A System Interconnection Device for Small-Scale Clusters

Ye Ren(✉), Young Woo Kim, and Hag Young Kim

Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Republic of Korea
{yeren,bartmann,h0kim}@etri.re.kr

Abstract. The performance of a physical cluster ultimately depends on two factors. One is the capability of individual computing nodes and the other is the networking speed among them. Recent processors are being greatly developed in terms of enhancing the processing speed while lowering the monetary cost. As for the networking technologies, nowadays dominant solutions have disadvantages such as high installation price and low protocol efficiency. Such drawbacks become the bottleneck of improving the 'performance per cost' ratio of the cluster as a whole. This paper proposes an alternative system interconnection device especially for application in small-scale clusters. The non-transparent bridges in PCI Express technology are employed to allow PCI Express packets to directly transmit across networked computing nodes. The performance is measured under two kinds of data transmission schemes, with two different benchmarking tools, respectively. Currently the proposed device delivers a peak unidirectional throughput of 8.6 gigabits per second.

Keywords: Cluster · Interconnect · PCI Express · Throughput

1 Introduction

One common way of building up a small-scale cluster is to combine multiple computing nodes in the form of either boxes or blades, using networking cables and switches. Based on this method, the performance of a cluster that nests those cloud computing resources ultimately depends on the two factors. One is the performance of each individual node and the other is the networking speed among nodes of the cluster.

Recently, processors are rapidly developed to deliver more powerful processing capability while at lower cost according with the Moore's Law [1]. General-purpose CPUs clustered together can allow for an impressive performance of hundreds of giga-FLOPS (giga FLoating-point Operations Per Second) or even several teraFLOPS. In the meantime, the interconnection technologies for clusters are dominated by InfiniBand and high-performance Ethernet [2]. Those dominant networking solutions can deliver high bandwidth, but they still have disadvantages such as high price for installation and low protocol efficiency in packet transmission. Such drawbacks of system interconnects have become the bottleneck to the improvement for the cluster in terms of achieving a higher 'performance per cost' ratio as a whole.

This paper proposes an alternative system interconnection device based on the PCI Express technology and the device is especially for application in small-scale clusters.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2016
Y. Zhang et al. (Eds.): CloudComp 2015, LNICST 167, pp. 344–353, 2016.
DOI: 10.1007/978-3-319-38904-2_36

In the present paper an interconnection device refers to the physical interface between a local computing node and an external cable, facilitating the communication with the remote standalone node connected on the other end of the cable. The proposed device employs the non-transparent bridges to allow PCI Express packets to directly transmit across computing nodes and it is easy and flexible for ordinary consumers to install and use. Two kinds of data transmission schemes are implemented which are the memory copy and the direct memory access (DMA). The performance is measured with two different benchmarking tools, respectively, and experiments show that the proposed device outperforms the Gigabit Ethernet and is capable to deliver a peak unidirectional throughput of 8.6 gigabits per second.

The rest of paper is organized as follows. An overview of the related research work on system interconnects are depicted in Sect. 2. Section 3 presents the detailed explanation about the technology deployed in the proposed interconnection device. Next, Sect. 4 elaborates the design and implementation of the proposed generally-applicable interconnection device. Section 5 shows performance evaluation results and discussions. Finally, Sect. 6 concludes the whole paper.

2 Related Work

There are already several existing solutions for high-performance system interconnection. The 10 Gigabit Ethernet [3] and other commercial models [4, 5] provide higher performance but also induce too much power consumption and expenses for small-scale use case. Moreover, they require switches even to build up a network with just a few nodes (e.g. 2–5 nodes).

