Performance and Energy Comparison of Electrical and Hybrid Photonic Networks for CMPs

Shoaib Kamil[1,2], Ankit Jain[1], Marghoob Mohiyuddin[1], John Shalf[2], John Kubiatowicz[1]
[1] EECS Department, University of California at Berkeley, CA
[2] CRD/NERSC, Lawrence Berkeley National Laboratory, Berkeley CA
{skamil | ankit | marghoob | kubitron}@cs.berkeley.edu
jshalf@lbl.gov

Introduction

Due to the slowing ability of chip designers to scale processors to faster speeds, future trends point towards chip multiprocessors (CMPs) in order to utilize the available transistors in a performance and power-efficient manner. In this work, we explore two possible directions for future on-chip networks: one based entirely on electrical routers, and another using a hybrid approach, combining a limited electrical network with an on-chip photonic network made possible by recent advances in 3DI CMOS technology [2]. By stacking memory and interconnect resources on CMOS layers above the processors, it is possible to integrate larger memories and faster interconnects with future CMPs. In this work, we make the following novel contributions: unlike previous examinations of photonic NoCs, we use both synthetic traces and actual application traces from SPMD-style scientific codes; we build cycle accurate simulators for the two architectures and construct simple analytic models that accurately predict energy and performance without the slow runtime of a simulator; and we show the importance of good process-to-processor mappings to obtain optimal interconnection performance. Results indicate photonic NoCs have exciting potential.

Architecture

This work explores NoCs for a 64-processor CMP, with processors arranged in a 2D planar fashion in a future 22nm process. The processors are assumed to be simple in-order 5 GHz cores with local memory stacked on higher layers of the 3D CMOS die; in the case of an optical interconnect, an additional layer of photonic elements and pathways is above the memory layers.

The fully-electrical NoC architecture is due to Dally [1] and consists of wormhole-routed 8x8 switches with virtual channels arranged in a CMesh topology, shown in Figure 1. Link latency for neighbors is a single cycle, while the express channel latency is two cycles; like [1], we assume the inter-router latency is two cycles. Area and cycle-time restrictions limit the link bandwidth to 128Kbit/s.

The hybrid photonic network is arranged in a mesh topology. Each 4x4 blocking photonic switch (and corresponding electrical control switch) can route a single path through the router and is made up of four *Photonic Switch Elements*, which consume no power when inactive and only 0.5mW when switched on. In addition to switching cost, a message transmission consumes energy in converting messages from electric to optical and back. A path must be allocated and torn down through the electronic control network, but once allocated, a path can route a message at an end-to-end rate of 192Kbit/s with no intermediate routing latency. Details of the photonic network are in [2].

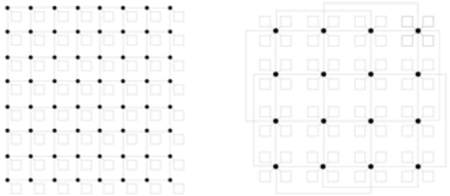

Figure 1: Mesh and CMesh Topologies.

Modeling & Simulation Methodology

Before building cycle-accurate simulators for the two architectures, it is prudent to construct simple energy and performance models that can help determine the viability of the networks.

For the electrical network, we assume latency is hidden through the use of virtual channels, and so use a bandwidth-only model (i.e., T_{msg} = $size_{msg}$ / BW). The overall time is estimated by routing each message and determining the most-used link; routing the total volume over this link is the bottleneck time. For energy, we assume messages incur energy usage at each link to cross the router and then the link. The parameters for this come from [1], scaled to 22nm using the ITRS Roadmap.

The hybrid network uses a similar model, but there are two networks to account for. The time for each message must include the latency to set up a path and tear it down, so

$$T_{msg} = Links \times Latency_{electrical} + \frac{size_{msg}}{BW_{photonic}}$$

Overall time is calculated by routing all messages and finding the most-used link; since messages must be serialized due to the blocking nature of the switches, the overall time is equal to routing all the messages through the bottleneck link.

Energy calculation requires accounting for the energy across the electrical network, the energy to switch a PSE on, the energy for the EOE conversions, and the active PSE energy for the duration of the message. Thus, energy is

$$E_{total} = \sum \begin{matrix}(Links \times E_{electrical} + E_{PSESwitching} \\ + T_{msg} \times E_{PSEActive} + E_{EOE} \times size_{msg})\end{matrix}$$

In addition to these models, we constructed cycle-accurate simulators using energy and cycle-time estimates from previous work [1,2,3], along with ITRS scaling factors. Using a custom MPI tracing layer, we obtained actual communication traces from three scientific applications: Cactus, an astrophysics application using a stencil operator; GTC, a particle-in-cell fusion reactor simulator; and MADbench, a cosmic microwave background analysis tool with heavy use of dense linear algebra. Along with a suite of standard synthetic traces, we model and simulate performance of these applications to compare the two NoC strategies. Our simulation and modeling methodologies ignore computation, using only phase information from the trace to determine message ordering.

