Characterization and Cost-Efficient Selection of NoC Topologies for General Purpose CMPs

Marta Ortín
U. of Zaragoza
Spain
ortin@unizar.es

Alexandra Ferrerón
U. of Zaragoza
Spain
ferreron@unizar.es

Jorge Albericio
U. of Zaragoza
Spain
jalberic@unizar.es

Darío Suárez
U. of Zaragoza
Spain
dario@unizar.es

María Villarroya-Gaudó
U. of Zaragoza
Spain
mvg@unizar.es

Cruz Izu
U. of Adelaide
SA 5005 Australia
cruz@cs.adelaide.edu.au

Víctor Viñals
U. of Zaragoza
Spain
victor@unizar.es

ABSTRACT

The importance of the interconnection network is growing as the number of cores integrated on a chip increases. Communication among nodes becomes a bottleneck and impacts system performance and power consumption. This work targets general purpose CMPs, where there is a rising concern about finding low-power alternatives.

We explore the implications of the interconnect choice on overall performance by comparing the behaviour of three topologies: ring, mesh, and torus. We also evaluate two additional ring configurations (one with increased bandwidth and another with reduced-pipeline routers) and concentrated versions of the topologies. Running full-system simulations allows us to carefully model the processors, memory hierarchy, and interconnection network, and execute realistic parallel and multiprogrammed workloads. We determine that the network diameter is critical for system performance and that a concentrated mesh offers the best area-energy-delay tradeoff for both 16 and 64-core chips. Traffic is very light and highly unbalanced, asserting the need for an heterogeneous network with more resources located in specific areas.

1. INTRODUCTION

Nowadays, a single chip may contain multiple processors and a significant amount of memory. A popular trend consists on interconnecting several nodes, each of them with a core and one or more levels of private and/or shared memory caches. Nodes communicate through an interconnection network that allows any pair of nodes to exchange information and has a major impact on overall performance, energy consumption, and area. We focus on general purpose CMPs, where there is high need for low-power chips. Our conclusions also apply to energy-efficient supercomputers, like the Mont Blanc high-performance platform, which might be built with embedded ARM processors[1].

There are few works that study the interconnect by modelling in detail the processors, memory hierarchy, and interconnection network. Some of those analysis are performed with synthetic traffic or application traces that do not capture the behaviour of a real execution. In this work, we contribute to previous research by simulating parallel and multiprogrammed workloads with real applications, carefully modelling all the components mentioned earlier. This allows us to study the effect of the interconnection network configuration on the whole system and the interactions between the memory subsystem and the interconnect.

We present an analysis of three topologies with varying degrees of complexity, performance, power, and area (bidirectional ring, mesh, and torus), with full-system simulation of a CMP with 16 and 64 cores. Our aim is to extract meaningful conclusions that will indicate the weaknesses of current configurations and guide our future research. We show that low-resource topologies like the concentrated mesh and ring are more area-energy-delay efficient than others with more links and routers for both parallel and multiprogrammed workloads. Besides, traffic is not uniformly distributed across the chip, indicating that we should focus on heterogeneous networks for the design of future architectures.

The rest of this document is organized as follows: Section 2 describes the CMP architecture; Section 3 explains the methodology and summarizes our results; Section 4 presents the state of the art and Section 5 concludes the paper.

2. CMP ARCHITECTURE FRAMEWORK

This section presents the CMP architecture we are modelling and a detailed description of the interconnection network.

2.1 General Description of the Architecture

Our study focuses on homogeneous CMPs. The system is composed of several tiles connected by an interconnection network. Each tile has a core with a private first level cache (L1) split into data and instructions and a bank of the shared second level cache (L2), both connected to the router. Some tiles in the edges of the chip also include a memory controller. Table 1 summarizes the key parameters of the architecture.

We use a directory-based MESI coherence protocol. All the traffic that traverses the interconnection network is a direct consequence of the memory activity, either to move cache lines (instructions or data) to the tile that needs them or for coherence management. That is why it is important to model the caches realistically, even though our main interest

[1] http://www.montblanc-project.eu/

INA-OCMC '13, January 23, 2013, Berlin, Germany.
Copyright 2013 ACM 978-1-4503-1784-9/13/01 ...$15.00.

Table 1: Main characteristics of the CMP system.