PCI Express (PCIe) [6] is a promising technology to use in system interconnection devices and this is already indicated by many previous works. Ravindran [7] examined the technology of "PCI Express External Cabling 1.0 Specification" and proposed a new technical specification named "the local mode specification" to ensure a high-bandwidth, low-latency connection between devices with independent PCI domains. Hanawa et al. proposed communication links using PCIe Gen2 (2nd generation) for embedded systems [8] and he also demonstrated an effective interconnect for direct communication between accelerators over nodes via PCIe external cables [9]. Byrne et al. [10] evaluated the benefits of using PCIe for power-efficient networking with a test-bed cluster where each node links to a central PCIe switch board and independently manages their own devices. Krishnan [11] proposed a PCIe based hardware with software for integrating IO expansion and clustering functionalities onto one PCIe interconnect, in which the solution consists of host PCIe adapters, expansion switch, and cluster switch. Such hardware would be too complicated and over-powerful in case of only two or three computing nodes are to be clustered by a consumer. In addition to the above works, a standalone system interconnection device named PCIeLINK (PCI Express Link) [12] has been proposed which can be used to integrate multiple general-purpose CPUs.

This paper elaborates the research on the proposed PCIeLINK and presents a more developed implementation with better performance compared with the prior one.

A detailed description about the adopted technology by PCIeLINK is included in this paper as well.

3 Using PCI Express for System Interconnects

3.1 Advantages and Disadvantages of PCI Express for System Interconnects

PCIe is an industrial standard system bus that delivers high-performance and low latency packet transmission between the root complex and the enumerated PCI endpoints (as shown in Fig. 1). Also, the PCIe slot is also the primary interface on a motherboard for a system to connect with various peripherals. The data rate per lane for PCIe Gen2 is 4 Gbit/s, for Gen3 is 7.877 Gbit/s, and the PCIe specification is evolving to enable higher data rates. Prevalent PCIe connectors adopt 8 or 16 lanes which contribute to a total theoretical bandwidth as good as that of many widely used system interconnection devices for supercomputers. Besides, the PCIe as an interconnect technology consumes lower power per port and also costs less money for unit price than most common commodity products [13]. Therefore, PCIe is considered as a competitive and alternative technology for system interconnections.

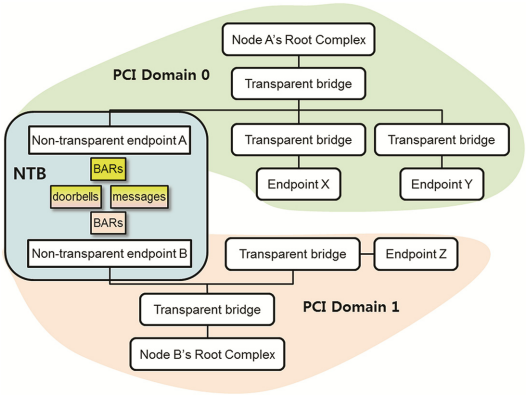

Fig. 1. PCI enumeration on two systems and the communication between the two PCI domains through the non-transparent bridge (NTB).

The PCIe standard was initially developed to allow one root complex to enumerate the PCI endpoints in one local PCI address space (or PCI domain) and provide connections between processors and IO devices within a single computing system. Therefore, there exit some restrictions in both electrical and systemic perspectives to directly apply PCIe as a system interconnect. If two PCIe domains are connected with each other using a common PCIe device directly, then the asynchronous clocking in the two different systems will hinder PCIe protocol from working properly. A more critical problem is the PCIe hierarchy enumeration problem. When two systems are powered up, BIOS of each side enumerates devices in its PCI hierarchy (PCI domain) and assign BDF (Bus, Device and Function) numbers. If two identical systems are connected with a typical

PCIe device, then the BDF number conflicts occur, which causes the two systems both fail to build up a workable PCI hierarchy.

3.2 Data Transmission Mechanism Through Non-transparent Bridges

In order to isolate the different electrical clocking and also separate the PCI bus hierarchies between individual systems, the PCIe non-transparent bridge (NTB) can be employed. As shown in Fig. 1, a typical NTB consists of two PCIe non-transparent (NT) endpoints as well as the doorbell and message registers in between to exchange status and facilitate the PCI inter-domain communication. The device discovery process of each side stops at the non-transparent endpoint during enumeration, so that the other side of NTB is not influenced at all. Doorbell registers are used to initiate interrupts on the other side of NTB and message registers can be read and written by processors on both sides of NTB. Several prior researches [10, 14, 15] about developing novel system interconnections have been conducted by employing the NTB technology.

