Implementation and Performance Evaluation of M-VIA on AceNIC Gigabit Ethernet Card*

In-Su Yoon[1], Sang-Hwa Chung[1], Ben Lee[2], and Hyuk-Chul Kwon[1]

[1] Pusan National University
School of Electrical and Computer Engineering
Pusan, 609-735, Korea
{isyoon, shchung, hckwon}@pusan.ac.kr
[2] Oregon State University
Electrical and Computer Engineering Department
Owen Hall 302, Corvallis, OR 97331
benl@ece.orst.edu

Abstract. This paper describes the implementation and performance of M-VIA on the AceNIC Gigabit Ethernet card. The AceNIC adapter has several notable hardware features for high-speed communication, such as jumbo frames and interrupt coalescing. The M-VIA performance characteristics were measured and evaluated based on these hardware features. Our results show that latency and bandwidth improvement can be obtained when the M-VIA data segmentation size is properly adjusted to utilize the AceNIC's jumbo frame feature. The M-VIA data segmentation size of 4,096 bytes with MTU size of 4,138 bytes showed the best performance. However, larger MTU sizes did not necessarily result in better performance due to extra segmentation and DMA setup overhead. In addition, the cost of M-VIA interrupt handling can be reduced with AceNIC's hardware interrupt coalescing. When the parameters for the hardware interrupt coalescing were properly adjusted, the latency of interrupt handling was reduced by up to 170 μs.

1 Introduction

Gigabit Ethernet based clusters are considered as scalable, cost-effective platforms for high performance computing. However, the performance of Gigabit Ethernet has not been fully delivered to the application layer because of the TCP/IP protocol stack overhead. In order to circumvent these problems, a group of user-level communication protocols has been proposed. Examples of user-level communication protocol are U-Net [1], Fast Message [2], Active Message [3] and GAMMA [4]. The Virtual Interface Architecture (VIA) [5] has emerged to standardize these different user-level communication protocols. Since the introduction of VIA, there have been several software and hardware implementations of

* This work has been supported by Korean Science and Engineering Foundation (Contract Number: R05-2003-000-10726-0) and National Research Laboratory Program (Contract Number: M10203000028-02J0000-01510).

H. Kosch, L. Böszörményi, H. Hellwagner (Eds.): Euro-Par 2003, LNCS 2790, pp. 995–1000, 2003.

VIA. M-VIA (Modular VIA) [6] is a software implementation that employs Fast or Gigabit Ethernet as the underlying platform.

This paper discusses the implementation of M-VIA on the AceNIC Gigabit Ethernet card. The AceNIC Gigabit Ethernet card has several notable hardware features which are jumbo frames and interrupt coalescing. Therefore, this paper presents a study of what effects jumbo frames and interrupt coalescing features have on the performance of M-VIA.

2 M-VIA Overview

M-VIA is implemented as a user-level library and at least two loadable kernel modules for Linux. The core module (*via_ka* module) is device-independent and provides the majority of functionality needed by VIA. M-VIA device drivers implement device-specific functionality. In M-VIA device drivers, the *via_ering* module includes operations such as construction and interpretation of media-specific VIA headers and mechanisms for enabling VIA to co-exist with traditional networking protocols, i.e., TCP/IP. In this paper, we present our implementation of M-VIA on the AceNIC by developing a new AceNIC driver module (*via_acenic* module) for the M-VIA. Also, the *via_ering* module was modified to support different M-VIA segmentation sizes.

3 AceNIC Hardware Features

3.1 Jumbo Frames

Although jumbo frames are available to transfer large data, the original M-VIA segmentation size was designed to support the standard Ethernet MTU size of 1,514 bytes. When M-VIA transfers data, the *via_ering* module organizes data into pages and then each page is divided into the M-VIA segmentation size. Then, the *via_acenic* module writes the physical address, length, and other information of each data segment on the AceNIC's descriptor. Finally, each data segment is transferred to the AceNIC's buffer via DMA. Since segmentation and DMA setup require substantial amount of processing time, it is important to reduce the number of data segments.

In our implementation, the M-VIA segmentation size was adjusted to utilize AceNIC's jumbo frame feature. M-VIA segmentation size of 8,958 bytes is obtained by subtracting M-VIA data header of 42 bytes from an MTU of 9,000 bytes. With large MTU and segmentation size, the number of M-VIA packets is significantly reduced. When M-VIA segmentation size is made equal to the page size, the *via_ering* module needs to only generate one segment for each page. In this case, we can use an MTU of 4,138 bytes, which is obtained by adding a data header to M-VIA segment size of 4,096 bytes.

3.2 Interrupt Coalescing

Interrupt coalescing delays the generation of an interrupt until a number of packets arrive. The waiting time before generating interrupt is controlled by setting the AceNIC's internal timer. The internal timer starts counting clock ticks from the time when the first packet arrives. The number of coalescing clock ticks can be changed by modifying module parameters. With the hardware interrupt coalescing, the host can amortize the interrupt handling cost over a number of packets and thus save host processing cycles.

