Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 56 (2015) 403 – 408

Procedia Computer Science

The 2nd International Workshop on Design and Performance of Networks on Chip (DPNoC 2015)

The Impact of Traffic Localisation on the Performance of NoCs for Very Large Manycore Systems

Sharifa Al Khanjari*, Wim Vanderbauwhede

School of Computing Science, University of Glasgow, Glasgow, UK

Abstract

The scaling of semiconductor technologies is leading to processors with increasing numbers of cores. The adoption of Networks-on-Chip (NoC) in manycore systems requires a shift in focus from computation to communication, as communication is fast becoming the dominant factor in processor performance. In large manycore systems, performance is predicated on the locality of communication. In this work, we investigate the performance of three NoC topologies for systems with thousands of processor cores under two types of localised traffic. We present latency and throughput results comparing fat quadtree, concentrated mesh and mesh topologies under different degrees of localisation. Our results, based on the ITRS physical data for 2023, show that the type and degree of localisation of traffic significantly affects the NoC performance, and that scale-invariant topologies perform worse than flat topologies.

Keywords: Manycore; NUMA; Network on Chip; Locality

1. Introduction

With the drive towards exascale computing and the resulting need for reduction in power consumption and optimization of performance per Watt, a growth in the number of cores per chip is to be expected. Already, the Intel Xeon Phi has 240 hardware threads. In ten years, according to the International Technology Roadmap for Semiconductors (ITRS), we can expect a thousand-core CPU to fit into an area of less than 1 mm^2 and consume only a few Watts. A further trend is 3D-stacked memory [1,2], already the Xeon Phi uses this technology and integrates it with the CPU using flip-chip. Further integration leading to memory stacked on top of the CPU is in active research.

Consequently, future manycore platforms can reasonably be expected to have such essential distributed memory architecture. Because of the large difference in access time between memory on top of a core and memory on a far removed core, a message passing style of programming similar to the approach used in NUMA architectures and HPC clusters is to be expected. In fact, it is likely that users will want to deploy legacy MPI code on these novel platforms,

* Corresponding author. Tel.: ++44-141-330-4256 ; fax: +0-000-000-0000.
E-mail address: s.al-khanjari.1@research.gla.ac.uk

because rewriting large HPC codebases is a very large effort. However, other message-passing approaches such as Erlang would be equally suitable for this type of architecture. Regardless of the programming language, it is clear that there is a need for programming models that exploit locality to avoid the long latency of communication between remote cores.

The NoC architecture for such manycore CPUs for exascale systems is of particular interest. Besides the standard mesh (as used in e.g. the Tilera manycore system and the Intel SCC) and the ring topology used in the Intel Xeon Phi and recent Intel Xeons[3], many NoC topologies have been proposed and evaluated. Some aimed to provide improvements on the ring topology, such as the Spidergon[4] or our own Quarc[5]; some aimed to improve on the mesh, e.g. the concentrated mesh[6]. There has also been some interested in scale-invariant topologies such as the quadtree[7]. Recently, locality-based traffic has been receiving increased attention[8,9]. We contend that a combination of locality-aware task placement and a locality-based communication topology can greatly improve performance of message-passing style applications.

For this particular work we have taken as a starting point the common patterns used in Computational Fluid Dynamics codes. Most of the run time of e.g. a weather simulation is spent in solving differential equations. These algorithms make extensive using of stencils for neighbour-based interaction, but they also involve whole-system reductions and this reduction step is often what determines the performance. Such reductions are most efficiently done using a tree to compute and aggregate partial results, rather than by a single process. Thus they produce a different type of localised traffic pattern.

In order to investigate the effect of the topology on the performance for similar computational patterns, we propose abstract models of locality with different distance metrics, which let us control the degree of locality as well as the shape of the local area, and thus provides general insights into the suitability of a given topology for a given locality model.

2. Non Uniform Traffic Scenarios Based on Locality of Computation

We propose a simple hierarchical model for locality-based traffic. To model locality, we group the cores of the chip and create hierarchical groups to encompass the whole system. Using $0 < l \leq n$ for the levels of the hierarchy, we can express the probability for communication across level-l as:

$$\begin{aligned} p(l) &= (1-\alpha)\alpha^{l-1}, 1 \leq l < n \\ p(n) &= \alpha^{n+1} \end{aligned} \tag{1}$$

The parameter α relates to the locality of the processes making up a message-passing based task. It expresses the probability that a message has to travel a certain distance in the hierarchy. Larger α means lower locality: if $\alpha = 0.8$, then according to equation (1) 80% of the message will have a destination outside the first cluster, and 80% of that portion outside the second cluster, etc. When $\alpha = 1$, it means that most of the traffic will be sent to the last level cluster. This is worse than a uniform traffic where all destinations have equal probability.

