla.

Los resultados de las ejecuciones con VCT pueden ser consultados en las Figuras 6.13 y 6.14. Debido a los beneficios en la latencia de la red que proporciona

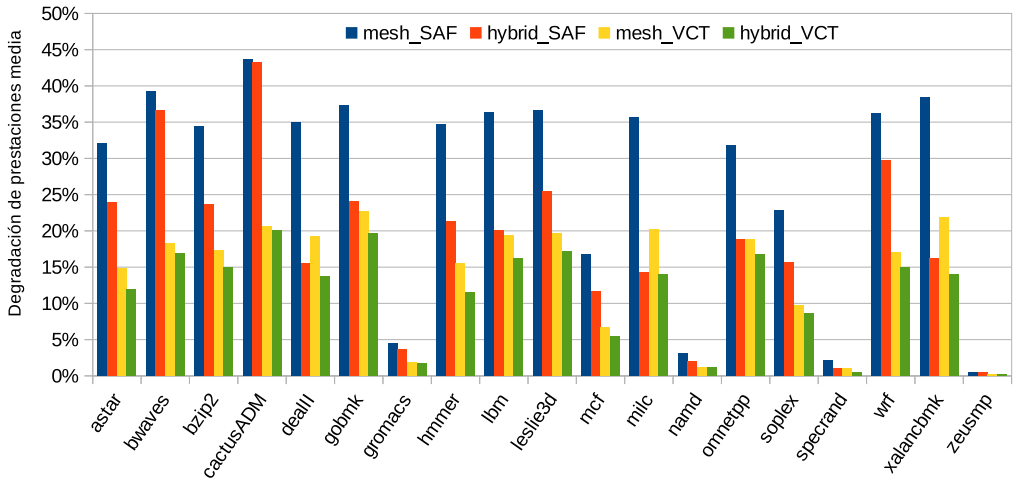


Figura 6.15: Degradación de prestaciones media de las aplicaciones en las cuatro configuraciones estudiadas.

VCT, la reducción en la degradación de prestaciones en la red híbrida es en este caso menor que en el anterior. Sin embargo, siguen existiendo mejoras en los resultados ya que se reduce el porcentaje medio de degradación de todas las cargas en torno a un 4 %.

Aunque la red híbrida reduce la degradación de prestaciones, existen ciertos efectos intrínsecos a la utilización del anillo que no deben ser pasados por alto. Por ejemplo, en el caso de la aplicación `lbm`, las ejecuciones realizadas en las posiciones 15 y 31 apenas han experimentado mejoras respecto a las ejecuciones en el resto de ubicaciones. Lo mismo ocurre con la ejecución en las posiciones 31 de `gobmk` y 15 y 31 de `dealII`. Estos comportamientos se deben a que el anillo óptico, pese a operar a una alta frecuencia, cuenta con un ancho de banda reducido (4 bits/ciclo), lo que provoca que este alcance niveles de saturación cuando recibe ráfagas de peticiones prolongadas.

Pese a esto, la red híbrida propuesta contribuye, junto con VCT, a reducir el porcentaje medio inicial de degradación del 27 % experimentado por la malla con SAF hasta un 11 %. Esto se puede observar en la Figura 6.15, donde se presentan las degradaciones medias de las cuatro combinaciones empleadas.

En esta última Figura queda patente cómo la red híbrida propuesta presenta la menor degradación de prestaciones entre todas las configuraciones con las que se han realizado los experimentos. Ahora bien, en las combinaciones de malla con VCT y red híbrida con SAF los resultados son dispares. Las aplicaciones `milc`, `xalancbmk` y `dealII` presentan mejores resultados en la red híbrida con SAF que en la malla con VCT. En estas cargas, debido al uso intensivo que se hace de