Both sides of the NTB have their own PCI address space, i.e. PCI domain. The NTB needs to support address translation in order to direct a packet from one side to pass across the NTB and reach its destination on other side. The packet transmitting mechanism is explained based on the two-node situation as shown in Fig. 2.

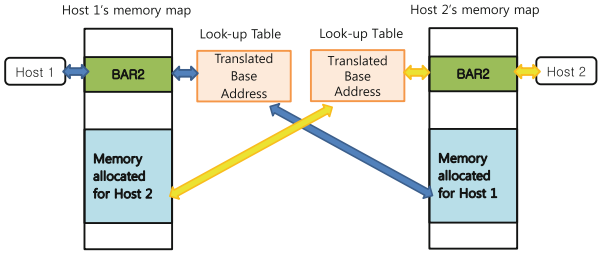

Fig. 2. Data transmission scheme between two hosts

Each NT endpoint is associated with six BARs (Base Address Register) in its configuration space. BAR0 (plus BAR1 in case of 64-bit addressing) maps this NT endpoint's configuration space into the enumerating local host's virtual memory region so that the local host can access the configuration space of this NT endpoint. BAR2 (plus BAR3 in case of 64-bit addressing) maps the endpoint-requested memory into the local host's memory region and this BAR2 (or plus BAR3) mapped region is used to serve as the aperture for local host system to access some specified memory region of the remote system on the other side of NTB. A look-up table is associated with BAR2 (or plus BAR3), and the table entries are manually programmed so that those aperture memory regions are mapped to the target memory region of the remote node. As a result, when the NT endpoint receives a memory operation (read or write) from its local host and the target address of the operation falls in the aperture, the target address of such an operation will get translated as per the associating look-up table entry, so that the memory operation goes across the NTB and is performed on the prepared memory region

on the remote host. The BAR4 (plus BAR5 for 64-bit addressing) opens up an aperture for accessing remote host's memory in the same manner as the BAR2 and BAR3, except that the mapped region can also be used to access different remote hosts' configuration spaces.

4 Design and Implementation of PCIeLINK

4.1 Hardware Design and Implementation

PCIeLINK [10] was designed to have one eight-lane PCIe standard interface for connection with the local host and four four-lane external QSFP (Quad Small Form-factor Pluggable) modules for connection with remote systems. Therefore, the PCIeLINK delivers up to 16 Gbps (Gbit per second) data rate on each of its external ports and 32 Gbps on its host interface in case of using PCIe Gen2.

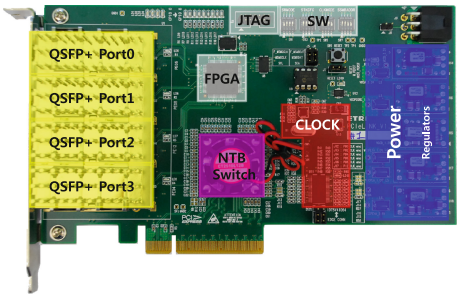

Fig. 3. The implemented PCIeLINK board [10].

The PCIeLINK board was implemented with standard full profile PCIe form factor as shown in Fig. 3. As has been designed, the board has one host connector to be inserted into the PCIe slot of the local host's motherboard and four external ports to connect with remote hosts. Every two PCIeLINKs can be connected by a QDR (quad data rate) cable without using any networking switch in between, which is easy to implement and also indicates a cheaper installation budget for a small cluster with only two to five nodes.

The board is initially configured with a firmware by which the board was divided into five different partitions, where each partition belongs to an independent PCI address domain. Also by using the firmware, the host interface is configured to operate in PCI-to-PCI (P2P) bridge plus NT endpoint plus DMA combined mode, while the external ports are all configured to operate in NT endpoint mode. The firmware also sets the value of several primary registers on the PCIeLINK to regulate the BAR's address translation method (e.g. direct address translation or lookup table translation), requirements of the claiming of memory region (e.g. prefetchable or non-prefetchable), the size of the requested memory region by the NT endpoint, and the memory addressing bits (32-bit or 64-bit addressing), etc.

