Electronic Preprint for Journal of Information Processing Vol.22 No.3

Regular Paper

A Portable Method for Improving Available Bandwidth of PC Cluster

Takafumi Fukunaga[1,a)]

Received: September 16, 2013, Accepted: February 14, 2014

Abstract: The Installation and maintenance costs of dedicated interconnection networks for PC cluster are still expensive, and they tend to increase the specializations and complexities in comparison with Ethernet due to dedicated protocols and libraries to draw their hardware performance. The porting works from Ethernet system to dedicated system need lots of time and manpower. This paper proposes a simple and portable method, PMCME, that improves the PC cluster performance only by loading the proposed module. The existing systems can easily and cheaply introduce PMCME. The basic idea is that the performance of PC clusters increases by improving the total bandwidth of the streams running concurrently on each node even if the bandwidth of each stream does not increase. PMCME performs better than IEEE802.3ad (LACP) without the LACP supported switches. LACP performance is influenced by the network parameters such as IP addresses and MAC addresses because it uses them as hash keys for distribution policy. On the contrary, PMCME shows the stable performance regardless of them.

Keywords: IEEE802.3ad (LACP), multi ports NIC, multi cores, bonding module

1. Introduction

Ethernet is widely employed to interconnect PC cluster nodes because of its high cost-effectiveness. However, even if 10G Ethernet is employed, the communication performance cannot catch up with sharp increase in computing performance which increases in proportion to the number of cores. This paper proposes the portable method to improve the total bandwidth available at each node. The proposed method (hereafter called PMCME: Portable method using both Multi-core and Multiple Ethernet ports) is easily installed to each node by loading proposed module and attaching several NICs to node. The existing applications receive the benefit of PMCME without any additional effort. The existing NIC and its drivers are also available in PMCME as before. PMCME has high portability. In particular when PC cluster employs multiple switches, PMCME shows its full abilities because it does not require an interconnection (hereafter, called link) between switches as described below.

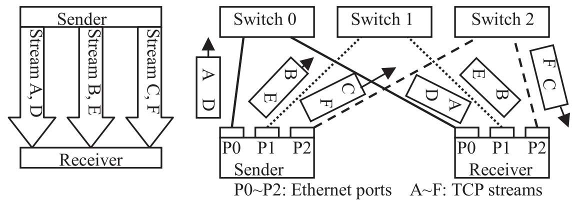
The basic idea of PMCME is to improve the total bandwidth of streams running concurrently on each node by distributing the high bandwidth streams among multiple Ethernet ports almost equally as shown in **Fig. 1** (a). In the case of six streams and three Ethernet ports, each Ethernet port (hereafter, called Eport) of the sender transmits two streams and each Eport of the receiver receives two streams. Figure 1 (b) shows the communication form in the case above. P0 Eport of the sender communicates only with P0 Eport of the receiver through Switch 0 according to correspondence information table prepared in advance. The packets from P0 of the sender never reach P1 or P2 of the receiver. P1 and P2 also communicate in the same way as P0. The links between switches, which increase the latency and can cause a communication bottleneck, are removed. The proposed method is implemented by modifying Bonding module in Linux. The number of modified code in the existing module is slightly 100 lines or so.

The target of PMCME is Ethernet family including 10G. This paper evaluates using Gigabit Ethernet due to lack of 10G hardware. The evaluation on 10G Ethernet will be conducted after getting hardware.

2. Related Works

Powerful networks such as Myrinet [1], Infiniband [2], QsNet [3], and RHiNET [4] employ specialized protocols and libraries in order to obtain full hardware performance. They aim to improve not only the bandwidth but also the latency by making the best use of dedicated hardware. The proposed PMCME aims to improve the total bandwidth available of each node using existing Ethernet hardware and TCP/IP protocol stack.

There are many previous works to improve communication performance on PC cluster with Ethernet. GAMMA [5], MultiEdge [6], EMP [7], GigaE PM [10], PM/Ethernet [15], PM/Ethernet-HXB [11], and MPI/QMP [9] only work on limited Ethernet NICs since they use modified NIC drivers. iWARP [8]



(a) Division of streams' paths (b) No link between switches

Fig. 1 Basic idea.

[1] Kumamoto Prefectural College of Technology, Kumamoto 869–1102, Japan

a) t-fukunaga@kumamoto-pct.ac.jp

needs expensive Ethernet NICs specially made for them. EMP, GigaE PM, and MPI/QMP require programmable NICs.

MultiEdge, iWARP, MPI/QMP, PM/Ethernet, and PM/Ethernet-HXB among them use bonding technique of multiple Eports. MultiEdge, iWARP, and PM/Ethernet distribute the packets of single stream to multiple Eports (paths) in a round-robin fashion for improving the bandwidth of each stream. PMCME differs from them in that it distributes the packets to multiple Eports (paths) in units of streams for improving the total bandwidth of streams concurrently running on each node. Multi-path communication of the previous works comes with the disadvantage of out-of-order arrivals of packets because the packets of each stream traverse different paths. Whereas PMCME causes less out-of-order arrivals because all packets of each stream are transmitted using single path. MPI/QMP has Mesh connections and PM/Ethernet-HXB has Hyper Crossbar connections. Main targets of them are the applications using frequent neighboring communications such as Lattice QCD code. PMCME are widely targeted at bandwidth sensitive applications.

RI2N/DRV increases the bandwidth between two nodes using two Eports at both the sender and receiver with two switches, so-called segment division. The information for the receiver is inserted between Ethernet header and IP header as new header at the sender. Ethernet frame is modified for RI2N/DRV. Also, because it is so hard to reorder the packets which alternately arrive at two Eports, the latency performance is lower than the round-robin distribution of the existing Bonding driver in Linux. RI2N/DRV also makes no mention of the performance using more than two Eports.

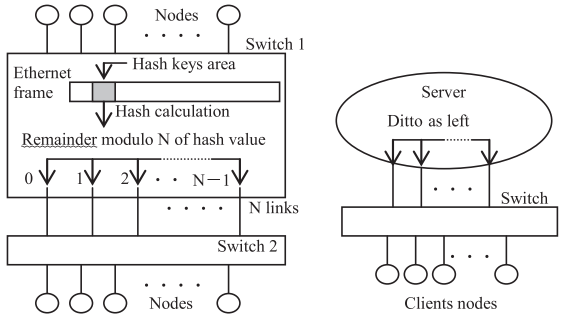
IEEE802.3ad (LACP) and round-robin distribution are more similar to PMCME because they improve the bandwidth without requiring the special hardware and protocols. They are compared with PMCME in Evaluation section.

