Impact of Topology and Link Aggregation on a PC Cluster with Ethernet

Takafumi Watanabe [†1], Masahiro Nakao [†2], Tomoyuki Hiroyasu [‡3], Tomohiro Otsuka [*4], Michihiro Koibuchi [#5]

[†] *Graduate School of Engineering, Doshisha University*
1-3 Tatara Miyakodani, Kyo-tanabe, Kyoto, 610-0321, Japan
[1] twatanabe@mikilab.doshisha.ac.jp
[2] mnakao@mikilab.doshisha.ac.jp

[‡] *Department of Life and Medical Sciences, Doshisha University*
1-3 Tatara Miyakodani, Kyo-tanabe, Kyoto, 610-0321, Japan
[3] tomo@is.doshisha.ac.jp

[*] *Graduate School of Science and Technology, Keio University*
3-14-1, Hiyoshi, Kohoku-ku, Yokohama, 223-8522, Japan
[4] terry@am.ics.keio.ac.jp

[#] *National Institute of Informatics/The Graduate University of Advanced Studies/JST*
2-1-2 Hitotsubashi, Chiyoda-ku, 101-8430, Tokyo, Japan
[5] koibuchi@nii.ac.jp

***Abstract*—In addition to its use in local area networks, Ethernet has been used for connecting hosts in the area of high-performance computing. Here, we investigated the impact of topology and link aggregation on a large-scale PC cluster with Ethernet. Ethernet topology that allows loops and its routing can be implemented by the VLAN routing method without creating broadcast storms. To simplify the system configuration without modifying system software, the VLAN tag is added to a frame at switches in our implementation of topologies. Each host creates VLAN interfaces that have different local network addresses on a physical interface, so that a switch learns the MAC addresses of hosts in a PC cluster by broadcast. Evaluation results showed that the performance characteristics of an eight-switch network are comparable to those of an ideal 1-switch (full crossbar) network in the execution of High-Performance LINPACK Benchmark (HPL) on a 225-host PC cluster. On the other hand, evaluation results using NAS Parallel Benchmarks indicated that topologies achieved by the proposed methodology showed performance improvements of up to about 650% as compared to the simple tree topology. These results indicate that topology and link aggregation have marked impacts and commodity switches can be used instead of expensive and high functional switches.**

I. Introduction

Inexpensive commodity hardware and simplicity of management make Ethernet a popular choice not only for local area networks (LANs) but also for interconnects in PC clusters. In particular, with the use of layer-2 non-blocking switches of Gigabit Ethernet (GbE) and standardization of 10GBASE-T with twisted-pair cable (IEEE 802.3an-2006), Ethernet has seen widespread application as interconnects in high-performance computing (HPC), and its performance is close to that of expensive System Area Networks (SANs), such as Myrinet.

However, many PC clusters with Ethernet adopt a simple tree topology. Although systems with GbE account for 57% of the TOP500 Supercomputer ranking for scientific applications [1], to our knowledge these do not employ topologies containing loops, such as the torus topology. This is because Ethernet that uses the spanning-tree protocol (STP, IEEE-802.1D) does not allow topologies that include loops. A tree topology tends to result in imbalanced traffic around the root. To mitigate the problem of imbalanced traffic, a bundle of multiple links between the same pair of switches around the root is sometimes employed with the link aggregation standard (IEEE 802.3ad). However, in the case of large-scale PC clusters, as many ports of a switch are used for a single channel by link aggregation, the topology tends to be a low-dimensional tree with large diameter.

The VLAN application for employing multiple paths between source and destination switches has been proposed in addition to the link aggregation [2][3]. As the VLAN-based method disables STP, which avoids loops of links, topologies including loop structures, such as a Fat tree and Torus, can be achieved. As each VLAN takes a tree topology, it avoids broadcast storms. When we build a large-scale PC cluster system without VLANs, management of host MAC addresses will become complicated, because it is difficult to resolve a broadcast storm by addition of a host, failure of a switch, and misoperation without VLANs.

