

A Configurable Network Protocol for Cluster Based Communications using Modular Hardware Primitives on an Intelligent NIC *

Ranjesh G. Jaganathan

Keith D. Underwood[†]

Ron Sass

Parallel Architecture Research Lab
ECE Dept. / Clemson University
Clemson, SC 29634-0915

Sandia National Laboratories[†]
P.O. Box 5800 MS-1110
Albuquerque, NM 87185-1110

E-mail: {jranjes, rsass}@parl.clemson.edu, kdunder@sandia.gov

Abstract

The high overhead of generic protocols like TCP/IP provides strong motivation for the development of a better protocol architecture for cluster-based parallel computers. Reconfigurable computing has a unique opportunity to contribute hardware level protocol acceleration while retaining the flexibility to adapt to changing needs. Thus, it is possible to provide application-specific protocol processing to improve performance and to reduce space utilization. Reducing space utilization permits the use of a greater portion of the FPGA for other application-specific processing.

This paper focuses on work to create a set of components that can be put together as needed to obtain a customized protocol for each application. The components are parameterizable, increasing the protocol's flexibility. To study the feasibility of such an architecture, hardware components for the reconfigurable logic on the NIC were built such that they can be stitched together as required to provide the required functionality. Feasibility is demonstrated using four different protocol configurations in this paper. The different configurations illustrate trade-offs between chip space and functionality.

1. Introduction

The Adaptable Computing Cluster (ACC) integrates RC and a commodity NIC, forming an Intelligent Network Interface Card or INIC, such as the GRIP2 card described in [1]. Introducing RC in the network datapath has been shown to have a significant impact on performance and is cost-effective for several classes of applications [5]. The INIC

can operate in several modes. It is the communications assist mode we wish to explore.

There are several reasons for wanting a communication assist in the network interface card. First, any protocol processing performed by the card is computation that the host does not have to do. In addition, because the card is aware of the protocol, it can interact with the network without host intervention. This has two important consequences. First, it reduces the latency of the overall protocol because short messages such as an ACK do not have to cross the peripheral bus twice. Second, the host does not have to service an interrupt or perform a context switch — thus saving several operations that contribute factors to the latency of the response and are pure overhead. While simple unicast messages are demonstrated in this paper, the architecture is generic enough that it is easily extended to support more complex, collective communications such as barrier synchronization [4], multicast, and reductions [2].

Another advantage of a protocol-aware INIC design is that it can be more efficient at transferring data to and from the host. A number of situations can arise to cause small packets including a limited network MTU and small communications with multiple nodes. These small messages, whether from one source or multiple sources, can often be reassembled in the INIC and transferred to the host memory via one DMA rather than as a sequence of short DMA transfers. Furthermore, as network speeds continue to grow, the delay involved in transferring data over the common PCI bus will become a critical bottleneck. Migrating the protocol to the INIC mitigates these effects.

Unlike traditional Beowulf clusters which use TCP, a different protocol is needed to reap the benefits of the INIC. TCP offers a reliable, in-order message transfer service. Although most parallel algorithms are built around the assumption of in-order, reliable messages, this is not always required by the algorithm.

The work described here concentrates on using the INIC as a communication assist for Beowulf-class architectures

*This work was supported in part by the National Science Foundation under NSF Grant EIA-9985986 and NSF Grant NAG5-11329.

[†]Dr. Underwood was supported by a NASA GSRP Fellowship under grant number NGT5-85

Table 1. A comparison of space and functionality for the various protocol configurations

Protocol Configuration	Reliable Yes/No	Ordered Yes/No	Dup elim Yes/No	Number of FlipFlops	Percentage of FlipFlops	Number of 4 LUTs	Percentage of 4 LUTs
CONF1	No	No	No	1643	13.10 %	1977	15.76 %
CONF2	Yes	No	No	2165	17.26 %	3380	26.95 %
CONF3	Yes	No	Yes	2534	20.20 %	3784	30.15 %
CONF4	Yes	Yes	Yes	2650	21.13 %	3933	31.14 %

running message-passing parallel codes with modern, high-speed networking such as Gigabit Ethernet. The design of a component-based, configurable network protocol for the INIC is described. Several modules have been implemented to demonstrate that the architecture is feasible.

2. Evaluation

Four different protocol configurations were put together to study feasibility. Each of the protocols evaluated were tested on a prototype Intelligent Network Interface (INIC). The experimental setup consists of the adaptable computing cluster (a Beowulf cluster with an INIC on each of the nodes). The prototype INIC used was a PCI-bus ACE2 card. The ACE2 card contains two Xilinx XC4085XLA FPGAs, a microSPARC chip, and a Gigabit Ethernet card.

This platform is an experimental prototype used in our ACC project. As such, it has a number of weaknesses that make it generally uncompetitive as a high-performance network interface. Thus, the primary concern here is to show feasibility: a configurable network protocol can be realized on an INIC. Secondary concerns include flexibility and resource trade-offs.

Each of the protocol components were implemented as VHDL entities. Four different configurations were put together to obtain varying functionality. Components in configuration 1 implement a simple unreliable packet transfer protocol. In configuration 2, the components implement a reliable, unordered message transfer protocol without the elimination of duplicate packets. Such protocols are appropriate when reliability is necessary but the order of packets (and packet redundancy) do not matter. The third configuration is a reliable, unordered message transfer protocol with the elimination of duplicate packets. In configuration 4, a complete reliable, ordered delivery protocol is implemented. **Table 1** shows the different functionalities provided and CLB space consumed by each of the protocol configurations.

3. Conclusion

This paper presents a configurable architecture for protocol acceleration in cluster computers. Such an architecture

provides excellent flexibility for cluster network interfaces. Hardware protocol options can be configured on a per application basis to maximize protocol performance and minimize the area requirements.

Using an application-specific hardware protocol in the ACC has shown a significant performance gain for certain applications[3]. The component based protocol approach reduces the programming time and cost for future ACC applications. It not only improves network performance by hardware acceleration, it also eliminates overhead on the host processor improving overall application performance. Optimized components can reduce chip space leaving much hardware as possible for application acceleration without sacrificing network performance.

This paper demonstrated the feasibility and flexibility of an RC-based communication architecture. The key characteristic of the architecture is the ability to modularize a communications protocol and selectively assemble those modules to form specialized protocols. Future work will focus on building a next generation prototype and analyzing the performance advantages of the various protocol configurations.

References

- [1] P. Bellows, V. Bhaskaran, J. Flidr, T. Lehman, B. Schott, and K. D. Underwood. GRIP: A reconfigurable architecture for host-based gigabit-rate packet processing. In *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, April 2002.
- [2] V. Kumar, A. Grama, A. Gupta, and G. Karypis. *Introduction to Parallel Computing: Design and Analysis of Algorithms*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1994.
- [3] K. Underwood, R. Sass, and W. Ligon. Acceleration of a 2d-fft on an adaptable computing cluster. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines*, April 2001.
- [4] K. D. Underwood. *An Evaluation of the Integration of Reconfigurable Hardware with the Network Interface in Cluster Computer Systems*. PhD thesis, Clemson University, Aug. 2002.
- [5] K. D. Underwood, R. R. Sass, and W. B. Ligon, III. Cost effectiveness of an adaptable computing cluster. In *Proceedings of the 2001 Conference on Supercomputing*, Nov. 2001.