3. Design of PMCME

This section begins by looking at how LACP and round-robin distribution work and explains the design of PMCME in order to make clear the differences between PMCME and the existing techniques (i.e., LACP and round-robin distribution). LACP and round-robin distribution are implemented in original Bonding module in Linux.

3.1 IEEE802.3ad (LACP) and Round-robin Distribution

LACP provides functions to control bundling of several physical Eports together to form a single logical channel for both communication load-balancing and fault tolerance. Load balancing provides an increase in communication bandwidth both between switches and between a server and a switch. **Figure 2** (a) shows the outline sketch of LACP between switches. The packets, which arrived in switch 1 from nodes in the upper side, are distributed to multiple Eports in switch 1 so that switch 1 transmit the packets to switch 2 with multiple paths. The decision as to which Eport should be used for transmitting each packet depends on the value of hash key area of each packet. First, the hash value is calculated from this area's several values (e.g., destination and source MAC address) with an exclusive OR, etc. Then the value is divided by the number of Eports and the remainder is used to designate Eport number. Eventually, LACP distributes the packets to multiple Eports in units of streams. Figure 2 (b) shows the outline sketch of LACP between server and switch. The distribution of packets is performed in the server in the same way as the former. In this case, LACP improves the bandwidth between the server and switch, that is to say, LACP contributes to a client/server system. Thus, LACP is originally suitable to links between switches or client/server systems.



(a) LACP between switches (b) LACP between server and switch

Fig. 2 Outline of LACP.

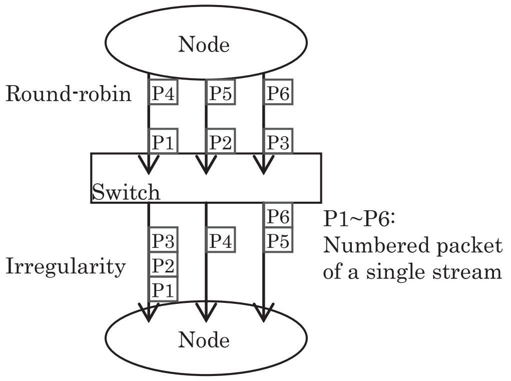


Fig. 3 Outline of round-robin distribution.

Meanwhile, the sender implementing round-robin distribution transmits the packets of a single stream using multiple Eports in a round-robin fashion. **Figure 3** shows the example of packets flow when the sender and receiver are respectively connected with three Eports. The sender accurately transmits the packets P1 to P6 in a round-robin fashion. However, the switch forwards their packets to multiple Eports irregularly because all three Eports of the receiver have the same MAC addresses due to specification of round-robin distribution. The packets flow between the switch and receiver in this figure is just one example. The switch transmits each packet from the switch Eport that learns most recently the receiver's MAC address by ACK packet from the receiver. Round-robin distribution can achieve a bandwidth of over 1 Gbps even when sending a single stream on Gigabit Ethernet. However, a large number of SACK packets are transmitted from the receiver to the sender in order to inform out-of-order arrivals of packets according to TCP standard. The frequent out-of-order arrivals of packets affect the communication performance and waste CPU power at both sides.

The reason why round-robin distribution causes frequent out-of-order arrivals is explained with **Fig. 4**. The packets of a sin-

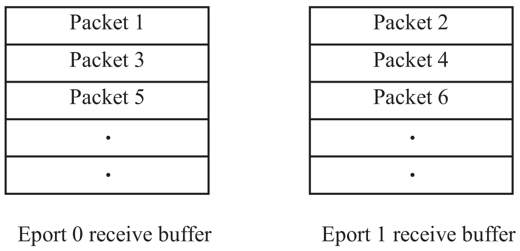


Fig. 4 Receiver NIC ring buffer.

gle stream separately reach the multiple Eports at the receiver, that is to say, the packets received by each Eport are not continuous. Note that the condition in Fig. 4 is just one example. Most of high-speed NICs handling heavy traffic process several packets at a time, either on the card itself or in an in-memory DMA ring, either by the drivers based on polling or by receive interrupt moderation. In this example, Packet 1, Packet 3, and Packet 5 might be processed collectively before Packet 2, Packet 4, and Packet 6. Then out-of-order arises. This bundle processing can take a substantial amount of load off the processor. However, in round-robin distribution, this way causes a large amount of overhead to process out-of-order arrivals of packets. Naturally, the frequency of out-of-order is higher when the bandwidth is higher.

3.2 Design of PMCME

The implementation of PMCME is achieved only by modifying the program code relative to the transmission. There is no additional code in the receiving function. **Figure 5** denotes the sending side equipped with 4 Ethernet ports, on which 8 streams created by 4 applications are running. Whether each stream created by those applications is high bandwidth or not is judged by the criterion described later, but it should be noted that "high bandwidth stream" in this paper indicates the stream which has much data left in its send buffer. Let us assume that Stream A, C, E, F, G, and H underlined in Fig. 5 are high bandwidth streams. Since specific Eport is allocated to each of those high bandwidth streams in a round-robin fashion in units of streams, all packets of each stream are transmitted only by using allocated single Eport and the numbers of streams in each Eport are approximately equal, namely, equal load-balancing is achieved. Whereas the packets of the streams not judged as high bandwidth, Stream B and D in Fig. 5, are transmitted using all Eports in a round-robin fashion in units of packets. This scheme is the simple way to avoid flooding because each Eport of the switch is periodically able to relearn MAC addresses by ACK packets from the receiver.

The decision as to whether a stream is high bandwidth or not, criterion for decision in Fig. 5, is determined by the amount of data left on its send buffer which have not completed transmission yet. The term "complete transmission" above means having received an ACK from the receiver. **Figure 6** shows a TCP socket send buffer and related members. The snd_una, snd_nxt, and write_seq member in the figure respectively indicate the first sequence waiting for an ACK, the next sequence to be sent, and tail sequence of data copied from the user application. Criterion for judging whether the packet belongs to the high bandwidth uses the value calculated by the subtraction of snd_una from write_seq (shaded area in Fig. 6). If that value is larger than a certain threshold, the packet is regarded as a high bandwidth stream's packet. This condition is based on the notion that the amount of data left in send buffer is directly related to the level of necessity to transmit the packets.

