

An Architecture of high performance cluster network: Maestro2

Keiichi Aoki*, Shinichi Yamagiwa**, Masaaki Ono*, Koichi Wada*, Luis Miguel Campos**

*Institute of Information Sciences and Electronics, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573 JAPAN. (e-mail: k1@padc.mmpc.is.tsukuba.ac.jp)

** PDM&FC, Rua Latino Coelho, 87, 1050-134, Lisboa, Portugal

Abstract—Cluster computers have become the platform of choice to run high performance parallel computing applications. However, most cluster computers use conventional wide-area network technologies to interconnect the individual processing elements. Therefore, there is a disparity between the wide-area network technologies used and the network technology required by cluster computing, leading to a severe degradation in performance. Maestro network technology attempts to solve this disparity. This paper describes improvements to the basic Maestro architecture. In particular it describes a new technique used by Maestro2 (*Continuous network burst*) and shows its impact in the overall performance of Maestro cluster networks.

Index Terms—High performance Network Architecture, Cluster computing, Network Protocols, Performance evaluation

I. INTRODUCTION

We have developed a high-performance network for cluster computing, called *Maestro cluster network* [6]. It has been developed to solve performance disparities between the obtainable communication performance using conventional WAN or LAN-based network technologies and the requirements of cluster applications. Although Maestro networks offer low latency and high throughput [6], as required by high performance parallel computing applications, during the development of the technology we have identified two issues that restricted the overall performance of Maestro cluster networks. The first issue occurs at the link layer and it is related with extending the continuous transfer of data. The second issue occurs in the switch itself and it is related with reducing idle time. We propose two new techniques to solve the two issues mentioned above. The first technique is called *continuous network burst*. This technique allows the protocol to continue the network burst as long as there are packets in the network buffer of the link layer. The second technique is called *out of order switching*. In out of order switching, the switch processes subsequent messages when it can not continue to transfer any given message. This technique reduces idle time of switch buses and increase throughput.

We have implemented the new link layer protocol in FPGA as MLC-X chips. Additionally, we have developed network in-

terfaces which connect to host processors via MLC-X chip, and a switch box that connects to network interfaces via LVDS cables [5]. Together these components compose Maestro2 cluster network.

This paper describes the first technique, *continuous network burst*, the overall architecture of Maestro2, and shows basic performance evaluation results.

II. MAESTRO CLUSTER NETWORK

A. Issues with WAN or LAN-based network

A cluster [1] is usually physically located in a closed environment. Therefore high-level communication guarantees such as multiple error correction and layered connection quality maintenance are superfluous in inter-cluster communication. These guarantees cause severe communication performance degradation which in turn leads to an overall performance degradation of parallel application program. Our aim is to improve the communication performance in clusters by bridging the difference between communication strategies used for wide area networks and the ones required by cluster networks.

The communication overheads in WAN-based network technology are as follows:

- 1) The frame header in the physical layer includes redundant information. In the case of Ethernet, the header occupies 36bytes of the frame. If the data size is small, say 4bytes, 90% of the bytes transmitted are header information.
- 2) The sender sends each communication unit even if those units are smaller than maximum size allowed by the link layer. This increases the number of transmissions and transmission startup cost and thus degrades the communication throughput.
- 3) Message transmission is postponed until the whole message is prepared at the link layer. Conventional link layer hardware sends a message only after the entire message is written into the buffer for transmission. This serializes the communication processes and degrades throughput.

These overheads degrade communication performance in particular and overall performance of cluster systems in general, when WAN-based network technologies are used. If these overheads are reduced or even eliminated it is possible to significantly increase communication and overall system performance.

B. Maestro cluster network

We have developed *Maestro cluster network* [6] to address the disparity described earlier. In the Maestro project, we have proposed two key techniques, named *network burst* and *pipelined transfer*.

1) Network burst

To address the second issue mentioned in the last section, we have implemented Maestro Link Protocol in the link layer of the network interface hardware. Maestro Link Protocol reduces the overhead by executing the following steps:

1. Divide a message into small data units. We call these data units *packets*.
2. The sender sends packets continuously, as many as the receiver can receive, in burst after an initial arbitration phase.

Network burst is able to send messages in long bursts for as long as the receiver can receive them, without having to acquire the physical medium more than once. Via experimentation we have confirmed that network burst is able to reduce the overhead caused by the accumulated effect of arbitration.

2) Pipelined transfer

In the third problem described in the previous section, with conventional network technologies, the sender can not process the next message until it completely finishes sending the current message at the link layer. The link layer controller of Maestro divides messages into small units, hands them over to each hardware components in order and makes a data pipeline from the sender's buffer to the receiver's buffer.