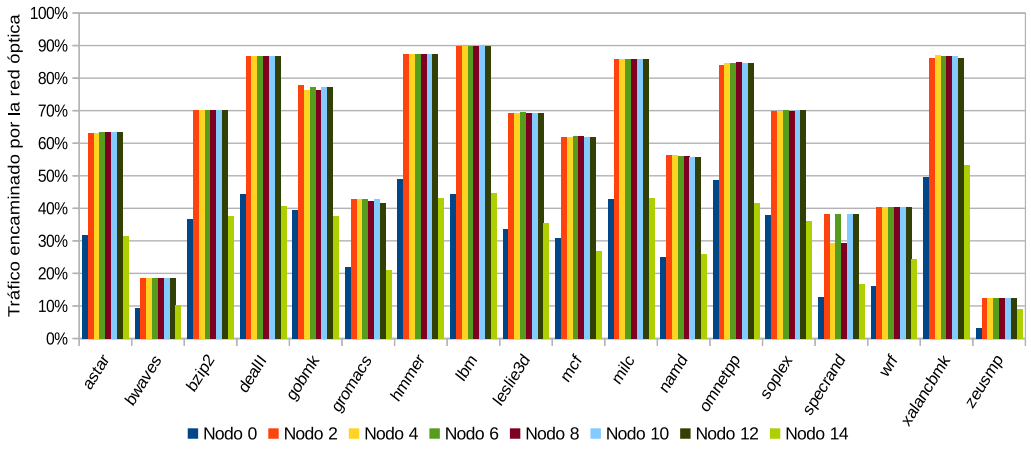


Figura 6.16: Porcentaje de tráfico encaminado por la red fotónica en cada nodo cuando se utiliza SAF.

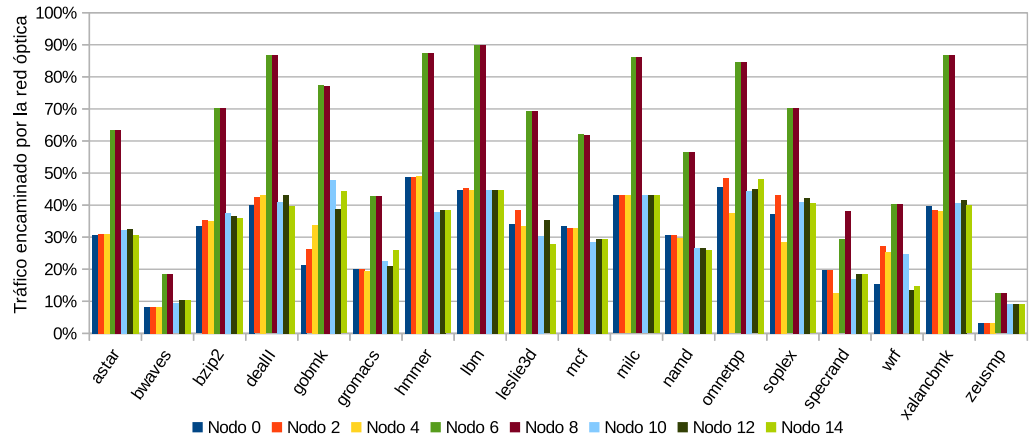


Figura 6.17: Porcentaje de tráfico encaminado por la red fotónica en cada nodo cuando se utiliza VCT.

la red, nuestra propuesta mejora las prestaciones en mayor medida que VCT pese al ancho de banda reducido con el que cuenta el anillo.

En resumen, el estudio realizado demuestra la eficacia de la red híbrida propuesta como complemento de la técnica de conmutación VCT. Ambos mecanismos consiguen reducir el impacto negativo que suponen en las prestaciones las elevadas distancias en la red entre los núcleos de cómputo y los controladores de memoria. La red híbrida propuesta permite así compensar situaciones en las que VCT no es suficiente por sí mismo, en casos como aplicaciones que presen-

tan numerosos accesos a memoria o en los que la distancia al controlador crece excesivamente.

6.2.2 Distribución del tráfico en la red híbrida

El modelo de selección de red presentado en los apartados 4.1.1 y 5.2.6 tiene como objetivo decidir qué red es la encargada de transmitir cada mensaje. Como ya se ha explicado anteriormente, este modelo tiene en cuenta las latencias teóricas estimadas tanto para SAF como para VCT. Esto se debe a que la latencia experimentada por un mensaje cuando se está trabajando con SAF es mucho mayor, por lo que en esta situación es mejor utiizar el anillo desde un mayor número de núcleos. Para comprobar la eficacia del modelo implementado, en este apartado se presentan las gráficas correspondientes a la distribución de tráfico que tiene lugar en la red híbrida en las ejecuciones anteriores. Las gráficas presenta el porcentaje de bytes que han sido transmitidos por la red óptica respecto del total, mostrando así el porcentaje de tráfico que ha sido entregado por esta red.

