KIPS Trans. Comp. and Comm. Sys.
Vol.8, No.3 pp.65~72 pISSN: 2287-5891

https://doi.org/10.3745/KTCCS.2019.8.3.65

Implementation of Ring Topology Interconnection Network with PCIe Non-Transparent Bridge Interface

Sang-Gyum Kim† · Yang-Woo Lee† · Seung-Ho Lim††

ABSTRACT

HPC(High Performance Computing) is the computing system that connects a number of computing nodes with high performance interconnect network. In the HPC, interconnect network technology is one of the key player to make high performance systems, and mainly, Infiniband or Ethernet are used for interconnect network technology. Nowadays, PCIe interface is main interface within computer system in that host CPU connects high performance peripheral devices through PCIe bridge interface. For connecting between two computing nodes, PCIe Non-Transparent Bridge(NTB) standard can be used, however it basically connects only two hosts with its original standards. To give cost-effective interconnect network interface with PCIe technology, we develop a prototype of interconnect network system with PCIe NTB. In the prototyped system, computing nodes are connected to each other via PCIe NTB interface constructing switchless interconnect network such as ring network. Also, we have implemented prototyped data sharing mechanism on the prototyped interconnect network system. The designed PCIe NTB-based interconnect network system is cost-effective as well as it provides competitive data transferring bandwidth within the interconnect network.

Keywords : BHPC, PCIe Non-Transparent Bridge, Interconnect Network

PCIe Non-Transparent Bridge 인터페이스 기반 링 네트워크 인터커넥트 시스템 구현

김 상 겸† · 이 양 우† · 임 승 호††

요 약

HPC(High Performance Computer)은 다수의 계산노드를 초고성능 상호연결망으로 연결하여 클러스터 시스템으로 구성된 시스템이다. 이러한 HPC 시스템에서 사용하는 계산 노드 간의 연결 네트워크 기술로는 Infiniband, Ethernet 등의 기술이 많이 사용된다. 최근 PCIe 표준의 발전으로 인해서 컴퓨터 호스트는 고속의 주변 장치 디바이스를 주로 PCIe Bridge 인터페이스에 연결하여 사용한다. PCIe 표준 기술 중 컴퓨터 노드 간의 직접 연결하는 방식으로 Non-Transparent Bridge(NTB) 기반의 인터콘넥션 표준이 존재한다. 그러나 NTB의 기본 표준은 두 노드 간에 분리된 메모리를 제공하는 방식이기 때문에 다중 노드를 직접 연결하기 위해서는 추가된 구성 방법이 필요하다. 본 논문에서는 다중 NTB 포트에 직접 연결된 다수의 호스트들 간에 무스위치 네트워크를 구성하여 NTB 통신을 이용한 데이터 공유 방법의 설계와 구현에 대해서 다룬다. 각 호스트에 연결된 두 개의 NTB포트를 이용해서 링 네트워크를 구성하고, 링 네트워크 상에서 NTB 인터컨넥션을 이용한 데이터 공유 방식의 구현을 하였다. 이와 같이 PCIe NTB 기반 무스위치 네트워크를 통해서 기존의 인터커넥트 네트워크에 비해서 Cost-Effective한 HPC 상호연결망을 구성할 수 있다.

키워드 : HPC, PCIe Non-Transparent Bridge, 인터커넥트 네트워크

1. 서 론

HPC(High Performance Computing) 시스템은 시간이 많이 소모되는 고급 과학 계산 응용 프로그램이나 대량의 데이터 처리를 요구하는 컴퓨팅 작업을 수행하기 위해서 많은 수의 계산노드(Computation Node)를 초고성능 상호연결망(Interconnection Network)로 연결하여 클러스터 시스템으로 구성하고 있다. 그러므로, HPC 시스템은 계산노드를 연결하는 상호연결망(Interconnect Network)은 큰 대역폭을 보장해 줄 수 있는 기술이 필요하다. 즉,

※ The Research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2016R1C1B2009914). 이 논문은 한국과학기술정보연구원(KISTI)의 2018년 주요사업 위탁연구지원사업의 지원에 의해서 이루어진 것임. 이 연구는 한국외국어대학교 교내학술연구비의 지원에 의하여 이루어진 것임.
※ 이 논문은 2018년도 한국정보처리학회 추계학술발표대회에서 'Non-Transparent Bridge 기반 링 네트워크 통신 방식 구현'의 제목으로 발표된 논문을 확장한 것임.
† 비 회 원 : 한국외국어대학교 컴퓨터 · 전자시스템공학부 학사과정
†† 종신회원 : 한국외국어대학교 컴퓨터 · 전자시스템공학부 교수
Manuscript Received : December 3, 2018
Accepted : January 9, 2019
* Corresponding Author : Seung-Ho Lim(lim.seungho@gmail.com)

이러한 상호연결망 기술은 HPC 시스템의 성능과 안정성을 결정하는 중요한 기술 요소이라고 할 수 있다.

현재 대부분의 HPC 시스템의 Interconnect Network 기술은 Infiniband 및 Ethernet 기술에 의존하고 있으며, 이 두 기술에 대한 기술 의존성이 많이 있다. 이에 대해서, 최근 주요 HPC 관련 연구기관들은 자체 Interconnect 기술을 확보하기 위해서 많은 연구를 진행하고 있으며 이를 기반으로 Top500[1]의 주요 HPC를 선보이고 있다. 예로써 중국의 텐허는 TH Express-3[2]를 일본의 Kei는 Tofu[3]를 사용하고 있으며, 미국 또한 타이탄의 Cray Gemini와 Intel의 Omni-Path와 같은 독자적인 상호연결망 시스템을 사용하고 있다.