For stable management of the MAC addresses of hosts,

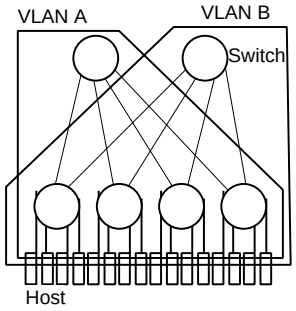

Fig. 1. VLAN-based routing method

the VLAN-based routing method [2][3] is one of the best methods of IEEE-802.1Q tag VLAN technology. The VLAN technology was originally intended to divide hosts into two or more logical local area networks, when all hosts originally belong to a single physical network. It has been widely used in not only LANs, but also in QoS control in advanced Internet backbones [4]. The VLAN-based routing method uses VLANs for improvement of network throughput. As shown in Figure 1, each host belongs to two or more VLAN groups, and we assign a link set that is different in each VLAN. Hosts can communicate with each other, and a path is selectable by specifying the proper VLAN ID.

In this study, we implemented our VLAN-based routing method [5] in a large-scale PC cluster with eight switches the system software of which cannot handle VLAN technology. We attempted to improve performance of the large-scale PC cluster by improving Ethernet topology and routing.

To simplify the system configuration without modifying the system software of the PC cluster, the VLAN tag is added to the frame at switches in the implementation of topologies. Each host creates VLAN interfaces that have different local network addresses on a physical interface, so that a switch learns the MAC addresses of hosts in a PC cluster by broadcast.

The VLAN-based routing method simply configures a layer-2 switch as follows; it allocates the VLAN sets, disables STP, and optionally enables link-level flow control for increasing the throughput. As it can be implemented on commercial Ethernet switches with the above configuration, it has a high degree of portability.

This paper is organized as follows. Section II describes related work. Section III describes the outline of the 225-host PC cluster, and implementation of the VLAN-based routing method. In Section IV, we describe performance evaluation of the 225-host PC cluster. We present our conclusions in Section V.

II. Related Work

VLAN-based routing, which enables employment of various topologies that include loops, was proposed to obtain well-distributed multiple paths among switches without creating broadcast storms [2][3]. Miura et al. developed a Linux device driver to treat VLAN tags in Ethernet frames, and they evaluated the VLAN routing method using TCP/IP [6].

We investigated a method to assign VLAN in various topologies [7], and proposed a switch-tagged VLAN routing method for PC clusters the system software of which does not treat VLAN technology by performing VLAN tagging at a switch [5] and evaluated our system on a 32-host PC cluster. We used the switch-tagged VLAN routing method as the baseline of this work, and details are discussed in the following section.

There have been a number of previous studies on cycle-free routing algorithms [8][9][10] and their implementation on Ethernet by statically registering the MAC addresses of hosts without VLAN technology. However, it is difficult for Ethernet without VLAN that does not express channel dependency of the routing to manage frames when a broadcast storm occurs, because the spanning-tree protocol (STP) that imposes poor performance scalability is disabled.

Another study realized a hyper-crossbar network in a large-scale PC cluster system using the PM/Ethernet-HXB communication library. PM/Ethernet-HXB played a role in routing and it uses no VLANs [11].

III. Implementation of VLAN Routing Method

A. PC cluster

In this study, we implemented a switch-tagged VLAN method on a 225-host PC cluster called SuperNova cluster system at Doshisha University, Japan. SuperNova cluster is a large-scale computing cluster, which ranked 93rd in the TOP500 supercomputer ranking in 2003 [1]. It used 1-switch (non-blocking crossbar) of Force10 Networks E1200. At present, eight Dell PowerConnect6248 (Gigabit Ethernet $\times$ 48 port, non-blocking) switches are used to connect 225 hosts, as shown in Figure 2.

Table I lists the specifications of each host in the PC cluster, and the topologies considered in the evaluation is described in Section IV.

