

Reconfigurable Optical Interconnection Networks for Shared-Memory Multiprocessor Architectures

Wim Heirman

ELIS Department, Ghent University, Belgium

While parallel processing used to be limited to supercomputing or large web and database servers, the era of the multicore processor is now bringing parallel computing to desktop PCs, laptops and game consoles. Putting all these processors to good use on solving a single problem, requires an interconnection network that supports communication at high bandwidths and low latencies.

Current technologies, using electrical signaling, are reaching the end of their capabilities. Moreover, the traffic on such networks is very irregular, making some parts of the network highly saturated while other parts are barely used. The precise patterns also depend on the application, and can even change during the run time of a single application. This makes it very difficult to build an efficient communication network.

This doctoral dissertation explores the possibilities of optical, reconfigurable networks. Optical connections are one part of the solution to the communication problem, since they allow for much higher data rates. They are currently being investigated by academic and industrial labs around the world, including big players such as Intel and IBM [1]. It is expected that optically communicating multiprocessor systems will be on the market in just a few years. Multicore processor chips, with on-chip optical networks, may follow in the next five to ten years. Additionally, novel optical components such as tunable lasers or micro-ring resonators allow the network to be reconfigured at runtime to match the current traffic pattern [2]. This way, network connections are utilized more efficiently, which both increases the network's performance and reduces power usage.

Traffic patterns: The speed of reconfiguration that can be obtained using cheap optical components is relatively low. The network can therefore not be reconfigured to provide minimal latency for each memory access. We therefore looked for locality in the communication, and found that network traffic often remains very similar for longer periods of time. This fact can be exploited by reconfiguration: by changing the distribution of available bandwidth in the network, such that the most voluminous traffic streams can use the fastest connections, latency can be brought down significantly. All this can be done in an automatic, application-invisible way, staying true to the idea of the shared-memory abstraction which isolates the programmer from the implementation details of the underlying machine.

A reconfigurable network architecture: To achieve this we designed the following reconfigurable network architecture: a

base network with fixed topology is augmented with reconfigurable extra links (elinks). The base network has a regular topology such as a mesh, torus or hypercube, and connects all processors. The reconfigurable elinks are placed such that they provide a direct, high-bandwidth connection between those processor pairs that communicate with the highest intensity. The locations of these elinks are changed after every reconfiguration interval, which usually has a length in the order of milliseconds. This way, most of the high-volume burst traffic is carried by the elinks, while the base network is still available for background traffic, or during reconfiguration of the elinks. This architecture resembles existing work such as [3], but differs in the sense that reconfiguration is now automatic and driven by measured traffic patterns, rather than by the programmer or compiler.

Speeding up design-space explorations: Reconfiguration adds even more parameters to an already huge network design space. Moreover, performance is now much more dependent on the temporal behavior of the network traffic. Existing network design techniques do not account for this temporal behavior. We therefore extended these methods, and proposed new methods that do allow a quick but sufficiently accurate evaluation of reconfigurable network architectures. Our methods are also usable on non-reconfiguring designs, were they can improve higher accuracy than older methods that do not take temporal traffic behavior into account.

Performance evaluation: Finally, we explored the performance of our proposed reconfigurable network architecture, under a wide range of workloads and architectural parameters. We combined large-scale explorations, using our faster methods described before, with detailed studies using highly accurate execution-driven simulations. Overall, we found that reconfigurable networks could improve network performance significantly. Depending on the traffic pattern of the application, latency could be lowered by up to 40% on 64-processor networks. Since reconfiguration can already be present in the system, for instance for reliability reasons, this performance improvement can be obtained at a very low added cost.

REFERENCES

- [1] L. Schares et al. Terabus: Terabit/second-class card-level optical interconnect technologies. *J. STQE*, 12(5):1032–1044, Sept. 2006.
- [2] I. Artundo et al. Selective optical broadcast component for reconfigurable multiprocessor interconnects. *IEEE J. STQE*, 12(4):828–837, July 2006.
- [3] K. Barker et al. On the feasibility of optical circuit switching for high performance computing systems. In *SC '05*, p. 16, Nov. 2005.