고성능 수퍼컴퓨터를 자체적으로 개발하기 위해서는 기술 의존성이 높은 인터커넥트 기술인 Infiniband나 Ethernet 인터컨넥트가 아닌 대체 인터컨넥트 기술로써, Low Power와 High Performance 기능을 달성할 수 있는 PCIe 기반 기술을 고려할 수 있다. PCIe(Peripheral Component Interconnect Express)[4, 5, 10] 기술 표준은 컴퓨터 CPU와 주변 장치를 고속으로 연결하여 IO 성능을 높이기 위한 인터페이스로 인텔에 의해서 개발된 기술 표준으로써 최근 PCIe 버전 5까지 제안되고 있으며 현재 널리 사용되는 표준은 PCIe3 이다. PCIe 인터페이스 기반의 네트워크 스위칭 기술의 경우, PCIe 스위칭 칩 제조사의 패브릭 네트워크 기능에 대한 기술지원이 원활하게 이루어지지 못함에 따라 PCIe 스위칭 기술을 인터커넥션 네트워크로 활용하기 위한 대체 방안으로는 PCIe 표준기술 부분 중에서PCIe NTB기술을 활용하는 방안을 연구 개발할 필요가 생기게 되었다.

PCIe NTB는 PCI-Express bridge chip에서 지원하는 모드 중 하나로서[6, 7], 2대 컴퓨터의 서로 분리된 메모리 시스템을 같은 PCI-Express fabric으로 연결시키는 기술이다. 그런데, 현재까지 진행된 PCIe NTB 기반 무스위치 인터컨넥션 네트워크 기술은, 2대의 호스트간에 서로 분리된 메모리에 접근하도록 구성하는 연결방식에 대해서 연구 개발이 이루어져 왔다. 3대 이상의 호스트간에 서로 부리된 메모리를 구성하도록 하는 다중 호스트 기반의 NTB 인터컨넥션 기술에 대한 연구는, 스위칭 기반의 NTB 연결방식에 대한 연구 및 직접적인 호스트간 연결에 의한 클러스터 구성에 대한 연구가 진행중이나 그 결과물이 미비한 상황이다[8, 9].

본 연구개발에서는 PCIe의 NTB기술을 활용해서 다중 호스트 기반의 PCIe 무스위치 인터컨넥션 네트워크(Switchless Interconnection Network)에 통신 구성 방법 및 데이터 공유 방법에 대해 연구한다. 구체적으로는 다중 포트 기반의 호스트간 직접적인 NTB 연결을 통하여 링 토폴로지를 구성하고, NTB 기반의 다중 호스트 클러스터 내에서 데이터 공유 방법에 대한 설계 및 구현을 하도록 한다.

본 논문의 구성은 다음과 같다. 2장에서 본 연구에 대한 배경을 소개하고, 3장에서 본 연구에서 구현한 PCIe NTB 기반 무스위치 인터커넥트 네트워크를 설명한다. 4장에서 실험결과에 대해서 설명하고 5장에서 결론을 맺는다.

2. 배 경

2.1 PCIe Non-Transparent Bridge

PCIe NTB는 PCI-Express bridge chip에서 지원하는 모드 중 하나로서, 2대 이상의 컴퓨터의 서로 분리된 메모리 시스템을 같은 PCI-Express fabric 으로 연결시키는 기술이다. PCIe NTB는 TB(Transparent Bridge)와 마찬가지로 독립적인 PCI bus(PCI 또는 PCI Express bus)에 대해서 데이터 전송 경로(path)를 제공한다는 점에서 유사하다. 그러나, TB와의 가장 큰 차이점은 NTB가 사용될 경우 bridge의 하향부분(downstream side)에 위치한 장치들은 상향 부분(upstream side)에서는 보이지 않는다는 점이다. 이는 bridge의 하향부분(downstream side)에 위치한 인텔리전트(intelligent)한 시스템이 자신의 downstream side에 위치하는 서브시스템 내 각종 장치들을 독립적으로 관리할 수 있다. Fig. 1은 PCie NTB Interconnect에 의해서 두 CPU 간의 연결에 대한 구성도와, NTB로 연결된 두 Host CPU간의

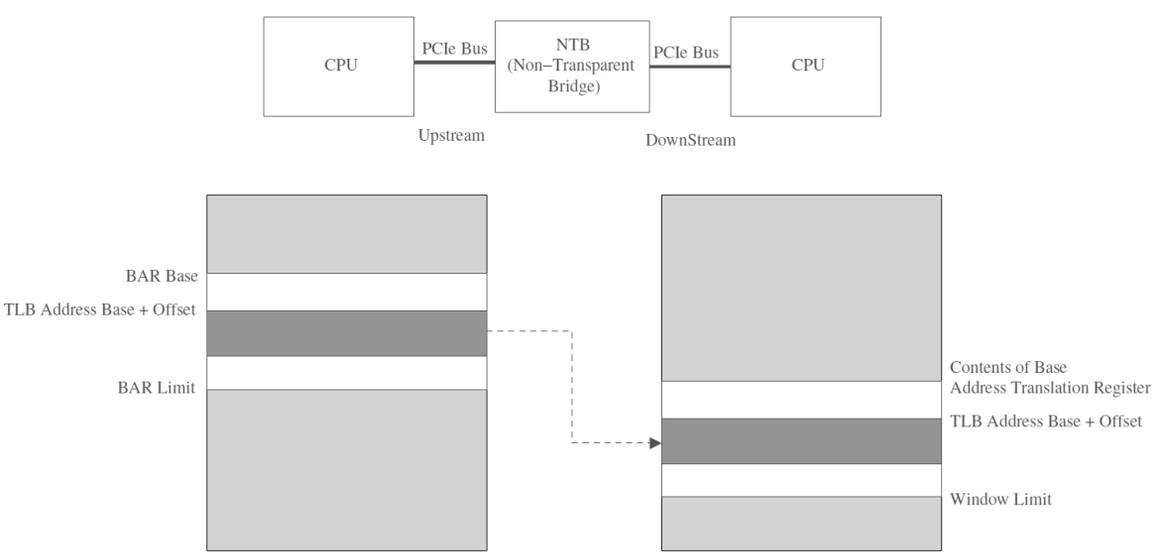


Fig. 1. PCIe NTB(Non-Transparent Bridge)-based Interconnect and Address Translation Process