Cores	16 and 64 cores, Ultrasparc III Plus, in order, 1 instr/cycle, single threaded, 2GHz frequency
Coherence protocol	Directory-based, MESI, directory distributed among L2 cache banks
Consistency model	Sequential
L1 cache	32KB data and instruction caches, 4-way set assoc, 2-cycle hit access time, 64B line size Private, pseudo-LRU replacement policy
L2 cache	Distributed, 1 bank/tile, 1MB per bank, 16-way set assoc, 7-cycle hit access time, 64B line size Shared, inclusive, interleaved by line address, pseudo-LRU replacement policy
Memory	4 memory controllers, distributed in the edges of the chip (both for 16 and 64-core architectures) 160-cycle latency

Table 2: Main characteristics of the interconnection network.

General	Two virtual networks (requests and replies), 2 virtual channels (VCs) per virtual network
Routers	4-stage pipeline: routing and input buffering, VC allocation, switch allocation and switch traversal Round-robin 2-phase VC/switch allocators 5-flit buffers per VC, enough to store an entire message (3-flits per buffer in the ring with higher BW)
Links	16-byte flit size (we also include a ring with higher bandwidth with 24B flit size), 1-cycle latency

is the interconnect [4, 6].

2.2 Interconnection Network

Our networks are built with simple 4-stage routers using wormhole credit-based flow control and dimension order routing. Table 2 shows the detail network configuration.

We compare three different topologies: mesh, torus, and ring. The *2D mesh* is a widespread choice for large-scale CMPs due to its regularity. A *torus* is a mesh with wraparound links to reduce the network diameter. It requires a larger area and consumes more energy. In contrast, we have included a *bidirectional ring*, built as a Hamiltonian cycle. Every connection is implemented with two links, one in each direction.

In the ring topology, the number of inputs/outputs to the outside of the tile is reduced to 2/2 (as opposed to the 4/4 used in mesh and tours), which results in a smaller number of buffers and simpler allocators and crossbar. For that reason, we include a ring configuration with increased bandwidth and the same router area as the torus (links and flits of 24B, abbreviated RING_FLIT24B), and another one with reduced latency, where we merge the switch allocation and switch traversal stages, resulting in a 3-cycle router (RING_3CYCLE_R). We have checked with DSENT that these modifications do not increase cycle time.

We also study concentrated topologies with a concentration factor of 4, which reduces the amount of resources of the network and might introduce contention. To avoid increasing the router radix, we use *external* concentration [3] with local routers. For the 16-core chips we implement a concentrated mesh (CMESH), depicted in Figure 1. With only four global routers, the concentrated ring topology is equivalent to the CMESH; the concentrated torus would have additional links, but we omit the results because the higher bandwidth does not benefit performance and increases power and area. For 64 cores, we model the CMESH, CTORUS, and CRING.

3. EVALUATION

This section summarizes the methodology and main contributions of our analysis for 16 and 64-core architectures.

3.1 Simulation Environment

We use Simics, GEMS, and an extended version of GARNET. We carefully model all the components of the chip and perform full-system simulation with simple single-threaded cores and directory-based coherence. To get the timing, area and energy expended by the network we use DSENT, a state-of-the-art circuit modelling tool (with 32nm technology).

