PCIe 스위치 기반 NTB의 통신 성능 분석

차광호[1], 주영인[2], 황준[2], 최현태[3]
[1]한국과학기술정보연구원
[2](주)신보
[3]국방기술진흥연구소
khocha@kisti.re.kr, yijoo@shinbo.kr, jhwang@shinbo.kr, gusxo1004@krit.re.kr

Communication Performance Evaluation of PCIe Switch based NTB

Kwangho CHA[1], Young-In JOO[2], Joon HWANG[2], Hyuntae CHOI[3]
[1]Korea Institute of Science and Technology Information
[2]SHINBO Co., Ltd.
[3]Korea Research Institute for defense Technology planning and advancement

요 약

다양한 AI 서비스들의 확산으로 특수 목적형 고성능 시스템에 대한 관심이 증가하면서, PCIe 버스를 인터커넥션 네트워크로 사용하는 사례가 늘고 있다. 본 연구에서는 NTB를 활용한 PCIe 네트워크의 통신 성능을 TCP를 사용하는 방법과 NTB 모듈에 직접 접근하는 방법으로 나누어 측정하고 그 결과와 의미를 분석하였다. 두 가지 방법들이 성능에서 상이한 패턴을 보이기 때문에 PCIe 네트워크를 사용할 시스템의 성능 요구 사항에 따라 그에 부합하는 통신 방법을 선택해야 할 것으로 예상된다.

1. 서론

컴퓨터 시스템에서 PCIe 버스는 오랫동안 시스템 버스의 역할을 수행하고 있다. 또한 최근에는 언어 모델, 자율 주행 등의 다양한 AI 기반 서비스들이 확산됨에 따라, 특수 목적형 고성능 시스템에 대한 관심이 증가하고 있는데 이런 특수 목적형 시스템에서는 운영환경의 제약으로 인하여 PCIe 버스를 직접 인터커넥션 네트워크로 사용하는 방법이 자주 이용되고 있다. 본 연구에서는 NTB를 사용하는 PCIe 네트워크의 통신 성능을 여러 방법으로 측정하고 그 결과와 의미를 분석하였다.

2. PCIe 버스와 NTB

PCIe(Peripheral Component Interconnect Express)는 PCI(1992년)와 PCI-X(1999년)와의 호환성 및 확장성을 고려해서 개발된 후속 버스로서 2002년 PCIe 1.0이 발표되었다. 메모리, IO 공간 및 설정 공간(Configuration Space)에 대한 소프트웨어적인 접근이 가능하고 표 1처럼 2022년 6.0 규격까지 발표되었으며 2025년 발표를 목표로 현재 7.0 규격이 정의되고 있다[2].

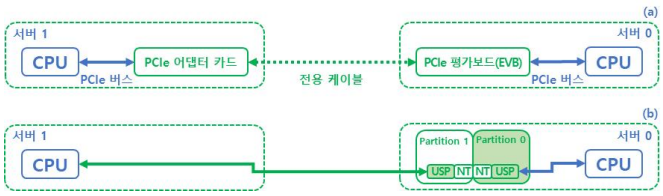
PCIe 버스의 주된 기술적 특징으로는 양방향 점대점 시리얼 통신 수행, 분기(Bifurcation) 확대를 통한 확장성 제공 및 패킷 방식의 전송 프로토콜 사용을 생각해 볼 수 있다[2].

NTB(Non-Transparent Bridging)는 PCIe 스위치를 사용하여 구현할 수 있는 특수 기능 중 하나로 특정 포트에 장착된 디바이스에 대한 격리 (isolate)기능을 제공한다. 원래 고장 감내 시스템의 구성 요소로 활용하기 위해 고안되었으며 각종 레지스터(Door Bell, Scratchpad, BAR 등)들과 주소 및 ID 변화 기능을 제공한다.

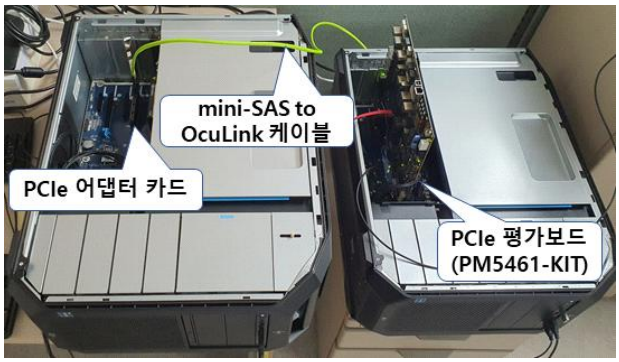
NTB 기능을 활용하면 PCIe를 노드 간 통신용 네트워크(인터커넥션 네트워크)로 사용하는 것이 가능해지며 HPEC(High Performance Embedded Computing) 서버와 같은 특수 목적형 고성능 시스템에서 PCIe 기반 인터커넥션 네트워크를 사용하고 있다[3].

