Performance Sensitivity of NUCA Caches to On-Chip Network Parameters

Alessandro Bardine*, Manuel Comparetti*, Pierfrancesco Foglia*, Giacomo Gabrielli*, Cosimo A. Prete*
Dipartimento di Ingegneria dell'Informazione, Università di Pisa
Via Diotisalvi 2, 56122, Pisa, Italy
{*alessandro.bardine, manuel.comparetti, foglia, giacomo.gabrielli, prete*}*@iet.unipi.it*

Abstract

Non Uniform Cache Architectures (NUCA) are a novel design paradigm for large last-level on-chip caches that has been introduced to deliver low access latencies in wire-delay dominated environments. Their structure is partitioned into sub-banks and the resulting access latency is a function of the physical position of the requested data. Typically, to connect the different sub-banks and the cache controller, NUCA caches employ a switched network, made up of links and routers with buffered queues; the characteristics of such switched network may affect the performance of the entire system. This work analyzes how different parameters for the routers, namely cut-through latency and buffering capacity, affect the overall performance of NUCA-based systems for the single processor case, assuming a reference organization proposed in literature. The results indicate that the sensitivity of the system to the cut-through latency is very high and that limited buffering capacity is sufficient to achieve a good performance level. As a consequence, we propose an alternative NUCA organization that limits the average number of hops experienced by cache accesses. This organization is better performing in most of the cases and scales better as the cut-through latency increases, thus simplifying the implementation of routers.

1. Introduction

Non-Uniform Cache Architectures have been proposed as a new design paradigm for large last-level on-chip caches [18] in order to reduce the effects of *wire delays*, which significantly limit the performance scaling of today's high clock frequency microprocessors [2]. This is achieved by the adoption of a storage structure partitioned into sub-banks, with each sub-bank being an independently accessible entity, and by the adoption of a fast interconnection network to connect the banks and the cache controller. The access latency exhibited by a NUCA cache is a function of the physical location of the requested line; i.e., a line belonging to a bank located near the cache controller will be accessed faster than another line belonging to a bank located farther away. The mapping between cache lines and banks can be either static or dynamic. The former approach leads to the Static NUCA (S-NUCA) scheme: a line can be located in a single specific bank, univocally determined by its address. The latter approach leads to the Dynamic NUCA (D-NUCA) scheme: a line can be located in one of a set of allowed bank locations, which collectively form a *bank set*, and each bank of the bank set behaves like a single way of a set-associative cache [18]. Lines can dynamically migrate from one bank to another, provided that it belongs to the pertaining bank set, and the migration is triggered by a certain number of consecutive line accesses. Different policies for mapping of data on bank sets and data migration in D-NUCA caches have been proposed and evaluated in literature [18].

A viable solution to connect the banks and the controller of a NUCA cache is represented by an on-chip network. The paradigm introduced by on-chip networks tends to favour the reuse of design and verification efforts, which is particularly important for modern VLSI processes: many digital design blocks, namely network links and routers, can be used repeatedly to form a complete communication infrastructure across the chip [16]. The resulting interconnection scheme is more scalable than traditional approaches based on broadcast mediums, such as busses and rings [6]. The intrinsic features of NUCA caches introduce constraints on the design of the on-chip network, in particular on the design of network routers. These constraints reflect the characteristics of the network itself, such as topology, routing, and flow control, but, primarily, they are influenced by the way with which last-level on-chip caches are accessed by the CPU. A fundamental property of the NUCA on-chip network is that it is self-throttling, as it is common for processor-to-memory interconnects [10]. In fact, non-

*Members of the HiPEAC European Network of Excellence on High-Performance Embedded Architecture and Compilation.

This work is partially supported by the SARC project funded by the European Union under the contract no. 27648.

1550-6533/08 $25.00 © 2008 IEEE
DOI 10.1109/SBAC-PAD.2008.20

blocking caches are able to support only a limited number of outstanding misses, therefore the number of simultaneous requests on the L2 or, more generally, on the last-level cache, is limited by the number of outstanding misses supported by the higher level. This number is determined by the number and size of the Miss Status Holding Registers (MSHRs) [19], which keep track of the pending misses, coalescing multiple outstanding misses for the same cache line into a single request to the following level of the memory hierarchy. From these considerations, we might expect the network traffic offered to the NUCA on-chip network to be quite moderate. Since the access latency is the fundamental performance metric of a NUCA cache, we also might expect that latency, instead of bandwidth, should be be the primary design goal for the switching elements of the network.

Different implementations of routers have been proposed for high performance on-chip networks [21], but it is not clear which is the most suitable router architecture to face the design costraints posed by a NUCA cache scenario. In order to characterize such design constraints, in this paper we analyze how the performance of a reference NUCA L2 cache [18, 4, 3] is influenced by different values of cut-through latency and buffering capacity of routers. Such parameters, in fact, permit to guide in the selection of an adequate router architecture [12].