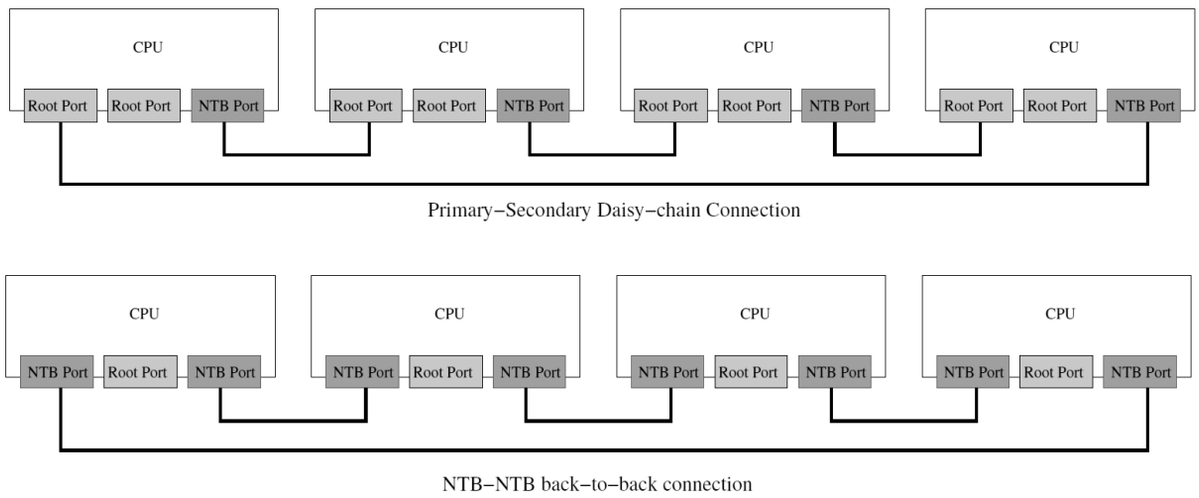


Fig. 2. PCIe NTB-based Switchless Interconnect Network Topologies

분리된 메로리 영역을 이용한 주소 변환 방식에 의한 데이터 공유 방법을 도식화한 것이다.

두 대의 컴퓨터를 NTB를 이용해서 연결하는 방법은 Fig. 2와 같이 Primary-Secondary 연결 방식과 B2B(Back to Back) 연결 방식을 생각할 수 있다. 첫 번째로 Fig. 2의 위쪽 그림과 같이, PCIe NTB는 또한 primary host의 PCI bus로 구성된 서브 시스템 계층구조에 secondary host를 연결하는 데 사용될 수 있다. 두 번째로, Fig. 2의 아래쪽 그림과 같이 Back to Back(B2B) NTB를 두 호스트 간에 연결하게 되면, 하나의 호스트의 primary address가 NTB에 의해서 secondary address로 변환되어 해당 호스트의 address space에 보이지 않게 되며, 마찬가지로, secondary address 역시 다른 쪽 host의 NTB에 의해서 address map에 포함되지 않게 된다. 이 경우, 어느 한쪽에서 doorbell 인터럽트를 발생시키거나 scratchpad register를 통해서 interprocessor communication을 하기 위해서는 secondary side의 BAR 0/1를 접근하여 해당 doorbell register와 scratchpad register에 값을 쓸 수 있다.

3. NTB 기반 Switchless Ring 네트워크

앞서 언급한 바와 같이 NTB는 기본적으로 두 대의 호스트가 서로 다른 메모리 영역을 통해서 데이터를 공유하거나 전송하는 방식을 제공한다. 즉, 이를 더 확장하여 세 대 이상의 다중 호스트간에 NTB 인터페이스를 통하여 데이터를 공유할 수 있는 시스템을 구성하기 위해서는 기존의 NTB로는 어려우며 부가적인 구현이 필요하다. 본 장에서는 기본적인 NTB 표준 인터페이스 기반으로 다중 호스트를 서로 연결하는 NTB 기반의 다중 호스트 시스템을 구성하는 방법과 구현에 대해 기술하도록 한다. NTB 기반 다중 호스트를 구성하기 위해서, 하나의 호스트에 두 개 이상의 NTB 포트를 위한 호스트 아답터를 연결하고, 각 호스트의 포트를 양쪽의 다른 호스트에 연결하는 방식으로 세 대 이상의 호스트를 링 형식으로 연결하여 다중 호스트를 구성한다. Fig. 2는 NTB host adapter를 이용하여 다중 호스트를 묶어서 Ring과 같은 형태의 Interconnect network system 구성을 함께 도식화한 것이다. 그림과 같이, 이러한 연결 구성 방식도 위쪽 그림과 같은 Primary-Secondary 연결 방식으로 구성할 수도 있으며, 아래쪽 그림의 NTB Back-to-Back 방식으로 구성할 수도 있다.