<표 1> PCIe 버스의 주요 제원[1]

PCIe 버전	데이터 전송률 (GT/s)	인코딩 방식	대역폭 (단방향 x16 기준, GB/s)
1.0	2.5	8b/10b	4
2.0	5.0	8b/10b	8
3.0	8.0	128b/130b	15.75
4.0	16.0	128b/130b	31.51
5.0	32.0	128b/130b	63.01
6.0	64.0	FLIT (PAM-4)	~128



(그림 1) 테스트 베드: (a) 물리적 구성, (b) 논리적 구성



(그림 2) 테스트 베드 구성

<표 2> 실험 환경 세부 구성

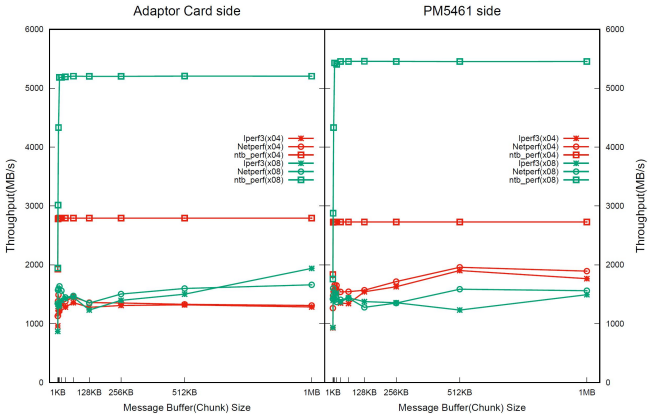
서버 노드	CPU	Intel Xeon w5-3423
	메모리	64GB (2x32GB) DDR5
	운영체제	Ubuntu 22.04.4 LTS linux kernel 5.15.0-43-generic
PCIe 3.0 EVB		Microchip PM5461-KIT
PCIe 4.0 EVB		Microchip PM42100-KIT

3. 실험 환경 설정

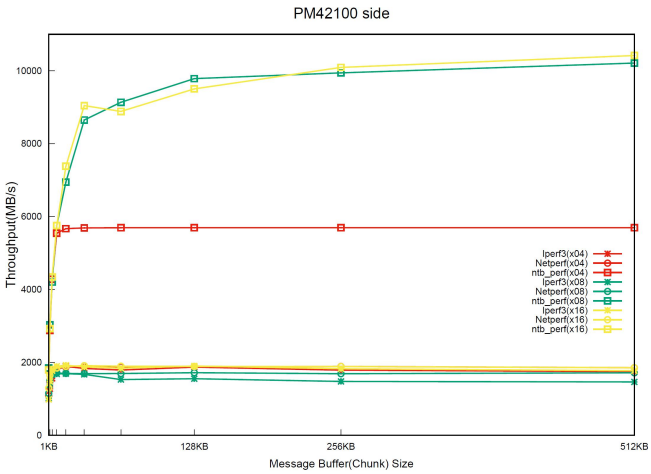
본 연구에서는 두 서버를 PCIe 평가보드상의 NTB 포트를 사용하여 연결하고 네트워크용 BMT 프로그램을 사용하여 통신 성능을 측정하였다. 이를 위해서 그림 1(a)와 그림 2와 같은 물리적인 연결을 구성하게 된다. 이때 그림 1(b)와 같이 서버 0의 PCIe 평가보드는 서로 다른 영역으로 나뉘어진 후 각 서버에게 할당된 PCIe 도메인을 NT 포트를 사용하여 연결하는 기능을 수행한다. 현 시점에서 확보 가능한 PCIe 3.0과 4.0 평가보드를 사용하였으며[1] 테스트 베드의 세부 구성 요소는 표 2과 같다.

