Using Heterogeneous Networks to Improve Energy Efficiency in Direct Coherence Protocols for Many-Core CMPs

Alberto Ros, Ricardo Fernández-Pascual, and Manuel E. Acacio
Universidad de Murcia, Spain
Email: {aros,rfernandez,meacacio}@ditec.um.es

***Abstract*—Direct coherence protocols have been recently proposed as an alternative to directory-based protocols to keep cache coherence in many-core CMPs. Differently from directory-based protocols, in direct coherence the responsible for providing the requested data in case of a cache miss (i.e., the *owner* cache) is also tasked with keeping the updated directory information and serializing the different accesses to the block by all cores. This way, these protocols send requests directly to the owner cache, thus avoiding the indirection caused by accessing a separate directory (usually in the *home* node). A *hints* mechanism ensures a high hit rate when predicting the current owner of a block for sending requests, but at the price of significantly increasing network traffic, and consequently, energy consumption. In this work, we show how using a heterogeneous interconnection network composed of two kinds of links is enough to drastically reduce the energy consumed by hint messages, obtaining significant improvements in energy efficiency.**

***Keywords*-Cache coherence, heterogeneous networks, direct coherence.**

I. Introduction

Most of todays mainstream multicore processors (chip-multiprocessors or CMPs) employ the well-known shared memory paradigm as the low-level communication abstraction. In a shared memory CMP, the cores communicate through load and store instructions to a shared address space. Each processor core uses one or several levels of private caches to reduce both the average memory latency and memory traffic. Although beneficial, the use of private caches leads to the possibility of incoherence when accessing shared data (i.e., two different cores can observe distinct versions of the same data). To prevent such situations from happening, a cache coherence protocol implemented in hardware is incorporated in current CMP designs.

The cache coherence protocol is responsible for tracking the state of each memory block, invalidating all copies when one core wants to write the block and creating several copies when different cores read it. In this way, the cache coherence protocol makes caches functionally invisible to software. This hardware-managed, implicitly-addressed, coherent caches memory model is expected to also be implemented in future many-core CMPs [1], [2].

On the other hand, future many-core CMPs will probably be designed as arrays of identical or close-to-identical building blocks (tiles) connected over an on-chip switched direct network [3]. Each tile contains at least one processing core, caches and a connection to the on-chip network. These tiled architectures have been claimed to provide a scalable solution for managing the design complexity, and effectively using the resources available in advanced VLSI technologies.

As the number of cores increases, the cache coherence protocol turns into a key element in the performance and power consumption of the whole CMP. Directory-based cache coherence protocols have been typically employed in systems with direct networks, as tiled CMPs are. The directory structure is distributed between the last-level cache banks (L2 in this work), usually included into the tags portion [4]. In this way, each tile keeps the sharing information of the blocks mapped to the L2 cache bank that it contains (i.e., the *home* node). Unfortunately, these protocols introduce indirection to obtain coherence information from the directory, thus increasing cache miss latencies.

To remedy this deficiency of directory protocols, and therefore, improve their performance, we proposed the direct coherence protocols [5], which are especially suited to many-core tiled CMP architectures [6]. In DiCo-CMP (i.e., direct coherence protocols for CMPs) the task of storing up-to-date sharing information and ensuring ordered accesses for every memory block is assigned to the cache that provides the block on a miss, i.e., the owner cache. Therefore, DiCo-CMP reduces the miss latency compared to a directory protocol by sending requests directly to the owner cache. To this end, the identity of the owner caches in DiCo-CMP is speculatively recorded in a small structure called L1 coherence cache, associated to each core, which is updated whenever the owner tile changes through control messages called *hints*. Although the use of hints ensures accurate owner predictions, it increases network traffic, and consequently, the amount of energy consumed in the interconnection network, thus compromising the energy efficiency of DiCo-CMP. To illustrate this problem, Fig. 1 shows the fraction of critical and non-critical messages generated by both the directory protocol used as baseline in this work (Dir-CMP) and DiCo-CMP. As it can be seen, on average, more than half of the messages that travel through the network in DiCo-CMP are non-critical (53.5%), while

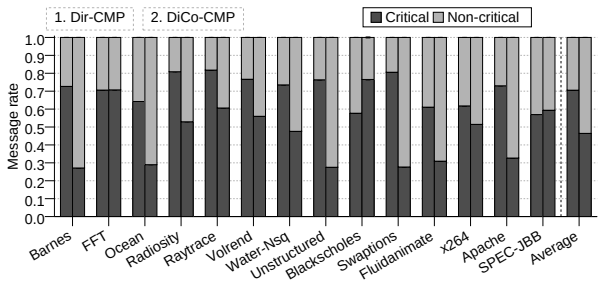


Figure 1: Critical vs. non-critical message ratio in Dir-CMP and DiCo-CMP

this number is significantly lower for Dir-CMP (29.5%).