C. Consideration of Maestro cluster network technology

Through the examination of Maestro cluster network technology, we identified four issues that restrict performance. They are:

1. Arbitration time at the physical layer
Maestro cluster network technology uses IEEE1394 physical layer. However IEEE1394 requires 640ns for arbitration. That corresponds to 16% of the latency of the link layer.
2. Half-duplex physical medium
Maestro cluster network technology uses only half-duplex medium. That reduces the obtainable throughput to half of its potential when two processors send data simultaneously.
3. Communication protocol in the link layer
Network burst terminates transfer invariably when the sender finishes transferring the amount of packets decided at the beginning of the transfer.
4. Serial switching algorithm in the switch box
Maestro's switch must process requests one by one even if there is no dependency between requests.

In other words, we still have room to optimize inter-cluster communication in Maestro cluster networks. Namely, arbitration time in the physical layer can be reduced and half-duplex physical medium can be improved by using full-duplex physical medium. Communication protocol in the link layer can

be improved by adding the functionality to dynamically notify the sender, of the amount of network buffer remaining on the receiver side and allow the continuous sending of data as long as the buffer capacity in the receiver's side is not zero. Finally, by allowing out of order requests to be processed in the switch box, we can also improve performance. In next section, we describe one of new techniques proposed, called *continuous network burst*.

III. NEW TECHNIQUE – CONTINUOUS NETWORK BURST

We propose the *continuous network burst* technique to tackle overhead in the link layer.

The original network burst implemented in the Maestro cluster network, terminated the transfer inevitably when the sender finished transferring the amount of packets decided at the beginning of the transfer. When the sender wants to transfer data continuously, it must acquire the physical medium repeatedly. This accumulative effect of arbitration reduces throughput and decreases overall performance. *Continuous network burst* decreases those overheads and increases the utilization ratio of the physical medium. It works by having the sender to check the amount of capacity of network buffer remaining in the receiver and to continue the network burst if the receiver is able to accept it.

Fig. 1 compares continuous network burst with the conventional network burst implemented in the Maestro cluster network. In continuous network burst, the sender inserts the *continue word* to notify the receiver it wants to continue sending data, and continues the transfer without additional arbitrations, if the receiver is able to receive more data. This technique reduces the number of arbitrations and achieves higher throughput than the conventional network burst.

IV. IMPLEMENTATION

Maestro2 cluster network is composed of network interfaces and switch boxes as shown in Fig. 2. Network interfaces are connected to each host processor via the PCI bus and exchange messages with host processor. A message is a communication unit composed of one or more packets. Each switch box is connected to up to 8 network interfaces via LVDS (Low Voltage Differential Signaling) physical layer [5] and its responsible for switching messages from the different network interfaces.

A. Network interface

The network interface (NI) is composed of a LVDS transmitter/receiver, 8Kbytes network buffer, MLC-X, PCI interface, PowerPC603e 300MHz[2], 64Mbytes SDRAM. MLC-X, PCI interface and network buffer are implemented in VirtexII FPGA chip. The continuous network burst technique is implemented in the MLC-X. MLC-X controls LVDS transmitter/receiver and is bidirectional. LVDS transmitter/receiver is connected to the switch box and transfer data at 3.2Gbps each way.

B. Switch box

The switch box (SB) includes eight LVDS transmitter/receiver, four SB interfaces, a shared transfer bus, SB manager and switch controller. SB interfaces are composed of a message analyzer, two MLC-X chips and network buffer. It communicates with network interfaces via LVDS connections. MLC-X and network buffer in the switch box are similar to the ones in the NI. The message analyzer is connected to the MLC-X. It picks up headers from messages and passes it to the SB manager. As described above, Maestro2 NIs and SB are connected via LVDS cables and exchange messages. Data is transferred between each module in pipeline manner.

V. EVALUATION

A. Environment

We prepared two host PCs with Maestro2 network interface to measure communication performance. Table I shows the test environment we prepared for the evaluations.

B. Basic performance

We show basic performance of Maestro2 measured by PINGPONG. We measured throughput and latency between two Network Interfaces when they are connected directly and when they are connected via a Switch Box. We measured time