Synthetic Communication Results

Our suite of synthetic communication patterns consists of random, bitreverse, neighbor, and tornado patterns. Using the models, the suite was tested for both small and large message sizes, as seen in Figure 2. The hybrid network always outperforms the electrical network in terms of energy, but it requires larger messages to overcome the added latency of the setup and teardown processes.

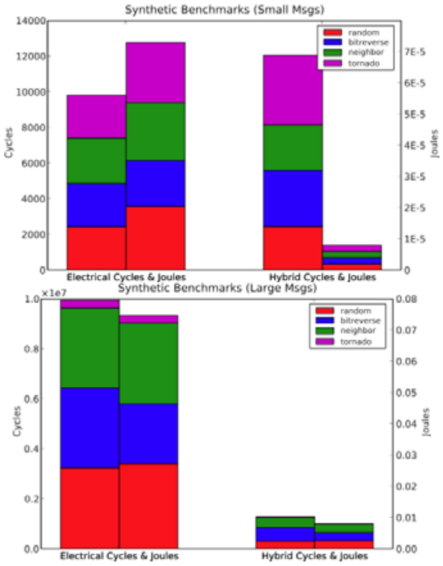

Figure 2: Synthetic Benchmark Results.

Application Results

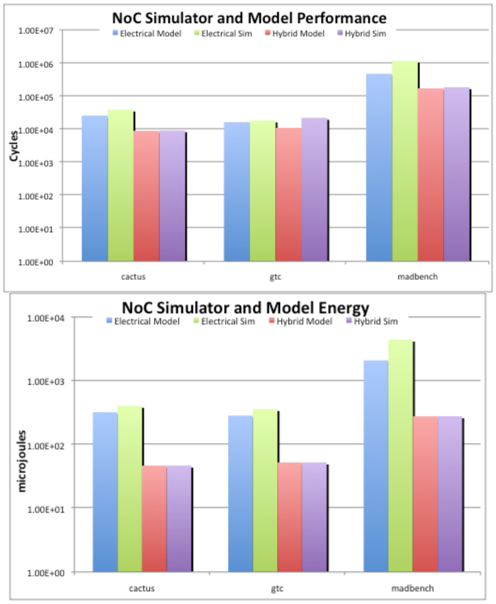

Figure 3: Application Simulation and Modeling Results.

For the three applications, we used both simulation and modeling to compare the two networks. Figure 3 shows the results. Note that the models are quite accurate for both energy and time. In addition, the hybrid network again outperforms the electrical network in terms of energy, but requires larger messages as in Cactus and MADbench to amortize the latency costs. For these SPMD-style applications, the hybrid network shows the potential to outperform an electrical NoC by an order of magnitude. Figure 4 shows the effect of process-to-processor mapping on the two NoCs: it is far more important for the hybrid network to find a good process-to-processor mapping, because otherwise link contention results in much slower performance and much higher energy cost due to failed path-setup messages on the electrical control network.

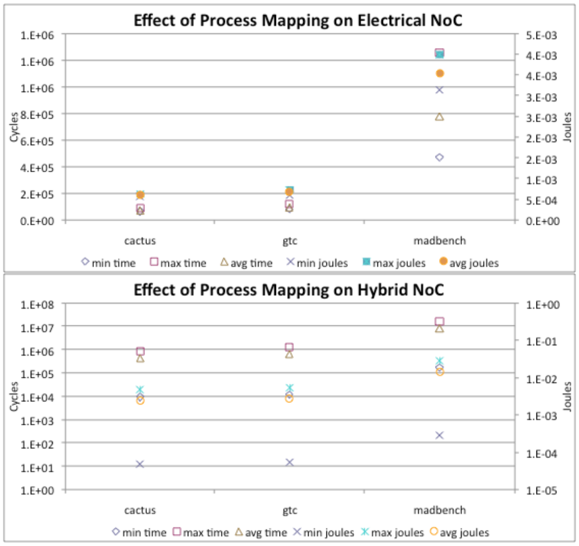

Figure 4: Effect of Process Mapping on the NoCs.

Conclusion

For future high-performance embedded CMPs, obtaining performance and power efficiency will require an appropriate network. This work shows that a hybrid photonic network can potentially yield orders-of-magnitude network energy savings and, with reasonable message sizes, large performance gains. Using a simple model, we accurately predict simulated interconnect performance, which allows network designers to explore large parameter spaces without resorting to costly cycle-accurate simulation. We show the need for a good process-to-processor mapping to obtain optimal performance. Future work will extend these methodologies to non-blocking photonic networks, which provide crossbar-like connectivity. In addition, a full system simulator with simulated cores is also planned.

References

[1] J. Balfour and W. Dally, "Design Tradeoffs for Tiled CMP On-Chip Networks," *Proceedings of the International Conference on Supercomputing*, 2006.

[2] A. Shacham, K. Bergman, L. Carloni, "On the Design of a Photonic Network-On-Chip," *Proceedings of the First International Symposium on Networks-on-Chip*, 2007.

[3] M. Erez, *Merrimac—High Performance, Highly Efficient Scientific Computing with Streams*," Ph.D. dissertation, Stanford University, 2006.