In this work, we address the energy efficiency problem of DiCo-CMP, demonstrating that the use of a heterogeneous network comprised of two kinds of links with varying physical characteristics greatly benefits this protocol. More precisely, we make two observations about the hint messages causing the problem: first, and most importantly, these messages are non-critical (i.e., delaying a hint message does not have direct impact on the cache miss that generates it); and second, hint messages are short. Based on these two observations we propose to employ a heterogeneous network that has a set of cheap low-power, low-bandwidth links for transmitting non-critical messages. We show that by using this network the network energy consumption is drastically reduced (by 25%). This is achieved without jeopardizing performance, since the delayed messages are not in the critical path of cache misses. It is important to note that since most of the messages generated by Dir-CMP are in the critical path of a cache miss, it can hardly benefit from such a network organization [7].

The rest of the document is organized as follows. In Section II we present a review of the related work. Section III describes the criticality of the coherence messages for directory and direct coherence protocols. Several heterogeneous networks focused on either reducing performance or reducing area are proposed in Section IV. Section V introduces the methodology employed in the evaluation and Section VI shows the performance and energy results obtained when applying heterogeneous networks to both protocols. Finally, Section VII draws some conclusions.

II. Related Work

One of the major bottlenecks to high performance and energy efficiency in tiled CMP architectures is the high cost of on-chip communications. Previous works have demonstrated that a very significant fraction of this power is dissipated in the point-to-point links of the interconnection network [8], [7]. Thus, wires pose important power dissipation problems as technology shrinks and total die area increases.

Wires can be designed with varying latency and bandwidth properties by tuning characteristics such as wire width and spacing. Similarly, it is possible to design wires with varying latency and energy properties by tuning repeater size and spacing [9]. Therefore, using links that are composed of wires with different physical properties, different number of wires, and spacing between wires, we lead to a heterogeneous on-chip interconnection network.

As communication emerges as a larger power and performance constraint than computation itself, link properties should be exposed to architects in order to enable them to find ways to exploit these properties. Cheng *et al.* [10] proposed a heterogeneous interconnect made up of three wire implementations: power optimized wires (PW-Wires) that have fewer and smaller repeaters, bandwidth optimized wires (L-Wires) with bigger widths and spacing, and baseline wires (B-Wires). Then, coherence messages are mapped to the appropriate set of wires taking into account, among others, their latency and bandwidth requirements. The authors show that with such a heterogeneous interconnect, a reduction in both execution time and energy consumption is obtained for a CMP with a two-level tree interconnect topology. Unfortunately, they report insignificant performance improvements for the direct network topologies employed in tiled CMPs.

Subsequently, Flores *et al.* [7] simplified the design of the heterogeneous network by considering just two types of wires, low-latency (L-Wires) for critical messages and low-energy (PW-Wires) for non-critical ones. Along with this, they propose a *request partitioning* technique that allows every coherence message to be classified into two groups: 1) critical and short and 2) non-critical and long messages.

Differently from these prior works, this paper studies the ability of a heterogeneous network to improve energy efficiency, particularly, in DiCo-CMP. To this end, we propose two different heterogeneous network configurations composed of two kinds of wires: baseline wires (B-Wires) and power optimized wires (PW-Wires).

III. Heterogeneous Networks for Power-Efficient Coherence

Coherence messages exchanged between nodes can be classified according to their criticality. A message is considered critical if its delivery latency affects directly the latency of a cache miss. That is, when it is in the critical path of the coherence transaction that resolves a cache miss.

If we identify which protocol messages are critical and which ones are not, we can take advantage of a heterogeneous network for reducing energy consumption without increasing neither miss latency nor network traffic. The key observation is that messages out of the critical path of cache misses can be sent through low power links even if this implies a higher latency, as long as the consequent increase in latency for these messages is not so large that they arrive "too late".

When a message arrives too late, overall performance can be affected even if the latency of the miss that caused the

message to be sent is unaffected. This can happen due to two reasons:

- The message is required to arrive before a subsequent cache miss (different than the one that originated the message) can be attended. For example, in many directory-based protocols, if the directory cache receives two consecutive requests for the same address, it will not attend the second one until it receives a finalization message (called UNBLOCK) for the first one. Note that the UNBLOCK message sent for the first transaction is not critical for that miss, but it can become critical for the second one. We call these messages *indirectly critical*.
- The information conveyed by the message can become obsolete or the chance of taking advantage of that information can be lost. For example, if the information conveyed by *hint* messages arrives too late, the accuracy of predictions will decrease. However, since those messages are used by the protocol only for updating information used for predictions (soft state) which, by definition, can be inexact, no node will ever wait the arrival of one of those messages before attending any request. We call these messages *non-critical*.

We should note that even if a message is considered non-critical, its latency cannot be increased arbitrarily without affecting the execution time. If this were possible, it would mean that the protocol did not actually need such a message.

Considering this, the more non-critical messages a cache coherence protocol uses and the more they can be delayed without affecting the execution time, the more advantage we can take from a heterogeneous network for reducing its energy consumption.