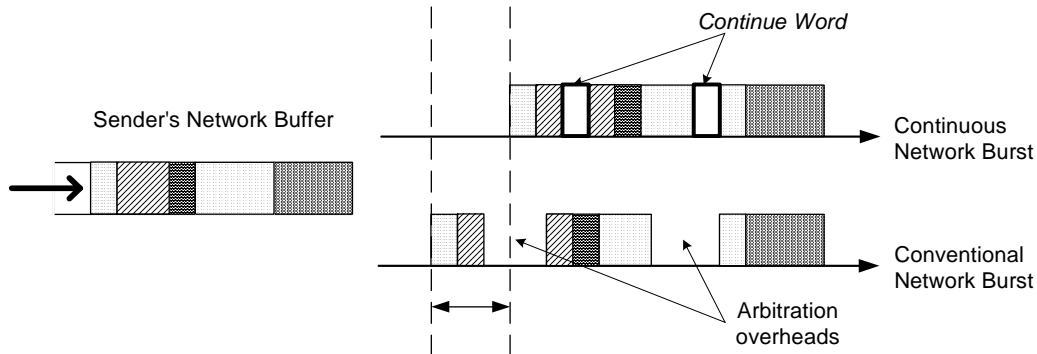


Fig. 1 The comparison with two network burst

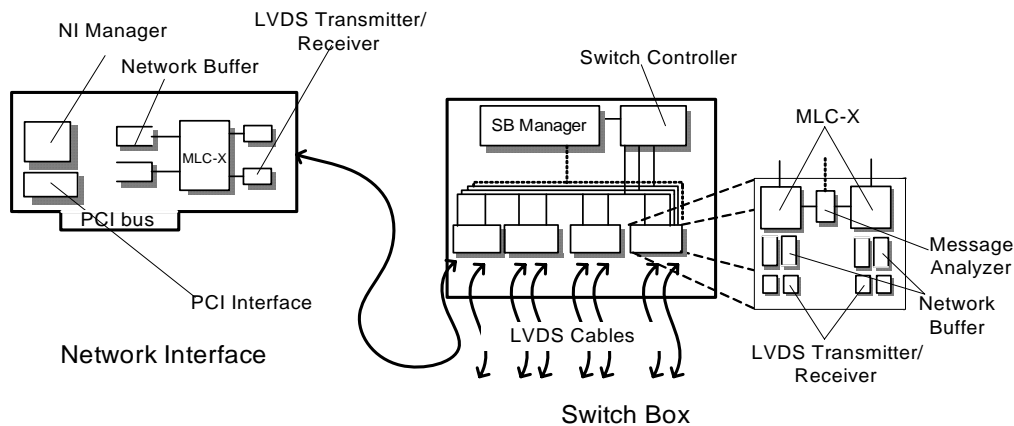


Fig. 2 Maestro2 cluster network

C. Effects of continuous network burst

To evaluate the performance of continuous network burst, we measured throughput and latency between two network

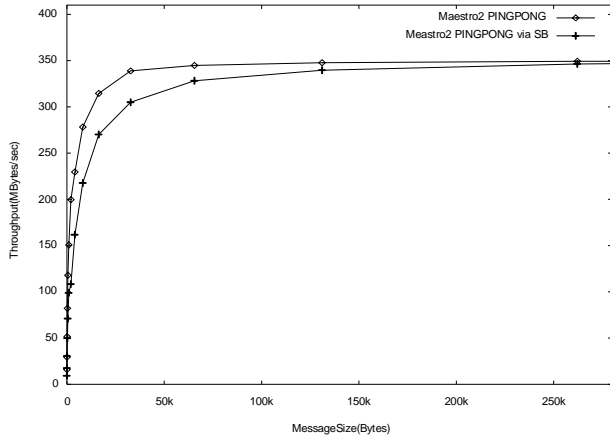
TABLE I
THE TEST ENVIRONMENT

THE TEST ENVIRONMENT	
Host PC 1	Intel PentiumIII 800MHz SUPERMICRO 370DE6 PC133 SDRAM 512MB
Host PC 2	Intel PentiumIII 1GHz SUPERMICRO 370DE6 PC133 SDRAM 512MB
OS	Linux kernel 2.4.7-10

by using the *time base register* of PowerPC, from the moment the sender's PowerPC starts sending messages (from SDRAM) to the moment the receiver receives the entire message (to SDRAM). Fig. 3 shows the basic performance of Maestro2 by varying the message size up to 512Kbyte when two NIs are connected directly. The minimum latency of Maestro2 is 1.9 μ sec when the message size is 32bytes. The maximum throughput of Maestro2 is 353Mbytes/sec. It is 88% of the physical medium capacity. When two NIs are connected via a switch box, the minimum latency is 3.3 μ sec. The maximum throughput is 352Mbytes/sec. In the case of large message, the graph effectively shows the message to be transmitted from one NI to the other in a pipeline manner.

interfaces with and without continuous network burst. Fig. 4 shows the performance comparison measured by PINGPONG communication. Performance results using continuous net-

work burst were calculated in the previous subsection.



