Impact of Interconnection Network resources on CMP performance

P. Abad, P. Prieto, J. Merino, L.G. Menezo and V. Puente
University of Cantabria

Abstract. *The main goal of this work is to analyze the tradeoffs when devoting more or less CMP system resources to the interconnection network. This preliminary study explores the effects that the resources (mainly buffer room and topology) have on the performance of a chip multiprocessor organized as a SNUCA. Results show that lowering throughput from 0.43 to 0.17 phits/cycle/router increases execution time by 25% on average. This provokes a reduction of approximately 30% in interconnection network power, but at the cost of increasing the energy consumed by the whole CMP system by 25%.*

1.- Introduction

In CMP (Chip Multi-Processor) architectures, the communication sub-system, or the interconnection network as it is also called, shares area with the rest of the system components (mainly processors and cache memory). For this reason, a large amount of computer architecture research in the context of these networks has been to do with complexity limitations [5][19]. Simplistic network architectures have been presented frequently as the best design choice in order to devote as many transistors as possible to the rest of the system. Scarce buffering, simplistic routing or flow control mechanisms have been presented as the only available choice for CMP architectures [20]. While important contributions have been made on low-load pipeline optimizations [11][12][15], most of these proposals obtain poor results under high-contention scenarios, limiting the maximum achievable throughput.

The main objective of this work is to clarify how much of the chip's real estate should be dedicated to the communication sub-system in order to achieve sufficient application performance. To answer this question, we will explore the effect of different network configurations on system performance, ranging from area-oriented implementations to performance-oriented structures.

Specifically, the raw performance in terms of throughput, router architectures, networks and mechanisms of very low complexity has been compared with other combinations that require more resources but achieve better throughput. These same architectures have been included as a part of a CMP and the complete system performance has been analyzed by running different applications.

Although detailed analysis of hardware cost is not included, the conclusion is clear. The use of simple architectures reduces the values of area and apparently also the energy consumed, but actually it causes the opposite effect because the execution time of applications increases. Since the interconnection network represents a relatively small percentage of the overall system, the reduction of its own energy consumption does not compensate for the loss caused to the whole system as a result of the increased time required to run applications.

That is, the search for efficient solutions for the interconnection network must be made in a comprehensive manner taking into account the effects on the rest of the system.

2.- Design-space exploration

In order to make the study as complete as possible, we have implemented a significant number of router architectures, covering a wide range of the routers actually proposed to be employed in CMPs:

- **WH MESH**: This basic wormhole router configuration performs flit-level flow control. Topology (Mesh) and routing (Deterministic) are fixed, regardless of the number of virtual channels or message types.
- **WH TORUS**: This router shares almost every characteristic with the upper configuration, changing only the topology. In this case a torus network is employed. In order to guarantee deadlock avoidance, every message type must have at least two virtual channels.
- **WH TORUS ADAP**: The ability to perform fully adaptive routing is added in this structure. Another extra virtual channel is needed in order to implement the deadlock avoidance mechanism. Working with torus topologies means a minimum of three virtual channels per message type.
- **VCT MESH**: Flow control is modified here, the division of buffering capacity into different virtual channels is avoided in order to be able to perform packet-level arbitration (For this kind of flow control, buffers must be large enough to store at least a whole packet).
- **VCT TORUS (ADAP)**: A final optimization of the flow control mechanism is performed (Bubble) in order to minimize the number of virtual channels needed to perform adaptive routing in a torus.

Obviously, there are important hardware implications derived from some design decisions. Aspects such as router pipeline and link width are continuously under discussion. As pipelines are moving fast to smaller FO4s [9], the appropriate number of router stages is not a clearly fixed parameter. However, routers with fused pipeline stages [15] do not seem to be a good solution for future proposals and processor clock cycle is an orthogonal variable to our design space. For these reasons, the preliminary solution adopted has been to work with 5-stage canonical designs, without considering pipeline optimizations. The tradeoffs between area, power and performance for different link widths must be quantified after obtaining hardware models for each width value.

Finally, we assume that the most successful architectures for CMP systems will be based on the use of shared memory and therefore they must provide hardware mechanisms to ensure consistency. The use of coherence protocols requires different classes of messages (for example, the communication protocol in Alpha 21364 has a dependency chain length of seven) and end-to-end deadlock avoidance mechanisms [18].

3.- Research results

Simulation Framework

Our framework allows us to perform full-system simulation with complex workloads based on detailed architectural models of the most relevant system modules. We employ an infrastructure composed of three connected simulation tools. The core of the simulation infrastructure is the full system simulator Simics [13], which has been architecturally augmented with the GEMS timing infrastructure [14]. GEMS provides detailed models of both the memory system (RUBY) and a state-of-the-art out-of-order processor (OPAL). In order to obtain detailed contention modelling for the interconnection network, the original network simulator of GEMS has been replaced with SICOSYS [17]. This simulator allows us to take into account most of the network implementation details with much higher precision.