Fig. 2에서 도식한 바와 같이, NTB를 이용하여 토폴로지를 구성하면, 각 호스트당 각각의 NTB를 이용해 연결된 다른 쪽의 호스트에 대해서 분리된 메모리 어드레스 영역을 통해서 데이터 공유를 할 수 있다. 그러나, 링 토폴로지에서 자신에게 직접 연결되지 않은 호스트와 데이터를 주고받거나 공유하기 위해서는 해당 목적지 호스트를 분별해주기 위한 Id 와 같은 정보들이 필요하게 된다. 즉, 링 토폴로지로 구성된 네트워크에서 각 호스트간 데이터 송수신 및 공유를 하는 방법은 각 호스트에 할당된 Id와 메모리 영역의 주소를 통해서 데이터를 송수신하거나 공유할 수 있다. 그러한 정보를 위해서 링 네트워크 구성 시에 각 호스트의 Id와 공유 메모리 영역을 할당하고 교환하는 단계를 수행하게 된다. Fig. 3은 초기 링 구성 시 각각의 호스트에서 설정해주는 Id와 메모리 윈도우를 나타낸 것이다. 그림과 같이 초기 설정 시 각각의 호스트는 Id를 부여받게 되며, 각각의 호스트의 좌, 우 NTB 포트당 연결된 호스트들의 NTB와 데이터를 주고받을 메모리 윈도우를 할당한다. 할당된 메모리 윈도우는 NTB의 Translation Register를 통해서 address translation 되어 상대방 호스트의 메모리에 접근할 수 있는 통로가 된다. 할당된 Id와 공유 메모리 윈도우는 각각 좌,우의 NTB와 연결된 호스트와 그 정보를 주고받는다. 이 때 정보교환은 ScratchPad Register를 통해서 이루어진다. 이렇게 정보를 주고받음으로써 링 네트워크에서 자신의 주변에 존재하는 호스트 Id와 공유 메모리 윈도우에 대한 테이블을 구성한다.

이렇게 셋업 단계를 거치면, 각 호스트는 데이터 전송을 원하는 호스트에게 데이터를 전송하거나 전송받을 수 있다. NTB로 구성된 Ring Topology 네트워크에서 데이터 전송 방향은 한 방향 통신을 통해서 데이터를 주고받으며, 데이터 전

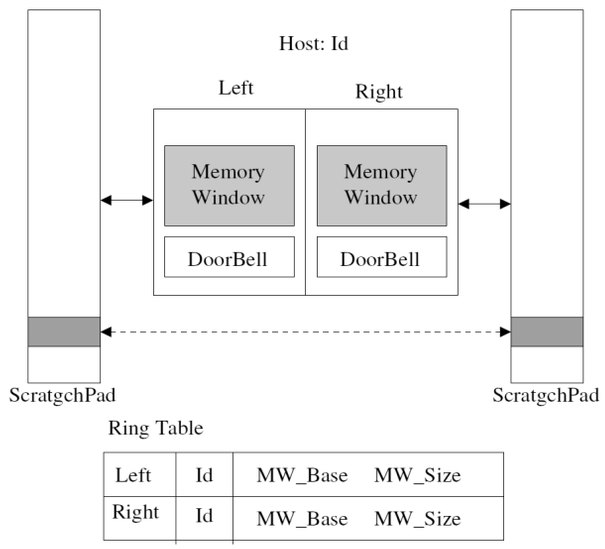


Fig. 3. Initial Ring Setup Procedure for NTB-based Ring Topology

송 방법은 연결된 호스트간의 NTB로 공유된 공유영역을 통해서 데이터 주소 변환을 통한 메모리 간 직접 복사하는 방식으로 데이터를 전송할 수 있게 된다. 전송하는 방식은 공유 데이터 영역의 메모리 복사 또는 DMA 둘 다 가능하다. 데이터 전송이 아닌 일반 정보를 주고받을 때에는, 둘 간에 연결된 NTB의 ScratchPad register를 통해서 전송할 수 있다. NTB의 한쪽 호스트에서 ScratchPad register에 쓴 데이터는 NTB의 상대측 호스트에서 해당 ScratchPad register에 직접 접근하여 데이터를 가져갈 수 있다.

데이터 송수신 프로토콜 및 전송 방법은 Fig. 4에 도식화하였으며, 다음과 같은 데이터 프로토콜을 통해서 수행된다. 특정 호스트에서 데이터를 전송하려고 할 때, 해당 호스트는 송신자가 되며, SrcId를 자기 자신으로 설정하며, 수신하려는 호스트의 Id를 지정하여 DestId로 설정한다. 데이터 송수신은 링 네트워크에서 둘 간의 위치에 따라서 약간 차이가 있다. 만약, DestId가 NTB 어드레스 변환을 통해서 데이터 직접 전송이 가능한 자신과 직접 연결된 호스트이면, 해당 데이터는 둘 간의 직접적인 공유 데이터 메모리에 데이터 쓰기 수행을 통해서 데이터 전송 시작한다. 그 후, SrcId, DestId, 메모리 주소, 전송 크기 등의 메타 정보를 ScratchPad Register를 통해서 전달한다. 데이터 전송이 완료되면, Doorbell Register를 통해서 해당 호스트에게 데이터 전송의 완료를 알리는 인터럽트를 발생시켜, 상대 호스트에게 전송된 데이터가 있음을 알린다. 인터럽트를 받은 호스트는 ScratchPad Register를 통해서 자신이 수신자인지 알게 되면, 자신이 수신자인 경우, 해당 메모리 주소로부터 직접 자신에게 전송된 데이터를 볼 수 있게 된다.

만약 자신과 직접 연결되지 않은 호스트, 즉 이웃이 아닌 멀티 홉 떨어져 있는 대상 호스트에게 데이터를 전송하려면, 직적접으로 데이터 전송을 할 수 없는 구조이므로, 이웃에게 데이터를 전달하는 방식으로 data propagation을 해주어야 한다. 즉, DestId에 전송하려는 호스트의 Id를 기재하여 NTB로 직접 연결된 이웃 호스트의 공유 메모리로 데이터를 전송한다.