The results indicate that NUCA performance is strongly dependent from router latency, implying that different implementations of routers can significantly affect the overall performance of the system. Buffer capacity is not an issue, as also a single-message sized buffer permits to achieve adequate performance. In addition, taking into account such considerations, we derive an alternative NUCA organization that is better performing and is able to reduce the strong dependency of performance from the router latency.

2. Related Work

The NUCA cache paradigm has been introduced by Kim *et al.* [18] for the single processor case. Chishti *et al.* [7] have proposed an optimization scheme, called NuRAPID, aiming at increasing the energy efficiency and the performance of NUCA caches in a single core configuration. Several studies have focused on the application of NUCA caches to CMP (Chip Multiprocessor) architectures, focusing on the evaluation of the best sharing degree [15], on the effectiveness of block migration for multithreaded workloads [5] and on optimizations for block placement and coherence management [8]. None of the former studies has explicitly focused on the impact of the network architecture on the overall performance.

Other studies have characterized the effect of network elements on performance and have indicated possible solutions, but none of them explicitely focused on cut-through

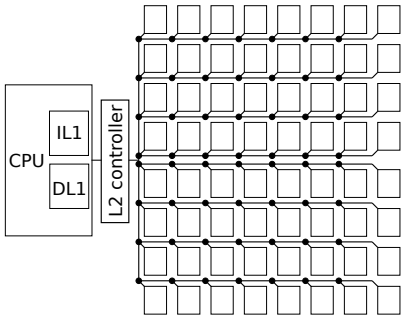

Figure 1. Partial 2D mesh topology. The NUCA structure represented here is made up of 64 banks (8x8). The black circles depict the network routers. For this study, the same topological organization is used for both S-NUCA and D-NUCA architectures. The CPU, with its instruction and data L1 caches, is attached through a bus to the L2 cache controller, which is the injection point of the NUCA on-chip network. For D-NUCA, each row of banks corresponds to a bank set.

latency and buffer size of routers. Muralimanohar *et al.* [21], by including network delay parameters in the full design space exploration of an S-NUCA cache, have shown how in a many-bank architecture the increased number of hops to reach a cache bank dominates the overall access latency. Their study also investigates the improvements on the performance of an S-NUCA employing different interconnection schemes distributed on multiple metal layers, assuming a 3 stage pipelined router, with an unloaded latency of 3 cycles and virtual channel capability. Their work has also presented an evaluation of the cache access time when the router latency assumes two different values, but the effects on the overall system performance are not reported. Muralimanohar *et al.* [22] have highlighted that the contention on network resources has a non negligible impact on the performance of NUCA caches. Jin *et al.* [17] have shown that the network traversal time is the main component of the latency of a NUCA access. In order to reduce latency ad improve overall performance, they have proposed a block management policy (Fast-LRU) and have evaluated some topological improvements with respect to a full mesh architecture.

On-chip networks have been introduced as a common communication infrastructure for system-on-chips [6, 9]. In the field of general purpose high-performance systems, on-chip networks have been proposed for different purposes. A typical application is represented by tiled CMP architectures [26, 31], which are based on a matrix of nodes (called tiles), with each node comprising a processing unit and, in most of the cases, a certain amount of cache memory. An on-chip network connecting the tiles is responsible to transport the data and synchronization messages, enabling chip-wide communications. Intel has adopted an on-chip network to connect the 80 cores of the tera-scale research

project called Teraflops [28]. The TRIPS processor prototype [24] has employed different on-chip networks to interconnect the execution units, the SRAM cache banks of an S-NUCA L2 cache and the DRAM controllers; since the prototype has been built with a 130nm manufacturing technology and its operating frequency is fixed at 500 MHz, the effects of wire delays encountered for that design are not so critical as in deep sub-micron manufacturing technologies, as pointed out by the same authors [24]. The impact of router delay on the performance of a grid processor employing an inter-ALU operand network has been analyzed in [25], and the results indicate that such network structures are highly sensitive to this parameter.

The infrastructure of on-chip networks for general purpose systems must support high operating frequencies, as it is common for modern chips, and several works have focused on the design of high-performance network routers for this kind of applications. Most of them focuses on a pipelined organization, which is able to guarantee a high throughput. For instance, Peh and Dally [23] propose a speculative router model with a 3-stage pipelined architecture with virtual channels. Clearly, the main design objectives for high-performance routers are to minimize latency, by reducing the number of pipeline stages, and to maximize throughput; Mullins *et al.* [20] have proposed an innovative router architecture with virtual channels that is able to deliver a flit in a single cycle, using speculation mechanisms, buffer bypassing capabilities and removing the arbitration logic from the critical path; Jin *et al.* [17] also adopt a single cycle router with virtual channel capability, but its architectural implementation has not been detailed. However, while such architectures have been proposed in literature, single stage routers are not yet an industrial reality [21] due to the issues that emerge in the digital design process. Therefore, it is important to analyze the performance sensitivity of NUCA caches to the router delay, since designing low latency network routers is not a trivial task.