En la Figura 6.16 se presentan los porcentajes de tráfico relativos a las ejecuciones con SAF. Como se puede observar, las ejecuciones ubicadas en los nodos 2, 4, 6, 15, 31 y 47 presentan un porcentaje mucho mayor de tráfico encaminado por la red fotónica. Esto se debe a que estos nodos superan, para tamaños de mensaje elevados (típicamente 72 bytes), el umbral de latencia teórico estimado hasta el controlador. El empleo de SAF hace que el umbral se alcance antes, por lo que únicamente los nodos ubicados en las posiciones 0 y 63 (colindantes con los controladores) presentan un porcentaje de tráfico por la red fotónica reducido.

Por otro lado, los resultados correspondientes a la red híbrida con VCT se muestran en la Figura 6.17. En este caso, únicamente los nodos 6 y 15 son los que presentan mayor porcentaje de tráfico encaminado por la red fotónica. Se aprecia, por tanto, el efecto de VCT en estos resultados. En este caso, la latencia teórica calculada por el modelo es mucho menor que al utilizar SAF, por lo que el umbral necesario para utilizar la red óptica se alcanza a una mayor distancia del controlador.

Capítulo 7

Conclusiones

El principal objeto de estudio de este Trabajo Fin de Máster ha sido la escalabilidad de las redes dentro del chip en procesadores multinúcleo. En vista de los futuros requisitos de escalabilidad que se espera de este tipo de procesadores, en este trabajo hemos evaluado qué limites puede suponer la red en los mismos. Para ello, hemos estudiado cómo afecta el aumento del diámetro de la red en las prestaciones de las aplicaciones que se ejecutan en el CMP. En primer lugar, los resultados correspondientes a la latencia de red muestran que ésta puede alcanzar cotas superiores de hasta el 90 % cuando el diámetro alcanza los 14 saltos entre nodos de cómputo y controlador de memoria. Este incremento en la latencia de red supone una degradación en las prestaciones excesiva para algunas cargas, en especial para aquellas que presentan frecuentes accesos a memoria.

Además, este estudio preliminar ha puesto de manifiesto la importancia de la técnica de conmutación que utiliza la red. Cuando la técnica seleccionada es Store & Forward, conforme crece el número de nodos de la red la degradación de prestaciones experimentada por las cargas hace que la escalabilidad sea insostenible. Sin embargo, cuando la técnica utilizada es Virtual Cut-Through, el incremento de latencia y la degradación de IPC de las cargas son menos significativos.

Pese a esto, los resultados obtenidos muestran claramente un límite en la escalabilidad de las mallas eléctricas tradicionales conforme aumenta el número de nodos. Basándonos en esta premisa, en este Trabajo Fin de Máster proponemos una red híbrida que utiliza tanto una malla bidimensional convencional como un anillo óptico para conectar los distintos nodos con los controladores de memoria. El objetivo de esta red híbrida es conseguir reducir la latencia de las comunicaciones de red que se realizan entre nodos y controladores muy alejados entre sí. De esta forma, bloques servidos por el controlador que se verían ralentizados en su transmisión por la malla son obtenidos rapidamente por medio del anillo óptico.

7.1 Contribuciones

Este Trabajo Fin de Máster presenta tres contribuciones principales en los campos de las redes dentro del chip y las redes fotoeléctricas.

La primera contribución es la caracterización de las cargas multiprogramadas SPEC2006 respecto a cómo se ven afectadas por la distancia que las separa del controlador de memoria. Esta caracterización se ha realizado teniendo en cuenta distintos factores. En primer lugar se ha estudiado cómo afecta la distancia a las diferentes cargas multiprogramadas ejecutadas en solitario en la red utilizando tanto la técnica de conmutación SAF como VCT. En este punto, los resultados muestran que no todas las cargas ven afectadas sus prestaciones de igual forma conforme se alejan del controlador de memoria. En particular, aquellas cargas que presentan un MPKI más bajo son las que menos impacto reciben en estos casos. Por contra, aplicaciones que presentan frecuentes accesos a memoria y que, por tanto, hacen un uso más prolongado de la red, pueden llegar a ver degradadas sus prestaciones en porcentajes de hasta el 60 %.

