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

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

ABSTRACT

The emergence of accelerator technology in current supercomputing systems is changing the landscape of supercomputing architectures. Accelerators like GPGPUs and coprocessors are optimized for parallel computation while being more energy efficient. Today's accelerators come with some limitations. They require a local host CPU to configure and operate them. This limits the number of accelerators per host. Another problem is the unbalanced communication between distributed accelerators. Network-attached accelerators are an architectural approach for scaling the number of accelerators and host CPUs independently. The design enables remote initialization, control of the accelerator devices, and host-independent accelerator-to-accelerator direct communication. Workloads can be dynamically assigned to CPUs and accelerators at run-time in an N to M ratio. An operative prototype implementation, based on the Intel Xeon Phi coprocessor and the EXTOLL NIC, is used to evaluate the latency, bandwidth, performance of the MPI communication, and communication time of the LAMMPS molecular dynamics simulator.

1. INTRODUCTION

The trend in supercomputing development indicates that the computational power of high performance computing (HPC) systems is increased by a factor of ten every decade. Considering this development cycle, devices like General-Purpose Graphics Processing Units (GPGPUs) and coprocessors become attractive choices for system designers. They are optimized for parallel computation while being more energy efficient. Heterogeneous systems come with some limitations. Current accelerators require a local host to configure and operate them. They are not designed to run autonomously and are incapable of sourcing or sinking network traffic. This leads to imbalanced workload distribution and communication. Amdahl's law [2] states that the scalability of a parallel code is limited by its sequential part. Future HPC systems have to be able to run applications with a varying degree of scalability and complicated communication patterns. Network-attached accelerators [6] are an architectural approach for scaling the number of accelerators and host CPUs independently. Accelerator nodes consist of an accelerator and a NIC. The configuration and operation is done over the network. The communication architecture is transparent to upper software and hardware layers. The commodity aspect of the accelerator is maintained. An operative prototype is implemented with Intel Xeon Phi coprocessors [4] and EXTOLL NICs [3].

2. COMMUNICATION ARCHITECTURE

The idea of network-attached accelerators (NAAs) is to improve the scalability between the number of accelerators and host CPUs within a system independently from the number of hosts. All accelerators can be used by any host within a cluster. The accelerator-to-accelerator direct communication does not require a local host. In addition, workload can be dynamically distributed between CPUs and accelerators at runtime in an N to M ratio.

2.1 Hardware Configuration

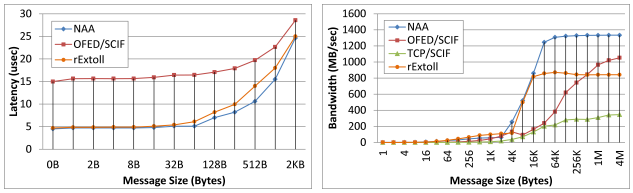
Current accelerators have to be configured by a CPU. In an NAA system, the configuration is enabled by the NIC. There are three different system entities: compute nodes (CN), booster interface nodes (BI), and booster nodes (BN). PCIe configuration packets are inserted to the NIC's outgoing host interface. The BI's CPU is able to configure the accelerator's PCIe interface via software, and to assign a memory-mapped I/O (MMIO) range to the PCIe endpoint. Accelerators are accessed by using loads and stores to locally reserved address ranges inside the BI's NIC. This has the additional benefit that the upper software and hardware layers remain unchanged. Local memory segments are exported to remote nodes to build a distributed shared memory. Loads and stores from the CPU to the exported segments are encapsulated into network transactions to the remote node. At the remote node, the packets are translated back into host interface requests. The MMIO ranges appear to be locally assigned to the BI node.

2.2 Software Design

The software layer substitutes the PCI kernel application programming interface (API) and maps the structures needed by the accelerator management software to the NIC's device structures. There are three main tasks that are provided by the software layer: device configuration and maintenance, MSI configuration, and interrupt handling. The software configures and maintains the mapped MMIO regions. The messaged signal interrupt (MSI) packet is configured in a way that the NIC is able to distinguish EXTOLL and accelerator interrupts based on the payload of the MSI packet.