Because many Ethernet cards do not support hardware interrupt coalescing, M-VIA implements this feature in software. When M-VIA sends intermediate data segments, it does not mark the completion flags of the descriptors, which are ignored by the interrupt handler. When the completion flag of the final descriptor is marked, it indicates the completion of the interrupt coalescing. M-VIA's software interrupt coalescing conflicts with AceNIC's because an interrupt can be generated by the expired timer before M-VIA marks the completion flag on the final descriptor. Therefore, we maximized the number of coalescing clock ticks to prevent interrupts from being generated during send operations. However M-VIA does not implement the interrupt coalescing when it receives data, and instead depends entirely on the receive interrupt handler of NIC. The cost of M-VIA's receive interrupt handling is reduced using the AceNIC's interrupt coalescing.

4 Experimental Results

The performance of M-VIA was measured using 800 MHz Pentium III PCs with 256 MB SDRAM and 33 MHz/32-bit PCI bus. The AceNIC Gigabit Ethernet card used was a 3Com 3C985B-SX. The PCs were running with Linux kernel 2.4. M-VIA latency and bandwidth were measured using *vnettest* program.

4.1 M-VIA Data Segmentation Size and Hardware MTU

Figure 1 shows the performance of M-VIA with various segmentation sizes. One interesting observation is that M-VIA segmentation size of 4,096 bytes shows a sawtooth shape of the curve. This is because fewer packets are generated when the data size is multiples of the page size. For M-VIA segmentation size of 8,192 bytes, the performance is worse than that of 4,096-byte case until the data size reaches approximately 50 KB. Although AceNIC was configured to carry 8,192-byte frames, the *via_ering* module segments data by page size. Therefore, an 8,192-byte frame requires two segmentations and DMA initiation processes resulting in extra overhead. However, the receiving side can benefit from larger MTUs for bulk data due to reduced number of interrupts. The segmentation size of 8,958 bytes shows even worse performance because of extra segmentation and DMA initiation costs. For data size from 8 KB to approximately 36 KB, 8,958-byte segment size resulted in even worse performance compared to the

1,472-byte case. This is due to the fact that small frames allow packets to be sent faster to the receiving side, while large frames have to wait until they are filled up. However, for data size larger than 36 KB, 8,958-byte case performs better because small frames generate more frequent interrupts on the receiving side.

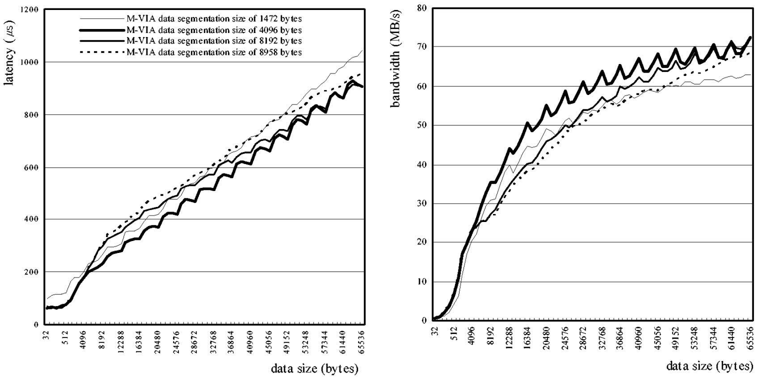

Fig. 1. Performance with various M-VIA segmentation sizes

4.2 Hardware Interrupt Coalescing Feature

The interrupt coalescing of AceNIC is controlled by a pair of parameters (*tx_coal*, *rx_coal*), which specify the number of coalescing clock ticks for transmission and reception, respectively. These parameters indicate the duration of packet send/receive before interrupting the host. Figure 2 shows the M-VIA latency for AceNIC using 1,514-byte MTU.

When the parameters are set to (1, 1), the interrupt coalescing is disabled for both transmission and reception. Thus, AceNIC invokes the interrupt handler routine as soon as a packet arrives. This leads to a minimum latency of 67 μs for a 32-byte data, but results in significantly longer latencies for larger messages. The maximum latency difference between (1, 1) and (2000, 90) is approximately 400 μs for 64 KB data. To evaluate the latency of the M-VIA's software interrupt coalescing, the parameters were set to (2000, 1). The *tx_coal* value of 2,000 is sufficient for M-VIA to complete its transmit operations before the expired timer generates an interrupt. This results in significantly lower latency than disabling the interrupt coalescing. The approximate minimum and maximum latency differences are 2 μs and 212 μs, respectively. To confirm that the hardware interrupt coalescing for receiving data improves performance, experiments with parameters set to (2000, 90) were performed. The value 90 was determined experimentally to give the best performance. For data sizes larger than 17 KB, lower latencies were observed compared to when the parameters were set to

(2000,1). The maximum latency difference between (2000, 90) and (2000, 1) is approximately 170 μs for sending a 64 KB data. For data sizes smaller than 17 KB, slightly higher latencies were observed because of the increased waiting time on the receiving side, but the difference was negligible.