La segunda contribución de este trabajo es el estudio de diversos factores que pueden incrementar la degradación de prestaciones experimentada por las distintas aplicaciones. Estos factores han sido la técnica de conmutación empleada, la utilización de técnicas de prebúsqueda agresiva y la presencia de contención en la red. Los experimentos realizados en técnicas de conmutación apuntan que VCT consigue reducir hasta en un 17 % la degradación media de todas las cargas cuando éstas se ejecutan a distancia 14 del controlador. Sin embargo, el uso de técnicas de prebúsqueda agresivas no reduce de forma significativa los efectos negativos de las elevadas distancias en la red. Es más, en algunas aplicaciones las prestaciones se ven aún más degradadas debido a la sobreutilización que estas técnicas provocan sobre la red. Finalmente, la presencia de contención ha demostrado ser un problema todavía mayor que la distancia que separa a las diferentes cargas del controlador, con porcentajes de degradación que superan el 20 % en todos los casos estudiados.

Por último, la tercera contribución de este trabajo es la propuesta de una red híbrida que pretende reducir la latencia de acceso al controlador de las aplicaciones que se encuentran demasiado lejanas al mismo. Dicha red híbrida cuenta con un modelo teórico de selección de red que estima la latencia de la malla bidimensional en función de la técnica de conmutación que ésta utiliza. Utilizando estos datos el modelo determina cuál de las dos redes disponibles es la mejor opción para realizar el acceso al controlador, escogiendo en todo caso la que menos ciclos necesita para entregar el paquete. Los resultados obtenidos muestran que la red híbrida que utiliza VCT presenta una degradación de prestaciones media del 11 %, mejorando en todo caso los resultados obtenidos en las ejecuciones sobre la malla eléctrica.

7.2 Trabajo futuro

Como trabajo futuro, planeamos extender la propuesta de red híbrida incluida en este Trabajo Fin de Máster teniendo en cuenta los distintos aspectos estudiados anteriormente. Las diferentes líneas de investigación a este respecto son las siguientes:

- Modelado de la contención. El modelo actual cuenta con un registro de las latencias que experimentan las operaciones de *load* del procesador. Esto da una noción acerca del estado en el que se encuentra la red en el momento de realizar la transmisión correspondiente. En implementaciones futuras se propone utilizar esta información para elaborar un modelo detallado de contención que estime de una manera más precisa qué red es la adecuada para el siguiente envío.

- Modelado de arbitraje óptico por *tokens*. Muchos de los esquemas ópticos expuestos requieren de arbitraje global o de técnicas de *buffering* cuando varios emisores seleccionan un mismo destino. La implementación en el entorno Multi2Sim de técnicas de arbitraje por tokens permitirá ampliar las opciones de reparto de wavelengths entre nodos y controladores, así como utilizar esquemas más dinámicos y flexibles.

- Red fotónica completa. La implementación de una red óptica completamente funcional permitirá estudiar de forma más detallada los requisitos de la misma en presencia de tráfico intenso. Esta implementación constituye la base para un estudio experimental de los diferentes esquemas de comunicación ópticos explicados en este trabajo.

- Mecanismo de *core-allocation* para la malla bidimensional. A la vista de los resultados obtenidos en diferentes cargas, se propone la implementación de un mecanismo que permita ubicar las aplicaciones en los nodos más adecuados. De esta forma, aplicaciones que apenas sufren por la distancia pero sí por contención pueden ser ubicadas en un nodo extremo de la red que cuenta con conexión óptica directa con el controlador de memoria. Por contra, aplicaciones muy sensibles a la distancia pueden ser ubicadas en nodos contiguos a los controladores de manera que se beneficien del buen rendimiento de la malla en distancias reducidas.

7.3 Publicaciones

Parte del trabajo de investigación presentado en este Trabajo Fin de Máster ha sido aceptado para publicación en la siguiente conferencia:

- J. Puche, J. Duro, S. Petit, M. Gómez y J. Sahuquillo, “Estudio de exploración sobre los beneficios de la tecnología fotónica en manycores”, en *XXVI Jornadas de Paralelismo* (JP2015).

Bibliografía

[1] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. Beausoleil, and J. Ahn, "Corona: System implications of emerging nanophotonic technology," in *Computer Architecture, 2008. ISCA '08. 35th International Symposium on*, June 2008, pp. 153–164.