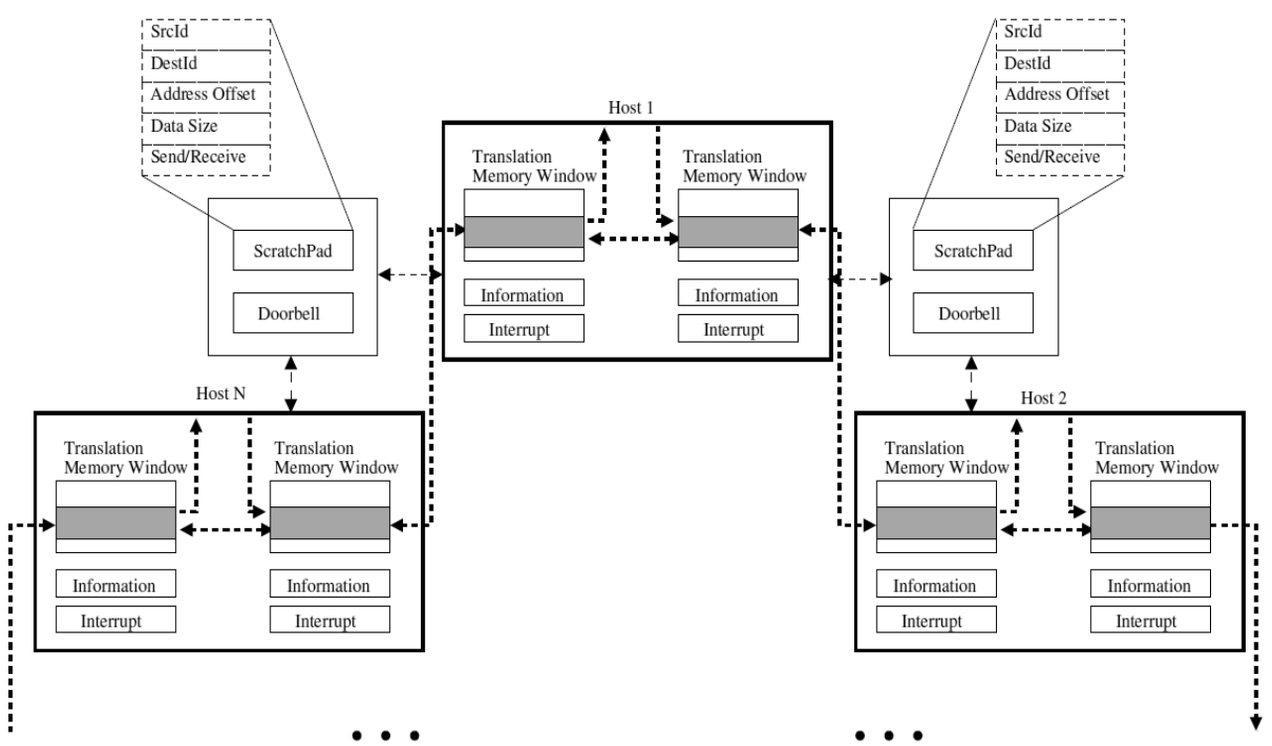


Fig. 4. Data Transfer between hosts with Memory Window, ScratchPad, and Doorbell for PCIe NTB-based Ring Network

또한, 직접 연결된 호스트끼리의 전송과 마찬가지로 SrcId, DestId, 메모리 주소, 전송데이터 크기 등도 propagation하기 위해서 ScratchPad Register로 이웃에게 보낸다. 그 후, 이웃에게 데이터 전송이 완료되었을 시에, Doorbell Register를 이용해서 이웃에게 인터럽트를 발생시킨다. 인터럽트를 받은 이웃 호스트는 DestId를 확인해서, 자신이 송신자가 아님을 확인하면, 전송받은 데이터를 다른 이웃에게 데이터를 포워딩해준다. 데이터를 포워딩하는 방식은 마찬가지로 다른 이웃과 공유된 메모리 영역을 통해서 DMA로 전송이 가능하다. 이와 같이 수신자 호스트가 데이터를 전송받을 때까지 계속 이어진다. 수신자 호스트가 인터럽트를 통해서 데이터 전송을 받고, ScratchPad Register를 통해서 수신자가 자신임을 확인하면, 비로소 데이터 전송이 완료된다. 이러한 방식으로 NTB 기반의 인터커넥트를 통해서 다중 호스트로 구성된 네트워크 시스템에서 각 호스트간의 데이터 공유를 할 수 있게 된다.

4. 실 험

구현된 NTB 기반 무스위치 링 네트워크의 대역폭 등 성능을 평가하기 위해서 실제 시스템을 구성하였다. NTB Ring Network Topology를 구성하기 위해서, 3대의 Host PC 각각에 두 개의 PCIe interface bus에 NTB Host Adapter를 연결하고, NTB Host Adapter를 서로 연결하여 링 네트워크를 구성하였다. 연결에 사용된 NTB Host Adapter Card는 PLX사의 PEX8733, 또는 PEX8749 NTB Chipset을 기반으로 KISTI에서 자체 제작한 NTB Host Adapter Card와, PLX사의 PEX8749 NTB Chipset 기반의 RDK보드를 활용하였다[12, 13]. PEX8749와 PEX8733 Chipset은 거의 동일한 NTB controller를 내장하고 있으며, 동일한 Device Driver와 DMA Driver 및 설정 관련 모듈을 사용하기 때문에 동일한 소프트웨어 모듈을 사용하여 구현 하는데에 문제는 없다.

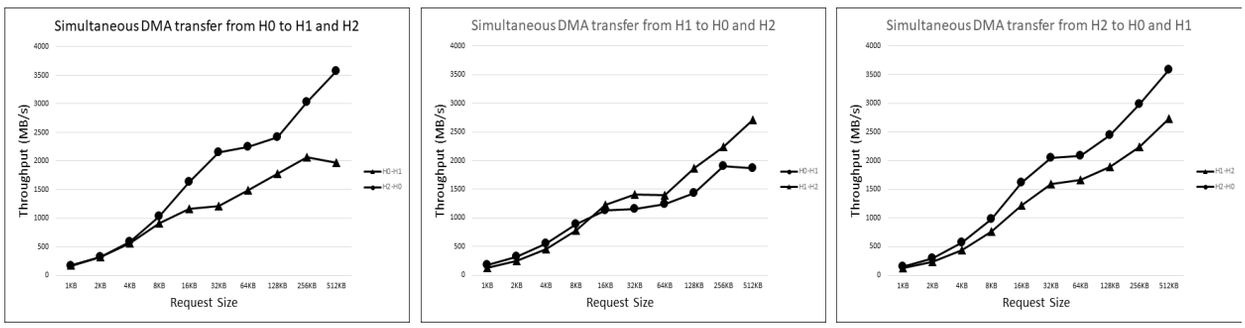
각 Host PC는 Intel i7-8700K 3.7GHz CPU와 16GB DDR RAM과 2개의 PCIe3.0x16 인터페이스 이상의 메인보드로 구성되어 있다. 각 Host PC의 2개의 PCIe3.0x16 인터페이스 각각에 PEX8749 Host adapter, PEX8733 Host adapter 및 PEX8749 RDK 보드를 연결하였다. 세대의 Host는 각기 다른 Host Adapter를 연결하도록 구성하였는데, Host0는 PEX8733 Adapter Card와 PEX8749 RDK를 연결하였고, Host1은 PEX8749 Adapter Card, PEX8749 RDK를 연결하였으며, Host2는 PEX8749 Adapter Card, PEX8733 Adapter Card를 각각 연결하였다.