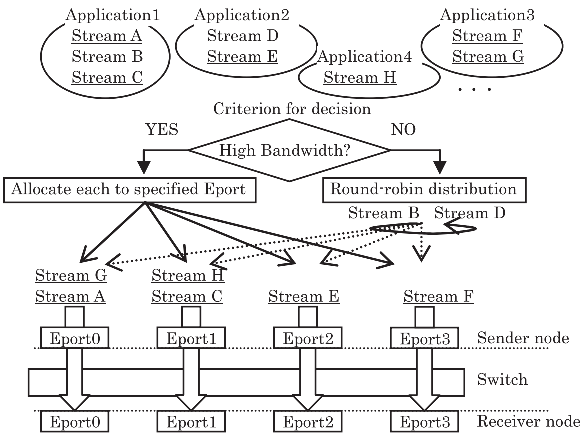


Fig. 5 Outline of PMCME.

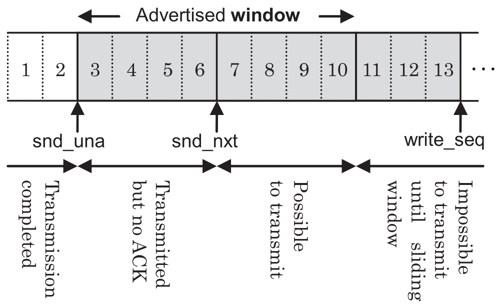


Fig. 6 Members of TCP send buffer.

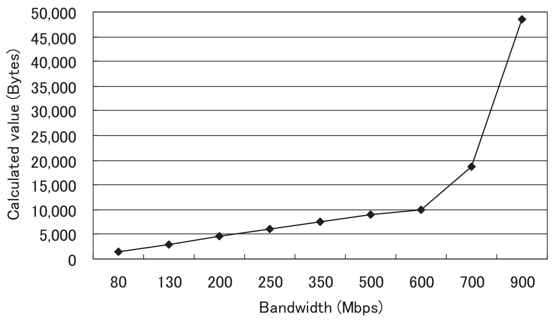


Fig. 7 Relation between calculated value and bandwidth.

The relation between the bandwidth of single stream and the value calculated by the subtraction of snd_una from write_seq are shown in **Fig. 7** to determine the value of threshold. As shown in later Section 5.1, existing round-robin distribution, which is second highest system in bandwidth performance next to PMCME, achieve a bandwidth of 2.0 Gbps when running 8 processes on 8-core node whose condition is often used in parallel processing. It amounts to this, that existing round-robin distribution is capable to handle around a bandwidth of 250 Mbps per process under that condition. Since the situation of send buffer varies depending on network conditions such as heavy traffic, the number of process, whether burst or ping-pong transfer, it is difficult to determine the value of threshold. This time the value of threshold is determined so that PMCME regards the streams of bandwidth over 250 Mbps as the targets to handle (i.e., high bandwidth streams), though that term is valid under no heavy traffic condition. As show in Fig. 7, the calculated value reaches to about 6,000 at a bandwidth of 250 Mbps, so 7,000 is adopted as the value of threshold to choose the bandwidth over 250 Mbps. Note that, under heavy traffic condition, the streams of bandwidth under 250 Mbps are easily regarded as high bandwidth streams. The main purpose of threshold is to choose the streams being apt to be delayed. In other world, in parallel processing with heavy traffic, it is to except very slow streams from the targets. It has been confirmed that ACK packets stream with no data and ping-pong transfer composed of packets of under MTU size (1,500) are not chosen by the value of 7,000 even under heavy traffic condition.

In addition, the destination and source MAC addresses of Ethernet header are modified to achieve communication load-balancing in the receiver side as shown in Fig. 1 (b). The Eports of the sender is associated with the Eports of the receiver in a one-to-one relationship by MAC table in which the relationships are defined. According to those relationships, the destination MAC addresses are replaced with the addresses obtained from MAC table just before transmitting the packet so that the packets are received by multiple associated Eports at the receiver.

The correspondence between each stream and Eport is managed by two tables, one of which holds the Eport number to transmit the stream, another is used for removing old entry unused in former table. Since the number of entries of former table per Eport is 100 at present, a hundred streams can be allocated to single Eport. The value can be increased by hand, if necessary. Upward scalability is guaranteed.

The first difference in bonding technique between LACP and PMCME is how to distribute the streams to multiple Eports. Although both are the same in that the streams are distributed in units of streams, LACP distributes each stream according to network parameters such as MAC address, IP address, TCP port number, etc., whereas PMCME allocates selected high bandwidth streams to multiple Eports in a round-robin fashion. If the assumption is made that the bandwidths of selected streams are almost same, PMCME achieves almost same load-balancing for multiple Eports. Even if not, PMCME can make good use of all of Eports. LACP does not always utilize all of Eport as shown in Section 5. The second difference is PMCME does not require the links between the switches because of a one-to-one correspondence between sender's Eports and receiver's Eports as described in Section 1.

The difference between round-robin distribution and PMCME is the unit of distribution. Round-robin distribution distributes the packets in units of packets so that a single stream is divided to multiple paths. Naturally, that condition causes out-of-order arrivals of packets as described in the foregoing subsection. The packets are distributed to different paths one after another, the out-of-order ratio is very high as shown in Section 5. In contrast, PMCME causes little out-of-order arrivals since all packets of a single stream are sent through a single path.

4. Implementation

The implementation of PMCME has been achieved by modifying the existing Bonding module in Linux that originally provides communication load-balancing and fault tolerance function. Once having loaded this module to system, a logical Eport is made, whose default interface name is bond0, and it is called master. The physical Eports are associated with the master by tool beforehand in order to transmit the packets actually. These associated physical Eports are called slaves. In sending process, the protocol stacks deliver the packets to the master, that is to say, the transmitting function in Bonding module. Next, that function calls the transmitting function in actual driver of slave selected according to the policy such as IEEE802.3ad and round-robin distribution. PMCME is added this module as new policy. **Figure 8** shows PMCME mechanism. The mechanism requires three tables. First table is required to register Eport number allocated to specified stream so that the following packets of the same stream can learn already allocated Eport number. This table is looked up by hash value calculated from TCP source port number. This table is depicted as Port_Info in Fig. 8 and table contents are denoted in **Table 1**. Second table maintains Port_Info table, that is to say, the entries which have become unnecessary (old) are re-

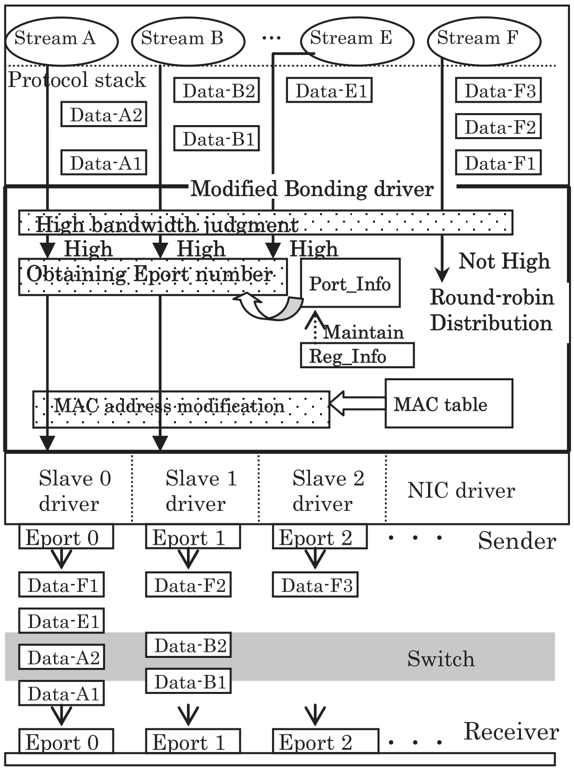


Fig. 8 Implementation of PMCME.