Concerning the buffer capacity of routers, it's important to analyze the buffering requirements of NUCA on-chip networks, since, apart from performance, the size of queues has a significant impact on dynamic and static energy consumption and on die area occupancy [29, 30].

3. On-Chip Network Architecture and Router Model

The analysis described in this paper assumes a reference NUCA structure which has been derived from previous work [18, 13, 4, 3]. The topology of the on-chip network is derived from a 2D mesh, which will be called *partial 2D mesh* in the following, since only a subset of the links of a full 2D mesh are employed in order to reduce the area overhead. Different topology schemes of direct networks, such as toroids, have not been considered for this study because 2D meshes map more effectively onto 2D silicon substrates: even if toroids can be mapped onto 2D silicon substrates, doing so in a wire-delay dominated environment would not introduce significant improvements, except for bandwidth gain, and will increase the area occupied by the interconnection fabric. The sole injection point of the network is the L2 cache controller, which is assumed to be directly attached to the external DRAM controller. The network links are bidirectional, so two traffic flows on opposite directions are completely independent from each other. The partial 2D mesh topology of the on-chip network is represented in Fig. 1.

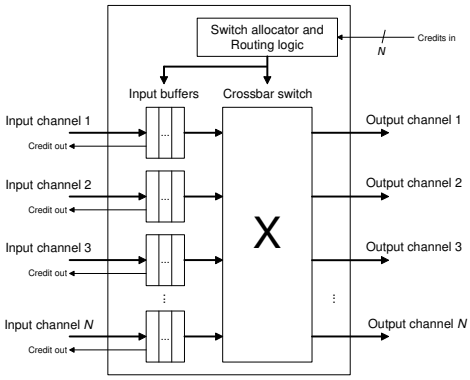

Figure 2. Reference architecture for network routers. The routers are assumed to be input-buffered. A crossbar switch is adopted to minimize contention on output channels. The credit signals are necessary to implement credit-based flow control.

The reference architecture for the NUCA on-chip network is based on a wormhole scheme, with routing and flow control policies working on a per-flit basis. The size of a flit is assumed to be equal to the link width. The routing scheme is deterministic, X-Y dimension ordered; for the NUCA architectures considered in this work a flit is first propagated along the vertical dimension (vertical links in Fig. 1), then it is propagated along the horizontal dimension (horizontal links in Fig. 1). For D-NUCA caches, since a bank set is mapped to a single row of banks, first a flit has to reach the pertaining bank set, then it is propagated to the nodes attached to the banks of its bank set, starting from the nearest one to the cache controller. This causes the global access latency to raise as the distance of the requested cache line from the first node of the pertaining bank set increases.

The general architecture of the network routers which have been modeled for this analysis is represented in Fig. 2. The network routers are assumed to be input buffered, and the buffers are managed on a per-flit basis in a FIFO manner. The flow control is credit-based, so each router must keep track of the status of the input queues of its neighbours. This is accomplished by adding two extra credit signals to the link width (*credit in* and *credit out* in Fig. 2).

When a router removes a flit from one of its queues, it asserts the corresponding *credit out* signal to notify the sender that more buffering space is available. On the other side, a sender is allowed to transmit a flit only if the number of collected credits is at least 1. In order to guarantee routing fairness, a round-robin scheme is used when multiple transmission requests for the same channel occur. In order to implement this feature, the router needs to keep track of the direction of the last packet sent for each output channel.

The variable router parameters that have been considered in this study are: 1) cut-through latency (expressed in number of clock cycles); 2) buffer capacity (expressed in number of flits per input queue). The cut-through latency is equal to the delay needed to transfer a flit from the source input channel to the destination output channel of a router (Fig. 2), assuming a no load condition. In order to calculate the hop latency, i.e. the latency to move from one node to the next, our model takes the sum of the cut-through latency and the link latency (delay introduced by the transmission of signals on wires). The wire length has been determined assuming the full width or height of the cache banks, depending on the link direction (horizontal or vertical). The methodology that has been applied to calculate the physical parameters is described in the next section. The router model that has been simulated checks the state of the input ports, of the input buffers and of the attached links at each clock cycle, and when no conflicts occur it triggers the necessary transmission events.

4. Physical Parameters

Computer architects often rely on existing analytical models to estimate the characteristics of VLSI circuits. One of such models has been adopted by the CACTI tool [32] to estimate the area, the access time and the energy consumption of on-chip SRAM caches. The analysis described in this paper is based on values obtained from CACTI 5.1 [27], which derives the technological parameters for devices and wires from the projections of the ITRS 2005 report [1].

