ExpressEther – Ethernet-Based Virtualization Technology for Reconfigurable Hardware Platform

Jun Suzuki, Yoichi Hidaka, Junichi Higuchi, Takashi Yoshikawa, and Atsushi Iwata
System Platforms Research Laboratories, NEC Corporation
1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa 211-8666, Japan
{j-suzuki@ax, y-hidaka@bq, j-higuchi@ax, yoshikawa@cd, a-iwata@ah}.jp.nec.com

Abstract

We propose ExpressEther as Ethernet-based virtualization technology for a reconfigurable hardware platform. It groups modularized hardware resources interconnected by an Ethernet, and transports a PCI Express (PCIe) packet between the grouped modules by encapsulating it into an Ethernet frame. The configuration of the group is dynamically reconfigured by an Ethernet connection, followed by standardized PCIe hot-plug event. Such reconfigurability enables sharing of a physical resource among computer entities. We demonstrate that an I/O device is shared by servers, using our developed prototype consisted of an interface card and I/O concentrator. A commercially available server and serial ATA card are used for the demonstration, without any change for an operating system, device driver, PCIe interface, and Ethernet switch. The benchmark of I/O performance shows at most 16% degradation, which is caused by the implementation matters of our prototype. No degradation is measured when data flow from an I/O to a server.

1. Introduction

Modular system platform architecture with industry standardized interfaces on each module has become one of the prevailing approaches to a hardware platform [1]. It benefits both system vendors and their customers. For system vendors, a common component reduces total production costs and a standardized development process shortens the time to market. For customers, multi-vendor competition reduces capital expenditures (CAPEX) and a standardized appliance reduces system operation costs (OPEX). The Advanced Telecom Computing Architecture (ATCA), standardized by the PCI Industrial Computers Manufacturers Group (PICMG) [2], is intended to bring modular architecture to networking appliances, whose architecture has been developed according to the proprietary standard of each vendor.

In addition to these benefits, we have showed low-layer reconfigurability introduces novel flexibility to hardware configuration of a computing and networking equipment [3]. It makes it possible to flexibly reconfigure its hardware platform by arbitrarily combining the consisting modules with each other, depending on various kinds of required services. Hardware-level system redundancy and scalability are also made possible.

The interconnection between resource modules is a key to provide the hardware flexibility mentioned above. However, in a conventional blade system, almost all the interconnections which have been implemented so far are conventional Ethernet or based on a proprietary standard. This limits the benefits of flexibility of a modular architecture by isolating hardware resources inside one module from those in the others.

In this paper, we propose ExpressEther as Ethernet-based virtualization technology for a reconfigurable hardware platform. It groups modularized hardware resources interconnected by an Ethernet, and transports a PCI Express (PCIe) packet between the grouped modules by encapsulating it into an Ethernet frame. The configuration of the group is dynamically reconfigured by an Ethernet connection, followed by standardized PCIe hot-plug event. The architecture is based on respective standardized interfaces, and is realized without any change for a conventional operating system, device driver, PCIe interface, and Ethernet switch.

The rest of the paper is organized as follows: Section 2 describes the requirements of the

This work was partly supported by Ministry of Internal Affairs and Communications (MIC).

Proceedings of the 14th IEEE Symposium on High-Performance Interconnects (HOTI'06)
0-7695-2654-3/06 $20.00 © 2006 IEEE

interconnection for a reconfigurable hardware platform. Section 3 provides an overview of the ExpressEther architecture. Section 4 describes the configuration of ExpressEther. Section 5 goes into the details of configuration. Section 6 describes our ExpressEther prototype. In Section 7, we present the results obtained from a demonstration of an I/O sharing function, which is one of the novel benefits introduced by ExpressEther, using our developed system. We also present the results regarding the performance of a shared I/O. We conclude in Section 8.