We have classified the coherence messages of the protocols considered in this work as described in the following two sections. We consider the directory implementation provided in the GEMS simulator [11] and the direct coherence implementation proposed in [6].

A. Dir-CMP Protocol

- **Critical messages:** All request messages except those related with replacements[1], and every response except UNBLOCK messages and the final message of a replacement (which also conveys data). Hence, this category includes request messages sent by nodes (both read and write requests), forwarded requests, response messages with data, invalidation messages and invalidation acknowledgments.
- **Indirectly critical messages:** Transaction finalization messages (UNBLOCK, as explained above), replacement finalization messages (WRITEBACK_DATA) and replacement acknowledgments. Dir-CMP uses three-phase write-backs from L1 to L2: first a writeback initiation message (PUT) is sent which is answered by the L2 with an acknowledgment (ACK) when it is ready for accepting the data (usually after performing a replacement from L2 to memory). While the PUT message is not critical, once the ACK is sent the home node becomes locked and cannot attend the next request for the address involved until the last message arrives to the *home* node.
- **Non-critical messages:** Writeback initiation messages (PUT). This initiation message is neither critical nor indirectly critical because both the directory and the L2 can attend requests for the same address while the ACK has not been sent. The delay of the PUT message only affects to the size of the writeback buffer.

[1]These would be in the critical path if there were no writeback buffer.

B. DiCo-CMP Protocol

- **Critical messages:** Just like in the case of Dir-CMP, all request messages and all forwarded request and response messages except those related to writebacks are critical.
- **Indirectly critical messages:** In DiCo-CMP writebacks are performed with a single message that carries the data. This message is indirectly critical, like the finalization message in Dir-CMP. Furthermore, DiCo-CMP uses a pair of messages (CHANGE_OWNER and an ACK) to inform the home node about which node is the current owner of the block, and hence, has its sharing information. These messages are also indirectly critical because some misses cannot be attended by the owner until it has received the ACK. DiCo-CMP does not use a finalization message (UNBLOCK) to end coherence transactions since transactions are serialized by the owner node which also provides the data and holds the directory information.
- **Non-critical messages:** DiCo-CMP uses hint messages which are non-critical. These messages are used to update the information stored in the L1 coherence cache. Since this information is only used for prediction, it can be incorrect without impairing the correctness of the protocol. A delayed hint message never delays the response of a request, but it may cause a miss in the owner prediction mechanism which will lead to sending the request to a node which is not the current owner, forcing that node to forward the request to the home node, and so, the miss may take longer to be handled due to the indirection, in addition to increasing the interconnection network traffic.

C. Discussion

Fig. 2 shows the network traffic (measured as number of flits) of both protocols classified by message criticality. We can see that the total traffic of DiCo-CMP is much higher than the total traffic of Dir-CMP, on average. However,

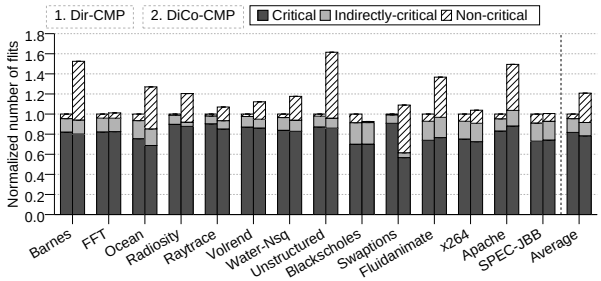


Figure 2: Comparison of the network traffic of Dir-CMP and DiCo-CMP classified by its criticality

the critical traffic is slightly lower due to the removal of forwarding messages.

This traffic distribution suggests that DiCo-CMP will obtain more advantage than Dir-CMP from the use of a heterogeneous network for reducing energy consumption. In fact, as we show in the following sections, a heterogeneous network will allow us to approximate the network consumption of DiCo-CMP and Dir-CMP, despite the lower traffic requirements of Dir-CMP, without losing the advantage in the execution time of DiCo-CMP with respect to Dir-CMP.

IV. Low-Power Heterogeneous Network

As mentioned in Section II, Flores *et al.* [7] proposes a heterogeneous network with two types of links and divides messages into two types to improve the execution time: critical short messages and non-critical long messages.

In this work, we also propose to use a heterogeneous network with two types of links: baseline links and low-power links. Differently from previous works, our heterogeneous network is optimized to carry two kind of messages: critical (including long messages) messages and non-critical and indirectly critical short messages.

The links of our interconnection network are made from two of the four kinds of wires proposed in [10]: B-Wires, which we use for the baseline network, and PW-Wires, which we use for the low-power network. PW-Wires have double latency than B-Wires, but they reduce dynamic energy consumption by 70% for each bit transmitted compared to B-Wires. Finally, both wires require the same area.