Table 1 Port_Info table.

	Description
PortNO	The Eport number to transmit the packet. All slave ports are numbered from 0 increasing 1.
NOperPort	Consecutive number of stream allocated to each Eport. The MAX value is 99 at present. The value go back to 0 when being over MAX (99).

moved by the second table. This table is a two-dimensional array indexed by Eport number and consecutive number in each Eport. Concurrently with registration into Port_Info table, its hash value is registered into this second table to remove its entry when becoming old (unused). Since the number of entries (streams) that can be registered in each Eport is limited to 100, the old (probably unused) entry registered more than 100 entries ago is overwritten with a new one. Then Port_Info entry corresponding to overwritten one need to be removed to avoid that a new stream use its old entry by mistake. Because TCP source port number is used to calculate Port_Info's hash value. The TCP source port number is reused by a new stream after some time. Even if the old entries are in use (i.e., long-life streams), there is no matter, because they will be registered again (i.e., relocation) if needed. This action has another good effect. The relocation softens a lack in load-balance caused by the differences in streams' life-time. This second table is depicted as Reg_Info in Fig. 8. The third table is called MAC table which is required to achieve load-balancing at the receiver as described in Section 3 and later in this section.

In Fig. 8, each packet is judged using high bandwidth criterion described in the foregoing section. Next, the packet is evaluated whether its TCP port number has already been registered to Port_Info table. If not, the specified Eport selected in a round-robin fashion is registered Port_Info table. On the other hand, if the table already has a corresponding entry to the packet, the packet and following same stream's packets are transmitted from Eport designated by its entry. Because the selection of Eports is conducted in a round-robin fashion, the communication load of each Eport is almost balanced. The old (probably unused) entries are removed from Port_Info table using Reg_Info array as described above.

Just before transmitting the packets, both the destination and source MAC addresses are modified in order to achieve communication load-balancing at the receiver. Let X be the Eport number. The packets transmitted from the slave Eport X at the sender always reach the slave Eport X at the receiver without exception as shown in Fig. 8. Although the original Bonding module overwrites all the slave Eports' MAC addresses with the first slave's one in LACP and round-robin distribution mode, PMCME not, namely each slave Eport keeps its original MAC address. The association between the Eports of both sides is defined in MAC table as shown in **Table 2**. The entries are inserted by the tool just after loading of the module. Destination node identifier in Table 2 is MAC address of the master of the destination node and it is used as a search-key1. Destination Eport identifier is the Eport's consecutive number in the destination node, and it is used as a search-key2. New destination MAC address is the address to replace the original destination MAC address in Ethernet header. The combination of search-key1 and 2 is used to look up the entry. The value of search-key1 can be obtained from outgoing packet. The value of search-key2 is the same as Eport number used at the sender because of a one-to-one correspondence between sender Eports and receiver Eports. When the entry is found, the destination MAC address of the packet is replaced with new destination MAC address in the entry except for the packets transmitted from the first slave Eport, that of which originally indicates the first slave's Eport in the receiver. The source MAC address of the packet, which is the same as the first slave's one despite slave number in original specifications, is replaced with the actual slave MAC address.

Table 2 MAC table.

	Description
Destination node identifier (Search-key1)	MAC address of master of the destination node.
Destination Eport identifier (Search-key2)	Eport's consecutive number of the destination node.
New MAC address	Destination MAC address to newly replace.

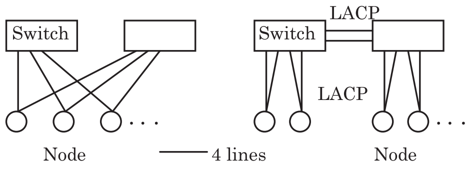
5. Evaluation

First, this section evaluates the basic communication performance, one-to-one, one-to-many, many-to-one, many-to-many bandwidth, and latency. Second, the performance of parallel processing are evaluated using NPB benchmarks. The subsections which measure above performances only give the results of evaluated systems. The comparisons between evaluated systems are discussed in the following subsections.

The four systems evaluated are follows:

1. Proposed method (labeled as PMCME).
2. IEEE802.3ad (LACP) of the existing Bonding module (labeled as LACP).
3. Round-robin distribution of the existing Bonding module (labeled as Round).
4. Normal transmission using one Eport (labeled as 1Port).

The benchmarks run on a Gigabit Ethernet cluster of 8 nodes. Each node is equipped with 2 Quad-Core Opteron 2.4 GHz CPUs (8 cores per node), 16-Gbyte main memory and 8 Gigabit Ethernet ports (Intel PRO/1000 $*$ 8). The used operating system is Linux 2.6.24. Two switches, NETGEAR GSM7248R 48 Eports and GS748TP 48 Eports both supporting LACP, are used to connect 8 nodes, each with 8 communication lines (UTP cables). Although different models of switches are used due to lack of hardware, these two switches' basic performances are almost the same, so this disparity does not affect the results of the experiments. When using LACP function, both of switches and nodes employ source/destination MAC addresses based hash algorithm because of the only policy they support. In general, this policy is widely used. The connection forms of PMCME, LACP and Round-robin system are shown in **Fig. 9**. One full line in the figure denotes 4 communication lines, that is to say, each node is connected to switch using 8 communication lines (8 Eports). In



(a) PMCME, Round-robin (b) IEEE802.3ad (LACP)

Fig. 9 Connection form of evaluated systems.

LACP, the number of available Eports connecting two switches is 8 because of a limitation on the number of Eports available in the specification for switches. This is not only true of NETGEAR switches, but also for many vendors' switches [13], [14]. All nodes of 1Port system are connected to one switch (GSM7248R), each using 1 Eport.