For each of the considered L2 cache architectures, i.e. UCA, S-NUCA and D-NUCA, we assumed a fixed cache size of 8 Mbytes and a line size of 64 bytes. For the UCA architecture, we modeled access time and cycle time for different cache configurations obtained varying the associativity. We found that the best tradeoff between high associativity (hence lower miss rate) and low latencies for this UCA model is a degree of associativity of 4. For NUCA caches we also modeled bank sizes and latencies with CACTI. The experiments that were conducted led to the following optimal configurations for NUCA: D-NUCA with 64 banks, 8x8, globally behaving like an 8-way set associative cache; S-NUCA with 32 banks, 8x4, with each bank being 4-way set associative (Tab. 1). Such configurations differ from the ones proposed by Kim *et al.* in [18] mainly because we referred to different technology projections [1]. All the configurations assume serial access to tag and data arrays, as this is a common design choice for reducing energy consumption of large on-chip caches.

Table 1. Configuration parameters for the CPU and the memory hierarchy.

Parameter	Value
Technology node	65nm
CPU	Alpha 21264
Fetch/Issue/Commit width	4 / 4 int. + 2 f.p. / 11
Functional units	4 int. ALUs, 4 int. MUL/DIVs 1 f.p. ALU, 1 f.p. MUL
Instr. L1 cache	64 KB, 2-way s.a., 64 B line 1 cycle hit latency
Data L1 cache	64 KB, 2-way s.a., 64 B line 3 cycle hit latency, 2 ports
Cache MSHR size	8 entries, each points up to 4 targets
Main memory latency	300 cycles
UCA L2 cache	8 MB, 4-way s.a., 64 B line access time = 37, cycle time = 4 (cycles)
S-NUCA L2 cache	8 MB, 64 B line 32 banks (8x4), each one 4-way s.a. with acc. time = 13, cycle time = 3 (cycles)
D-NUCA L2 cache	8MB, 64 B line 64 banks (8x8), each one direct mapped with acc. time = 11, cycle time = 2 (cycles)

A fundamental parameter of the NUCA on-chip network is the latency of wires, i.e. the latency of transmissions on network links. Firstly, the length of links were determined according to the physical dimensions of banks, which were derived from CACTI. Then, we calculated link latencies applying the delay-optimal repeated wire model proposed by Ho [14]. Since CACTI employs this same model to determine the latency of wiring connections for traditional UCA caches, this approach leads to a general uniformity of the performed analysis, improving the accuracy and the validity of this study. Furthermore, we used the same technological parameters for wires used by CACTI, which in turn are derived from the projections of the ITRS report; from the two proposed projection scenarios, i.e. aggressive or conservative, we selected the conservative one. We assume that network links (for NUCA) and high-level interconnections (for UCA) are both based on semi-global wires. After the application of the selected physical model, the flit transmission latency on a link fits into 2 clock cycles for S-NUCA and into 1 cycle for D-NUCA. It is worth noting that since the on-chip network assumed in this study works syncronously, the latency of wires should not be far from an integer multiplier of the clock cycle time, in order to maximize the utilization of the links. The configurations described above obey to this principle, and the sizing of cache banks derived by our study leads to an efficient utilization of the communication infrastructure.

The entire analysis is based on the 65nm technology

node. The different architectures considered in this work assume a 16 FO4 (fanout of four) clock cycle time, which roughly corresponds to a 5 GHz operating frequency [1].

5. Simulation Methodology

The simulation platform adopted for this study is based on an extended version of the cycle-accurate execution-driven simulator *sim-alpha* [11]. The original version of sim-alpha has been augmented to reproduce the behavior of a single processor system backed by a UCA, S-NUCA or D-NUCA L2 cache. The level of detail of the NUCA architecture model allows to specify different parameters for the on-chip network, including the latency of transmissions on links, and the buffer capacity and latency of routers. The behavior of network routers reflects the characteristics of their reference architecture described in Section 3. The original sim-alpha model for cache banks has been augmented to support a customizable cycle time, i.e. the minimum interval between two consecutive requests that can be issued: both access time and cycle time can now be specified for cache banks, thus offering a better modeling accuracy (this can be relevant when applications exhibit an high L1 miss-rate and/or a high degree of "burstiness" of L2 accesses). In this study we assume that UCA caches employ the necessary latching logic to support multiple on-going accesses, spaced out by the proper cycle time, as the reference cache architecture modeled by CACTI does.

The simulated systems are based on the Alpha 21264 microprocessor (Tab. 1). Each of the considered systems assumes a single CPU with splitted L1 instruction and data caches, and an on-chip 8 Mbytes L2 cache. The architecture of the L2 cache is varied between UCA, S-NUCA and D-NUCA, assuming the configuration parameters reported in Tab. 1, which were determined as described in Section 4.

