Architectural study of the opportunities for reconfigurable optical interconnects in distributed shared memory systems.

C. Debaes[1], I. Artundo[1], W. Heirmann[2], J. Dambre[2], Khoi Bui Viet[1] and H. Thienpont[1]

1 Department of Applied Physics and Photonics,
Vrije Universiteit Brussel, Pleinlaan 2, 1050 Etterbeek, Belgium
2 Universiteit Ghent, 9000 gent, Belgium

An intriguing aspect of optical interconnects from an architectural point of view is their ability to reconfigure the topology in a data-transparent way. We focus in this work on the potentialities of such dynamically reconfigurable interconnects in particularly for parallel shared-memory machines and we identify the time-scales at which this inter-processor network should be able to reconfigure to result in a performance gain. By performing full-system simulations of the shared memory architectures with parallelized benchmark applications, we show the existence of a considerable amount of bursts which last up to tens of milliseconds, indicating a possible match with current technology for reconfigurable optical interconnects.

Introduction

The problems we solve on computers tend to become exponentially bigger as years are passing by. Since the processing power of one processor is and will always be limited, computer scientists have since long been studying the possibilities to use more than one processing element in parallel. The efficiency at which such a computer system operates depends on its running application. Some programs can be well-partitioned in concurrent sub-talks that do not have a large need for communication, leading to a limited or straightforward demand of the intercommunication network. This class of programs is mostly programmed using the message passing paradigma, where the programmer is orchestrating the data-flow within the machine by sending explicit messages between the different processing elements.

Other programs, on the contrary, show a strong coupling between the different components. They are preferable run on a system with *distributed shared memory (DSM)* machines since this approach strongly simplifies the required programming. In this class of machines, the memory is physically distributed among the different processor nodes but is logically accessible as one global memory range. By far the commercially most valuable type of DSM machines is the cache-coherent Non Uniform Memory Access (ccNUMA) machine. As memory is distributed over different nodes, these machine have non-uniform memory access times. The local copies of the caches are kept coherent with cache control messages and data messages that are flowing over the network. Hence, the interprocessor communication network is a part of the memory hierarchy. This configuration therefore makes the performance and the latency of such a network of the uttermost importance: when one of the processor experiences a read-miss, it will have to stall until the requested data will find its way over the network.

The required bandwidth in DSM systems grows proportionally with the number of processors, with bandwidths between individual node pairs of up to tens of Gb/s. Access times in large shared memory machines currently range from 100 ns to and a few microseconds, a number that has remained fairly constant for some years. Since the clock speeds of CPUs continue to increase exponentially, this result in a growing gap between instruction speed and memory access times. *Most of this latency is spent by messages traversing the network. Hence, there is a large margin for improvement by addressing the performance limitations of current memory networks.*

Optical Networks

To overcome some of the limitations of conventional interconnection technologies within the DSM machines, optically interconnected electronics have been proposed as a solution. Indeed, a whole range of physical reasons can be identified why optics as a wire replacement technology could be beneficial as opposed to conventional interconnects. The high carrier frequency, for example, will lead to links that have negligible frequency dependent loss or cross-talk. Other advantages for optical interconnects include: very dense integration of photonic pin I/Os, simple impedance matching and the ability for wavelength division multiplexing (WDM), immunity for electromagnetic interferences and ground loops and many others [1].

However, the most intriguing aspect, from an architectural point of view is that the optical communication paths can be easily combined, splitted, routed and reconfigured by suitable optical devices and optical components. This paves the way for reconfigurable optical interconnects and, hence, an intercommunication network that can change its topology over time.

A first objective of this research is to investigate whether reconfigurable optical interconnects can improve the performance of distributed shared memory (DSM) multiprocessors. Since no reconfigurable interconnect technology – be it optical or electrical - will enable to switch a topology instantaneously, it is clear that this performance benefit will depend highly on the time-scale at which traffic patterns on the multiprocessor are changing as no traffic can be transmitted over a network while it is being reconfigured.

Simulation environment and architectural parameters

To identify the time-scales of the reconfigurable networks, one has to measure the behavior of the internode communication. In a first attempt, one could emulate the traffic that is passing over the network by statistically generated packets. Although this method can give an idea of the capacity of a network, it has been shown that one can only gain a limited amount of information on the performance of real programs with these simple statistical simulations. It will therefore be necessary to generate the traffic of the real-life programs by full system simulators that simulate the execution of the complete program instruction by instruction. Although there exists already a couple of full system simulators, such as RSIM[2] and SimOS[3], we have adopted the commercial available SimICs[4] simulator. We are targeting a system closely resembling a SunFire 6800 server with the Sun Fireplane interconnect and sixteen 1.2Ghz ultraSparc III processors. Where appropriately, we have kept distance from the real hardware disposition to make our study more general. There is however one aspect of the cache protocol which we followed closely: the use multiple *snoopy cache coherence domains*.

There exist currently two major approaches to deal with the cache coherence problem. The first and most simple protocol, the *snoopy protocol*, relies on every message being broadcasted to all nodes[1]. This is usually implemented using a common bus, onto which each request is transmitted. All other nodes are constantly listening on the bus to check for relevant transactions. This protocol consumes a large amount of traffic and is therefore not scalable to a high number of processor counts. The sunFire server can support only a maximum of 24 processors with this snoopy protocol. At larger processor counts, the system is divided into so-called 'snoopy coherence domains'. If data is requested outside the snoopy coherence domain, a higher level of the interconnection network is supplying the request via a more complex *directory based protocol*. It is that traffic on this higher level aggregated interconnection network which we model in this work. Such a segmented topology is in our belief an accurate approach of future DSM machines. Indeed, the current trends to multiple processor cores on single chips will almost certainly lead to islands with tightly coupled processors connected with a higher-level interconnect network. To keep the simulation time within bounds, we have limited the number of processor to a maximum of 16 and varied the size of the cache coherence domain between 2, 4 or 8 processors.