B. Applying Switch-Tagged VLAN Routing Method

To achieve various topologies in the PC cluster, we employed the switch-tagged VLAN routing method [5]. To simplify the system update without modifying the system software of the PC cluster, the VLAN tag is added to the frame at switches [5]. Each host creates VLAN interfaces that have different local network addresses on a physical interface,

Fig. 2. Overview of SuperNova cluster

TABLE I
SPECIFICATIONS OF EACH HOST

CPU	AMD Opteron 1.8GHz × 2
Chipset	AMD 8131+8111
Memory	PC2700 Registered ECC 2GB
OS	Debian GNU/Linux 4.0
Kernel	2.6.18-4-amd64
MPICH	1.2.7p1

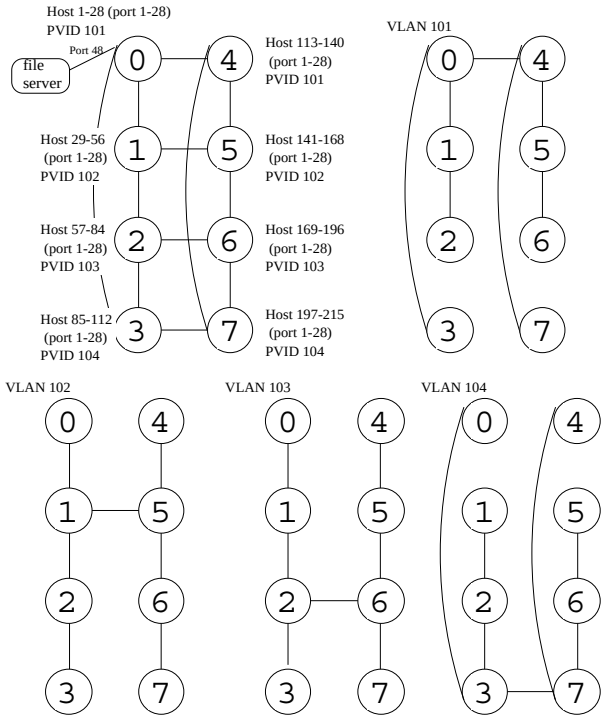

Fig. 3. Example of switch-tagged VLAN routing

so that a switch learns the MAC addresses of the hosts by broadcast.

1) Switch-tagged Routing Method: In the switch-tagged VLAN routing method, all paths from a host belong to a single VLAN regardless of the destination host. Both VLAN tagging and untagging operations are performed at each switch port connected to a host according to the following configuration.

- Set PVID of each port to the ID of the VLAN that is used by the connected host when sending frames.
- Register each port as an "untagged" member of all VLANs in the whole network.

A source host transmits a normal (untagged) frame in the usual way by specifying the IP or MAC address of the destination host. When a frame from a host enters the port of a switch, it is tagged with the PVID of the port and is regarded as a frame belonging to the VLAN indicated by the ID tag number. The frame is then transferred by the L2 Ethernet mechanism as well as normal VLAN-based routing. Finally, the frame is untagged when it leaves a port connected to the destination host, because the port is an "untagged" member of the VLAN. The destination host thus receives the usual untagged frame, which can be handled easily by the communication library.

In this way, all hosts can communicate with each other on various topologies, even if a VLAN tagging operation is not supported by the communication library on the hosts.

An example of this switch-tagged routing is shown in Figure 3. Each circle in the figure represents a switch. In Figure 3, frames from hosts 1-28 are tagged with the VLAN ID tag #101 in the port of switch 0 and forwarded along VLAN #101 to their destinations. The VLAN ID tag #101 is untagged when frames leave the switch connecting to the destination hosts.

A host does not process VLAN tags, and communication between all the hosts is achieved in various topologies that include loops.

2) Management of MAC addresses: The switch-tagged routing method introduces problems with regard to MAC address self-learning on switches. Ethernet switches usually learn unknown MAC addresses when receiving frames. However, as the VLAN-based routing method can employ multiple VLANs, there are many cases where a path from host A to B and that from B to A use different VLANs. In such cases, the intermediate switches of both paths cannot learn the destination MAC address, because MAC address self-learning is performed on each VLAN independently. The original VLAN-based routing is also subject to this problem. Thus, in this implementation, we proposed and used MAC-address self-learning according to the following procedure.