5.1 Bandwidth

The user application level bandwidths are evaluated for above 4 systems using Netperf-1.2.7 with a frame size of 1,518 bytes. The following types of bandwidth are evaluated: unidirectional, bidirectional bandwidth between 2 nodes, one-to-many (from 1 node to 7 other nodes), many-to-one (from 7 nodes to 1 node), and many-to-many (from each of 8 nodes to 7 other nodes) total bandwidth. These bandwidths indicate the sum of bandwidths of all streams. PMCME, LACP, and Round-robin are evaluated with varying the number of processes per node as 1, 2, 4, 6, 8 processes (hereafter, 1p, 2p, etc.), because multi-core node generally executes multiple tasks concurrently to make good use of CPU power. **Figure 10** shows the unidirectional and bidirectional bandwidths. PMCME achieved the best performance of all. LACP does not improve in both performances because MAC address based hash cannot achieve load-balancing in one-to-one communication. In LACP, all of the packets between two nodes have the same destination and source MAC address because all slaves of the sender or receiver have the same MAC address.

Similarly, PMCME shows the best performance in all cases of one-to-many, many-to-one, and many-to-many bandwidths as shown **Figs. 11–13**. Especially, PMCME dramatically outperforms the others in many-to-one and many-to-many bandwidths. Many-to-many and many-to-one communication are essential when executing the parallel applications which use total exchange operations and gather operations. In one-to-many bandwidth, the performance of Round is close to PMCME. Because Round distributes the packets of a single stream to multiple slaves in a round-robin fashion at the sender, the load-balancing of the sender is achieved almost completely. Round-robin distribution is originally suitable for one-to-many communication such as client/server systems.

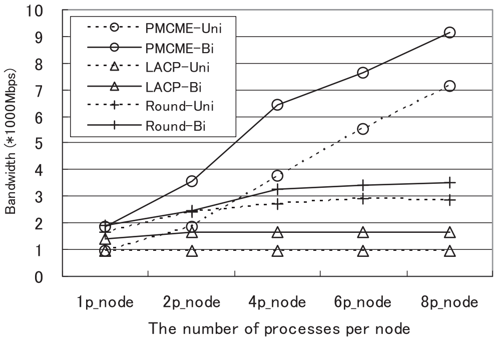


Fig. 10 Unidirectional and bidirectional bandwidth.

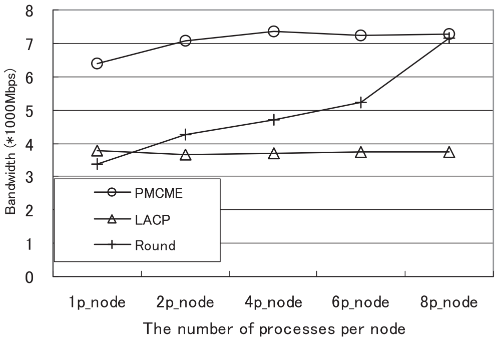


Fig. 11 One-to-many bandwidth.

5.2 Latency

This subsection evaluates the latency performance between two nodes with Netperf-1.2.7 benchmark. Experimental results are shown in **Fig. 14**. LACP (1 SW) and LACP (2 SW) respectively indicate the latency between two nodes connected to the same switch and that connected to separate switches interlinked by 8 Eports with LACP. Accordingly, the later includes the latency between the switches. PMCME achieves much better result than LACP and Round. Although the latency performance of PMCME (39.4 μs) is slightly lower than 1Port (37.4 μs) due to additional program codes, the slight degradation doesn't matter so much because the bandwidth sensitive applications hardly communicate on a ping-pong style. Another reason for better result in 1Port is because data cache miss rate of 1Port is slightly

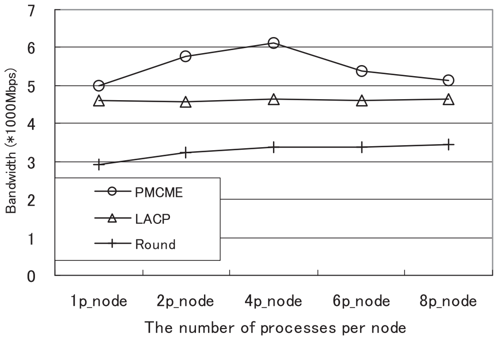


Fig. 12 Many-to-one bandwidth.

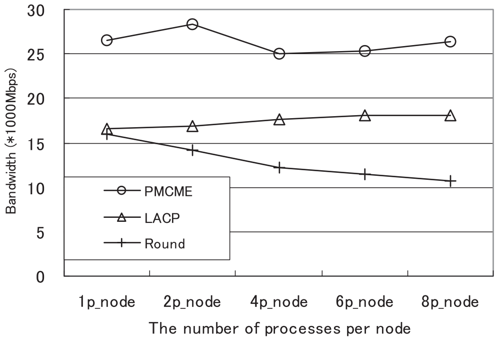


Fig. 13 Many-to-many bandwidth.

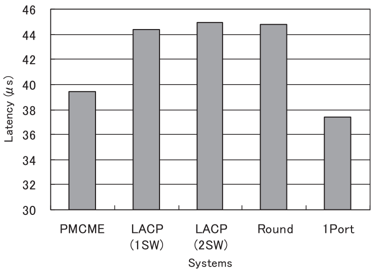


Fig. 14 Latency.

lower than that of PMCME. According to the results obtained by Oprofile-0.9.3, data cache miss rate of NIC driver of 1Port and PMCME are respectively, on average, 1.09% and 1.46% when executing one Netperf benchmark per node. 1Port always uses only one Eport both for transmitting and receiving, while PMCME usually uses two Eports, one for transmitting and another for receiving. Data cache miss rates tend to increase proportionally with number of used Eports. Round using 8 Eports shows 2.35% data cache miss rate, the highest rate of the 4 systems.

When considering the execution of bandwidth sensitive applications, PMCME has a good influence on the average latency because the packets waiting much longer in send buffer have priority in transmitting.

5.3 Parallel Processing Performance

This subsection shows the evaluation results of parallel processing performances using seven MPI benchmarks in NAS Parallel Processing Benchmarks (NPB) 3.3. class B. Although PMCME is extensively applicable to TCP/IP applications, MPI benchmarks in NPB are used in this subsection because they are used frequently by researchers to evaluate the parallel processing performance. In addition, they include massive data transfer benchmarks which are suitable to evaluate the influences by the increase of bandwidth. **Figure 15** shows the rate of performance increase in FT, LU, MG, CG, IS, BT, and SP as compared with serial execution. 1p, 4p, and 8p denote the number of processes running concurrently on each node.