구현된 무스위치 Ring Network에서 각각 연결된 호스트 사이의 데이터 전송 속도와 네트워크 트래픽 간의 상관관계를 알아보기 위해서, NTB DMA 전송 테스트를 실시하였다. 먼저, 단일 호스트간 DMA 전송 속도를 측정해보기 위해서 각 호스트 간의 독립적인 DMA 전송 실험을 진행하였다. 단일 호스트 간 DMA 전송 실험은 데이터 전송이 오직 두 호스트 간에만 이루어지고 이외의 다른 데이터 전송을 하지 않은 것을 의미한다. 두 번째로 한 DMA 전송 실험은 하나의 호스트에서 연결된 두 호스트에게 동시에 DMA 데이터 전송을 하는 실험이다. 세 번째 실험은 네트워크에 연결된 모든 호스트 간에 동시에 NTB를 통한 DMA 데이터 전송을 수행하는 실험을 진행했다. 각각의 DMA 전송 실험에서 전송되는 DMA 데이터의 크기는 1KB에서부터 512KB까지 단계적으로 증가하면서 진행하였다.

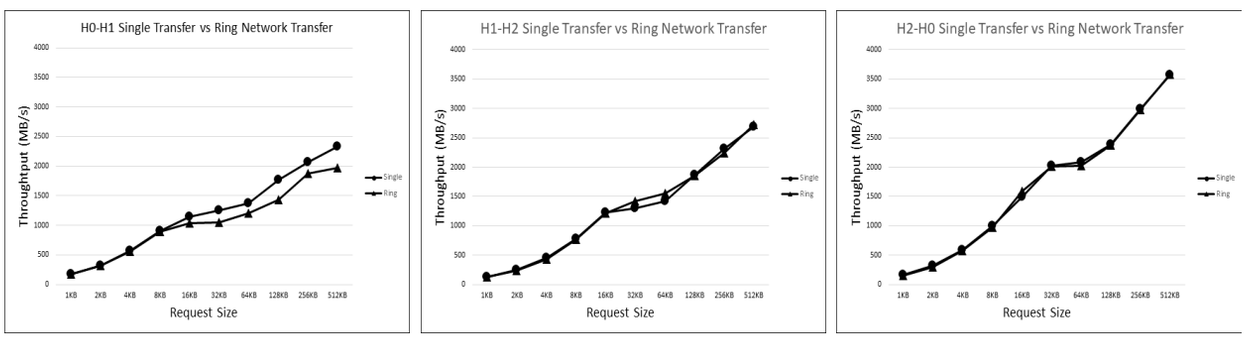
Fig. 5에서 도식화한 각 그래프는 각 호스트간 독립된 DMB 데이터 전송과 Ring Network에서의 세 호스트간 동시에 DMA를 전송하는 실험 결과를 그래프로 작성한 것이다. 그림에서 PE#는 해당 호스트의 Id이고, H#-H#은 Host 들간의 NTB 연결을 나타내는 것이다.

먼저 일반적인 실험 내용에 대해서 그 내용을 기술하면 다음과 같다. Fig. 5에서 보는 바와 같이 각 호스트 간 DMA 데이터 전송 속도는 전송 크기가 늘어날수록 커지는데, 512KB의 경우 최대 3500MB/s(3.5GB/s) 정도의 속도가 측정되었으며, 이는 H2-H0 간의 PEX8733 Host Adapter를 사용하였을 때 측정되는 전송 속도이다. 이외의 다른 NTB 연결은 PEX8749 Chipset으로 연결되어 있는데, 이때 측정된 전송속도는 최대 2.6GB/s 정도의 최고 전송 속도를 나타내었다. 이는 본 논문에서 구성한 NTB 기반 Ring Network가 서로 다른 NTB Chipset을 사용한 heterogeneous 한 환경으로, 연결 구성에 따라서 약간 다른 전송 속도를 보임을 확인할 수 있다. 그리고, 그래프에 도식화하지는 않았지만, 1MB 이상의 크기에 대해서 DMA 전송을 수행했을 때 오히려 전송 속도가 크게 저하됨을 확인할 수 있었다. 그 이유는, 본 실험에서 구성된 NTB 모듈의 PCI DMA buffer의 크기가 최대 1MB인 host adapter device가 존재하는데, 이런 물리적인 buffer의 크기로 인해서 1MB 이상의 DMA에 전송에 대해서 정상적인 DMA 전송이 이루어지지 않은 것으로 유추할 수 있었다. 그러한 이유로 인해서 모든 전송의 실험은 512KB를 최대 크기로 설정하고 실험을 진행하였다.