4.2 Software Design and Implementation

Each node in the PCIeLINK-based cluster is identified by a unique system ID number. The device driver developed for the PCIeLINK board encompasses three layers, i.e. base layer, library layer, and virtual Ethernet interface layer as shown in Fig. 4.

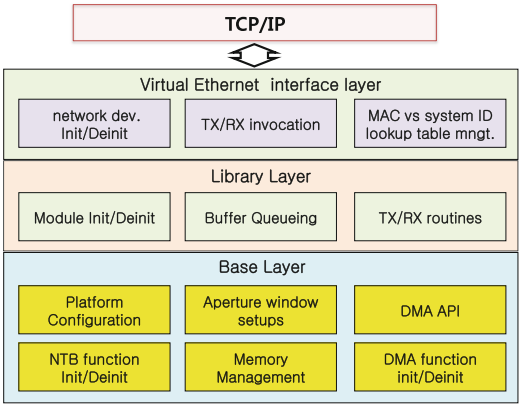

Fig. 4. The layered structure of the implemented device driver.

The base layer abstracts away those tedious hardware registers and provides primitive functions which are called by upper-layer functions to directly manipulate the hardware. Specifically, the base layer module consists of functional blocks which enable NT endpoint initialization and de-initialization, hardware platform configuration, memory management, NT aperture window setting-ups, DMA function initialization and de-initialization, the DMA APIs that are flexible to use by upper layer, and the interrupt management. The library layer module sets up queue models and functions that operate those queues. The library module also prepares wrapped-up functions for the upper layer to manipulate the packet transmission and reception. The virtual Ethernet interface module allocates an Ethernet device structure with random MAC address and registers itself as a network device, so that the above TCP/IP layer sees the PCIeLINK device the same way as a regular Ethernet card. The virtual Ethernet interface module also has functional blocks to carry out the mapping between the network device's MAC address and the PCIeLINK's system ID number as well as the invocation of the lower-layer provided packet transmission routines. The device driver is implemented to transmit packets using either the memcpy mechanism or the DMA mechanism.

5 Evaluation and Discussion

5.1 Description of Test Environment

The initial experiment only implements the back-to-back topology in which two identical nodes are interconnected using two PCIeLINKs as illustrated in Fig. 5. The detailed description of the test environment is given in Table 1. PCIeLINK is very easy to be

installed into the standard PCIe slot of a motherboard. Since a PCIeLINK board has four external ports, they are capable to interconnect a small number of hosts without using the networking switch. If a PCIe non-transparent networking switch is provided, more nodes can connect to the PCIe-networked cluster, though an upper limit exists for the total number of nodes due to the PCIe specification.

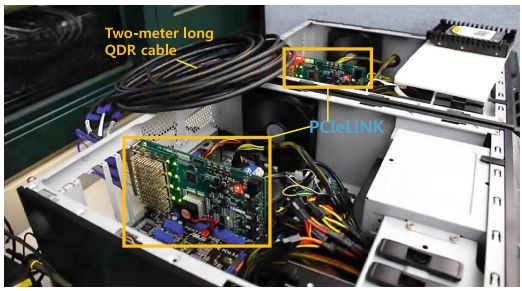

Fig. 5. The experimental platform constructed with two nodes in back-to-back topology.

Table 1. Test environment description.

Items	Description
CPU	x86 architecture, 3.4 GHz × 4 cores
Memory	DDR3 1333 MHz × 2 channels, 4 GB
NTBs on PCIeLINK board	PCIe Gen2
Motherboard slot	PCIe Gen3
OS	32-bit Open-source operating system

In this paper, the performance of PCIeLINK is compared against that of the traditional Gigabit Ethernet. The measurement approaches include using the proprietary ping-pong testing program within the device driver layer and the benchmarking tools over the TCP/IP. Experiments were made by using different testing methods, i.e. memory copy and DMA, respectively and the final results were comprehensively compared and analyzed.