2. Requirements for interconnection

The followings are the requirements of the interconnection for a regonfigurable hardware platform:

- Link bandwidth scalability
- Switching module density scalability
- Network topology scalability
- End-to-end latency
- Reliable end-to-end packet transmission
- Simple automatic configuration
- Compatibility to conventional system
- Continued support in the future

Although some interconnection standards have been proposed to date– e.g., InfiniBand [4] or Advanced Switching Interconnect (ASI) [5] – ExpressEther, by utilizing an Ethernet technology, meets the above requirements much better with regard to the followings:

- Bandwidth: 10G-Ethernet has become a mature technology and achieving higher bandwidths (e.g., 100 Gb/s) is the target of intensive research [6]; a link aggregation protocol [7] can also be used to enhance link bandwidth.
- Switching scalability and end-to-end latency: An Ethernet switch chip with latency on the hundred-nanosecond order and 24 ports is commercially available today [8].
- Reliable transmission: We have developed a retransmission scheme which is added to an Ethernet layer. The retransmission mechanism is implemented to the hardware of an edge node.
- Automatic configuration: The feature of automatic configuration is succeeded from Ethernet.
- Compatibility to conventional system: A conventional operating system, device driver, PCIe interface, and Ethernet switch can be used as they are.
- Continued support: The continued standardization of PCIe and Ethernet guarantee the lowest costs to the solution of interconnection and continued support in the long term.

3. ExpressEther overview

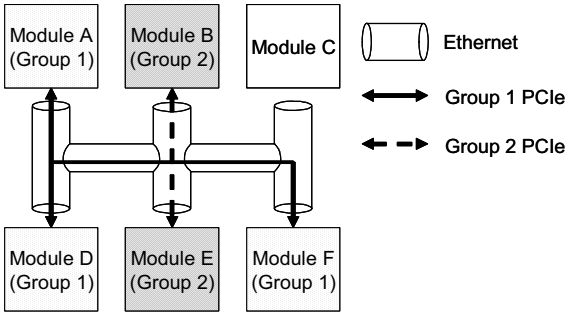


Fig. 1. ExpressEther concept.

Figure 1 shows the concept of ExpressEther. Modules A through F can be a server, a line card, a switch card, an IO card, etc. Each module is interconnected by a conventional Ethernet. They are flexibly reconfigured to constitute a set of modules to be a virtualized system which provides each required service. Once the group of modules is set up, the modules start to communicate with each other by industry standardized PCIe. ExpressEther provides a transport function for PCIe packets by encapsulating them within an Ethernet frame and performs tunneling between the connected modules. An Ethernet frame with IEEE 802.1D VLAN-tag [9] is used in the system and the module groups are isolated by VLAN. The group organization is flexibly reconfigured by altering the VLAN of each module.

ExpressEther also provides system redundancy and hardware scalability. When an active module goes down, the stand-by module is put into the "active" VLAN group to continue providing services. On the other hand, when a hardware resource is in lack, a new resource module is added to complement the system.

Using VLAN is one of the simplest ways to realize grouping and partitioning modules without any change for conventional Ethernet.

One of the novel benefits introduced by ExpressEther is that an I/O can be shared among different servers. From another point of view, a server can use an I/O connected to the Ethernet when it is required by the providing services. This I/O sharing mechanism reduces the total costs of I/O resources and enhances I/O utilization efficiency. It also simplifies the procedures to replace and maintain I/O resources.

4. Configuration

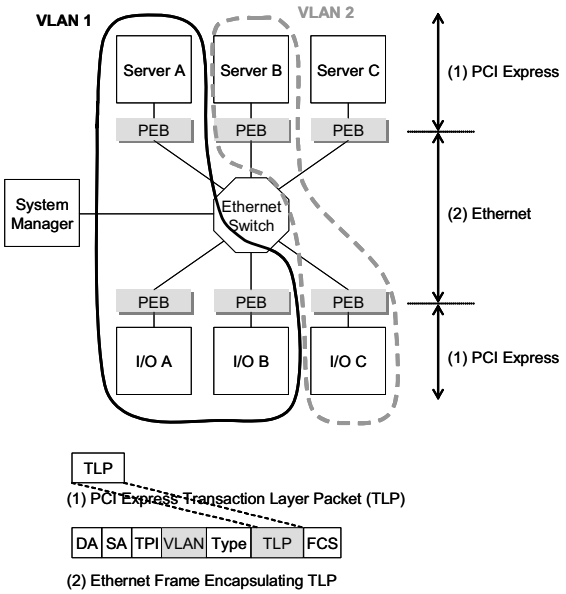


Fig. 2. ExpressEther Configuration. (PEB: PCIe-to-Ethernet bridge)