The complete framework will enable us to perform exhaustive full-system simulation with complex workloads and detailed modeling of the most relevant system modules at architectural level.

Table 1: Main Synthetic Traffic Characteristics.

Topology	8×8 mesh or 8×8 torus
Message Size	5 phits
Link width	128 bits
Cycles simulated	200,000 (20,000 warm-up)
Routing algorithm	DOR or Adaptive
Flow Control	WH, VCT or Bubble
Message types	4 and 6
Buffer size (per VC)	3 to 10 phits
Traffic Patterns	Uniform, bit-reversal, perfect-shuffle, matrix-transpose.

Synthetic workloads

In the first part of the study we have performed a preliminary exploration of the upper limit of performance benefits, without taking into account hardware implications of parameter changes. Using synthetic traffic patterns we have performed a wide space exploration employing the Sicosys simulator alone. For these simulations both router pipeline stages and frequency have been kept constant for every simulation done. The main parameters employed are shown in Table 1.

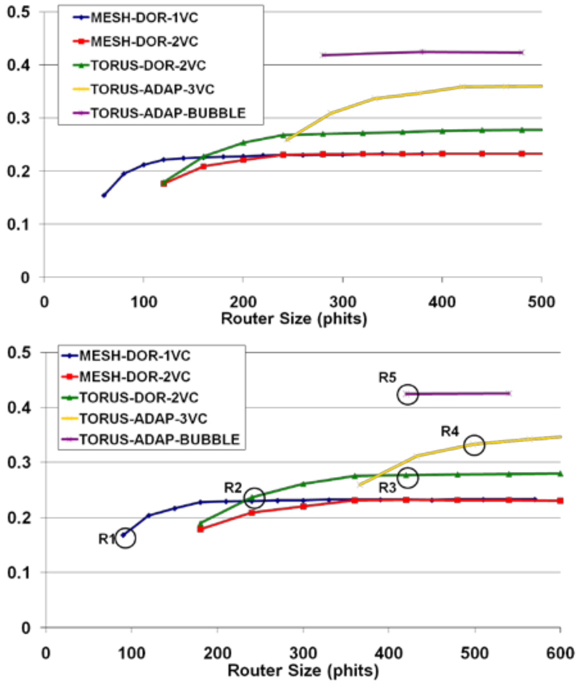

Figure 1: Average throughput for different router configurations. 4 (up) and 6 (down) message types.

The main results obtained for these experiments are shown in Figure 1. The x-axis represents the total amount of buffering devoted to the router, while the y-axis represents the average value of throughput (phits/cycle/router) for the four traffic patterns employed. In Figure 1 (below), R1 represents the simplest implementation possible; Minimal buffer size of 3 phits, DOR routing, WH flow control and a mesh topology. This means 60 phits per router for the configuration using 4 message types (3 phits/buffer × 5 buffers × 4 messages types) or 90 phits per router if 6 message types are necessary. R2 represents the potential benefits of increasing buffer capacity from 3 to 8 phits, but with the same configuration. Buffering is said to be the most area and power-consuming part of a router, but it is also a fundamental factor to obtain throughput improvements. As Figure 1 shows, for a wormhole router, buffering capacity itself is a determining factor for network performance. Regardless of the number of virtual channels or message types, increases in buffering capacity translate into throughput improvements (up to a certain limit). R3 and R4 represent the performance increases when topology and routing are modified. Implementing more efficient routing mechanisms (such as adaptivity), or making use of other point-to-point topologies (such as torus), requires the inclusion of extra virtual channels (and therefore extra buffering) in order to work correctly (deadlock avoidance), but, on the other hand, the improvements in performance can be significant if we make use of these optimizations.

Wormhole is the most common flow control mechanism for this kind of networks. Flit-level flow control gives a high level of freedom when choosing buffer size, and works with minimal-area routers (1 flit per buffer is enough for correctness) (We assume the same size for both a flit and a phit and therefore we use them interchangeably). The problem is that, as we have seen in Figure 1, if a reasonable (the knee of the curves) amount of throughput is needed, buffers must store at least 1 full packet (five phits). It is also well known [6] that the inclusion of extra virtual channels in a wormhole router helps to alleviate HOL (Head-Of-Line) blocking problems, increasing throughput without increasing storage capacity. Notwithstanding, that affirmation does not hold when the network has to deal with reactive traffic (such as in cache coherent systems, one of the most probable scenarios in CMP systems) and a large message dependency chain (each class of message should use different resources in order to avoid end-to-end deadlock). When the number of message types becomes high enough (4-6), adding extra virtual channels to each type has no beneficial effects on performance.