1) Create virtual interfaces that correspond to all VLANs used in each host. In the case of the Linux operating system, the vconfig command can be performed to make the virtual interface. For example, in Figure 3, as vlan #101-104 are used, interfaces eth0.101-eth0.104

are created on each host.

2) Give a unique network (IP) address to each virtual interface so that the virtual interfaces use IP addresses that have different segments.
3) Broadcast an ICMP or an UDP message once for every virtual interface.

At step (3), at each host, the ping command (ICMP echo req.) can be employed in each VLAN segment, and the MAC address of the source host is registered at the address table on each switch. Notice that when the source host uses the ping command, it cannot receive the pong (ICMP echo reply) corresponding to the ping. The VLAN tag of the frame will be removed when the ping is output from the switch to the destination host. For example, if a source host sends the frame from interface eth0.101 (VLAN #101), the destination host receives it from the eth0 physical interface and discards it.

As the procedure is used only for learning the MAC addresses of hosts, parallel computing communication is unaffected by these virtual interfaces on the hosts.

As hosts are rarely inserted or removed in a PC cluster, the aging time of switches, which defines the retention time of the MAC address table, should be large or infinite to maintain the table configuration.

IV. EVALUATION

In this section, we describe the evaluation results of various topologies and routing achieved in the PC cluster with Ethernet. We implemented various topologies and routing using the advanced switch-tagged routing method introduced in the previous section.

First, we evaluated a simple tree topology, as shown in Figure 3. Then, we also evaluated a 4×2 torus (deadlock-free routing)(3-bit hypercube), a 4×2 completely connected network (deadlock-free routing), a 8×1 ring, and a 4×2 mesh (deadlock-free routing), which are also shown in Figure 3.

In each topology, we varied the number of links between switches from one to six using link aggregation. In the case of the torus, we used four VLANs and adopted dimension order routing. In all topologies examined, the IEEE 802.3x link level flow control was enabled at each switch and the number of links between switch and host was one.

The LINPACK and NAS parallel benchmarks are performed by MPICH 1.2.7.p1 [12] for interprocess communication with IP packets.

A. LINPACK

Figure 4 shows the results of the High-Performance LINPACK Benchmark (HPL)[13] in a PC cluster. In Figure 3, "Tree" represents the simple tree topology of VLAN #101, "Compl" represents a completely connected network, "Torus" represents the 4×2 torus, "Mesh" represents the 4×2 mesh, and "Ring" represents the 8×1 ring. In addition, the figure in parentheses is the number of the links between switches. HPL is one of the implementations of Highly Parallel Computing of LINPACK Benchmark [14], and this class requires accuracy of the numerical solution.

We compiled MPICH using pgcc/pgf 7.1 with the options -fastsse -tp k8-64. We used GotoBLAS1.22 compiled using gcc 4.2.2/pgf 7.1 for the operation library. In HPL, it is possible to tune the parameters best suited to the features of the system [15]. The main parameters of HPL used for this measurement are shown in Table II.

TABLE II
THE MAIN PARAMETERS OF HPL

N	180000
NB	240
(P, Q)	(18, 25)
BCAST	increasing-1ring(modified)

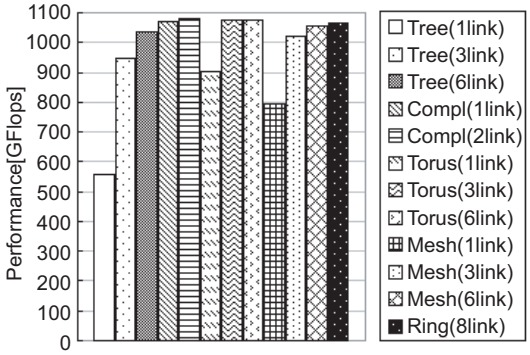

Fig. 4. LINPACK results

As verification of the results obtained from the calculation failed in the case of Tree(1link), Compl(2link), Mesh(1link), Mesh(6link), and Ring(8link), these results are references. As the SuperNova cluster is still used by scientists utilizing huge amounts of computation power, it could only be borrowed for a short time, and thus it was not possible to evaluate HPL repeatedly for every topology as long as the verification succeeded.