The configuration of ExpressEther is illustrated using Figure 2. A PCIe-to-Ethernet bridge (PEB) performs the encapsulation and decapsulation of a PCIe packet. A server and an I/O assigned to its server belong to the same VLAN. Note that the assignment of an I/O to a server can be flexibly changed by setting the VLAN of the I/O to that of the server. To the server, this assignment is automatically interpreted as a PCIe standardized hot plug event. The VLAN is set by a system manager in the Ethernet region. By these mechanisms, a static I/O sharing is realized with a commercially available server and I/O. Besides, this function is realized with a conventional operating system, device driver, PCIe interface, and Ethernet switch. The mechanism of the currently standardized PCIe I/O virtualization in PCI-SIG [10] will enable simultaneous I/O sharing among different servers in an ExpressEther-based system.

ExpressEther can also be applied to I/O redundancy and I/O scalability of a system platform.

5. Configuration details

In this section, we describe the detail configuration of ExpressEther. (Table I summarizes the features).

TABLE I
EXPRESSETHER CONFIGURATION FEATURES

Item	Feature
supported protocol of I/O interconnection	PCI Express
PCIe space isolation between servers	VLAN
I/O assignment	system manager performs assignment of I/Os by setting their VLAN
maximum number of servers in a system	4096
maximum number of I/Os assigned to single server	256
Ethernet region to PCIe topology	PCIe switch
device driver for server	not needed
device information frame	periodically broadcast by PEB
I/O configuration by server BIOS	supported for commercially available BIOS
PCIe hot plug	supported for commercially available OS
PCIe hot removal	supported for commercially available OS
trigger of hot plug and hot removal	periodically broadcast device information frame in VLAN of server
reliable TLP transmission between PEBs	supported

5. 1. PCI Express Switch over Ethernet

For the PCIe topology of each server, the region of the Ethernet including the tunneling PEBs is equal to a single PCIe switch, because a PEB connected to a server operates as an upstream PCI-to-PCI bridge in a PCIe switch, while a PEB connected to an I/O acts as a downstream PCI-to-PCI bridge. As a result, one PCIe switch over an Ethernet is configured for each individual VLAN. Figure 3 compares the configured PCIe switch over an Ethernet to a conventional PCIe switch. A maximum of 256 I/Os can be assigned to each server. This is limited by the sum of the device number field (5 bits) and the function number field (3 bits) of a PCIe space.

5. 2. Device Driver

The I/O sharing function provided by ExpressEther does not require any change for a device driver in a server. The Ethernet region appears as a conventional PCIe switch to the PCIe topology of a server, and a conventional PCI driver can manage the system. A PEB reacts in the same way as a PCI-to-PCI bridge in a PCIe switch does.

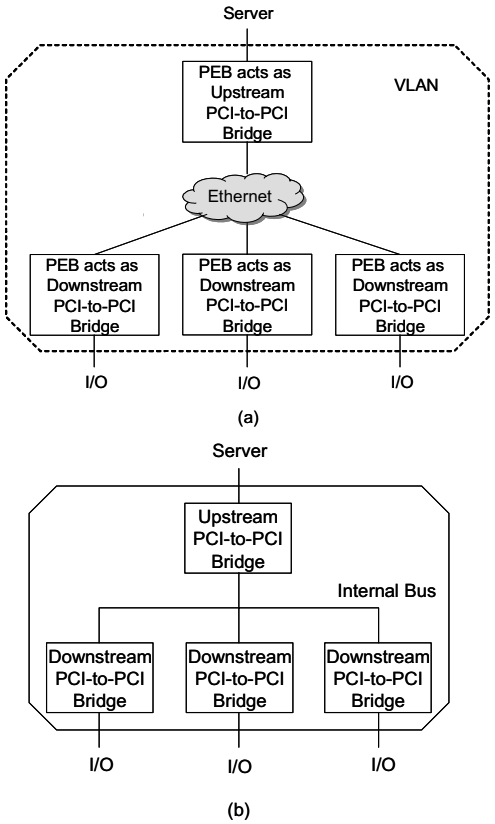


Fig. 3. Block diagram of (a) PCIe switch over Ethernet, (b) conventional PCIe switch.