In this paper, we use two different instances of this locality-based model to evaluate the performance of the NoC topologies.

- In the first model (Group Clustering), we group the cores of the chip per four, and create hierarchical groups to encompass the whole system, in a scale-invariant fashion. In this case $n = log_4(N)$ with N the number of cores, and each level contains 4^n cores. This is a generalisation of reduction traffic.
- In the second model (Ring Clustering), we group the cores of the chip in concentric rings around the sender core. In this case the number of cores per level is $8n$ as long as the rings don't meet the edge. This is a generalisation of nearest-neighbour (stencil) traffic.

3. Network Topologies

The network topology describes how routers are connected with each other and with the cores. For manycore systems, the communication cost is increasingly important. The topology has a major impact on the scalability and

Table 1. Table of notations

Symbol	Description	Symbol	Description
N	number of cores	N_L	number of links
n_B	number of buffers	N_R	number of routers
n_{VC}	number of virtual channels	N_B	number of buffers

Table 2. Cost Model for 1024 cores

	Mesh		Fat Quadtree		Cmesh	
N_L	$2\sqrt{N}(\sqrt{N}-1).n_{VC}$	4094	$N.log_4(N)$	5120	$\sqrt{N}(\frac{\sqrt{N}}{2}-1).n_{VC}$	960
N_R	N	1024	$\frac{N-1}{3}$	341	$\frac{N}{4}$	256
N_B	$5n_B n_{VC} N$	163840	$n_B(N-4)(\sqrt{N}-2)+2n_B\sqrt{N}$	490624	$2n_B n_{VC} N$	65536

the performance of the network. With this motivation, we analyse the network performance and communication locality in flat and hierarchical topologies. In this section we describe the three different types of network topologies namely mesh, concentrated mesh and fat quadtree.

Mesh. The mesh topology has been the most popular NoC topology so far and it has been used in most of the recent manycore chips such as Intel SCC 48-core[10], TFlops 80-core[11], Tilera 64-core[12]. It organises the routers in a grid, one router per core. Addresses of routers and cores can be easily defined as x-y coordinates in mesh. The mesh has a radix (number of ports) of 5. Deadlock is avoided by using a deadlock free routing algorithm, e.g. XY routing.

Concentrated Mesh. The concentrated mesh (cmesh) has been introduced by[6]to preserve the advantages of a mesh with decreased diameter. The number of cores sharing a router is called the concentration degree of the network (4 in this work). The cmesh topology requires fewer routers resulting in reduced hop count and consequently improved latency. It has a radix of 8. Routing is the same as in the normal mesh.

Fat Quadtree. The fat tree connects routers in a tree with the cores at the leaves. To avoid congestion towards the root of the tree, a fat tree use an increasing number of links as described in[7]. A fat quadtree of size N is a structure that can be regarded as a rooted 4-ary tree of height $log_4(N)$. In this way it exactly reflects the group clustering model. The advantage of the fat quadtree over the mesh is that the communication diameter of a fat quadtree is only $O(log_4 N)$ compared to $O(\sqrt{N})$ for the mesh. The fat quadtree uses nearest-common ancestor routing. Packets are adaptively routed up to the common ancestor and deterministically down to the destination. The fat quadtree is deadlock-free.

4. NoC Area Overhead

4.1. Cost Model

We present the cost model of mesh, cmesh and fat quadtree in terms of link complexity, number of routers and buffers. Table 1 shows the notations used for the cost model.

Link Complexity is the total number of links in the topology. Note that mesh and cmesh has two virtual channels while fat quadtree has no virtual channels. In section 4.2, we will compute the wire overhead for mesh, cmesh and fat quadtree. Total number of buffers in mesh and cmesh are straight forward as in each router there are 5 and 8 ports, respectively. The total number of buffers in fat quadtree is more complex since the buffer size is doubling at every level because the wire lengths are doubling at every level. Table 2 shows the cost model and the values for 1024 cores.