PMCME achieves the best performance of all. In FT, MG, IS, BT, and SP, 8p cases of PMCME achieve a processing speed respectively 11.2, 13.8, 3.3, 18.2, and 10.6 times faster than serial execution, and 1.67, 1.56, 2.04, 1.23, and 1.26 times faster than the second highest ones. In LU and CG, 4p cases of PMCME similarly achieve 20.1 and 5.3 times faster than serial execution, and 1.33 and 1.25 times faster than the second highest ones. Especially, PMCME has the best effect in Is and FT, which respectively achieve 2.04 and 1.67 times faster than the second highest ones. Both benchmarks frequently execute total exchange operation called All-to-All collective communication. The reason for high increase in IS and FT performance is because they concurrently create a large number of high bandwidth streams, more than three hundred streams at 8p, which execute frequent All-to-All collective communications. Since PMCME particularly exhibits high performances in many-to-many communication as shown in foregoing subsection, All-to-All communications in IS and FT are accelerated.

In LU and CG, 4p are clearly faster than 8p. In LU, 1p of 4 systems show almost the same performance and the CPU execution times per each LU process reach over 90% of the whole execution times, that is to say, the idle times waiting the messages are slight. These results show the bottleneck of 1p is CPU power shortage, not communication capability shortage. This view is also made sure by the fact that the performance of 8 processes running on single node (8 cores) is almost the same as 1p in where single process are running on each node of 8 node cluster communicating with each other. Since a bottleneck is shifted from CPU power to communication capability at 4p, CPU execution times decrease by 55% to 35% at that time. Then, PMCME achieves distinguished increase in performance because of improvements of communication capabilities. There are two reasons why the LU performance of 8p is less than 4p. First, the number of transmitted and received packets at 8p is approximately 1.7 times larger than that at 4p. Second, because the number of streams created by LU per node is small, 5 at 4p and 10 at 8p, the numbers of streams allocated to each Eport are unbalanced compared to FT and IS, which create more than a hundred streams.

Since CG conducts the communication patterns of butterfly structure which is equivalent to All-reduce, the increase of processes cause the increase of butterfly structure stages, that is to say, the amounts of communication times increase. Also, in each butterfly structure stage, since each process transmits the packets to the specified destination, the number of streams running concurrently in each stage is small. PMCME cannot make good use of 8 Eports in CG. Consequently, CG also degrades at 8p.

In FT, MG, and BT on PMCME, still more improvements (1.38, 1.14, 1.17 times, respectively) are achieved by increasing the number of processes per node from 8 to 16, which exceeds that of cores per node. This fact shows the main bottlenecks of those applications are the bandwidth and PMCME can relieve those bottlenecks.

5.4 Comparison of PMCME with LACP

First, PMCME have a wide applicability compared to LACP. PMCME does not require the switch and node supporting IEEE802.3ad (LACP). PMCME works well with the low-priced

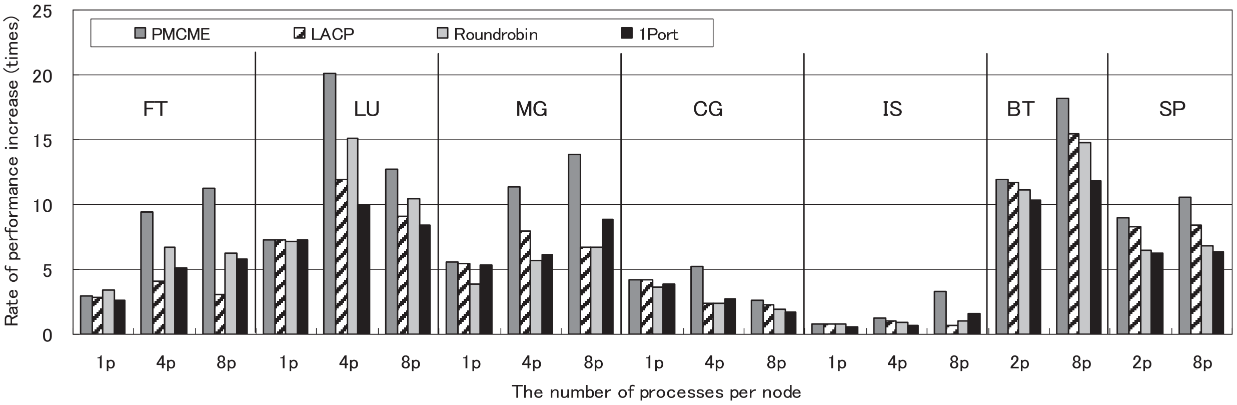


Fig. 15 Parallel processing performance.

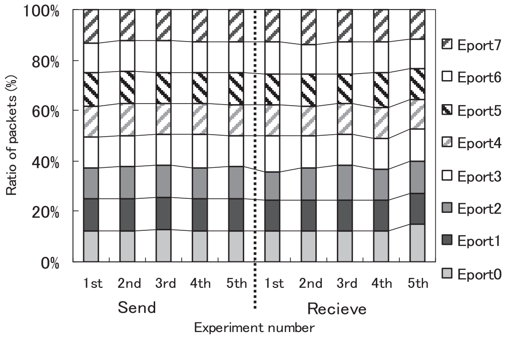


Fig. 16 Ratio of packets per port on FT using PMCME.

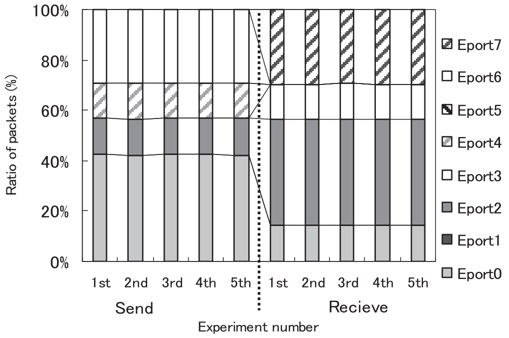


Fig. 17 Ratio of packets per port on FT using LACP.

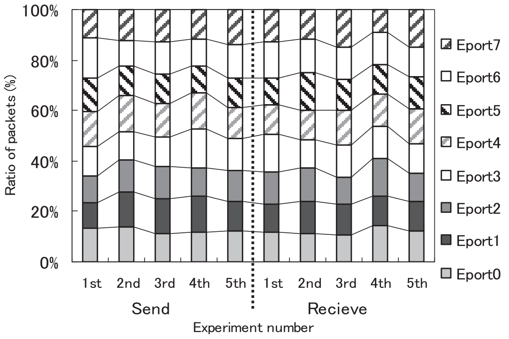


Fig. 18 Change in packets ratio in LACP (TCP port).