We propose two approaches to design the heterogeneous network. In the first one (called *Latency-aware*), we increase the total area of the network with respect to the base architecture (with a homogeneous network) because we add extra PW-Wires alongside the B-Wires. In the second one (called *Area-aware*), we replace some B-Wires with PW-Wires, which occupy the same area. Obviously, the second configuration will have less bandwidth available for critical messages. Also, we have decided to optimize the low-power network in the second approach for carrying only hint messages. In this way, since hint messages are always broadcast and do not need a destination address, we need to replace fewer B-Wires with PW-Wires, so the bandwidth available for critical messages is less affected.

Table II: System parameters

Memory parameters	
Cache hierarchy	Non inclusive
Block size	64 bytes
L1 data and instruction cache	64KB, 4 ways
L1 access latency	2 cycles
Shared distributed L2 cache	512KB/tile, 8 ways
L2 access latency	2 (tag) y 6 (data) cycles
Directory cache	Unlimited
L1 coherence cache	1KB, 4 ways, 2 cycles
L2 coherence cache	1KB, 4 ways, 2 cycles
Memory access latency	160 cycles
Network parameters	
Topology	2D mesh (4×4)
Routing technique	Deterministic X-Y
Flit size	16 bytes
Message size	3 flits (data), 1 flit (control)
Switching and routing latency	2 and 2 cycles
B-wires latency	2 cycles/flit
PW-wires latency	4 cycles/flit

We consider several configurations for the network which are shown in Table I, where area and power values have been determined according to [10]. In these configurations, we vary the number of wires of the low-power network. In *Latency-aware 2×* and *Area-aware 2×*, the low power network has enough wires to send a control or hint message using just one flit (in our implementation a flit is never split in several phits), hence obtaining twice the latency than the baseline network. In the remaining configurations, the number of PW-Wires is reduced to minimize the area used by the low-power network in exchange of bandwidth.

V. Evaluation Environment

We evaluate Dir-CMP and DiCo-CMP over the previously described heterogeneous network by means of the Virtutech Simics [12] full-system simulator extended with Multifacet GEMS [11], that provides a detailed memory system and interconnection network timing model. We have implemented the heterogeneous network on top of GARNET [13], already included in the GEMS toolkit. The simulated system is a 16-core tiled CMP. Table II shows the main parameters of our baseline system. Note that Dir-CMP has an unlimited directory cache, while DiCo-CMP has limited coherence caches. Although this gives some advantage to Dir-CMP, the focus of this paper is not the comparison about these two protocols, but their benefits for heterogeneous networks.

We perform the evaluation with a wide range of benchmarks from different suites: *Barnes* (16K particles), *FFT* (64K complex), *Ocean* (514×514 ocean), *Radiosity* (room, -ae 5000.0 -en 0.050 -bf 0.10), *Raytrace* (teapot, optimized by removing unnecessary locks), *Volrend* (head), and *Water-Nsq* (512 molecules) belong to the SPLASH-2 [14] benchmark suite; *Blackscholes* (simmedium), *Fluidanimate* (simmedium), *Swaptions* (simmedium), and *x264* (simmedium) are from the PARSEC [15] benchmark suite; *Unstructured* (Mesh.2K) is a scientific application with irregular access patterns; and finally, *Apache* (1000 HTTP transactions) and *SPEC-JBB* (1600 transactions) are two commercial applications [16]. Simulation results correspond

Table I: Interconnection network configurations

Configuration	Links	Low power links		Extra area
		Dynamic power (per message)	Latency (per message)	
Base (Homogeneous)	192 B-Wires	—	—	0%
Latency-aware 2×	192 B-Wires & 64 PW-Wires	0.3×	2×	33.3%
Latency-aware 4×	192 B-Wires & 32 PW-Wires	0.3×	4×	16.7%
Latency-aware 8×	192 B-Wires & 16 PW-Wires	0.3×	8×	8.33%
Latency-aware 16×	192 B-Wires & 8 PW-Wires	0.3×	16×	4.17%
Area-aware 2×	144 B-Wires & 48 PW-Wires	0.3×	2×	0%
Area-aware 4×	144 B-Wires & 24 PW-Wires	0.3×	4×	-12.5%
Area-aware 8×	144 B-Wires & 12 PW-Wires	0.3×	8×	-18.8%
Area-aware 16×	144 B-Wires & 6 PW-Wires	0.3×	16×	-21.9%

to the parallel phase of these benchmarks, and at least three simulations with different random seeds have been performed for each data point.

VI. Results

In this section we study the behavior in terms of performance and energy consumption of Dir-CMP and DiCo-CMP when they are implemented over a heterogeneous network that uses two kinds of links: ones employing B-wires and another ones employing PW-wires.

Particularly, we study two different sets of configurations for the heterogeneous network. In the first one (Section VI-A), we add extra PW-wires to the base interconnect for sending non-critical and indirectly critical messages. This increases the area required by the interconnection network but does not affect the critical traffic. In the second one (Section VI-B), we replace some B-wires with PW-wires. In this way, the network area does not increase with respect to the baseline configuration, since B-wires and PW-wires have the same area [10]. However, this area-aware scenario comes at the cost of increasing the number of flits needed for sending data messages through the network.