먼저, Fig. 5(a)는 각각의 호스트에서 링 네트워크에 접속된 각 NTB 호스트와 동시 데이터 전송을 할 경우, 각 호스트당 전송 대역폭을 도식화한 것이다. 그래프에서 보면, 각 호스트에 연결되어 있는 서로 다른 호스트간의 전송 대역폭이 각각 다른 것을 확인할 수 있는데, 이것은 각 호스트에 연결되어 있는 NTB Adapter의 Chipset이 다르기 때문으로 해석할 수 있다. Host0의 경우 PEX8749 Adapter와 PEX8732 Adapter를 이용해서 각각 Host1와 Host2로 연결되어 있는 환경인데, Fig. 5(a)에서 보는 바와 같이 Host2-Host0가의 데이터 전송 대역폭이 Host0-Host1에 비해서 큰 것을 볼 수 있다. Host1의 경우는 PEX8749 Chipset을 이용해서 둘 다 연결되어 있기 때문에 Host0-Host1간의 대역폭이 Host1-Host2간의 대역폭과 유사하게 나타남을 확인할 수 있다. 다음으로, Fig.5(b)는 호스트간 데이터 전송 대역폭을 단일 호스트 간에서 발생한 대역폭과, Ring Network에서 동시 데이터 전송을 수행할 때 발생하는 단일 호스트 간 대역폭과 비교하여 그래프로 도식화한 것이다. 그래프에서 보는 바와 같이 동시 데이터 전송을 수행할 때 측정된 호스트 간 대역폭이 동시 데이터 전송이



(a) Experimental Results for Data Transfer between single host VS. Ring network



(b) Comparison between Data Transfer between two hosts with Single Connection VS. Ring Network

Fig. 5. Experimental Results for DMA data transfer between hosts with Single Connection VS. Ring Network

발생하지 않고 호스트 간에만 데이터를 전송할 때보다 작게 나옴을 확인할 수 있다.

마지막으로 각 호스트 간 개별적으로 데이터 전송을 위한 DMA를 실행했을 때와 Ring Network에서 모든 호스트가 동시에 DMA를 실행했을 때 각 호스트에서 측정된 DMA 대역폭을 분석하여 Fig. 6에 도식화하였다. Fig. 6의 첫 번째 그래프에서 보는 바와 같이, 각 호스트 간 데이터 전송 실험을 하였을 경우, 각기 연결된 NTB Host Adapter의 chipset에 따라서 성능 차이가 나긴 하지만 대체적으로 전송 속도가 증가하는 것을 확인할 수 있다. Fig.6의 두 번째 그래프는 Ring Network에서 Host0-Host1-Host2 동시 DMA 전송하는 실험결과를 도식화한 것이다. 이러한 데이터로부터 동시 데이터 전송이 단일 전송에 비해서 전송에 미치는 영향을 분석할 수 있다. Ring Network에서 다중 호스트의 동시 데이터 전송은 호스트의 CPU에서 처리해야 하는 작업량이 늘어남에 따라서 단일 호스트 당 대역폭이 감소함을 확인할 수 있다. 그 전송 대역폭의 감소 폭은 단위 전송의 크기가 증가할수록 점점 증가함을 확인할 수 있는데, 이것은 단일 호스트가 단일 데이터 전송에 활용되는 대역폭의 증가량이 다중 접속 및 전송에 의한 대역폭의 증가량에 비해서 가파르기 때문에 상대적으로 확대되는 것이라 볼 수 있다.

이는, 처리해야 하는 데이터양이 많아질수록 동시 처리해야 하는 호스트의 오버헤드가 상대적으로 많이 늘어난다는 것을 의미한다. 그러나, 그래프의 결과에서 확인할 수 있듯이, 다중 접속 시에도 호스트 간 절대적인 데이터 전송 대역폭의 감소는 많지 않음을 알 수 있다. 즉, 무스위치 링 네트워크에서

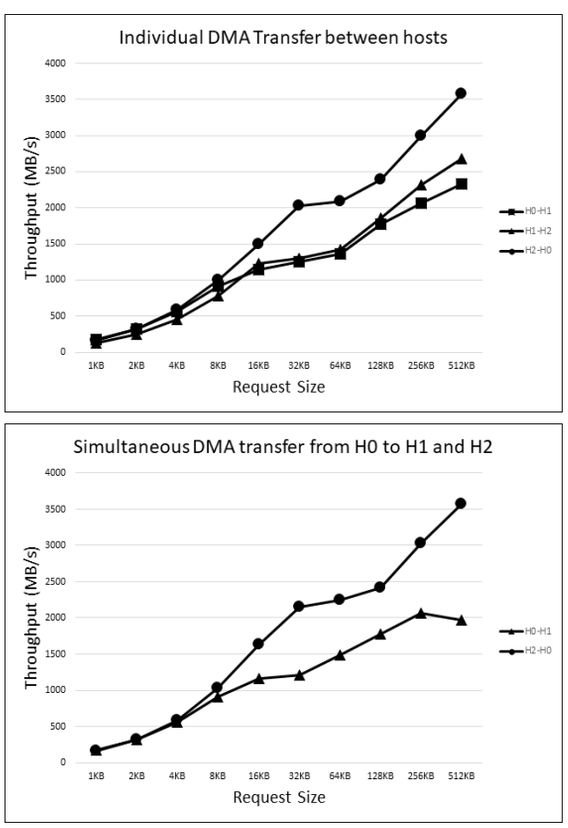


Fig. 6. Experimental Results and Comparison Between Each Individual Data Transfer Between Two Hosts VS. Simultaneous Transfer Among Three Hosts in Ring Network

단일 호스트에서 다중 포트에 대한 동시 데이터 처리의 오버헤드가 크지 않아서 기존 NTB port가 낼 수 있는 대역폭 대부분을 동시에 활용할 수 있다는 의미라 할 수 있다. 단일 호스트 간의 전송은 두 호스트에서만 사용할 수 있는 네트워크이며, 본 과제에서 수행하는 NTB 기반 무스위치 네트워크 시스템의 목표는 다중 호스트 간의 데이터 공유 및 전송 가능성에 대한 것이므로, 무스위치 네트워크에 연결된 호스트 끼리의 단위 시간당 다중 동시 데이터 전송량은 전체 네트워크의 전송량으로 환산할 경우, 훨씬 전송량이 늘어남을 의미한다. 그러므로 무스위치 네트워크에서도 대역폭의 증가를 활용할 수 있을 것으로 기대한다. 그러나, 링 및 2D Mesh와 같은 무스위치 네트워크는 호스트간 데이터 전달에 의해서 Latency는 길어질 가능성이 있으므로 이를 잘 고려해야 할 것으로 보인다.