5. 3. Device Information Frame

A PEB periodically broadcasts an Ethernet frame containing the information of a PEB and the device the PEB is connected to. This information includes the ID of the PEB on the other side of a connection over an Ethernet, and the kind of an I/O device. This frame is also used as the keep-alive frame of an assigned I/O sent to a server.

5. 4. Hot Plug and Device Configuration

There are two ways for a server to configure an assigned I/O in ExpressEther. One is by an industry-standardized PCIe hot plug event, and the other is by presetting the read-only memory (ROM) attached to a PEB. The latter is for the quick configuration of an assigned I/O.

In a standardized PCIe hot plug method, an I/O is assigned to a server during its operation. A new assignment is performed by the system manager which set the VLAN of an I/O. When the VLAN of an I/O is set to that of a server, the device information frame of the I/O is received by the PEB of the server. This triggers the setup of the PEB of the server to properly tunnel a PCIe packet between the PEBs of the server and the I/O. After that, the PEB interrupts the operating system of the server to trigger a PCIe hot plug event. Because this event follows the standardized PCIe specification, it can be performed by a conventional PCI driver. On the other hand, the standard PCIe hot removal is triggered when the PEB of the server stops to periodically receive the broadcast device information frame by the PEB of the assigned I/O. By this mechanism, the I/O can be assigned to another server.

In the presetting ROM method, the information for an I/O assigned to a server is already written to the ROM on the PEB of the server. When the server is booted, the information in the ROM is loaded to a PEB chip and the configuration cycle of the BIOS of the server can access the assigned I/O over the Ethernet.

5. 5. Connectivity between PEBs

PCIe performs link-by-link retransmission and flow control. This mechanism guarantees an operating system that a PCIe packet is unlikely to be discarded during its transmission.

To provide this function, ExpressEther implements a frame retransmission mechanism between connected PEBs. The details of this mechanism will be given elsewhere.

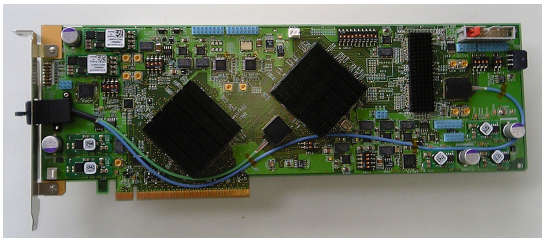
Besides this retransmission mechanism, the automatic configuration of Ethernet enables switch redundancy when a system prepares a redundant switch.

6. Developed system

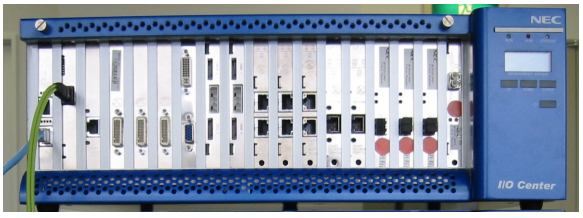
We have developed an ExpressEther prototype consisted of the two newly developed devices: a PEB card which is an ExpressEther interface for a server and inserted to a PCIe slot, and an I/O concentrator accommodating a conventional PCIe I/O card shared among different servers. Figure 4 shows photographs of these devices.

6. 1. PEB Card

A PEB card performs a PCIe packet encapsulation on a server side. Its form factor follows the PCIe specification [11] and is inserted into the conventional PCIe slot of a server. The maximum throughput of the PCIe interface is x16 by lane number while the Ethernet interface consists of 8 10G optical ports to



(a)



(b)

Fig. 4. Developed devices: (a) PEB card, (b) I/O concentrator.

support 80-Gb/s traffic. The 10G Ethernet optical ports are implemented using our originally developed small optical module, Petit [12]. FPGAs are implemented on a PEB card for PEB function and to enable reliable transmission between connected PEBs.