네트워크 성능 측정에는 ipref3[4], Netperf[5] 및 ping과 같이 보편적인 방법으로 많이 사용되는 인터넷 성능 측정 벤치마크들을 먼저 사용하였다. 이는 특별한 미들웨어나 소프트웨어 도구의 도움없이 리눅스 배포판에 포함되어 있는 드라이버와 라이브러리를 사용해서 NTB 장치를 TCP 방식으로 통신하는 것이

1) Microchip社는 PCIe 버전별로 대표적인 EVB를 1~2개 정도 제공하고 있다.



(그림 3) PCIe 3.0 평가보드에서의 대역폭 측정 결과



(그림 4) PCIe 4.0 평가보드에서의 대역폭 측정 결과

가능하기 때문이다.

이와 같은 TCP over NTB에 기반을 둔 성능 측정뿐만 아니라 ntb_perf도 실험에 사용하였다. ntb_perf는 리눅스 소스 코드에 포함되어 제공되는 벤치마크 프로그램으로[6] NTB로 연결된 두 호스트간의 전송 성능을 측정할 수 있다. iperf나 Netperf와는 달리 커널 모듈의 형태로 동작하는 성능 측정 도구로서 커널 영역과 사용자 영역 간의 메모리 복사가 제외되어 있다는 한계가 있기는 하지만 NTB 기능을 직접 사용하는 통신 성능을 측정할 수 있다는 장점이 있다.

4. 성능 평가 및 분석

그림 3은 PCIe 3.0 평가보드(PM5461-KIT)을 사용한 대역폭 측정 결과이다. PCIe 버스의 분기 설정을 x4와 x8로 변경하면서 성능을 측정하였다. TCP over NTB 방식의 경우, 분기 설정과 상관없이 대역폭이 2GB/s를 넘지 못하는 모습을 보여 주었다. 반면 ntb_perf의 경우, 분기 설정으로 레인 수가 증가하는 경우, 측정되는 대역폭도 증가하였고 이론 성능 대비

약 64~69%의 성능을 보여 주었다.

그림 4는 PCIe 4.0 평가보드(PM42100-KIT)를 대상으로 측정한 결과이다. 이 환경에서도 TCP를 사용하는 방식은 분기 설정과 상관없이 측정 대역폭이 2GB/s를 넘지 못했다. ntb_perf의 경우, x4와 x8에서의 측정 대역폭 성능이 이론 성능 대비 약 70%와 63%에 도달하면서 어느 정도의 확장성을 보여 주었으나 x16에서의 성능이 x8의 성능과 유사하여 2배로 증가시킨 추가 레인들을 제대로 활용하지 못하는 결과를 보였다.

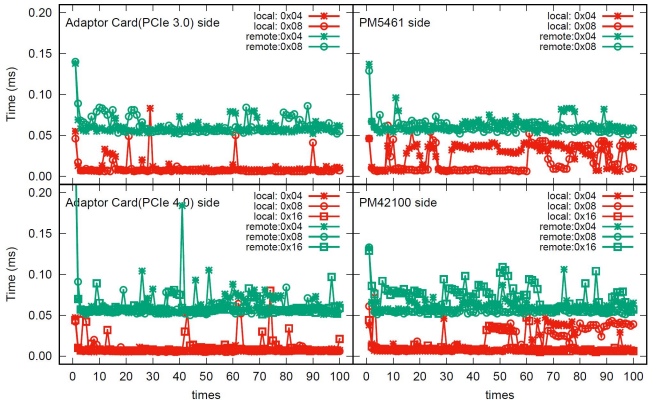
TCP를 사용하는 iperf나 Netperf을 사용하는 대역폭 측정에서 여러 실험 조건과 상관없이 특정 성능을 넘지 못하는 결과를 감안하면 TCP over NTB가 기존 TCP/IP를 사용하는 어플리케이션을 쉽게 NTB 네트워크 환경에서 구동시킬 수 있다는 장점은 있으나 성능의 확장성을 위해서는 적합하지 않은 방법이라는 사실을 유추할 수 있다. 반면 ntb_perf의 경우, x16 분기에서 확장성에 제약이 있다는 문제가 있기는 하지만 이론 성능 대비 63~70%의 성능을 보여 주었기 때문에 NTB 네트워크를 보다 효율적으로 사용하기 위해서는 ntb_perf처럼 NTB 전용의 레지스터를 직접 제어해서 통신하는 방식이 보다 적합할 것으로 판단된다.