5. 결 론

HPC 시스템은 다수의 계산노드를 고성능 인터컨넥트로 연결하여 클러스터 시스템으로 구성된 컴퓨팅 시스템이다. 대부분의 HPC 시스템은 다수의 계산노드(Computation Node)를 Infiniband, Ethernet과 같은 고가의 고성능 상호연결망(Interconnection Network)로 연결하여 클러스터 시스템을 구성하고 있으나, 기술 종속적인 측면으로 인해서, 이를 대체할 수 있는 인터커넥트 기술의 확장 및 적용이 필요하다. 두 개의 노드 간에 메모리 공유를 통해서 쉽게 데이터 송수신을 할 수 있는 인터커넥트 기술로서 PCIe NTB 기술이 있다. 그런데, 기본적인 PCIe NTB는 두 노드 간의 분리된 메모리 영역을 통해서 데이터 공유를 할 수 있기 때문에, 다수의 노드를 연결하여 구성하기가 쉽지 않다. 본 논문에서는 다중 NTB 포트로 구성된 다수의 노드를 상호 연결하여 링 네트워크를 구성한 후, 네트워크 내의 노드 간의 상호 데이터를 공유할 수 있는 시스템을 구현하였다. 이러한 구현을 통해서 PCIe NTB 기반의 Cost-Effective하면서 고속의 데이터 공유가 가능한 클러스터 구성이 가능하다.

References

[1] Top500.org.: Interconnect Family Statistics [Internet], http://top500.org/statistics/list, 2015.

[2] Tianhe-3 Tianhe-3 Exascale Supercomputer Prototype [Internet], https://medium.com/syncedreview/one-billion-billion-tianhe-3-exascale-supercomputer-prototype-passes-tests-7d30aa97aca2.

[3] POST-K Supercomputers Tofu D interconnect [Internet], https://www.nextplatform.com/2018/09/14/slicing-into-the-post-k-supercomputers-tofu-d-interconnect/

[4] Helal, A.A., Kim, Y.W., Ren, Y., and Choi, W.H., "Design and implementationof an alternate system inter-connect based on PCI Express," *J. Inst. Electron. Inform. Eng.*, Vol.52, No.8, pp.74-85, 2015.

[5] Liu, J., Mamidala, A., Vishnu, A., and Panda, D.K., "Evaluating infiniband performance with PCI Express," *IEEE Micro*, Vol.24, No.1, pp.20-29, 2005.

[6] Heymian, W., "PCI Express multi-root switch reconfiguration during system operation," M. Eng. Thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, 2011.

[7] Krishnan, V., "Towards an integrated IO and clustering solution using PCI express," *2007 IEEE International Conference on Cluster Computing* [online], pp.259-266, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4629239, 2007.

[8] Mohrmann, L., Tongen, J., Friedman, M., and Wetzel, M., "Creating multicomputer test systems using PCI and PCI Express," *IEEE AUTOTESTCON* [online], pp.7-10, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5314043, 2009.

[9] Choi, M. and Park, J.H., "Feasibility and performance analysis of RDMA transfer through PCI Express," *J. Inform. Process. Syst.*, Vol.13, No.1, pp.95-103, 2017.

[10] Rota, L., Caselle, M., Chilingaryan, S., Kopmann, A., and Weber, M., "A new DMA PCIe architecture for Gigabyte data transmission," *Real Time Conference(RT), 2014 19th IEEE-NPSS* [online], pp. 1-2, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7097561, 2014.

[11] Richter, A., Herber, C., Wild, T., and Herkersdorf, A., "Resolveing Performance Interference in SR-IOV Setups with PCIe Quality-of-Service Extensions," *Euromicro Conference on Digital System Design*, pp.454-462, 2016.

[12] ExpressLane PEX8749 PCI ExpressGen 3 Multi-Root Switch with DMA Data Book, PLX Technology, 2013.

[13] Cheol Shim, Kwang-ho Cha, and Min Choi, "Design and implementation of initial OpenSHMEM on PCIe NTB based cloud computing, *Cluster Computing*, pp.1-12, DOI: https://doi.org/10.1007/s10586-018-1707-0, 2018.

[14] J. Respondek, "Numerical approach to the non-linear diofantic equations with applications to the controllability of infinite dimensional dynamical systems," *International Journal of Control*, Vol.78, No.13, pp.1017-1030. 2007.

김 상 겸

http://orcid.org/0000-0002-6702-4598

e-mail : interpost94@gmail.com

2013년～현 재 한국외국어대학교 컴퓨터・전자시스템공학부 학사과정

관심분야 : Storage Network, Interconnect Network, Machine Learning

이 양 우

http://orcid.org/0000-0001-7490-4473
e-mail : ddonhlyw@naver.com
2013년～현 재 한국외국어대학교 컴퓨터・전자시스템공학부 학사과정
관심분야 : Operating System, Storage Network, Embedded System

임 승 호

http://orcid.org/0000-0003-3096-0785
e-mail : lim.seungho@gmail.com
2001년 KAIST 전기 및 전자공학과(학사)
2003년 KAIST 전기 및 전자공학과(석사)
2008년 KAIST 전기 및 전자공학과(박사)
2008년～2010년 삼성전자 메모리사업부 책임연구원
2010년～현 재 한국외국어대학교 컴퓨터・전자시스템공학부 교수
관심분야 : Operating System, Storage Network, Interconnect Network, Flash Memory