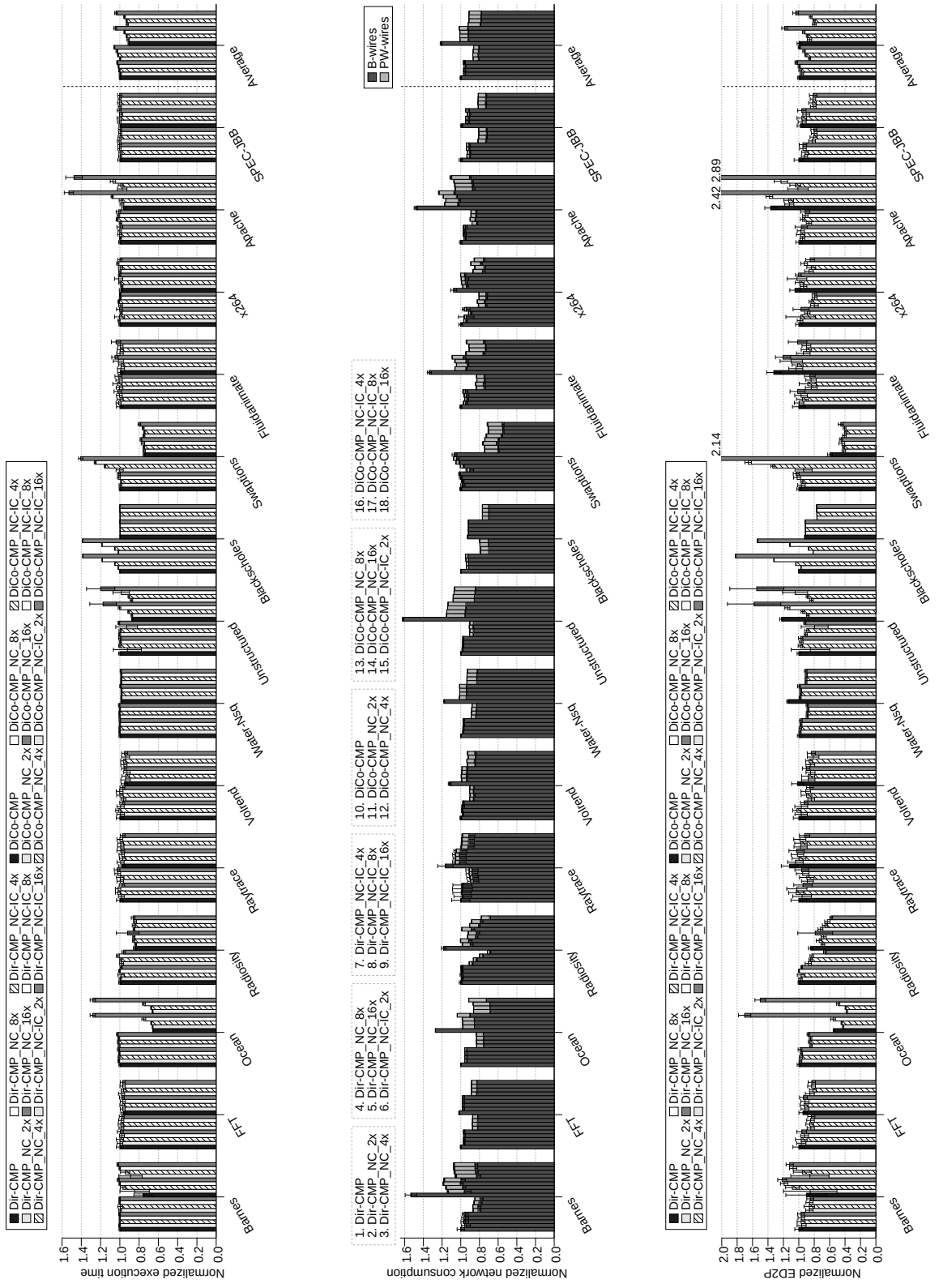
A. Latency-aware configurations

The first set of configurations that we study preserves the base links and adds 64 PW-wires (i.e., 8 bytes) per link (see Table I). Since we preserve all the wires in the base network, the latency of messages traveling through these links is not affected. On the other hand, we add enough PW-wires to allow control messages to be sent in a single flit through power-aware links. Since these PW-wires have double latency than B-wires, we are interested in using them only to transmit messages that are not in the critical path of cache misses. Additionally, we also study configurations that reduce the number of PW-wires at the cost of also reducing the flit size (i.e., increasing the message latency). In this way, we can reduce the area overhead of the heterogeneous network. The different configurations are labeled with the relative latency with respect to the base configuration.