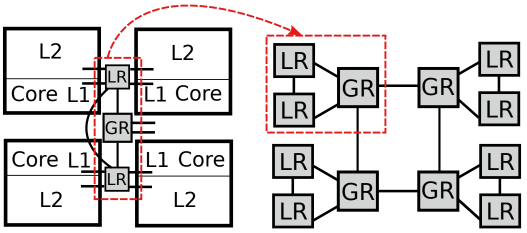


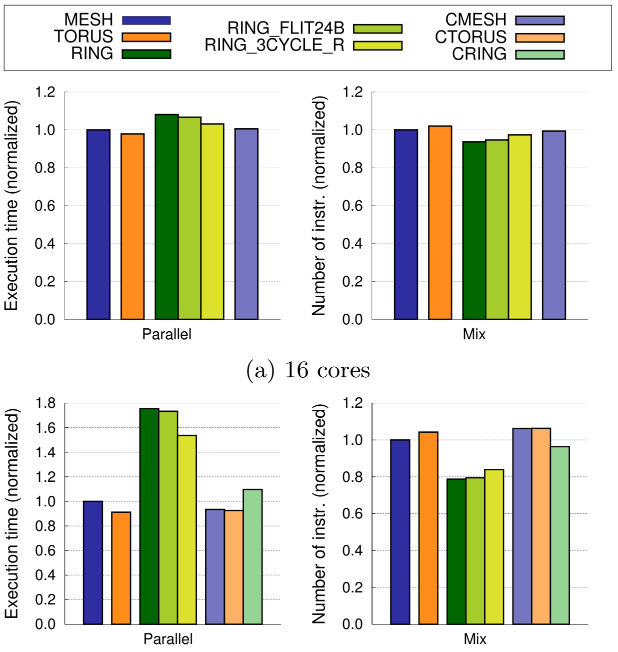
Figure 1: Connection of the nodes to the routers within a four-node cluster (left) and organization of all local and global routers (right) for a concentrated mesh in a 16-core chip. LR and GR stand for local router and global router, respectively.

3.2 Workloads

CMPs can execute parallel applications to reduce execution time, and multiprogrammed workloads (execution of independent programs on each core) to increase throughput. We use a selection of shared-memory parallel applications from PARSEC (`blackscholes`, `canneal`, `fluidanimate`, `swaptions`, and `x264`) and SPLASH2 (`barnes`, `fmm`, `ocean`, `radiosity`, `volrend`, and `water-spatial`). For the multiprogrammed workloads, we choose 16 applications with large working sets from the SPEC CPU2006 suite. To build the workload for the 16-core architectures we execute applications once, binding each of them to a different core to avoid migration. For the 64-core architectures we use each of the applications four times.

3.3 Performance

To compare the impact of the network configurations on performance, we analyse the number of processor cycles it takes for the parallel workloads to complete the parallel section; for the multiprogrammed workload, we check how many instructions get executed in 500 million cycles. Figure 2 represents the average execution time for the parallel applications and the number of completed instructions for the multiprogrammed workload, both normalized to the mesh. In 16-core architectures differences between topologies are much smaller, with the ring with 3-cycle routers and the



(a) 16 cores

(b) 64 cores

Figure 2: Average execution time for the parallel applications (left, the lower the better) and number of completed instructions for the multiprogrammed workload (right, the higher the better), both normalized to the mesh, for 16 and 64 cores.

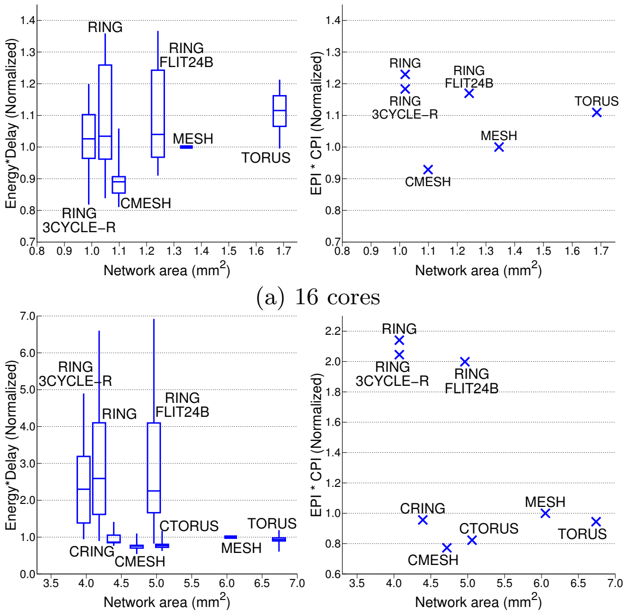
CMESH being very similar to the mesh, and the torus performing only slightly better. In 64-core applications, the performance of the ring topologies drops significantly while the concentrated topologies stay very close to the mesh and torus.

The differences in performance are a direct consequence of the number of hops it takes a message to go from its source to its destination. For that reason, the diameter of the network is critical and concentrated topologies achieve better results with less routers and links.

