Network-Attached Accelerators: Host-independent Accelerators for Future HPC Systems

Sarah Neuwirth, Dirk Frey, and Ulrich Brüning
Institute of Computer Engineering (ZITI) – University of Heidelberg, Germany; {sarah.neuwirth,dirk.frey,ulrich.bruening}@ziti.uni-heidelberg.de

Introduction

Current Heterogeneous Systems

- Clusters with accelerators
- Accelerators are host-centric
- No integrated network interconnect
- Static assignment (1 CPU : N accelerators)
- PCIe can become a bottleneck
- Explicit programming required

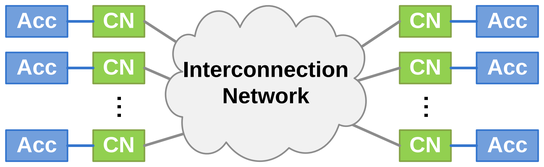


Fig. 1: Heterogeneous system.

Research Objective

Divide Architecture in Two Parts

- Cluster based on multi-core-chips
  - Executes scalar code
- Booster based on many-core technology
  - Runs highly scalable code
  - EXTOLL: Switchless direct 3D torus [1]

=> Components-off-the-shelf philosophy

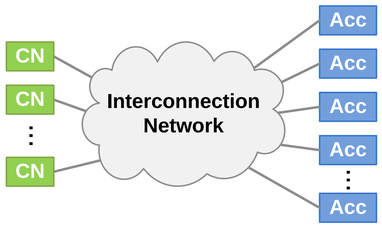


Fig. 2: Booster-based architecture.

Goals of the Architecture

- Accelerator directly connected to network
- Static and dynamic workload assignments
- Scale the number of accelerators and host CPUs independently in an N to M ratio

Network-Attached Accelerators

Communication Model

- Simplified, transparent user view
- Direct accelerator-to-accelerator communication
- Dynamic workload distribution
- Accelerators accessible from any node without host interaction

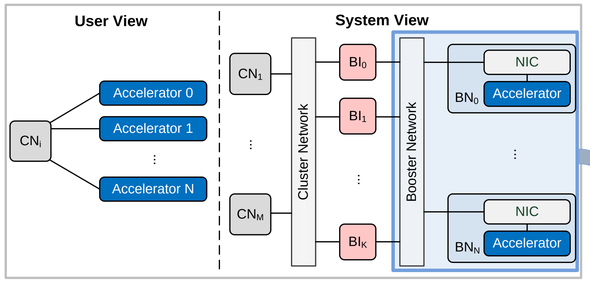


Fig. 3: Communication model and 3D torus topology [2].

Software Design

- Transparent to implementation
- No accelerator driver changes
- Responsible for device & MSI configuration and IRQ handling

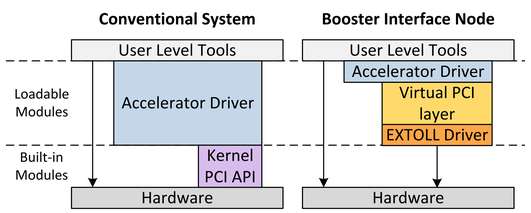


Fig. 5: Comparison of the software stacks.

Accelerator Access

- Distributed shared memory to communicate
- MMIO range can be anywhere in the network
- Loads and stores are encapsulated into network transactions

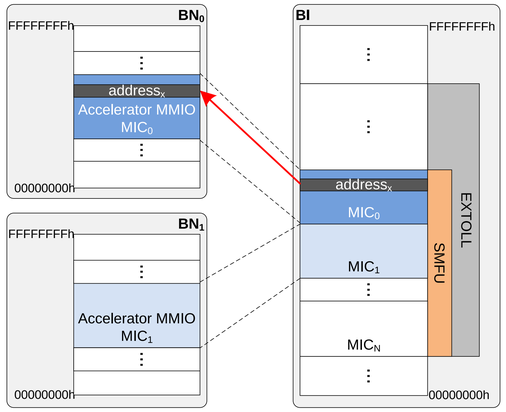


Fig. 4: Memory mapping between BNs and BI.