As shown in Figure 5, torus topologies with six links between switches outperformed the mesh and tree topologies in the some applications, while Figure 4 shows that the torus topology with six links had similar performance to the tree with six links, torus and mesh with three links. Although the torus topology with six links is needed to improve the performance of applications in the NAS parallel benchmarks, the impact of topologies on the HPL was not large.

The peak performance of the SuperNova cluster is 1.620 TFlops, and the completely connected topologies with two

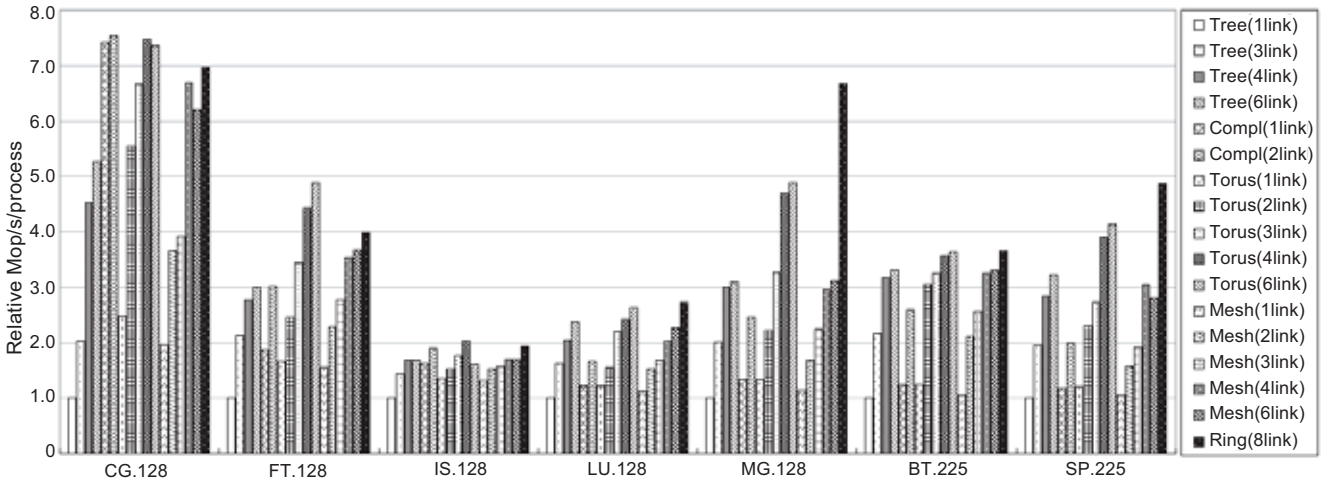

Fig. 5. NAS Parallel Benchmarks results

links between switches achieved the best performance (1.081 TFlops) in this evaluation. Topologies achieved by the proposed methodology showed improvements in performance of up to 93% relative to the tree topology with a single link between switches.

The completely connected topology with two links achieved only up to 4% performance improvement relative to the tree topology with six links, indicating that link aggregation is quite effective to improve the performance of the HPL. The tree topology with six links used 42 links in total, while the completely connected topology with a single link used 28 links. Thus, the proposed methodology allowed efficient use of the network resources (Table III).

TABLE III
COMPARISON OF A PERFORMANCE PER LINK BETWEEN SWITCHES

Topology	Tree(6link)	Compl(1link)
Links between switches	42	28
Performance(GFlops)	24.6	38.2

Table IV shows a comparison of performance using Force10 Networks E1200 switch in 2003 [16]. E1200 has a back plane of 1.44 Tbps and is a high-performance switch in which non-blocking communication is possible between a maximum of 336 nodes.