To calculate the wire link overhead, we get the links width $Link_{width}$ as in equation 2, where (W) is the wire pitch, (N_{bits}) is number of bits in parallel for one packet and $N_{layers}$is number of layers.

$$Link_{width} = \frac{W \times N_{bits}}{N_{layers}} \tag{2}$$

Table 3. Technology parameters for 10nm CMOS (2023) based on ITRS 2011

Year	2013	2023
Chip size at production	140 mm^2	111 mm^2
Global wire delay	1 ns/mm	33.8 ns/mm
Estimated core size	6.8 mm^2	0.3 mm^2
Estimated wire delay	2.6 ns	17.5 ns

The number of vertical wires in a fat quadtree can be obtained $Vertical_{wire} = 2^{(log_4 N-1)} log_4 N$, and for the mesh $Vertical_{wire} = \sqrt{N}$ where N is the number of cores. Starting from Eq. 2 and the number of vertical wires, we can compute the area overheads as follows:

$$Area_{Wire} = 2 \times Width_{VerticalLink} \times Width_{Chip} \tag{3}$$

$$Area_{Chip} = Area_{Core} \times N + Area_{Wire} \tag{4}$$

$$AreaOverhead = \frac{Area_{Wire}}{Area_{Chip}} \tag{5}$$

4.2. Technology Node Assumptions

To ensure realistic simulations of the next decade's manycore systems, we assume the 10 nm process in 2023 as projected by the International Technology Roadmap for Semiconductors. Table 3 lists the physical parameters for this technology node from the 2011 ITRS data. We used the die size of the 64-core Tile64 from[13] (433.5 mm^2) to estimate the core size and scaled it to the 2023 node.

4.3. Overheads for a 1024-core chip, 10nm (2023) node

In terms of buffer space overhead, the mesh will require 5.1 KB of storage per core, the cmesh 2 KB and the fat quadtree 15.3 KB. This is only a very small fraction of the total size of the chip: e.g. just the per-core L2 cache on the 60-core Xeon Phi is already 512KB. With the above assumptions, the area of a 15.3KB SRAM buffer would be 0.14% of the estimated core size (memory density 37.6 MB/mm^2). In terms of wire overhead, our cost model shows that the wire area overhead for the fat quadtree would be 0.3% of the estimated chip size for a 1024-core chip (wire pitch 17 nm). These results are very important as they indicate that for this type of manycore architecture, the NoC overhead is negligible, which means that the choice of the NoC can be based solely on performance.

5. Evaluation and Discussion

We simulated the different topologies on 1024-core chip (placed in a regular 32 × 32 grid) using HNOCS (Heterogeneous Network-on-Chip Simulator). The original HNOCS uses a mesh topology with wormhole switching with virtual channels and XY routing. We extended HNOCS with the cmesh and fat quadtree topologies and their routing algorithms, as well as our locality-based traffic distributions.

We model the process-to-process communication using Poisson-distributed traffic. Two virtual channels are used for the mesh and cmesh while for fat quadtree one physical channel is used for the lowest-level links and it quadruples at each level to simulate a fat quadtree. Hence, the fat quadtree has no virtual channels. The wire delay is proportional to the distance between the routers so in a fat quadtree it doubles at each level. The destination was selected using different degrees of localisation. Table 4 summarises the simulation parameters used in our simulations.

We evaluated the performance of the three topologies as a function of the transmission rate and the localisation degree α, where $\alpha = 1$ means no locality, and $\alpha = 0$ is fully local traffic. Figures 1 and 2 show the results of our experiments in terms of latency and throughput. The results show that for group clustering the cmesh performs best. One might expect the fat quadtree to perform better as the group clustering matches its topology. However, the layout of the quad tree results in fewer hops but longer paths, and in the 10nm CMOS process the wire delay is dominant