Prototype Implementation

- High-density booster node card (BNC) with two Intel Xeon Phis (61 cores) & NICs

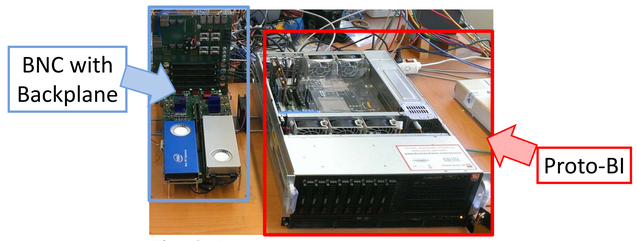
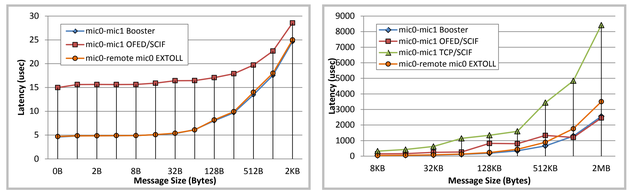


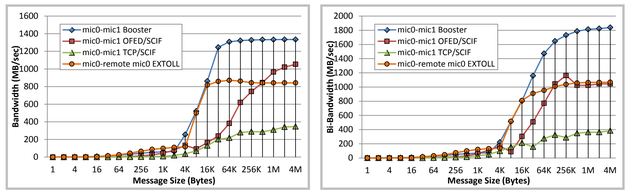
Fig. 6: Prototype system.

Results I – MPI Performance



(a) Small messages. (b) Large messages.

Fig. 7: Latency MIC-to-MIC internode half round-trip.

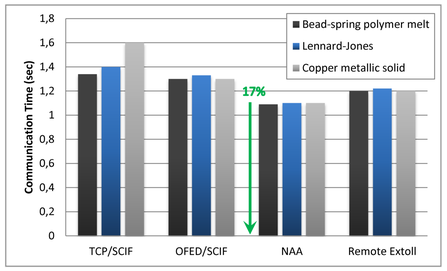


(a) Bandwidth. (b) Bidirectional bandwidth.

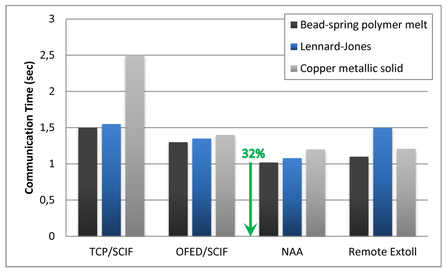
Fig. 8: Internode MIC-to-MIC Bandwidth and bi-bandwidth.

Results II – Application-level Evaluation

- MPI version of the LAMMPS application
- Equal Thread-to-MIC distribution
- Communication time between MICs can be improved by up to 32%



(a) 32 Threads, 16 Threads/MIC.



(b) 64 Threads, 32 Threads/MIC.

Fig. 9: LAMMPS performance using a bead-spring polymer, Lennard-Jones, and copper metallic solid benchmark.

Conclusion

- Novel communication architecture
- Scales the number of accelerators and CPUs independently
- Host-independent direct accelerator-to-accelerator communication with very low latency
- High-density implementation with promising MPI performance

References

[1] H. Froening, M. Nuessle, C. Leber, and U. Bruening, *On Achieving High Message Rates*. In CCGrid '13 (pp. 498-505).
[2] S. Neuwirth, D. Frey, M. Nuessle, and U. Bruening, *Scalable Communication Architecture for Network-Attached Accelerators*. In HPCA '15 (pp. 627-638).

The research leading to these results has been conducted in the frame of the DEEP (Dynamically Exascale Entry Platform) project, which has received funding from the European Union's Seventh Framework Programme for research, technological development, and demonstration under grant agreement no 287530.

More information available at www.deep-project.eu