3.4 Area, Energy and Delay

When making design choices for future architectures we need to consider performance, power, and area. For parallel applications, we calculate Energy*Delay (ED); for multiprogrammed workloads, where we simulate a constant number of cycles, we use EPI*CPI [2]. Figure 3 depicts area versus normalized ED or EPI*CPI for 16 and 64-core architectures. To display the variance across the parallel applications, we represent the results with candlesticks, which show the minimum, the quartile 25, the median, the quartile 75, and the maximum values. Ideally, we would like our configuration to be in the bottom left corner of the graphs.

For 16 cores, the CMESH offers the lowest values for energy and delay, with a small area (only 8% bigger than the ring and 18 and 35% smaller than the mesh and torus, respectively). For 64 cores, the ED and EPI*CPI increase substantially for the ring topologies with all workloads. Performance drops much more significantly with more cores due to the increased hop count. Therefore, networks with lower diameter perform better when integrating a larger number of cores. In this case, the CMESH still offers the best tradeoffs. Our results show that overdimensioning the network is not the best solution: a simple topology like the CRING is better than the torus from all standpoints.



(a) 16 cores

(b) 64 cores

Figure 3: Area versus Energy*Delay for the parallel applications (left) and EPI*CPI for the multiprogrammed workload (right) for 16 and 64 cores, normalized to the mesh. Candlesticks for the RING and RING_3CYCLE_R have been moved slightly on the horizontal axis for better visualization, both have an area of 1.0mm for 16 cores and 4.1mm for 64 cores.

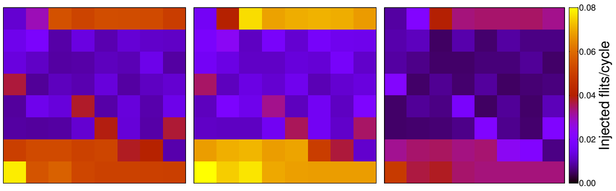
We also see that the deviation of the results varies among topologies and is bigger with 64 cores. It is proportional to the variation in network latency, which increases with the average distance of the network and hop latency. Performance loss is higher for certain applications in which the thread distribution generates disadvantageous traffic patterns for the ring topology.

3.5 Traffic Distribution

We have analysed the number of injected flits for all our configurations and workloads. We have noted that traffic is unevenly distributed in the interconnect, which means that some resources will be needed more often than others. In this section, we present results for `blackscholes` out of all the parallel applications and focus on a 64-core chip. Conclusions still hold for all applications and 16-core configurations and are supported by link utilization. The distribution remains constant when we change the network topology, so we illustrate our conclusions with only the mesh, torus, and ring.

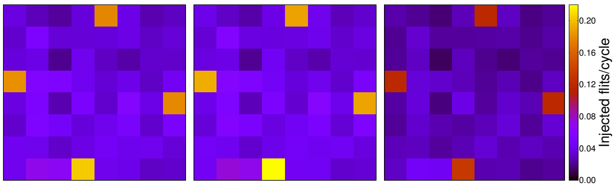
Figure 4 depicts a heat map of injected flits per cycle for each node for `blackscholes` executed on 64 cores. All the traffic is generated by the memory subsystem, so every action has a reaction (request-reply, invalidation-ack). Hence, the heat maps also indicate which nodes are receiving messages more often. The number of flits per cycle is bigger for the torus because a very similar amount of traffic gets injected in a much shorter period of time; we see the opposite effect in the ring topology. Nevertheless, the distribution of

[2] EPI=Energy per Instruction, CPI=Cycles per Instruction



(a) Mesh (b) Torus (c) Ring

Figure 4: Injected flits per cycle and node for the `blackscholes` application executed in 64 cores.



(a) Mesh (b) Torus (c) Ring

Figure 5: Injected flits per cycle and node for the multiprogrammed workload application executed in 64 cores.

traffic is the same regardless of the topology: some nodes inject more flits than others. This is because certain L2 banks are being accessed more frequently than others, depending on the physical distribution of the data touched by each application.

Figure 5 shows the same plots for the multiprogrammed workload. In this case, we see four clear hotspots in the edges of the chip, where the memory controllers are located. The multiprogrammed workload accesses main memory more often than parallel applications. Apart from that, the rest of ideas we introduced for parallel workloads are still valid.

In both cases, we note that the network is lightly loaded, even around the most active nodes; furthermore, some parts are idle most of the time. Considering all applications and configurations, parallel and multiprogrammed workloads inject an average of 0.021 and 0.064 flits per cycle, respectively. This explains why the concentrated topologies reduce network distance without a significant increment on network contention.