6. 2. I/O Concentrator

An I/O concentrator accommodates up to 18 commercially available I/O cards. The FPGA with the PEB function is implemented inside the box. By this kind of equipment, I/Os are gathered into one location and arbitrarily assigned to the servers connected by an Ethernet.

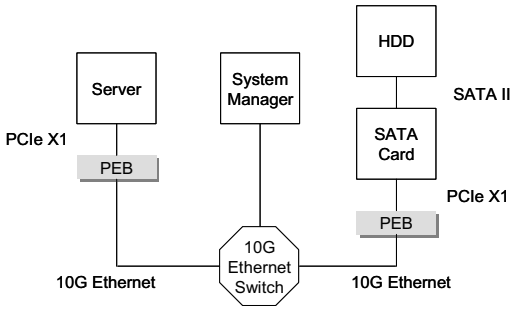
7. Experimental evaluation

We demonstrated the I/O sharing function, and evaluated the performance of a shared I/O using our developed prototype.

7. 1. Experimental Setup

Figure 5 shows the photograph and block diagram of the experimental setup. A storage device was used for a tested I/O. A server and an HDD were interconnected by an Ethernet using a PEB. The HDD and a Serial ATA (SATA) card were accommodated by the I/O concentrator.

(a)



(b)

Fig. 5. Experimental setup: (a) Photograph, (b) Block diagram.

The server and a PEB were connected using a PCIe x1 link. The link width of the Ethernet was 10 Gb/s, and a 10G Ethernet switch was used to interconnect the two PEBs. The link between a PEB and the SATA card was also a PCIe x1 link. The SATA card and the HDD were connected by a SATA II link.

The storage performance benchmark was determined by measuring the time until the copy command cp of Linux in the server was completed. The size of the file used in the experiment was 712 MB. The measured time was compared to the case when the SATA card was directly inserted into the PCIe slot of the server.

7. 2. Results

We demonstrated the basic function of the I/O sharing mechanism. The VLAN of the server and the HDD were first assigned different values. After the system manager set the VLAN of the HDD to that of the server, the HDD was hot plugged to the PCIe space of the server. On the other hand, when the HDD was assigned a different VLAN from that of the server, it was hot removed. By these mechanisms, I/Os connected to the Ethernet can be shared among servers. This scheme enables hot plug /out of I/Os by simply

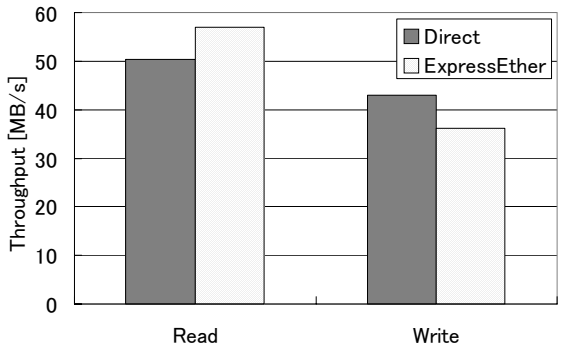


Fig. 6. Benchmark results for storage performance with ExpressEther

changing their assigned VLAN, which enhances the reconfigurability and reliability of a hardware platform.

Next, we evaluated the I/O performance when it was connected to the server in the configuration shown in Figure 5. Figure 6 shows the benchmark results for the storage performance. "Direct" indicates the case when the SATA card is directly inserted into the PCIe slot of the server. "Read" is the result when the file is read from the HDD over the Ethernet, and "Write" is the result when the same file is written to the HDD. The write performance of ExpressEther is slightly worse than that of the direct case. Similar results are obtained when we used the block transfer command of Linux in the server.