5.2 Experiment Results and Performance Evaluation

Figure 6 demonstrates the unidirectional throughput results tested under the same environment with different tools and data transmission mechanisms using the same set of increasing sizes of transmitted messages. The peak value of each set of testing results is marked in the figure. A dominant pattern is reflected that PCIeLINK delivers higher saturated throughput than Gigabit Ethernet, regardless of the testing method or the data moving mechanism the driver used. The NetPIPE benchmark shows that PCIeLINK delivers a peak throughput which is, using DMA and memcpy mechanism respectively, 5.4 times and 1.6 times of the highest performance supported by Gigabit Ethernet. Likewise, the highest throughput of PCIeLINK with DMA and memcpy mechanism

respectively measured by Netperf benchmark is 4.4 times and 1.8 times of that of the conventional Gigabit Ethernet. Specifically, the highest throughput is 8613 Mbps (megabit per second) at the message size of 2M bytes, which is achieved from the ping-pong testing program over PCIeLINK with DMA mechanism. This highest throughput measured of PCIeLINK with DMA is almost 9.2 times of the measured summit throughput of Gigabit Ethernet (941 Mbps by Netperf benchmarking), and for the ping-pong test of PCIeLINK with memcpy the result is 7.2 times. The highest throughput measured from PCIeLINK is 53.8 % of the designed wire speed of an external port, and this indicates that there is still much space to further improve the current driver. The reason that the ping-pong testing measures a higher performance than the NetPIPE and the Netperf benchmarking tools is that the proprietary ping-pong program runs within

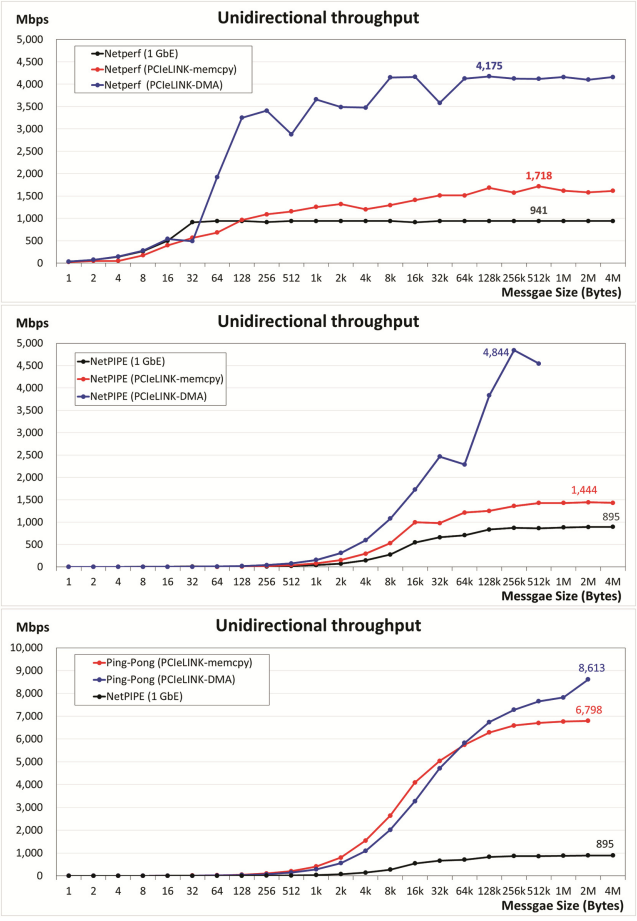

Fig. 6. The comparison of unidirectional throughput between Gigabit Ethernet (i.e. 1 GbE in the legend) and the proposed PCIeLINK. PCIeLINK-memcpy represents the results obtained with the memcpy data moving mechanism and PCIeLINK-DMA denotes the results obtained with the DMA data moving mechanism in the device driver. Peak values are noted in the figure.

the driver code, so that unlike the benchmarking tools which run over the TCP/IP protocol stack the ping-pong testing measures the raw time of data transmission between the link layer of the two nodes, saving the time consumed by copying data from and to the upper layers within the local node.