As shown in Table IV, the highest performance (1.081 TFlops) obtained in this experiment with 225 hosts was not so far from that with an ideal 1-switch (full crossbar) connecting 256 hosts. Moreover, such performance can be measured with the combination of eight commodity switches far more cheaply than with E1200 using 62 less CPUs than the number used in 2003. Furthermore, in this evaluation, a high effective performance rate (Rmax/Rpeak) of 66.7% was achieved, which was higher than that using E1200 in 2003.

However, it is necessary to take not only the improvement of the topologies but the influence of differences in the versions of the compiler and operation library or the cables used into consideration in Table IV. For example, an enhanced category 6 (CAT6E) cable provides performance of up to 500 MHz, five times that of enhanced category 5 (CAT5E). In the TOP500 Supercomputer ranking [1] announced in June 2008, the highest effective performance rate (Rmax/Rpeak) of a PC cluster using Gigabit Ethernet was 73.6%, followed by 63.8%. Although it is difficult to obtain results with an execution performance rate exceeding 60% in a large-scale PC cluster using Gigabit Ethernet, we obtained a value of 66.7%, which far exceeds 60%, in this study.

TABLE IV
COMPARISON WITH USING FORCE10 E1200 SWITCH

Switch	Powerconnect6248	E1200
Number of Processors	450	512
Cable	CAT6E	CAT5E
Number of Switches	8	1
Rmax(TFlops)	1.081	1.169
Rpeak(TFlops)	1.620	1.843
Rmax/Rpeak(%)	66.7	63.4
Nmax	180000	220000

B. NAS Parallel Benchmarks

We measured the application performance of NAS Parallel Benchmarks 3.2 [17] in each topology realized by the VLAN-based routing method. We let problem size of each benchmark be Class C, and the number of execution processes of each application can perform a maximum of 128 or 225 calculations in 225 hosts. CG, FT, IS, LU, MG, BT, and SP benchmarks were used for the application. Code was compiled using gcc 3.3.6/g77 3.4.6 with the -O3 option. The measured

benchmark performance (Mop/s/process) in each topology is shown in Figure 5, normalized relative to the performance of Tree(1link). The topology notation is the same as that in Figure 4. The notation "CG.128" used in the application shows that the number of execution processes in the CG application was 128. In each application, the improvement of high effective performance value in each topology that includes loops was improved by about 94%-650% in comparison with Tree(1link). Changing the topology to Compl(2link) from Tree(1link), although the results of Tree(6link) showed an improvement in performance of about 420% as compared with those of Tree(1link), a further improvement in performances of 230% was attained using the CG method. Moreover, in other applications, the results of the topology using the proposed method showed a minimum improvement in performance of about 40% in comparison with the results of Tree(6link).

In the case of the NAS Parallel Benchmarks, in each application, tuning of the network topology was efficient to improve the performance, and these evaluation results showed that good performance was acquired in comparison with the case of a simple tree topology. In all applications, as the topology tuned using VLANs acquired a high performance value, the impact of the VLAN routing method on Ethernet was large.

V. CONCLUSIONS

PC clusters with Ethernet are now the main architecture of parallel computers in the field of HPC. However, there have been few reports of the evaluation of topologies and routings on real PC clusters. In this study, we investigated the impact of topology and link aggregation on a 225-host PC cluster with Gigabit Ethernet. Ethernet topology that allows loops and routing can be implemented by the VLAN routing method without creating broadcast storms.

To simplify the system configuration without modifying the system software of the PC cluster, the VLAN tag is added to the frames at switches in our topology implementation. Each host creates VLAN interfaces that have different local network addresses on a physical interface, so that the switches can learn the MAC addresses of hosts by broadcast.

The results showed that the performance of the eight-switch network was comparable with that of an ideal 1-switch (full crossbar) network in execution of High-Performance LINPACK Benchmark (HPL). On the other hand, results of evaluations using NAS Parallel Benchmarks indicated performance improvements of up to about 650% by the topologies achieved by the proposed methodology relative to the simple tree topology. These results indicated that topology and link aggregation have marked impacts and that commodity switches can be used instead of expensive and high functional switches.