Different implementation policies have been proposed for D-NUCA: mapping policies, line search policies and migration policies [18]. Mapping policies involve how data are mapped onto bank sets, i.e. on which blocks every data line is allowed to reside. Line search policies determ the way in which the banks of a bank set are searched for the data, e.g. sequentially or by means of a broadcast request. The adoption of a smart search array near to the controller has also been proposed in [18], in order to lower the miss detection latency. Migration policies set the promotion trigger, i.e. the number of accesses after which a line is promoted, and the promotion distance, i.e. the number of banks traversed upon a promotion. In order to keep the number of variable parameters reasonably low, in this study a specific set of policies has been selected for D-NUCA, leading to a configuration that is a good tradeoff between performance and complexity. The selected policies are: simple mapping, with each row of banks making up a bank set; broadcast search without smart search array; promotion in the adjacent bank upon each hit (1 bank/1 hit).

Table 2. Benchmarks selected for this study.

Application	Suite	FFWD	RUN
art	SPECFP	2.2B	200M
applu	SPECFP	267M	650M
bt	NPB	800M	650M
bzip2	SPECINT	744M	1.0B
cg	NPB	600M	200M
equake	SPECFP	4.459B	200M
galgel	SPECFP	4.0B	200M
gcc	SPECINT	2.367B	300M
mcf	SPECINT	5.0B	200M
mesa	SPECFP	570M	200M
mgrid	SPECFP	550M	1.06B
parser	SPECINT	3.709B	200M
perlbmk	SPECINT	5.0B	200M
sp	NPB	2.5B	200M
twolf	SPECINT	511M	200M

For the NUCA on-chip network, the link width is 128 bits for each direction, as the flit size; this means that a flit can be transmitted on a link once in a row.

Our analysis was performed assuming the workload listed in Tab. 2, which comprises applications from the SPEC CPU2000 and the NAS Parallel Benchmarks suites. To reduce the overall simulation time, for each application we selected a representative phase of the entire execution, as in [18]. Tab. 2 reports, for each benchmark, the number of instructions which were skipped from the start (FFWD) and the number of simulated instructions (RUN).

6. Results

For NUCA caches, we performed a set of simulations varying the cut-through latency from 0 to 5 clock cycles. When the cut-through latency is 0, we assume that the hop latency is given only by the link latency, whose length is equal to the width/height of cache banks; however, all the internal activity of the routers is modeled in detail as for the other cut-through latency values. For each simulation point, we also varied the buffering capacity of routers, selecting three different configurations: 5 flits per input queue, 10 flits per input queue, and infinite buffering capacity (ideal case). These values were selected because the size of one of the most frequently occurring packet type on the network is 5 flits: 1 header flit, storing control parameters and the cache line address, plus 4 flits for storing the cache line content (64 bytes). All the collected results are compared against the ones achieved by the L2 UCA architecture, whose configuration has been described in the previous section. In the following, the average IPC (Instructions Per Cycle) over the entire workload is selected as synthetic metric to quantitatively represent the performance level of the systems under evaluation.

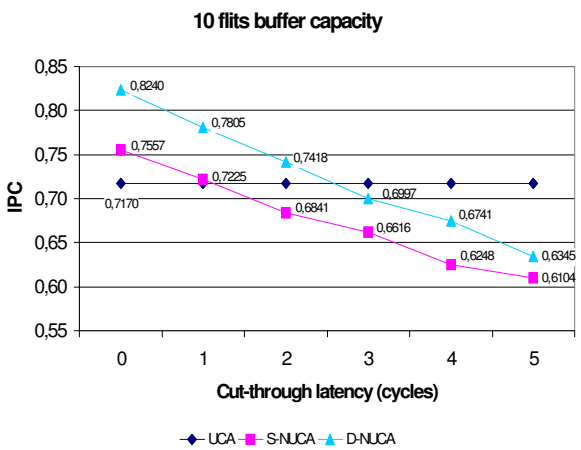

Figure 3. IPC vs. cut-through latency. S-NUCA and D-NUCA architectures, with the buffering capacity at 10 flits are compared with a traditional UCA-based system.

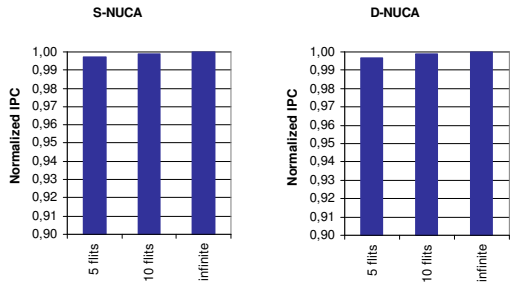

Figure 4. IPC vs. buffer capacity for S-NUCA and D-NUCA due to limited buffering capacity. The IPC is normalized w.r.t. the ideal case with infinite buffering capacity. Data assume 1 cycle cut-through latency.