Number of Bytes	32	64	128	256	512	1024	2048	4096	8192	16384
Latency of PINGPONG(ns)	1920	2070	2370	2970	4140	6480	9780	17010	28080	49680
Latency of PINGPONG via SB(ns)	3270	3480	3990	4890	6870	9870	18030	24150	35880	57840
Throughput of PINGPONG(Mbytes/sec)	16	29	52	82	118	151	200	230	278	315
Throughput of PINGPONG via SB(Mbytes/sec)	9	18	31	50	71	99	108	162	218	270

Fig. 3 Basic Performance of Maestro

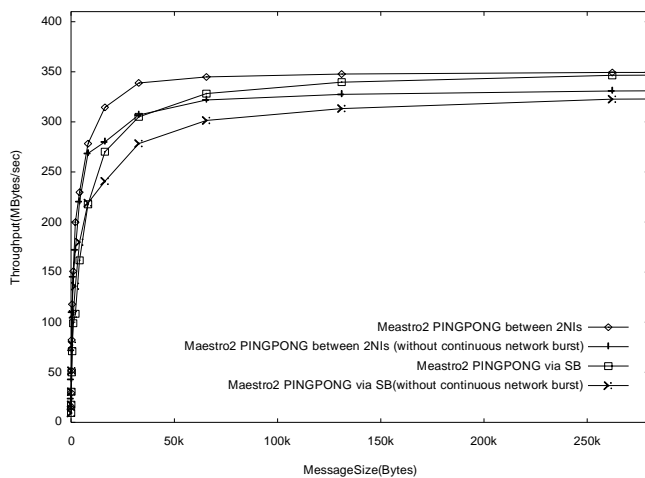


Fig. 4 Performance comparison with continuous network burst and no continuous network burst

In conventional network burst, the maximum throughput is 334Mbytes/sec when two NIs are connected directly and 328Mbytes/sec when they are connected via a switch box. Therefore, throughput using continuous network burst is 6% higher than without continuous network burst when NIs are connected directly, and is 7% higher when connected via a SB.

From these evaluation results, we confirmed that *continuous network burst* increases utilization ratio of the physical medium and increases throughput.

VI. COMPARISON WITH OTHER NETWORKS

Ethernet [3] Network Interface, accepts Ethernet frames and transfers them at the link layer. The length of Ethernet frames is variable (up to 1500 bytes). As such, it is impossible to continuously transfer more than 1500 bytes at the link layer. Gigabit Ethernet[6] can aggregate multiple IP datagrams into an Ethernet frame and transfer it in order to increase the utilization ratio of the physical medium. However Gigabit Ethernet must acquire the physical medium between frames because it is not able to send consecutive frames in bursts.

Myrinet[2] provides the so-called “B” bit in the physical medium and transfer data in burst with STOP and GO flow control. The receiver activates the “B” bit to tell the sender to stop sending data when the received data exceeds the STOP limit of the receiver’s buffer. Therefore, Myrinet transfers data in bursts similar to Maestro2 *continuous network burst* until the receiver activates its “B” bit. However, it is necessary to insert an extra physical wire for the “B” bit because the flow control algorithm of Myrinet requires a special line for each channel. Additional buffers are also required in the receiver to receive data until the signal of “B” bit is asserts at the sender. Maestro2 however does not require special line for flow control.

VII. CONCLUSIONS

In this paper we considered several possibilities to improve the performance of Maestro cluster networks. We described one of such possibilities, the *continuous network burst*. From performance evaluation, we confirmed that continuous network burst is effective in improving the physical medium utilization and throughput.

For future work, we plan to obtain results using a specialized messaging passing library (Maestro Message Passing) and measure performance using MPI benchmarks.

ACKNOWLEDGEMENT

This research was supported by Japan Society for the Promotion of Science, a Grant-in-Aid for Scientific Research(C), 14580361, 2002 and by the Portuguese Government through Agência de Inovação via a grant under Programa POCTI – Medida 1.2

REFERENCES

- [1] M.Baker,R.Buyya,and D.Hyde. Cluster computing: A high-performance contender. IEEE Computer, July 1999.
- [2] Nannette J.Boden, Danny Cohen, Robert Felderman, Alan E. Kulawik, Charles L Sietz, Jacov N. Seizovic, and Wen-King Su. Myrinet - a gigabit-per-second local-area network. IEEE Micro, Vol.15, No.1,1995.
- [3] Robert Breyer and Sean Riley. Switched and Fast Ethernet, Second Edition. Ziff Davis Press, 1996.
- [4] Motorola. MPC603e & EC603e Microprocessor User's Manual. 1997.
- [5] National Semiconductor. LVDS Owner's Manual & Design Guide (2nd Edition).
- [6] Stephen Saunders. Gigabit Ethernet HANDBOOK. McGraw-Hill Professional, 1998.
- [7] Shinichi Yamagiwa, Munehiro Fukuda, and Koichi Wada. Design and Performance of Maestro Cluster Network. In proceedings of IEEE International Conference on Cluster Computing (CLUSTER2000), November 2000.