As described in Section III, there are several degrees of criticality for the network messages. This section also studies which messages should be transmitted through PW-wire links. Particularly, we analyze the effect of sending either just non-critical messages (*NC* label) or both non-critical and indirectly critical messages (*NC-NI* label) through PW-wires. We compare these two options with the baseline configuration where every message is sent through B-wires.

Fig. 3a shows the execution time for the described configurations normalized with respect to Dir-CMP over a homogeneous network. First, we can observe that, employing a homogeneous network, DiCo-CMP reduces the execution time by 9.3%, on average, when compared to Dir-CMP. When we send non-critical messages (or even indirectly critical ones) through PW-wires the execution time increases only slightly for both Dir-CMP and DiCo-CMP. This slowdown grows as the latency of PW-wires increases (by reducing the number of wires in the link). In fact, it is more noticeable in DiCo-CMP, particularly for the $16\times$-latency, where it gets worse performance than Dir-CMP. This is because the late arrival of hint messages translates into many owner mispredictions. In general, DiCo-CMP tolerates well a $8\times$ latency for messages that are not critical, which will allow us to reduce the network consumption without too much area overhead.

Fig. 3b shows the dynamic energy consumed by the interconnection network for the evaluated configurations, normalized with respect to Dir-CMP over a homogeneous network. First, we can see that, for the homogeneous network, the dynamic network consumption in DiCo-CMP is 21% higher than in Dir-CMP, mainly due to the extra traffic generated by the hint messages. By sending messages that are not in the critical path of cache misses (e.g., hints) through PW-wires the dynamic energy can be severely reduced (70%). Particularly, if we consider a heterogeneous network and we send non-critical messages through PW-wires with $2\times$ latency, DiCo-CMP can save 17% energy consumption, on average, with respect to a homogeneous network. If we also send indirectly critical messages through PW-wires, the reduction in energy consumption grows up to 25%, thus consuming only 5% more than a directory protocol that also sends non-critical and indirectly critical messages through PW-wires. Notice that in this analysis we only consider dynamic consumption but not static consumption (leakage). Leakage is reduced as the number of wires in the low-power links is reduced. We achieve this by further increasing the latency of non-critical and indirectly critical messages ($4\times$, $8\times$, and $16\times$ configurations).



(a) Execution time (b) Network consumption (c) ED^2P

Figure 3: Evaluation results for the latency-aware configuration

Finally, Fig. 3c shows the value obtained for every configuration for the energy delay square (ED^2P) metric. Again, values are normalized with respect to Dir-CMP. We can observe the importance of sending indirectly critical messages through PW-wires in addition to non-critical messages. In fact, Dir-CMP is hardly affected when only non-critical messages are sent through PW-wires, since they are not very frequent in a directory protocol. When we also send indirectly critical messages through PW-wires with $2\times$ latency, Dir-CMP reduces ED^2P by 8.9%, on average, with respect to Dir-CMP. However, DiCo-CMP already obtains 10.3% of ED^2P improvement by just sending non-critical messages through PW-wires. If we also send indirectly critical messages through such wires, we improve ED^2P by 17.3% for $4\times$ latency links with respect to the base Dir-CMP configuration and 9.7% with respect to the best Dir-CMP configuration (*Dir-CMP_NC-IC_4*$\times$). These results prove that direct coherence protocols can get more advantage from heterogeneous networks than directory protocols.

As shown in Table I, the area overhead of the *4*$\times$ latency-aware heterogeneous network is 16.7% with respect to the homogeneous network. This will also result in extra leakage consumption. In order to avoid this area and consumption overhead we propose the area-aware configurations evaluated in the following section.

B. Area-aware configurations

Our second set of configurations replace some of the wires of the base network (B-wires) with PW-wires. Since both types of wires have the same area, the area of the network does not increase. Particularly, we replace only 48 B-wires (6 bytes) with PW-wires, since in this configuration we are interested in sending only hints through PW-wires. Hints are always broadcast and are the only message type in the low power links, therefore they do not need destination nor message type information, requiring only 6 bytes. Since the base links have changed from 24 bytes in the base case to 18 bytes, now the number of flits needed to transmit data messages (72 bytes) changes from 3 to 4. Control messages are unaffected. The heterogeneous network is labeled with the word *Hints* and the relative latency of PW-wires with respect to the base configuration.

Fig. 4a shows the execution time for the described configurations, normalized with respect to Dir-CMP. We can see that now the execution time increases by 3% when using the heterogeneous network (DiCo-CMP vs. DiCo-CMP_Hints_2$\times$) due to the increase in the number of flits of data messages. However, by only sending hint messages through PW-wires instead of sending other non-critical messages, we achieve a lower performance degradation when the link latency increases (5%).

Fig. 4b shows the dynamic power consumption of the network, normalized with respect to Dir-CMP. Since most non-critical messages in DiCo-CMP are hints, the savings in energy are significant, and therefore, the energy consumption approaches that of Dir-CMP.