These results ratify the idea of non uniform traffic derived from the behaviour of applications. They point out that synthetic traffic patterns should have located hotspots in both flit injection and destination distribution in order to reflect the real traffic load imposed on the network. Besides, a more efficient network would need more resources in some parts of the chip while saving power in others.

4. RELATED WORK

Several works model alternatives to the most commonly used router architectures, topologies, and flow control methods, but they base their proposals on network-only simulations of synthetic traffic and traces [1, 5].

Another approach consists on designing the network based on the behaviour of the memory subsystem and the coherence protocol [2, 7]. The ideas presented on those studies would achieve better results if coupled with more efficient topologies, as we have pointed out in this work.

There are very few papers which focus on the comparison of interconnection network configurations. Sanchez *et al.* explore the implications of interconnection network design for CMPs [6]. We complete their results including a simple topology (ring), multiprogrammed workloads and traffic distribution analysis.

5. CONCLUSIONS

Interconnection networks and cache hierarchy have a significant influence on system performance, area, and power consumption. Considering both aspects simultaneously helps to identify improvement opportunities. We have modelled in detail the processors, memory hierarchy, and network using full-system simulation and executing both parallel and multiprogrammed workloads. We have compared the behaviour of three network topologies: mesh, torus, and ring, including two additional ring configurations (one with more bandwidth and one with 3-cycle routers) and concentrated networks for CMPs with 16 and 64 cores.

We have demonstrated that performance is highly affected by the choice of the interconnect, specially in 64-core systems. The ring topologies perform worse due the increased hop count, which translates into higher network latency. The CMESH topology offers the best performance with low power consumption and area for all workloads considered and both 16 and 64-core chips.

We have shown that traffic is very light and not uniformly distributed on the network. For parallel applications, both the injection rate and the message destinations are more variable than those we see with synthetic traffic patterns; for multiprogrammed workloads, traffic is random with four hotspots at the memory controllers. This points out the need for an heterogeneous network, which could handle more traffic in some areas and save power in others

Acknowledgements

This work was supported in part by grants TIN2010-21291-C02-01 (Spanish Government, European ERDF), gaZ: T48 research group (Aragón Government and European ESF), Consolider CSD 2007-00050 (Spanish Government), and HiPEAC-3 NoE (European FP7/ICT 217068).

6. REFERENCES

[1] M. Koibuchi, H. Matsutani, H. Amano, D. F. Hsu, and H. Casanova. A case for random shortcut topologies for HPC interconnects. In *Proc of the 39th Int Symp on Comp Arch*, pages 177–188, 2012.

[2] T. Krishna, L.-S. Peh, B. M. Beckmann, and S. K. Reinhardt. Towards the ideal on-chip fabric for 1-to-many and many-to-1 communication. In *Proc of the 44th Ann IEEE/ACM Int Symp on Microarch*, pages 71–82, 2011.

[3] P. Kumar, Y. Pan, J. Kim, G. Memik, and A. Choudhary. Exploring concentration and channel slicing in on-chip network router. In *Proceedings of the 2009 3rd ACM/IEEE Int Symp on Networks-on-Chip*, pages 276–285, 2009.

[4] R. Kumar, V. Zyuban, and D. M. Tullsen. Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling. In *Proc of the 32nd Ann Int Symp on Comp Arch*, pages 408–419, 2005.

[5] A. K. Mishra, N. Vijaykrishnan, and C. R. Das. A case for heterogeneous on-chip interconnects for CMPs. In *Proc of the 38th Ann Int Symp on Comp Arch*, pages 389–400, 2011.

[6] D. Sanchez, G. Michelogiannakis, and C. Kozyrakis. An analysis of on-chip interconnection networks for large-scale chip multiprocessors. *ACM Trans. Archit. Code Optim.*, 7(1):4:1–4:28, 2010.

[7] S. Volos, C. Seiculescu, B. Grot, N. Pour, B. Falsafi, and G. De Micheli. CCNoC: Specializing on-chip interconnects for energy efficiency in cache-coherent servers. In *Sixth IEEE/ACM Int Symp on Networks on Chip*, pages 67 –74, 2012.