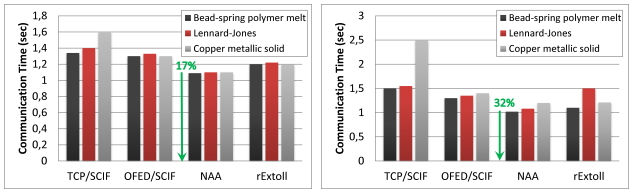
3. EVALUATION

A prototype is used to evaluate the network-attached accelerator approach. The internode MIC-to-MIC communication time is evaluated using OSU Micro-Benchmarks 4.3 [5] and the LAMMPS [1] molecular dynamics simulator.



(a) Latency. (b) Bandwidth.

Figure 1: Internode MIC-to-MIC communication performance using MPI.



(a) 16 Threads/MIC. (b) 32 Threads/MIC.

Figure 2: LAMMPS performance using a bead-spring polymer, Lennard-Jones, and copper metallic solid benchmark.

3.1 Prototype System

The BI prototype node is a standard server machine with two Intel Xeon E5-2630 processors running at 2.30GHz. An FPGA-version of the NIC is used that utilizes a Xilinx Virtex6 FPGA design with a 128bit-wide data path, running at 156.25MHz, and one x4 16Gbits/s EXTOLL link. The FPGA manages two Intel Xeon Phi coprocessors (MICs) with 8GB GDDR RAM each. The network link of the NIC is used to connect to the seventh link of the BN to get access to the 3D torus. The test environment contains two BN with two Altera StratixV FPGAs. The FPGAs implement an EXTOLL-compatible 128bit-wide data path running at 100MHz and seven x4 16Gbits/s links. Each StratixV NIC is connected to one MIC with an x8 PCIe Gen2 PCIe host interface. The StratixV FPGAs are connected via an x4 16Gbits/s link.

Intel MPSS 2.1.6720-16 is installed on the BI. The MPI performance is evaluated between two MICs connected over EXTOLL using OpenMPI 1.6.1 and directly connected to the BI utilizing SCIF, and OFED/SCIF using Intel MPI Library 4.1.3.049 and OFED-1.5.4.1.

3.2 Results

The point-to-point MPI benchmarks (*osu_latency*, *osu_bw*, *osu_bibw*) of the OMB are used for the evaluation. Early benchmarking of the prototype system shows that the half-round trip latency for small messages of up to 2kB in size is competitive with other solutions, see figure 1a. A minimum latency of 4.5μs can be achieved for internode MIC-to-MIC communication. The reached bandwidth of 1.3GB/s for large messages is close to the theoretical peak bandwidth of the FPGA, as displayed in figure 1b. Figures 2a and 2b display the communication time for three different LAMMPS benchmarks. LAMMPS is a classical molecular dynamics code. Running 64 threads on two Intel MICs with equal thread-to-MIC distribution shows a communication time improvement of about 32%.

4. CONCLUSION

The presented NAA approach is a novel architecture that scales the number of accelerators and CPUs independently. It facilitates accelerator-to-accelerator direct communication. The transparent implementation of the architecture enables dynamic workload mapping to accelerators and CPUs in an N to M ratio during runtime. Scalar code can be executed on the CPUs while highly scalable code is assigned to the accelerators.

The prototype system, using the Intel MIC technology, provides promising MPI communication layer results for internode latency, bandwidth, and communication time. Future work will include the setup and evaluation of an NAA system with the EXTOLL ASIC. In addition, the use of GPGPUs will be researched.

5. ACKNOWLEDGMENTS

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.

6. REFERENCES

[1] LAMMPS Molecular Dynamics Simulator, 2014.

[2] G. Amdahl. Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. In *AFIPS Conference Proceedings*, volume 30, pages 483–485, 1967.

[3] H. Froening, M. Nuessle, H. Litz, C. Leber, and U. Bruening. On Achieving High Message Rates. In *2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pages 498–505, May 2013.

[4] Intel Corporation. *Intel Xeon Phi Coprocessor System Software Developers Guide*, March 2014.

[5] Network-Based Computing Laboratory (NBCL), The Ohio State University. *OMB (OSU Micro-Benchmarks) README*, 2014.

[6] S. Neuwirth, D. Frey, M. Nuessle, and U. Bruening. Scalable Communication Architecture for Network-Attached Accelerators. In *IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pages 627–638. IEEE, 2015.