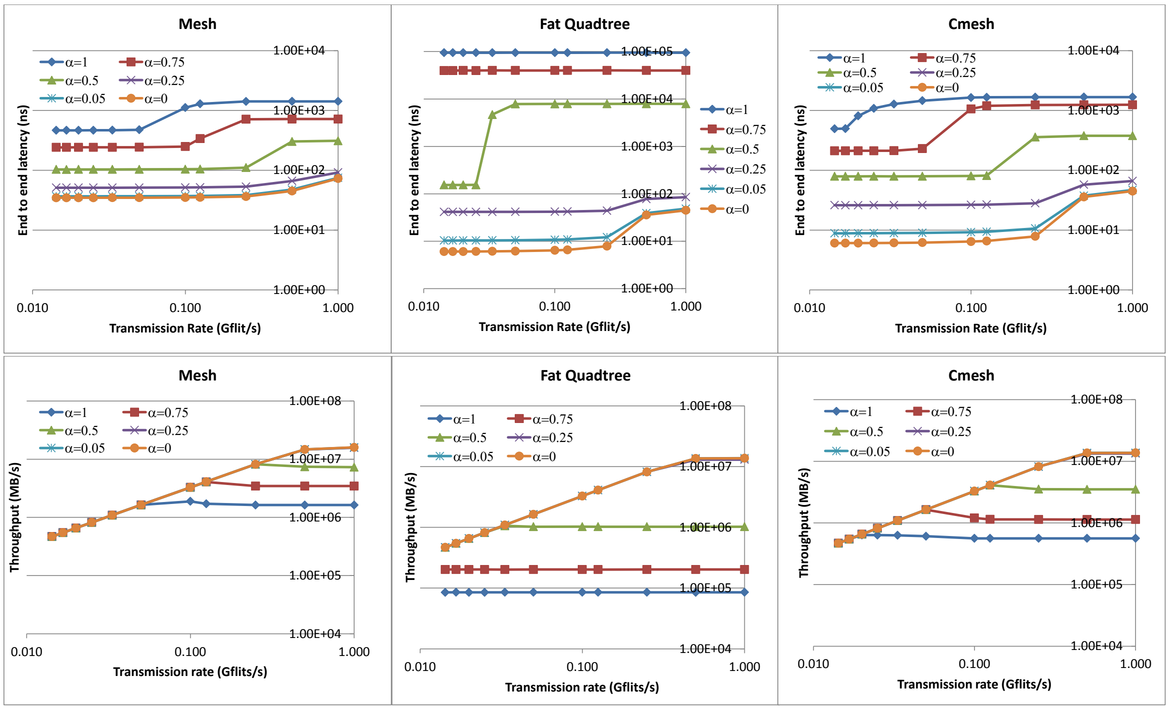

Fig. 1. Group clustering results (α = 1: no locality, α = 0: total locality)

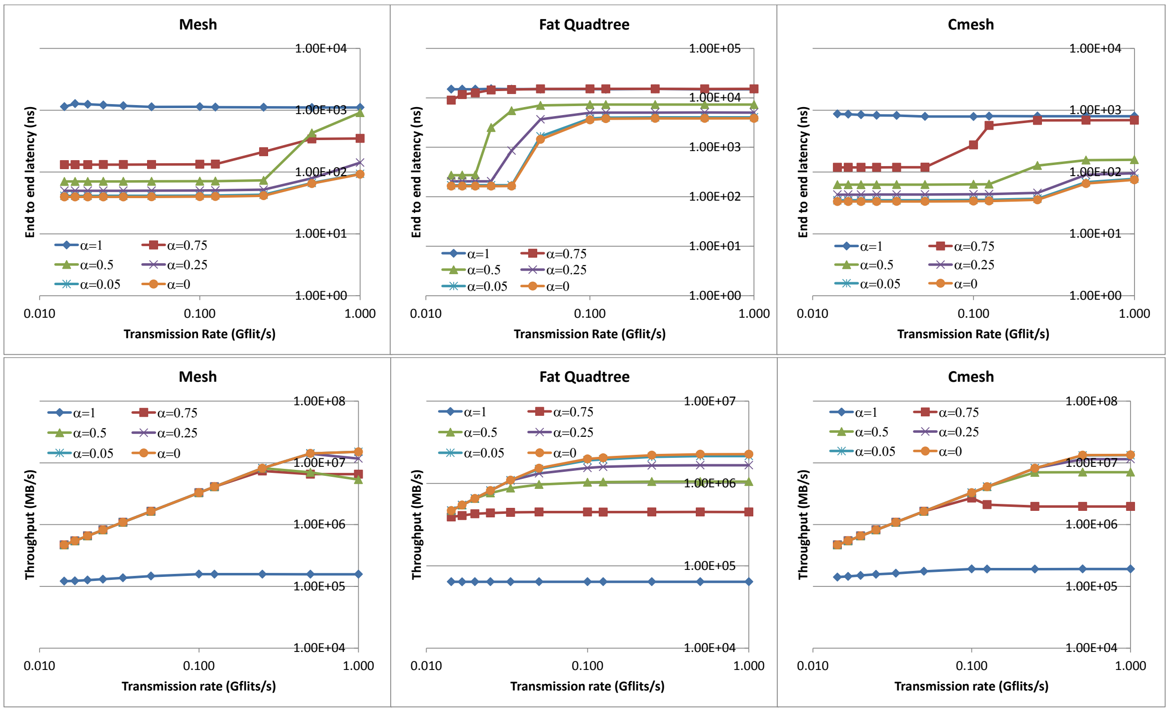

Fig. 2. Ring clustering results (α = 1: no locality, α = 0: total locality)