ACKNOWLEDGMENT

This work was partially supported by JST CREST (ULP-HPC: Ultra Low-Power, High-Performance Computing via Modeling and Optimization of Next Generation HPC Technologies).

REFERENCES

[1] Top 500 Supercomputer Sites, http://www.top500.org/.

[2] T. Kudoh, H. Tezuka, M. Matsuda, Y. Kodama, O. Tatebe, and S. Sekiguchi, "VLAN-based Routing: Multi-path L2 Ethernet Network for HPC Clusters," in *Proc. of 2004 IEEE International Conference on Cluster Computing (Cluster2004)*, Sep. 2004.

[3] S. Sharma, K. Gopalan, S. Nanda, and T. cker Chiueh, "Viking: A Multi-Spanning-Tree Ethernet Architecture for Metropolitan Area and Cluster Networks," in *Infocom*, Mar. 2004, pp. 2283–2294.

[4] S. Urushidani, S. Abe, K. Fukuda, J. Matsukata, Y. Ji, M. Koibuchi, and S. Yamada, " Architectural Design of Next-generation Science Information Network," *IEICE Transaction*, vol. E90-B, no. 5, pp. 1061–1070, May 2007.

[5] T. Otsuka, M. Koibuchi, T. Kudoh, and H. Amano, "A Switch-tagged VLAN Routing Methodology for PC Clusters with Ethernet," in *Proc. of the 2006 International Conference on Parallel Processing (ICPP-06)*, Aug. 2006, pp. 479–486.

[6] S. Miura, T. Okamoto, T. Boku, M. Sato, and D. Takahashi, "Low-cost High-bandwidth Tree Network for PC Clusters based on Tagged-VLAN Technology," in *Proc. of the 8th International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN 2005)*, Dec. 2005, pp. 84–93.

[7] T. Otsuka, M. Koibuchi, A. Jouraku, and H. Amano, "VLAN-based Minimal Paths in PC Cluster with Ethernet on Mesh and Torus," in *Proc. of the 2005 International Conference on Parallel Processing (ICPP-05)*, Jun. 2005, pp. 567–576.

[8] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks: an engineering approach*. Morgan Kaufmann, 2002.

[9] F. D. Pellegrini, D. Starobinski, M. G. Karpovsky, and L. B. Levitin, "Scalable Cycle-Breaking Algorithms for Gigabit Ethernet Backbones," in *Proc. of 23th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2004)*, Mar. 2004, pp. 2175–2184.

[10] A. Jouraku, M. Koibuchi, and H. Amano, " An Effective Design of Deadlock-Free Routing Algorithms Based on 2-D Turn Model for Irregular Networks," *IEEE Transaction on Parallel and Distributed Systems*, vol. 18, no. 3, pp. 320–333, Mar. 2007.

[11] T. Boku, M. Sato, A. Ukawa, D. Takahashi, S. Sumimoto, K. Kumon, T. Moriyama, and M. Shimizu, "PACS-CS: A Large-Scale Bandwidth-Aware PC Cluster for Scientific Computations," May 2006, pp. 233–240.

[12] MPICH, http://www-unix.mcs.anl.gov/mpi/mpich1/.

[13] HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers, http://www.netlib.org/benchmark/hpl/.

[14] The Linpack Benchmark, http://www.netlib.org/linpack/.

[15] T. Sasou and S. Matsuoka, "Performance tuning high-performance linpack (hpl)," *IPSJ SIG Notes*, vol. 91, no. 22, pp. 125–130, 8 2002.

[16] T. Hiroyasu, M. Miki, and H. Arakuta, "Construction of tera flops pc cluster system and evaluation of its performance using benchmark," *The science and engineering review of Doshisha University*, vol. 45, no. 4, pp. 187–198, 1 2005.

[17] The NAS Parallel Benchmarks, http://www.nas.nasa.gov/Software/NPB/.