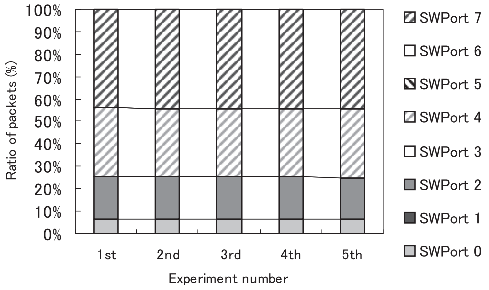


Fig. 19 Ratio of packets per switch port on LACP.

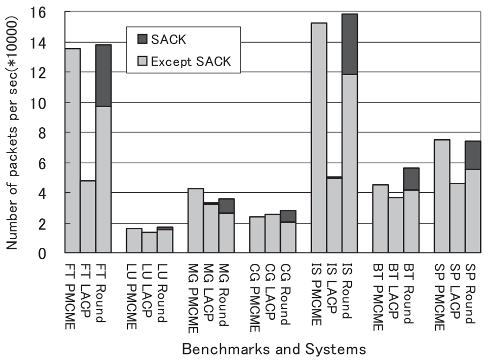


Fig. 20 Ratio of SACK packets.

non-intelligent switches and with any NIC drivers not supporting LACP. In addition, in PMCME, the number and speed of available Eports are not limited. On the contrary, in LACP, the number of Eports available is limited to 8 in actual implementations and all Eports are required to be of the same speed. Even if the number of Eports available is increased to satisfy full bisection bandwidth by upgrading LACP implementations, it introduces new difficulties that the same number of Eports as the number of Eports used for nodes is newly requited in each switch. In the case of this experiment, each switch is required 64 Eports. The number of Eports required increases sharply, as the number of nodes connecting to single switch increases.

Second, PMCME has advantages in load-balancing techniques. PMCME distributes the communication loads equally among all Eports as shown in **Fig. 16** which shows the ratio of transmitted/received packets per Eport to total packets in FT. On the contrary, in LACP, it is difficult to distribute the loads equally because the conditions of distribution are influenced by both the network parameters such as MAC address, IP address, TCP/UDP port number, etc. and implemented hash algorithms in switch. **Figure 17** shows the ratio in LACP when using the switch supporting MAC address based hash algorithm which most switches adopt. LACP cannot distribute the packets equally not only to sender's Eports but also receiver's Eports. As shown in Fig. 10, LACP cannot improve the bandwidth performance in one-to-one communication at all. This can be explained as follows. LACP assigns the same MAC address and IP address to all slave Eports in order to manage as a single logical Eport. Accordingly, in one-to-one communication, because MAC address based hash algorithm selects the same switch Eport to transmit the packets, the receiver receives all packets through a single Eport. **Figure 18** shows the ratio of transmitted/received packets per Eport using TCP port number as LACP hash key. The conditions of distribution also vary from execution to execution, although it is not big in this case.

Finally, PMCME does not require the interconnection links between the switches as shown in Fig. 1 (b) as far as the number of nodes does not exceed that of switches' Eports. This means the communication bottleneck between switches, which is easily caused by bandwidth intensive applications, can be removed. Meanwhile, in LACP, the ratios of the number of packets transferred by each of switch Eports unbalanced and the packets are transferred by limited Eports, not by all switch Eports as shown in **Fig. 19**. In this case, only 4 out of 8 switch Eports are used. The above facts signify the links between switches are likely to become a communication bottleneck in LACP.

5.5 Comparison of PMCME with Round-robin Distribution

First, PMCME can avoid frequent out-of-order arrivals of packets which multi-path communication tends to cause. **Figure 20** shows the total number of sending and receiving packets per second, and the number of SACK packets in it at 8p. SACK is one of TCP header options with which the receiver inform out-

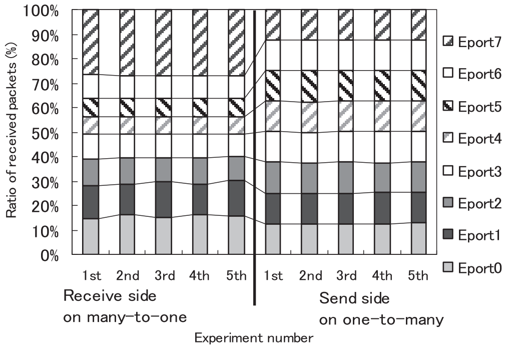


Fig. 21 Ratio of packets per Eport on Round-robin at many-to-one and one-to-many bottleneck sides.

of-order arrivals or lack of packets to the sender. It is clear that round-robin distribution (hereafter, Round-robin) causes frequent out-of-order arrivals accompanied with massive SACK packets, both of which waste the CPU power and network resources, and obstruct the smooth packets' flow. In Round-robin, the numbers of SACK packets rise in proportion to the total packets per second as shown in Fig. 20. As explained in Section 3.1, frequent out-of-order arrivals in Round-robin are inevitable. Consequently, they affect the performance of parallel processing. On the contrary, PMCME causes little SACK packets, a maximum of 0.15% in MG, because all packets of each high bandwidth stream are transmitted through designated path.

Second, Round-robin cannot achieve equal load-balancing at the receiver, whereas PMCME achieves as shown in Fig. 16. Many-to-one communication causes a bottleneck in the receiving side, and one-to-many causes a bottleneck in the sender side because the concentration of packets takes place at different sides. Round-robin shows the different conditions between above two types of communications. **Figure 21** shows the packets balances of Round-robin at above bottleneck sides, namely, the receiving side in many-to-one and sending side in one-to-many communication respectively. Round-robin cannot relieve the bottleneck of many-to-one due to load unbalancing in receiving Eports. On the contrary, the packets of the bottleneck side in one-to-many are equally distributed to multiple Eports. These facts signify Round-robin is suitable for one-to-many communication such as a client/server system. Whereas PMCME relieves both sides' bottlenecks because equal load-balancing is achieved at the both sides. PMCME is suitable for any types of communications.