To analyze this phenomenon, we captured the PCIe packets in flight using a PCIe protocol analyzer while the copy command was operated. The reason for the write performance degradation is explained as follows; the operating system in the server sets the direct memory access (DMA) engine in the SATA card when it accesses the HDD. When the server reads data from the HDD, the DMA engine sends a PCIe write request to the server. This is a posted request and the DMA engine does not have to wait until a completion packet is returned to send the next request. On the other hand, when the server writes data to the HDD, the DMA engine sends a PCIe read request to the server. This is a non-posted request and the DMA engine has to wait until the completion with data packet is returned. Therefore, in this case, the round trip time (RTT) between the server and the HDD can be the bottleneck of the write performance.

We analyzed each latency factor of the RTT in the configuration and found that the performance degradation resulted from the FPGA used to implement the PEB function. The replacement of the FPGA to an ASIC doubles the clock rate of the circuit from 125 MHz to 250 MHz. By this enhancement, we can conclude, based on the evaluation of the data flow between the server and the HDD, that the amount of the reduced latency is enough to solve the write performance degradation.

Note that these experiment was performed using a commercially available server and SATA card without the change of an operating system or a device driver. These results prove that ExpressEther provides the novel function of hot plug and I/O grouping without performance degradation.

8. Conclusion

We have proposed ExpressEther as Ethernet-based virtualization technology for a reconfigurable hardware platform. It groups modularized hardware resources interconnected by an already implemented Ethernet, and transports a PCI Express (PCIe) packet between the grouped modules by encapsulating it into an Ethernet frame. The configuration of the group is dynamically reconfigured by an Ethernet connection, followed by standardized PCIe hot-plug event. The separation of the groups is realized by VLAN. Using VLAN is one of the simplest ways to realize grouping and partitioning modules without any change for conventional Ethernet.

ExpressEther is based on respective standardized interfaces, and is realized without any change for a conventional operating system, device driver, PCIe interface, and Ethernet switch.

One of the novel benefits introduced by ExpressEther is that it enables an I/O sharing function between servers. We have demonstrated the experimental evaluation setup consisted of our developed interface card and I/O concentrator perform such an I/O sharing function with a commercially available server and SATA card. The benchmark of I/O performance showed at most 16% degradation, which is caused by the delay of the implemented FPGA, and is solved by utilization of an ASIC. No degradation was measured when data flowed from an I/O to a server. These results prove that ExpressEther provides the novel function of hot plug and I/O grouping without performance degradation.

We believe that ExpressEther, which is based on a conventional system and provides simple control scheme of hot plug / out and grouping of I/Os based on VLAN, guarantees the best solution to the interconnection for a reconfigurable hardware platform. We will continue to develop a system based on this architecture and confirm the effectiveness of various functionalities in the future.

The additional experimental results will be presented at the presentation of the conference.

9. References

[1] http://www.intel.com/design/network/papers/302641.htm

[2] PICMIG 3.0, "Advanced TCA Base Specification R2.0," Mar. 2005.

[3] N. Kami et al., "Scalable and Reliable Platform for Service-Oriented Networking and Computing Systems," MILCOM 2005, pp. 1-7, Oct. 2005.

[4] InfiniBand Trade Association, "InfiniBand Architecture Specification Release 1.1," Nov. 2002.

[5] ASI-SIG, "Advanced Switching Core Architecture Specification," Nov. 2004.

[6] P. J. Winzer et al., 107-Gb/s optical ETDM transmitter for 100G Ethernet transport," Post Deadline Session 1, ECOC2005, Sep. 2005.

[7] 802.3AD-2000 IEEE Standard for Information Technology – Local and Metropolitan Area Networks – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications–Aggregation of Multiple Link Segments, 2000.

[8] http://www.fulcrummicro.com/focalpoint.htm

[9] ANSI/IEEE Draft Standard P802.1Q/D11, "IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks," July 1998.

[10] PCI-SIG, "PCI Express IO Virtualization and IO Sharing Specification Revision 0.3," Oct. 2005.

[11] PCI-SIG, "PCI Express Card Electronical Specification Revision 1.1," Mar. 2005.

[12] T. Yoshikawa et al., "Optical interconnection as an IP macro of a CMOS library", Hot Interconnects 9, pp. 31-35, Aug. 2001.A.B. Smith, C.D. Jones, and E.F. Roberts, "Article Title", *Journal*, Publisher, Location, Date, pp. 1-10.