Fig. 4c shows the ED^2P metric for the configurations evaluated in this section, again normalized with respect to Dir-CMP. We can see that both the $2\times$ and the $4\times$ configurations reduce the ED^2P by 9% compared to Dir-CMP and DiCo-CMP. The $2\times$ configuration requires the same area as the homogeneous network. However, the $4\times$ configuration is able to reduce both area (by 12.5%) and ED^2P (by 9%) compared to a homogeneous network. Note that the area requirements of the $4\times$ area-aware heterogeneous network are 12.5% lower than the base homogeneous network.

VII. Conclusions

In this work, we have shown how direct coherence protocols can take much more advantage from heterogeneous networks than traditional directory protocols. The proposed networks have only two types of links: baseline and low-power links. Thanks to sending messages that are not in the critical path of cache misses through low-power links, we achieve an important reduction in the dynamic power consumption of the network, which constitutes a significant fraction of the total power of the chip.

Since the number of non-critical messages in direct coherence protocols (53.5%) is much higher than in directory protocols (29.5%), the former can utilize the low-power links more frequently. Also, since the increase of latency in non-critical messages has little effect in the execution time, we can save power with a negligible impact on performance while also reducing area.

Acknowledgments

This work has been funded by the Spanish Ministerio de Ciencia e Innovación (MICINN) under grant "TIN2009-14475-C04-02". We thank Pablo David Muñoz Sánchez for his work in the simulator used in this project.

References

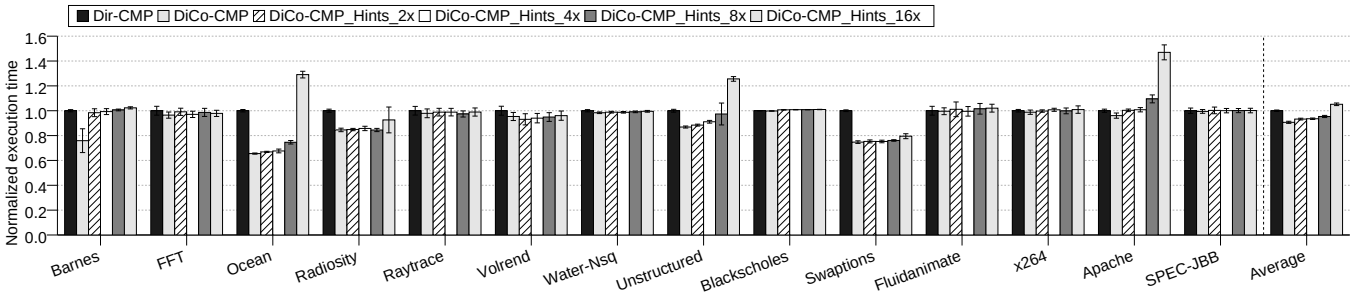
[1] J. Leverich, H. Arakida, A. Solomatnikov, A. Firoozshahian, M. Horowitz, and C. Kozyrakis, "Comparing memory systems for chip multiprocessors," in *34th Int'l Symp. on Computer Architecture (ISCA)*, Jun. 2007, pp. 358–368.

[2] M. M. Martin, M. D. Hill, and D. J. Sorin, "Why on-chip cache coherence is here to stay," *Commun. ACM*, vol. 55, no. 7, pp. 78–89, Jul. 2012.

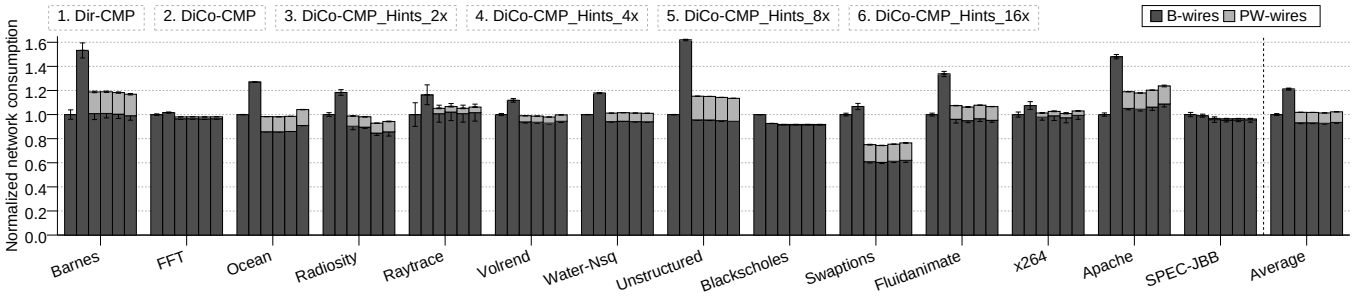
[3] S. Bell, B. Edwards, J. Amann, R. Conlin, K. J. andVince Leung, J. MacKay, M. Reif, L. Bao, J. Brown, M. Mattina, C.-C. Miao, C. Ramey, D. Wentzlaff, W. Anderson, E. Berger, N. Fairbanks, D. Khan, F. Montenegro, J. Stickney, and J. Zook, "TILE64^{TM} processor: A 64-core SoC with mesh interconnect," in *IEEE Int'l Solid-State Circuits Conference (ISSCC)*, Jan. 2008, pp. 88–598.