5.6 Influence by the Number of Eports Available

We evaluate the impact of the variation of number of Eports in each node. All the foregoing experiments are conducted with 8 Eports since LACP cannot support more than 8 Eports. However, PMCME and round-robin support more than 8 Eports, so their experimental results using 10 Eports as well as the results using 4 Eports are evaluated. Note that the number of links between switches in evaluating 4 Eports of LACP is 8. The rate of increase in performance on both cases compared to the results using 8 Eports are shown in **Figs. 22** and **23**. It is clear from the results of PMCME in the two figures that the performances of FT and IS, which carry out All-to-All communication, are obviously influenced by the number of Eports. Namely, an increase in the number of Eports improves the application performances, and a decrease degrades. The performances of LACP naturally decrease as the number of Eports available decreases. The round-robin distribution performance of LU in Fig. 23 increases in spite of a decrease in the number of Eport available. The number of streams in LU, 11 at 64 processes, is much fewer than FT, 395 at 64 processes. LU does not require much Eports. In addition, a decrease in the number of Eports decreases CPU busy rates because of a decrease in the number of out-of-order arrivals. Since LU needs CPU power more than the other, a decrease in CPU busy rate has a good effect on LU performance in round-robin distribution.

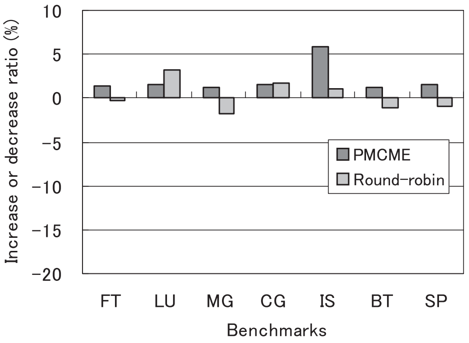


Fig. 22 Rate of increase when increasing the number of Eports from 8 to 10.

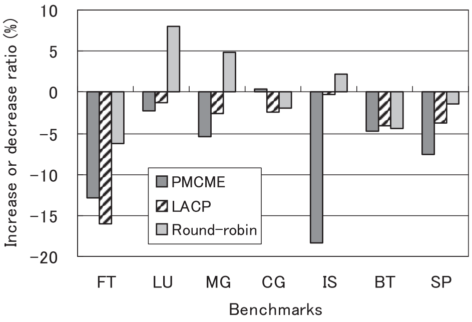


Fig. 23 Rate of increase when decreasing the number of Eports from 8 to 4.

6. Conclusion

IEEE802.3ad (LACP) and round-robin distribution are similar to PMCME in terms of distributing data packets to multiple Eports without the dedicated interconnection network. PMCME outperforms these methods in bandwidth, latency, and parallel processing performance. LACP performance is influenced both by the hash algorithms implemented in the switches and NIC drivers, and by the network parameters in the systems, while PMCME offers stable effect regardless any environments. The experiments using more nodes are planned in the future to examine the effects of PMCME with larger environments.

References

[1] Boden, N.J., Cohen, D., Felderman, R.E., Kulawik, A.E., Seitz, C.L., Seizovic, J.N. and Su, W.-K.: Myrinet: A gigabit-per-second local area network, *IEEE Micro*, Vol.15, pp.29–36 (1995).

[2] InfiniBand Trade Association: InfiniBand^{TM} Architecture Specification (2004), available from ⟨http://www.infinibandta.org⟩.

[3] Petrini, F., Feng, W.-C., Hoisie, A., Coll, S. and Frachtenberg, E.: The Quadrics Network: High-Performance Clustering Technology, *IEEE Micro*, Vol.22, No.1, pp.46–57 (2002).

[4] Watanabe, K., Otsuka, T., Tsuchiya, J., Nishi, H., Yamamoto, J., Tanabe, N., Kudoh, T. and Amano, H.: Martini: A Network Interface Controller Chip for High-Performance Computing with Distributed PCs, *IEEE Trans. Parallel and Distributed Systems*, Vol.18, No.9, pp.1282–1295 (2007).

[5] Ciaccio, G. and Chiola, G.: GAMMA and MPI/GAMMA on Gigabitethernet, *Proc. 7th EuroPVM-MPI Conference*, pp.129–136 (2000).

[6] Karlsson, S., Passas, S., Kotsis, G. and Bilas, A.: Multi-Edge: An Edge-based Communication Subsystem for Scalable Commodity Servers, *Proc. 21st International Parallel and Distributed Processing Symposium*, p.28 (2007).

[7] Shivam, P., Wyckoff, P. and Panda, D.K.: EMP: Zero-copy OS-bypass NIC-driven Gigabit Ethernet Message Passing, *Proc. Supercomputing ACM/IEEE 2001 Conference*, p.57 (2001).

[8] Grant, R.E., Rashti, M.J., Afsahi, A. and Balaji, P.: RDMA Capable iWARP over Datagrams, *Proc. 2011 IEEE International Parallel and Distributed Processing Symposium*, pp.628–639 (2011).

[9] Chen, J., Watson III, W., Edwards, R. and Mao, W.: Message Passing for Linux Clusters with Gigabit Ethernet Mesh Connections, *Proc. 19th IEEE International Parallel and Distributed Processing Symposium*, p.214 (2005).

[10] Sumimoto, S., Tezuka, H., Hori, A., Harada, H., Takahashi, T. and Ishikawa, Y.: The Design and Evaluation of High Performance Communication using a Gigabit Ethernet, *International Conference on Supercomputing '99*, pp.243–250 (1999).

[11] Boku, T., Sato, M., Ukawa, A., Takahashi, D., Sumimoto, S., Kumon, K., Moriyama, T. and Shimizu, M.: PACS-CS: A Large-Scale Bandwidth-Aware PC Cluster for Scientific Computations, *Proc. 6th IEEE International Symposium on Cluster Computing and the Grid*, pp.233–240 (2006).

[12] Miura, S., Hanawa, T., Yonemoto, T., Boku, T. and Sato, M.: RI2N/DRV: Multi-link ethernet for high-bandwidth and fault-tolerant network on PC clusters, *Proc. IEEE International Symposium on Parallel and Distributed Processing*, pp.1–7 (2009).

[13] Cisco: Understanding IEEE 802.3ad Link Bundling.

[14] Extreme Networks: Summit R 400-24t Data Sheet, available from ⟨http://www.andovercg.com/datasheets/extreme-networks-summint-400-24t.pdf⟩.

[15] Sumimoto, S., Tezuka, H., Hori, A., Harada, H., Takahashi, T. and Ishikawa, Y.: High Performance Communication using a Commodity Network for Cluster Systems, *Proc. 9th IEEE International Symposium on High Performance Distributed Computing*, p.139 (2000).

Takafumi Fukunaga received his Ph.D. (Engineering) degree from Kumamoto University, Japan, in 2009. He is currently a Professor at Kumamoto Prefectural College of Technology, Kumamoto. His research interests include networks, parallel and distributed computing, and data mining. He is a member of IPSJ and IEEE computer society.