TCP를 사용하는 경우의 지연시간을 테스트한 결과를 그림 5에 정리하였다. 노드 자신(local)과 NTB를 통하여 연결된 상대방(remote)을 대상으로 ping테스트를 수행한 결과인데 분기 설정과 PCIe 버전에 상관없이 유사한 결과를 보여 주었다. 즉 자기 자신에 대한 지연시간은 0.007ms를 상대방에 대한 지연시간은 0.053~0.057ms의 결과를 주로 보였다. 특이한 점은 PCIe 평가보드가 설치된 서버에서 자기 자신에 대한 지연시간의 편차가 다소 크게 나타난다는 것이며, 이는 측정 시점의 운영체제 상태에 영향을 받는 것으로 예상된다.

5. 결론

PCIe 버스를 사용하여 인터커넥션 네트워크를 구현하는 경우, NTB는 자료 공유 및 데이터 통신을 위한 주요 기능을 제공한다. 이러한 기능들은 NTB의 설정 및 제어를 위한 특수 레지스터들을 사용하여 동작하게 되며 NTB를 지원하는 PCIe 장치 드라이버와 NTB용 커널 모듈이 사용된다.

이러한 생소하고 낯선 일종의 특수 장치에 대한 접근을 좀 더 직관적이고 보편화된 방식으로 사용할 수



(그림 5) ping 측정 결과

있도록 지원하는 통신 방식이 TCP over NTB로서 리눅스 커널 배포판을 통해 주요 모듈들이 제공되고 있다. NTB 통신을 TCP로 지원하는 방식은 노드 간 확장성에 제약이 있기는 하지만 TCP를 사용하는 어플리케이션들에 대한 별도의 수정없이 NTB 네트워크을 이용해서 동작할 수 있다는 장점이 있다. 하지만 이번 성능 측정 실험 결과는 PCIe 버전과 PCIe 레인의 분기 등을 향상시키더라도 통신 성능은 개선되지 않는 한계를 보였다. 즉 여러 어플리케이션들에 쉽게 사용될 수 있으나 성능 효율성이 현저하게 떨어지는 단점도 보여 주었다.

한편 NTB 모듈을 직접 접근하도록 작성된 ntb_perf를 사용한 통신 성능은 PCIe 버전과 분기 설정을 증가시키면 통신 성능도 함께 개선됨을 보여 주었다. 물론 특정 조건을 넘어서는 구간에서는 성능 개선이 나타나지 않는 경우도 있었으나 일반적으로는 이론 성능 대비 약 70% 정도의 통신 성능을 보여 줄 정도로 높은 성능 효율성을 보여 준다고 할 수 있다.

물론 TCP 방식의 통신 성능과 ntb_perf의 통신 성능을 직접 비교하는 것이 적합하다고는 볼 수 없다. 그 이유는 ntb_perf는 사용자 레벨의 어플리케이션이 아닌 커널 모듈 방식의 테스트 프로그램으로 커널 파라메터와 debugfs를 사용자 인터페이스로 사용하고 있기 때문이다. 전송 데이터도 ntb_perf 모듈 내에서 생성되므로 사용자 레벨과 커널 레벨간 데이터 전달 과정을 포함하고 있지 않다. 그렇기 때문에 커널 계층과 사용자 계층 간의 데이터 이동에 따른 성능 저하가 발생할 수 있음을 고려해야 한다. 그렇다 하더라고 TCP를 이용하는 방법보다는 성능 차이가 크기 때문에 NTB를 사용하는 PCIe 네트워크를 효율적으로 사용하기 위해서는 NTB 모듈을 직접 접근하는 전용 통신 소프트웨어의 사용 또는 개발이 우선적으로 해결되어야 할 것으로 예견된다.

ACKNOWLEDGMENTS

이 논문은 22-2차 무기체계 부품 국산화 개발지원 사업의 지원을 받아 수행된 연구입니다. (과제번호 : C220027)

참고문헌

[1] Casey Morrison and Jonathan Bender, “Seamless Transition to PCIe 5.0 Technology in System Implementations,” PCI-SIG Webinar, Dec. 9, 2020.

[2] PCI-SIG, “PCI Express® Base Specification Revision 6.0.1,” 29 August 2022.

[3] HPEC High-Performance Embedded Computing, https://www.curtisswrightds.com/capabilities/technologies/hpec

[4] Iperf 3, https://iperf.fr

[5] Netperf, https://github.com/HewlettPackard/netperf

[6] The Linux Kernel Archives, https://www.kernel.org