Experimental methodology and the distribution of traffic bursts

In this work, we have measured the traffic burst distributions of different SLPASH benchmark applications (the Radix, Barnes and Cholesky application) and the apache web-server. We have therefore measured how much traffic was flowing between pairs of different cache coherence domains and typically analyzed the distribution of bursts. We consider a *burst* as the traffic on a link for which the communication demand (per 100'000 cycles or 670µs) is larger than the total averaged communication bandwidth. In figure 1, we give an example of such a distribution of bursts while running the apache server loaded by the SURGE benchmark. Besides a large number of short bursts, one can find a considerable fraction of longer bursts with durations of 100ms or more.

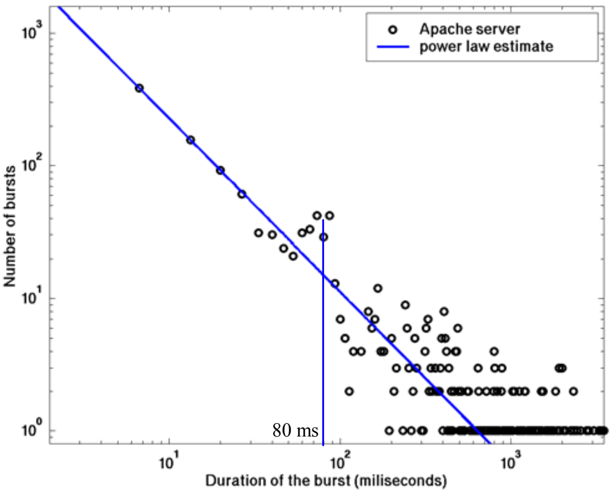

Figure 1 : Distribution of traffic burst of the apache web server.

In table 1 we summarize the results of the simulation. For each architecture with 2, 4 and 8 processors per domain we show the simulated time, the total number of transferred packages, the total average bandwidth and the maximal duration of a burst.

In all these benchmarks, we can clearly see the existence of a significant fraction of long bursts (up to several tenths milliseconds in length). However, for the reconfiguration to be beneficial the bursts should be detected and the network should be

[1] Only broadcasting all packets is not yet a necessary condition. The packets should also be guarantueed to come to their destinations in an ordered manner.

reconfigured in a timescale which is at least an order of magnitude smaller then the burst length. For the apache benchmark in particular, several UNIX processes were running simultaneously requiring the operating system to switch between them at regular intervals. As a result, we can see in figure 1 a peak of the number of bursts with a length of 80ms, the interval at which the operating systems handling a context switch between different processes.

Table 1: the resulting traffic demands of the simulated applications

	2 processors/domain			4 processors/domain				8 process/domain		
	Radix (2^{20},r4096)	Cholesky (Tk29)	Barnes	Radix (2^{20},r4096)	Cholesky (Tk29)	Barnes	Apache Web server	Radix (2^{20},r4096)	Cholesky (Tk29)	Barnes
Simulation time (s)	1.24	17.80	48.6	1.2	22.7	54	147	1.18	12.30	40.90
Total number of messages (millions)	76.1	991.4	1839	74.6	670	1315	1405	27.1	151.8	392.5
Average Total BW (Mpbs)	35.8	44.3	24.6	121	123	48.4	31.9	327	214	99.0
Maximum duration of a burst (seconds)	0.54	12	9.6	0.35	10.1	8.80	22.9	0.17	6.86	3.65
Average duration of traffic bursts (ms)	10.7	41.1	7.4	14.2	57.1	13.3	420	7.26	76.2	4.3

Conclusions

Using detailed simulation to obtain accurate traces of network traffic and subsequent analysis, we have shown that communication bursts occur between pairs of nodes and that the duration of these bursts can be several tens of milliseconds. These results lead to required reconfiguration times close to a millisecond, indicating a possible match with projected optical technologies

Acknowledgements

Christof Debaes is indebted to the FWO-Vlaanderen for his postdoctoral research fellowship. This work was funded by the IUAP Photon Network, FWO, GBOU, GOA and the OZR-VUB.

References

[1] D. A. B. Miller, “Rationale and Challenges for Optical Interconnects to Electronic Chips”, Proceedings of the IEEE, vol. 88, no. 6, pp. 728 749, June 2000.

[2] C.J. Hughes, V.S. V.S. Pai, P. Ranganathan, and S.V. Adve, "RSIM: simulating shared-memory multiprocessors with ILP processors," IEEE Computer magazine, vol. 35(2), pp. 40 - 49, 2002.A.B. Smith and D.E. Jones, "Title of paper", Journal name, vol. 10, pp. 1832-1838, 1992.

[3] M Rosenblum, E. Bugnion, S. Devine, and S. A. Herrod, "Using the SimOS machine simulator to study complex computer systems," ACM Transactions on Modeling and Computer Simulation (TOMACS), vol. 7(1), pp. 78 - 103, 1997.

[4] P.S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, and B. Werner, "Simics: A full system simulation platform," IEEE Computer magazine, vol. 35(2), pp. 50 - 58, 2002.