[2] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary, "Firefly: Illuminating future network-on-chip with nanophotonics," *SIGARCH Comput. Archit. News*, vol. 37, no. 3, pp. 429–440, Jun. 2009. [Online]. Available: http://doi.acm.org/10.1145/1555815.1555808

[3] C. Nitta, M. Farrens, and V. Akella, "Dcaf - a directly connected arbitration-free photonic crossbar for energy-efficient high performance computing," in *Parallel Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International*, May 2012, pp. 1144–1155.

[4] A. Shacham, K. Bergman, and L. Carloni, "Photonic networks-on-chip for future generations of chip multiprocessors," *Computers, IEEE Transactions on*, vol. 57, no. 9, pp. 1246–1260, Sept 2008.

[5] A. C. Bergman K. Carloni, L. P. Bibermani and H. G., *Photonic Network-on-Chip Design*. Springer-Verlag New York, 2014, vol. 68, no. 1.

[6] Q. Xu, D. Fattal, and R. G. Beausoleil, "Silicon microring resonators with 1.5-μm radius," *Opt. Express*, vol. 16, no. 6, pp. 4309–4315, Mar 2008. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-16-6-4309

[7] S. Abadal, A. Cabellos-Aparicio, J. A. Lazaro, E. Alarcon, and J. Sole-Pareta, "Graphene-enabled hybrid architectures for multiprocessors: Bridging nanophotonics and nanoscale wireless communication," in *Transparent Optical Networks (ICTON), 2012 14th International Conference on*, July 2012, pp. 1–4.

[8] J. Pang, C. Dwyer, and A. R. Lebeck, "Exploiting emerging technologies for nanoscale photonic networks-on-chip," in *Proceedings of the Sixth International Workshop on Network on Chip Architectures*, ser. NoCArc '13. New York, NY, USA: ACM, 2013, pp. 53–58. [Online]. Available: http://doi.acm.org/10.1145/2536522.2536525

[9] A. Shacham, K. Bergman, and L. Carloni, "On the design of a photonic network-on-chip," in *Networks-on-Chip, 2007. NOCS 2007. First International Symposium on*, May 2007, pp. 53–64.

[10] M. Petracca, K. Bergman, and L. Carloni, "Photonic networks-on-chip: Opportunities and challenges," in *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, May 2008, pp. 2789–2792.

[11] G. Kurian, J. E. Miller, J. Psota, J. Eastep, J. Liu, J. Michel, L. C. Kimerling, and A. Agarwal, "Atac: A 1000-core cache-coherent processor with on-chip optical network," in *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '10. New York, NY, USA: ACM, 2010, pp. 477–488. [Online]. Available: http://doi.acm.org/10.1145/1854273.1854332

[12] Y. Xu, J. Yang, and R. Melhem, "Tolerating process variations in nanophotonic on-chip networks," in *Computer Architecture (ISCA), 2012 39th Annual International Symposium on*, June 2012, pp. 142–152.

[13] Y. Pan, J. Kim, and G. Memik, "Flexishare: Channel sharing for an energy-efficient nanophotonic crossbar," in *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, Jan 2010, pp. 1–12.

[14] A. García-Guirado, R. Fernández-Pascual, J. M. García, and S. Bartolini, "Managing resources dynamically in hybrid photonic-electronic networks-on-chip," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 15, pp. 2530–2550, 2014. [Online]. Available: http://dx.doi.org/10.1002/cpe.3332

[15] D. Vantrease, N. Binkert, R. Schreiber, and M. Lipasti, "Light speed arbitration and flow control for nanophotonic interconnects," in *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*, Dec 2009, pp. 304–315.

[16] C. Batten, A. Joshi, V. Stojanovic, and K. Asanovic, "Designing chip-level nanophotonic interconnection networks," *Emerging and Selected Topics in*

Circuits and Systems, IEEE Journal on, vol. 2, no. 2, pp. 137–153, June 2012.

[17] C. Li, M. Browning, P. Gratz, and S. Palermo, "Luminoc: A power-efficient, high-performance, photonic network-on-chip," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 33, no. 6, pp. 826–838, June 2014.

[18] R. Ubal, J. Sahuquillo, S. Petit, and P. Lopez, "Multi2sim: A simulation framework to evaluate multicore-multithreaded processors," in *International Symposium on Computer Architecture and High Performance Computing.*, 2007, pp. 62–68.

[19] SPEC website. [Online]. Available: http://www.spec.org/