For all these reasons, we have also analyzed the effect of using packet-based flow controls such as Virtual Cut Through and Bubble Flow Control. With this latter method, the number of extra virtual channels needed to enable adaptivity and implement deadlock avoidance is lower, being more area efficient than wormhole-based approaches. R5 shows that more sophisticated flow control mechanisms (such as Bubble flow control) can make better use of the area increases to obtain even higher throughput results.

Full System Simulation

The next step in this preliminary evaluation was to check the effect of all these performance increases on the whole system performance. Up to now we have worked with a fixed CMP configuration, whose main parameters are listed at Table 2. The configurations of the routers' architectures correspond to the points R_i in Figure 1. As for the moment there is no hardware implementation to quantify network cost, for every network configuration it was assumed that the same amount of area would be devoted to the rest of the system. The workloads considered in this study are five applications as part of the NAS Parallel Benchmarks (OpenMP implementations) [7], four transactional benchmarks corresponding to the Wisconsin Commercial Workload suite [3], and three multi-programmed workloads belonging to the SPEC CPU2000 [8]. Results show (Fig.2) that reducing throughput from 0.43 to 0.23 phits/cycle/router increases execution time by 25% on average, e.g a chip multiprocessor SNUCA using a mesh-DOR-1VC-200phits buffer spend 25% more time for executing the set of applications than using a torus-adaptive-bubble-400phits buffer. Taking into account that the network only represents a small portion of the whole system, the increased time produced by the reduction of resources (e.g., a small decrease of L2's 16MB in our case) is practically negligible. However, the increase in power produced by an additional 25% runtime affects the whole system. Using a rough approximation relating area and energy, the interconnection network must be at least the 25% of the whole system and the improvement (reduction of area-complexity) would therefore be an unattainable 100% in order not to increase the total energy consumed.

Table 2: Simulated CMP parameters

Number of Cores	16
Window Size / outst. req per CPU	64/16
Issue Width	4
L1 I/D cache	Private, 32KB, 2-way, 64-Byte block, 1-cycle
L2 cache	16MB SNUCA, token coherence protocol, 16x16 banks, 4 banks/router
L2 cache bank	64KB, 16-way, 3-cycles, pseudo LRU, 64-Byte block
Main Memory	4GB, 260 cycles, 320 GB/s
Packet size	32 bytes/ 2 phits (command) 80 bytes/ 5 phits (data)
Network Topology	8x8 torus
Network link	128 bits / 1 cycle

In an attempt to better understand why simple structures have such a negative impact on system performance, in Figure 3 the throughput required by the Integer Sort application (IS) during its execution is plotted. There are phases of high traffic demand and others in which only a few packets cross the network. Clearly, when the throughput of messages required by the application saturates the network, it becomes the bottleneck and the application spends more time completing this phase, as is clearly seen in Figure 3. Only very low values of base latency and applications with low traffic demand would justify the use of very simple architectures.

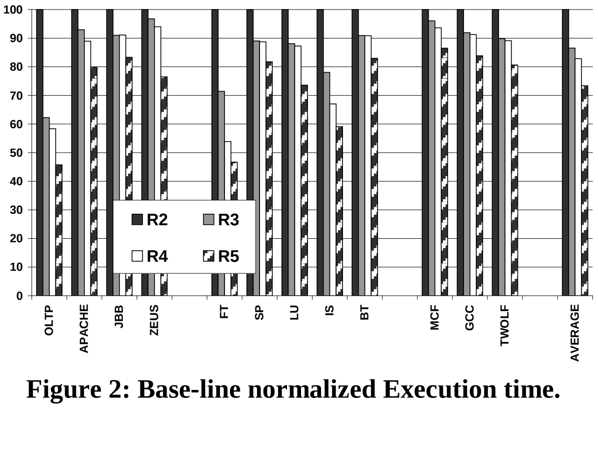

Figure 2: Base-line normalized Execution time.

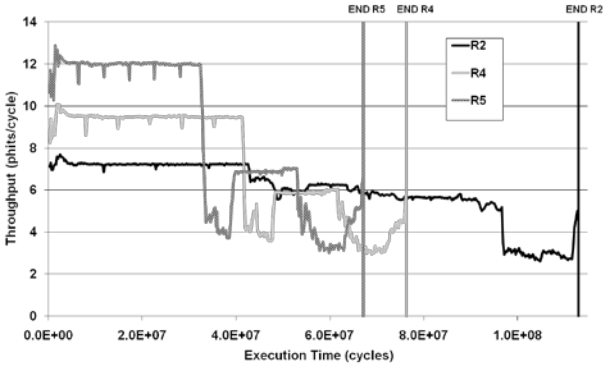

Figure 3: Throughput of the interconnection network demanded by the *IS* application.