Fig. 3 shows the average IPC for the entire workload as the cut-through latency increases from 0 to 5 cycles, for 10 flits buffering capacity. We can highlight that the overall system performance for NUCA is highly sensitive to cut-through latency. While D-NUCA always outperforms S-NUCA, the performance of NUCA-based architectures rapidly decreases from a simulation node to the next. For 2 cycles latency, S-NUCA is less performing than UCA, while the benefits of employing a D-NUCA are poor (only 3.5% improvement over UCA). From this node on, the extra efforts that would be needed to design the communication infrastructure commonly adopted by NUCA architectures would not be acceptable. The high sensitivity witnesses that the delay introduced by the on-chip network has strong effects on the overall system performance, while the latency of bank accesses becomes less influential as we move towards higher latencies for routers. However, if routers are able to deliver flits in a single cycle, D-NUCA caches offer a good improvement over UCA (+8.9%). This improvement is larger, as expected, for the null cut-through latency case (+14.9%). These results clearly show that multi-cycle router architectures are not adequate for NUCA and custom architectures should be used: even a 2 cycles routing delay introduces an unacceptable performance degradation.

Focusing on a single value for cut-through latency, e.g. 1 cycle, it is possible to quantitatively evaluate the performance degradation due to limited buffer capacity with respect to the ideal router case (infinite buffering capacity), for both S-NUCA and D-NUCA. Fig. 4 shows this degradation, reporting the average IPC over the entire workload, normalized with respect to the ideal router case with infinite buffering capacity. The resulting performance degradation is negligible even for the 5 flits buffering capacity; for both S-NUCA and D-NUCA the degradation is less than 0.1%. This result suggests that limited buffering capabilities don't jeopardize the performance improvements introduced by NUCA structures.

While the single cycle cut-through latency for routers is a reasonable operating condition for a NUCA cache, more aggressive implementations could reach a latency below one cycle. E.g., the speculative implementation proposed by Mullins *et al.* [20] is claimed to reach a 12 FO4 delay for the critical path. This latter and similar designs, however, must be supported by an underlying network architecture that doesn't restrict the latency of routers to be aligned on clock cycle boundaries. These implementations should assume only the hop latency, given by the sum of router latency and wire latency, to be aligned on clock cycle boundaries. While this study has focused on a quite aggressive 16 FO4 clock cycle time, modern CMP implementations opt for lower values of cycle time, i.e. 24 FO4 and more. For these scenarios, the aggressive architectures of routers operating below the single clock cycle could be profitable, thus leading to stronger performance benefits for NUCA architectures, especially for D-NUCA. However, the feasibility of these router implementations still needs to be investigated.

7. Reducing Sensitivity to Router Delay

The results shown in the previous section indicate that the latency introduced by the on-chip network has a significant impact on the average access latency of a NUCA cache and, as a consequence, on the overall performance of the system. One of the most effective ways to mitigate this effect is to reduce the average number of hops that cache accesses experience. This can be achieved by reducing the number of banks (assuming a constant cache capacity, this means that the size of banks is increased) or clustering the banks so that each cluster is attached to a network node, while keeping the bank size fixed. Since a wire-delay dominated environment puts strong constraints on bank size and

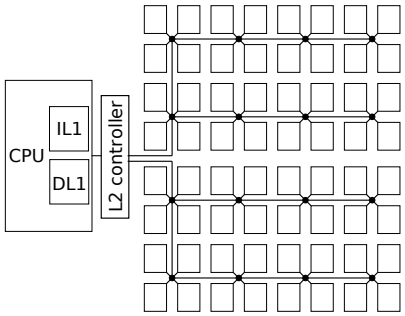

Figure 5. Partial 2D mesh topology with 4 banks per node (clustered approach).

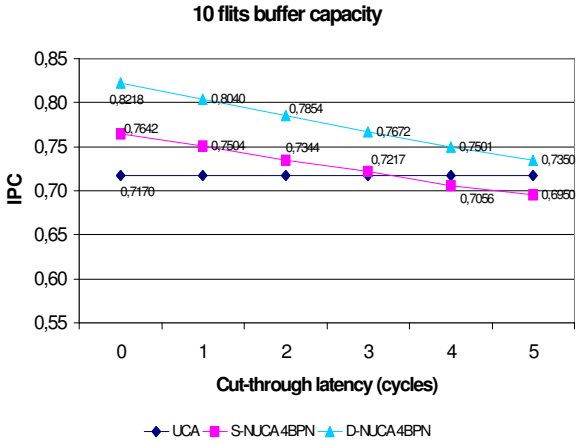

Figure 6. IPC vs. cut-through latency for the clustered scheme (4BPN = 4 banks per node).

topology, the only relevant scheme that we take into account is a clustered configuration with 4 banks per cluster (Fig. 5). The partitioning of the address space inside a single cluster is obtained by checking the least significant bits from the index field of the address.