When measuring the same device's performance in the same experimental environment with two different benchmarking tools, some other patterns are discovered. For example, for the same interconnect type (and the same data moving mechanism for PCIeLINK) the throughput measured by NetPIPE gets saturated later than that measured by Netperf. Another example is that, for the same message size, in general the throughput tested by Netperf is higher than that tested by NetPIPE. Such coherent patterns shown on both PCIeLINK and Gigabit Ethernet indicate that, like the Ethernet, the performance of a PCIeLINK-based cluster is consistent and reproducible. Lastly, there is an abnormal performance noticed when testing PCIeLINK with DMA using NetPIPE that the saturated throughput degraded and cannot sustain for large message sizes, which may due to the not adapted implementation of the DMA mechanism in the device driver or the overheated NTB on the PCIeLINK. Last but not the least, it is measured that PCIeLINK with DMA mechanism does not cause additional CPU usage as compared with that of the traditional Gigabit Ethernet in the test with TCP/IP based applications.

6 Conclusions

This paper has presented an alternative system interconnection device named PCIeLINK that provides high data transmission rate and leaves out switches especially for usage in small-scale clusters. The PCIeLINK is based on the non-transparent bridging technology and is very easy and flexible to use by ordinary users. Early implementation and evaluation are conducted, which demonstrates that PCIeLINK outperforms Gigabit Ethernet in terms of delivering a much higher throughput without consuming more CPU usage. Future work includes optimizing the device driver of PCIeLINK in order to achieve the hardware-supported speed as much as possible as well as comparing the power consumption between PCIeLINK and that of traditional interconnection devices.

Acknowledgments. This work was supported by the ICT R&D program of MSIP/IITP. [10038768, The Development of Supercomputing System for the Genome Analysis].

References

1. 50 Years of Moore's Law. http://www.intel.com/content/www/us/en/silicon-innovations/moores-law-technology.html
2. Interconnect Family System Share. http://www.top500.org/statistics/list/
3. Bencivenni, M., Bortolotti, D., Carbone, A., Cavalli, A., Chierici, A., et al.: Performance of 10 Gigabit ethernet using commodity hardware. IEEE Trans. Nucl. Sci. **57**(2), 630–641 (2010)
4. Interconnect analysis: 10 GigE and InfiniBand in high performance computing. White Paper, HPC Advisory Council (2009)

5. Koop, M.J. Huang, W., Gopalakrishnan, K., Panda, D.K.: Performance analysis and evaluation of PCIe 2.0 and quad-data rate InfiniBand. In: 16th IEEE Symposium on High Performance Interconnects, pp. 85–92, August 2008
6. PCI Express Base Specification, Revision 3.0, PCI-SIG, November 2010
7. Ravindran, M.: Extending cabled PCI express to connect devices with independent PCI domains, SysCon 2008. In: IEEE International Systems Conference, Montreal, Canada, April 2008
8. Hanawa, T., Boku, T., Miura, S., Okamoto, T., Sato, M., et al.: Low-power and high-performance communication mechanism for dependable embedded systems. In: International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems, pp. 67–73 (2008)
9. Hanawa, T., Kodama, Y., Boku, T., Sato, T.: Interconnection network for tightly coupled accelerators architecture. In: IEEE 21st Annual Symposium on High-Performance Interconnects, San Jose, USA, pp. 79–82, August 2013
10. Byrne, J., Chang, J., Lim, K.T., Ramirez, L., Ranganathan, P.: Power-efficient networking for balanced system designs: early experiences with PCIe In: HotPower 2011 Proceedings of the 4th Workshop on Power-Aware Computing and Systems, Article No. 3 2011
11. Krishnan, V.: Evaluation of an integrated PCI express IO expansion and clustering fabric. In: 16th IEEE Symposium on High Performance Interconnects, Stanford University, USA, pp. 93–100, August 2008
12. Ren, Y., Kim, Y.W., Kim, H.Y.: Implementation of system interconnection devices using PCI express. In: IEEE International Conference on Consumer Electronics, Las Vegas, USA, pp. 300–301, January 2015
13. Percival, D.: PCI express clustering. In: PCI-SIG Developers Conference, Israel (2011)
14. Bu-Khamsin, A.: Socket direct protocol over PCI express interconnect: design, implementation and evaluation. MS thesis, Simon Fraser University (2012)
15. Cooper, S.: Using PCIe over cable for high speed CPU-to-CPU communications. PCI-SIG Developers Conference (2008)