4.- Conclusions and Future Work

These results –although incomplete without the hardware implementation– clearly show that a simplistic interconnection network can be a huge performance hindrance. As a preliminary result, reducing L2 less than 5% is enough for achieving a competitive router structure.

That is, solutions must be integral. When trying to reduce the resources devoted to the interconnection network in order to devote more to other system components, it is necessary to bear in mind the negative effects that a “decreased” interconnection network provokes on the rest of the system.

In future work we will take into account the hardware implementation of the routers selected for the evaluation. That will allow us to accurately determine how much resources should be dedicated to the interconnection network to avoid a big negative impact on the performance of the whole CMP system.

5.- References

[1] Abad, P., Puente, V., and Gregorio, J.A. “Reducing the Interconnection Network Cost of Chip Multiprocessors”. Int. Symposium on Networks-on-Chip (NOCS), pp.183-192. February 2008.

[2] Abad, P., Puente, V., Gregorio, J.A., and Prieto, P. “Rotary router: an efficient architecture for CMP interconnection networks”. 34th Int. Symposium on Computer Architecture (ISCA), pp.116-125. June 2007.

[3] Alameldeen, A.R., Mauer, C.J., Xu, M., Harper, P.J., Martin, M.M.K., Sorin, D.J., Hill, M.D., and Wood, D.A. “Evaluating non-deterministic multi-threaded commercial workloads”. 5th Workshop on Computer Architecture Evaluation Using Commercial Workloads, pp.30–38. February 2002.

[4] Balfour, J., Dally, W. “Design tradeoffs for tiled CMP on-chip networks”. International Conference on Supercomputing (ICS), pp.187-198. June 2006.

[5] Burger, D. et al. “Scaling to the end of silicon with EDGE architectures”. IEEE Computer, Volume 37, No 7, pp.44-55. 2004.

[6] Dally, W. and Towles, B. “Principles and Practices of Interconnection Networks”. Morgan Kaufmann Publishers Inc. 2003.

[7] Jin, H., Frumkin, M., and Yan, J. “The OpenMP Implementation of NAS Parallel Benchmarks and its Performance”. NASA Ames Research Center, Technical Report NAS-99-01. October 1999.

[8] Henning, J. “Spec2000: Measuring CPU performance in the new millennium”. IEEE Computer, pp.28–35. July 2000.

[9] Ho, R., Mai, K., Horowitz, M. “The Future of Wires”. Proc. of the IEEE. pp. 490-504. April 2001.

[10] Katevenis, M., Vatsolaki, P., Efthymiou, A. “Pipelined Memory Shared Buffer for VLSI Switches”. SIGCOMM , pp.39-48. 1995.

[11] Kumar, A., Peh, L., Kundu, P., Jha, N. “Express Virtual Channels: Towards the Ideal Interconnection Fabric”. International Symposium on Computer Architecture (ISCA). June 2007.

[12] Kumar, A., Kundu, P., Singh, A.P., Peh, L.S., and Jha, N.K. “A 4.6 Tbits/s 3.6 GHz Single-cycle NoC Router with a Novel Switch Allocator in 65nm CMOS”. Int. Conf. on Computer Design (ICCD). Oct. 2007.

[13] Magnusson, P. et al. “Simics: A full system simulation platform”. Computer, Vol. 35, No. 2, pp.50-58. 2002.

[14] Martin, M.M.K. et al. “Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset”. SIGARCH Computer Architecture News, Vol. 33, No. 4, pp.92-99. November 2005.

[15] Mullins, R., West, A., and Moore, S. “Low-latency virtual-channel routers for on-chip networks”. 31st International Symposium on Computer Architecture (ISCA), pp.188-197. June 2004

[16] Puente, V., Beivide, R., Gregorio, J., Prellezo, J., Duato, J., and Izu, C. “Adaptive bubble router: a design to improve performance in torus networks”. Int. Conference on Parallel Processing, pp.58-67. 1999.

[17] Puente, V., Gregorio, J., and Beivide, R. “SICOSYS: an integrated framework for studying interconnection network performance in multiprocessor systems”. 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, pp.15-22. September 2002.

[18] Y.H. Song, T.M. Pinkston, “A Progressive Approach to Handling Message-Dependent Deadlock in Parallel Computer Systems”, IEEE TPDS, Vol. 14, No. 3, pp 259-275, March 2003.

[19] Taylor, M.B. et al. “The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs”. IEEE Micro, Vol. 22, Issue 2, pp.25-35. 2002

[20] Wentzlaff, D., Griffin, P., Hoffmann, H., Bao, L., Edwards, B., Ramey, C., Mattina, C., Miao, C., Brown III, J., Agarwal, A. “On-Chip Interconnection Architecture of the Tile Processor,” IEEE Micro, vol. 27, no. 5. pp. 15-31. Sep./Oct. 2007.