Focusing on D-NUCA architectures, which offer the highest performance level, modifying the network topology is not enough to significantly reduce the average access latency: since banks belonging to the same cluster have the same distance from the controller, assuming a 1 bank/1 hit migration policy is not effective in reducing data access latency. For this configuration we adopted a 1 cluster/1 hit promotion policy, in which accessed data are migrated directy in the next cluster towards the controller. This implies an alternative logical organization, which involves the way with which lines are mapped onto cache banks: while in the reference scheme each bank set is mapped onto a single row of banks, with the clustered approach each bank set is mapped onto a row of clusters. Each column of clusters now behaves like a single way of a set associative cache. Since our reference configuration was made up of 64 banks arranged into 8 rows and 8 columns, the clustered D-NUCA with 4 banks per node is globally 4-way set associative.

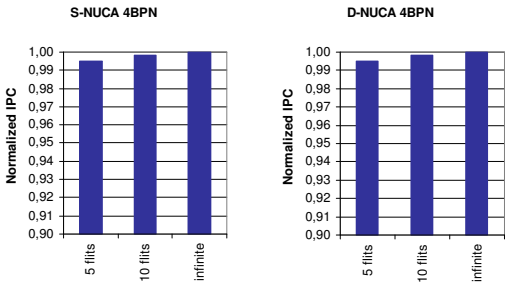

Figure 7. IPC vs. buffer capacity for the clustered scheme. Data assume 1 cycle cut-through latency.

Fig. 6 reports the performance achieved by the new scheme, when applied to both S-NUCA and D-NUCA architectures. Except for the null cut-through latency case, the clustered scheme always outperforms the reference one; for D-NUCA, the improvement over the UCA scheme is significant: +12.2% at 1 cycle cut-through latency, and +9.5% at 2 cycles. These results indicate that the minimal cut-through latency constraint can be relaxed, as this configuration is much more scalable w.r.t. the reference architecture. Fig. 7 shows the IPC loss due to limited buffering capacity for the clustered scheme, which is is negligible for both S-NUCA and D-NUCA.

The clustered scheme, while being better performing, reduces the number of routers, thus leading to a simpler implementation. The additional overhead is given by the additional ports to connect each router to its local banks (3 additional ports w.r.t. the previous scheme). However, a solution based on multiplexing a single port to connect to the local banks could be used, employing a simple arbiter.

8. Conclusions

This work investigates the impact of some on-chip network parameters on the overall performance of NUCA-based uniprocessor systems under a realistic workload. In particular, we have focused on cut-through latency and buffering capacity of network routers. We have assumed a UCA architecture as a reference performance level, in order to identify under which conditions the NUCA scheme outperforms the traditional UCA. The results indicate that NUCA systems exhibit a high sensitivity to router cut-through latency, thus meaning that router architectures with latencies of one cycle or less are needed; moreover, varying buffer capacity has almost negligible effects on the overall performance and the usage of network buffering resources is moderate. From these considerations, we have identified an alternative NUCA organization that performs better and is much less sensitive to variations on the router latency. This organization is based on the clustering of banks and assumes 4 banks per node, thus relaxing the strong constraint

on latency-oriented routers.

9. Acknowledgements

We wish to thank Stephen Keckler who furnished us with the initial version of the sim-alpha simulator and José Duato for his suggestion on our work.

References

[1] Int. Technology Roadmap for Semiconductors, 2005 Edition Report.

[2] V. Agarwal, M. S. Hrishikesh, S. W. Keckler, and D. Burger. Clock rate versus IPC: the end of the road for conventional microarchitectures. In *Proc. of the 27th Int. Symp. on Computer Architecture*, pp. 248–259, 2000.

[3] A. Bardine, P. Foglia, G. Gabrielli, C. A. Prete, and P. Stenstrom. Improving power efficiency of D-NUCA caches. *ACM SIGARCH Computer Architecture News*, 35(4):53–58, 2007.

[4] A. Bardine, P. Foglia, and C. Prete. Way adaptable D-NUCA caches. In *Proc. of the Poster Session of the 2nd International Summer School on Advanced Computer Architecture and Compilation for Embedded Systems*, pp. 213–216, 2006.

[5] B. M. Beckmann and D. A. Wood. Managing wire delay in large chip-multiprocessor caches. In *Proc. of the 37th Int. Symp. on Microarchitecture*, pp. 319–330, 2004.

[6] L. Benini and G. De Micheli. Networks on chips: a new SoC paradigm. *IEEE Computer*, 35(1):70–78, 2002.