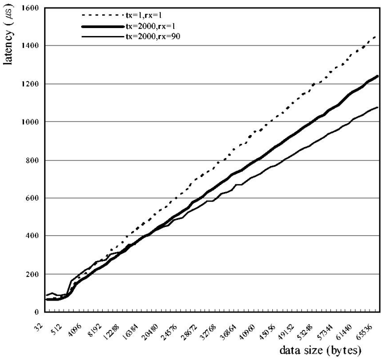

Fig. 2. M-VIA latency with hardware interrupt coalescing feature

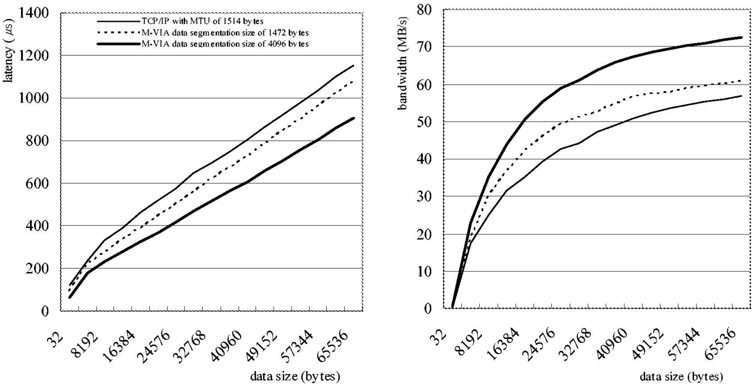

Fig. 3. M-VIA vs. TCP/IP

4.3 Comparison of M-VIA and TCP/IP

Figure 3 shows a comparison between M-VIA and TCP/IP in terms of latency and bandwidth. In this experiment, the 8,192-byte and 8,958-byte segment sizes were excluded because 4,096-byte segment size resulted in better performance. When both M-VIA and TCP/IP use the same MTU size of 1,514 bytes, M-VIA has lower latency than TCP/IP. M-VIA and TCP/IP have minimum latencies of 89 μs and 123 μs respectively. The minimum latency difference is 15 μs with 4 KB data. For data sizes larger than 16 KB, the latency difference is approximately 76 μs. M-VIA and TCP/IP have maximum bandwidths of 60.9 MB/s and 56.9 MB/s, respectively. Comparing M-VIA using MTU size of 4,138 bytes with TCP/IP, the minimum latency difference is 57 μs with 4 KB data and the maximum latency difference is 246 μs with 64 KB data. M-VIA has a maximum bandwidth of 72.5 MB/s with segmentation size of 4096 bytes.

5 Conclusion

We presented our implementation and performance study of M-VIA on the AceNIC Gigabit Ethernet card by developing a new AceNIC driver for M-VIA. In particular, we focused on AceNIC's jumbo frame and interrupt coalescing features for M-VIA. We experimented with the various M-VIA data segmentation sizes and MTUs. The M-VIA data segmentation size of 4,096 bytes with MTU size of 4,138 bytes showed the best performance. Comparing M-VIA using MTU size of 4,138 bytes with TCP/IP, M-VIA latency improves by approximately 57$\sim$246 μs and results in maximum bandwidth of 72.5 MB/s. Also the latency time of M-VIA's interrupt handling was reduced by up to 170 μs with the AceNIC's hardware interrupt coalescing.

References

1. T. Von Eicken, A. Basu, V. Buch, and W. Vogels: "U-NET: A User Level Network Interface for Parallel and Distributed Computing", Proc. of the 15th ACM Symposium on Operating Systems Principles (SOSP), Colorado, December 1995
2. S. Pakin, M. Lauria, and A. Chien: "High Performance Messaging on Workstation: Illinois Fast Message (FM) for Myrinet", Proc. of Supercomputing '95, December
3. T. von Eicken, D. E. Culler, S. C. Goldstein, and K. E. Schauser: "Active Messages: a Mechanism for Integrated Communication and Computation", 19th International Symposium on Computer Architecture, May 1992
4. G. Chiola and G. Ciaccio: "GAMMA: a Low-cost Network of Workstations Based on Active Messages", Proc. of 5th EUROMICRO workshop on Parallel and Distributed Processing, London, UK, January 1997
5. Intel, Compaq and Microsoft Corporations: Virtual Interface Architecture specification version 1.0, December 1997, http://developer.intel.com/design/servers/vi/
6. P. Bozeman and B. Saphir: "A Modular High Performance Implementation of the Virtual Interface Architecture", Proc. of the 2nd Extreme Linux Workshop, June 1999