[4] M. Zhang and K. Asanović, "Victim replication: Maximizing capacity while hiding wire delay in tiled chip multiprocessors," in *32nd Int'l Symp. on Computer Architecture (ISCA)*, Jun. 2005, pp. 336–345.

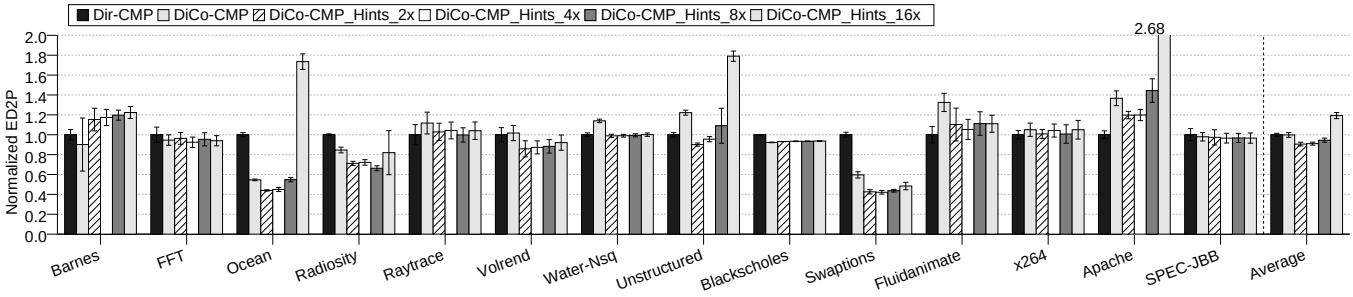
[5] A. Ros, M. E. Acacio, and J. M. García, "Direct coherence: Bringing together performance and scalability in shared-memory multiprocessors," in *14th Int'l Conference on High Performance Computing (HiPC)*, Dec. 2007, pp. 147–160.



(a) Execution time



(b) Network consumption



(c) ED^2P

Figure 4: Evaluation results for the area-aware configuration

[6] ——, "DiCo-CMP: Efficient cache coherency in tiled CMP architectures," in *22nd Int'l Parallel and Distributed Processing Symp. (IPDPS)*, Apr. 2008, pp. 1–11.

[7] A. Flores, J. L. Aragón, and M. E. Acacio, "Heterogeneous interconnects for energy-efficient message management in cmps," *IEEE Transactions on Computers (TC)*, vol. 59, no. 1, pp. 16–28, Jan. 2010.

[8] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, "Interconnect-power dissipation in a microprocessor," in *Int'l workshop on System Level Interconnect Prediction (SLIP)*, Feb. 2004, pp. 7–13.

[9] K. Banerjee and A. Mehrotra, "A power-optimal repeater insertion methodology for global interconnects in nanometer designs," *IEEE Transactions on Electron Devices*, vol. 49, no. 11, pp. 2001–2007, Nov. 2002.

[10] L. Cheng, N. Muralimanohar, K. Ramani, R. Balasubramonian, and J. B. Carter, "Interconnect-aware coherence protocols for chip multiprocessors," in *33rd Int'l Symp. on Computer Architecture (ISCA)*, Jun. 2006, pp. 339–351.

[11] M. M. Martin, D. J. Sorin, B. Beckmann, M. R. Marty, M. Xu, A. R. Alameldeen, K. E. Moore, M. D. Hill, and D. A. Wood, "Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset," *Computer Architecture News*, vol. 33, no. 4, pp. 92–99, Sep. 2005.

[12] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, and B. Werner, "Simics: A full system simulation platform," *IEEE Computer*, vol. 35, no. 2, pp. 50–58, Feb. 2002.

[13] N. Agarwal, T. Krishna, L.-S. Peh, and N. K. Jha, "GARNET: A detailed on-chip network model inside a full-system simulator," in *IEEE Int'l Symp. on Performance Analysis of Systems and Software (ISPASS)*, Apr. 2009, pp. 33–42.

[14] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 programs: Characterization and methodological considerations," in *22nd Int'l Symp. on Computer Architecture (ISCA)*, Jun. 1995, pp. 24–36.

[15] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *17th Int'l Conference on Parallel Architectures and Compilation Techniques (PACT)*, Oct. 2008, pp. 72–81.

[16] A. R. Alameldeen, C. J. Mauer, M. Xu, P. J. Harper, M. M. Martin, D. J. Sorin, M. D. Hill, and D. A. Wood, "Evaluating non-deterministic multi-threaded commercial workloads," in *5th Workshop On Computer Architecture Evaluation using Commercial Workloads (CAECW)*, Feb. 2002, pp. 30–38.