[7] Z. Chishti, M. D. Powell, and T. N. Vijaykumar. Distance associativity for high-performance energy-efficient non-uniform cache architectures. In *Proc. of the 36th Int. Symp. on Microarchitecture*, pp. 55–66, 2003.

[8] Z. Chishti, M. D. Powell, and T. N. Vijaykumar. Optimizing replication, communication, and capacity allocation in CMPs. In *Proc. of the 32nd Int. Symp. on Computer Architecture*, pp. 357–368, 2005.

[9] W. J. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In *Proc. of the 38th Conference on Design Automation*, pp. 684–689, 2001.

[10] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2003.

[11] R. Desikan, D. Burger, S. Keckler, and T. Austin. Sim-alpha: a validated, execution-driven Alpha 21264 simulator. Technical report, Department of Computer Sciences, University of Texas at Austin, 2001.

[12] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection Networks: An Engineering Approach*. Morgan Kaufmann, 2003.

[13] P. Foglia, D. Mangano, and C. A. Prete. A cache design for high performance embedded systems. *Journal of Embedded Computing*, 1(4):pp. 587–598, 2005.

[14] R. Ho. *On-chip wires: scaling and efficiency*. PhD thesis, Stanford University, 2003.

[15] J. Huh, C. Kim, H. Shafi, L. Zhang, D. Burger, and S. W. Keckler. A NUCA substrate for flexible CMP cache sharing. In *Proc. of the 19th Int. Conference on Supercomputing*, pp. 31–40, 2005.

[16] A. Jantsch and H. Tenhunen. Will networks on chip close the productivity gap? In *Networks on Chip*, chapter 1, pp. 3–18. Kluwer Academic Publishers, 2003.

[17] Y. Jin, E. J. Kin, and K. H. Yum. A domain-specific on-chip network design for large scale cache systems. In *Proc. of the 2007 IEEE 13th Int. Symp. on High Performance Computer Architecture*, pp. 318—327, 2007.

[18] C. Kim, D. Burger, and S. W. Keckler. An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches. In *Proc. of the 10th Int. Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 211–222, 2002.

[19] D. Kroft. Lockup-free instruction fetch/prefetch cache organization. In *Proc. of the 8th Annual Symp. on Computer Architecture*, pp. 81–87, 1981.

[20] R. Mullins, A. West, and S. Moore. Low-latency virtual-channel routers for on-chip networks. In *Proc. of the 31st Int. Symp. on Computer Architecture*, pp. 188–197, 2004.

[21] N. Muralimanohar and R. Balasubramonian. Interconnect design considerations for large NUCA caches. In *Proc. of the 34th Int. Symp. on Computer Architecture*, pp. 369–380, 2007.

[22] N. Muralimanohar, R. Balasubramonian, and N. Jouppi. Optimizing NUCA organizations and wiring alternatives for large caches With CACTI 6.0. In *Proc. of the 40th Int. Symp. on Microarchitecture*, pp. 3–14, 2007.

[23] L.-S. Peh and W. J. Dally. A delay model and speculative architecture for pipelined routers. In *Proc. of the 7th Int. Symp. on High-Performance Computer Architecture*, pp. 255–266, 2001.

[24] K. Sankaralingam et al. Distributed microarchitectural protocols in the TRIPS prototype processor. In *Proc. of the 39th Int. Symp. on Microarchitecture*, pp. 480–491, 2006.

[25] K. Sankaralingam, V. A. Singh, S. W. Keckler, and D. Burger. Routed inter-ALU networks for ILP scalability and performance. In *Proc. of the 21st Int. Conference on Computer Design*, pp. 170–177, 2003.

[26] M. B. Taylor et al. The Raw microprocessor: a computational fabric for software circuits and general purpose programs. *IEEE Micro*, 22(2):25–35, 2002.

[27] S. Thoziyoor, N. Muralimanohar, and N. P. Jouppi. CACTI 5.0. Technical report, HP, 2007.

[28] S. Vangal et al. An 80-tile 1.28TFLOPS network-on-chip in 65nm CMOS. In *Digest of Technical Papers, Int. Solid-State Circuits Conference*, pp. 98–589, 2007.

[29] H. Wang, L.-S. Peh, and S. Malik. Power-driven design of router microarchitectures in on-chip networks. In *Proc. of the 36th Int. Symp. on Microarchitecture*, pp. 105–116, 2003.

[30] H.-S. Wang, L.-S. Peh, and S. Malik. A power model for routers: modeling Alpha 21364 and InfiniBand routers. *IEEE Micro*, 23(1):26–35, 2003.

[31] D. Wentzlaff et al. On-chip interconnection architecture of the Tile processor. *IEEE Micro*, 27(5):15–31, 2007.

[32] S. J. Wilton and N. P. Jouppi. CACTI: an enhanced cache access and cycle time model. *IEEE Journal of Solid-State Circuits*, 31(5):677–688, 1996.