Table 4. Simulation parameters

Number of virtual channels	Mesh/Cmesh: 2	Fat quadtree: 1
Flit size (bytes)	32	
Buffer size (flits/VC)	16	
Wire delay (ns)	Mesh/Cmesh: 17.5/35	Fat quadtree:$35 \times 2^{l-1}$, $1 \leq l < n$
Channel datarate (Gb/s)	128	

(30× worse than the 2013 node). With ring clustering, mesh and cmesh also perform better than the fat quadtree. Overall, group clustering results in lower latencies than ring clustering; this is an important result for the placement of neighbours in stencil computations.

6. Conclusions

We have investigated the overhead and performance of flat (mesh, cmesh) and scale-invariant (fat quadtree) NoC topologies for future manycore systems with thousands of cores under group clustering and ring clustering localisation models. We show that the overhead of the NoC on a thousand-core system in 10nm CMOS is negligible for all three topologies. We show that the degree of locality and the clustering model strongly affects the performance of the network. Scale-invariant topologies such as the fat quadtree perform worse than flat ones (esp. cmesh) because the reduced hop count is outweighed by the longer path delays, as a consequence of the high wire delay in the 10 *nm* CMOS process. Our results clearly show the importance of traffic localisation for very large manycore systems.

References

1. G. H. Loh, 3d-stacked memory architectures for multi-core processors, in: ACM SIGARCH Computer Architecture News, Vol. 36, IEEE Computer Society, 2008, pp. 453–464.
2. S. K. Lim, 3d-maps: 3d massively parallel processor with stacked memory, in: Design for High Performance, Low Power, and Reliable 3D Integrated Circuits, Springer, 2013, pp. 537–560.
3. A. Duran, M. Klemm, The intel® many integrated core architecture, in: High Performance Computing and Simulation (HPCS), 2012 International Conference on, IEEE, 2012, pp. 365–366.
4. M. Moadeli, A. Shahrabi, W. Vanderbauwhede, M. Ould-Khaoua, An analytical performance model for the Spidergon NoC, 21st IEEE International Conference on Advanced Information Networking and Applications (2007) 1014–1021.
5. M. Moadeli, P. Maji, W. Vanderbauwhede, Quarc: A high-efficiency network on-chip architecture, in: Advanced Information Networking and Applications, 2009. AINA'09. International Conference on, IEEE, pp. 98–105.
6. J. Balfour, W. J. Dally, Design tradeoffs for tiled cmp on-chip networks, in: Proceedings of the 20th annual international conference on Supercomputing, ACM, 2006, pp. 187–198.
7. C. E. Leiserson, Fat-trees: universal networks for hardware-efficient supercomputing, Computers, IEEE Transactions on 100 (10) (1985) 892–901.
8. R. Das, S. Eachempati, A. K. Mishra, V. Narayanan, C. R. Das, Design and evaluation of a hierarchical on-chip interconnect for next-generation cmps, in: High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on, IEEE, 2009, pp. 175–186.
9. P. P. Pande, C. Grecu, M. Jones, A. Ivanov, R. Saleh, Effect of traffic localization on energy dissipation in noc-based interconnect, in: Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on, IEEE, 2005, pp. 1774–1777.
10. J. Howard, S. Dighe, S. R. Vangal, G. Ruhl, N. Borkar, S. Jain, V. Erraguntla, M. Konow, M. Riepen, M. Gries, et al., A 48-core ia-32 processor in 45 nm cmos using on-die message-passing and dvfs for performance and power scaling, Solid-State Circuits, IEEE Journal of 46 (1) (2011) 173–183.
11. S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, T. Jacob, et al., An 80-tile 1.28 tflops network-on-chip in 65nm cmos, in: Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International, IEEE, 2007, pp. 98–589.
12. S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. MacKay, M. Reif, L. Bao, J. Brown, et al., Tile64-processor: A 64-core soc with mesh interconnect, in: Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International, IEEE, 2008, pp. 88–598.
13. C. Killebrew, et al., L2 cache to off-chip memory networks for chip multiprocessor, Ph.D. thesis, Department of Electrical Engineering and Computer Sciences, University of California (2008).